



## **Cyber Security (CYBER); Cyber Resilience Act (CRA); Cybersecurity requirements for Virtualisation Execution Stack (VES) and Container Execution Stack (CES), including hypervisors and container runtime systems**

Exercising one of the **Principles of International Standardization – Openness** – and taking them to a different level, ETSI CYBER-EUSR has conducted open consultations on the vertical standards in support of the Cyber-Resilience Act, at a much earlier stage than it is usual in the standardization world. In this context, please keep in mind that the present document is an INTERIM draft, which expectably will be subject to substantial changes before their target publication date in the second semester of 2026.

ETSI CRA vertical standards v1.1.1 are being finalized and undergoing Public Enquiry via the National Standardization Organizations (NSO). Therefore, starting from 16 April 2026, ETSI CYBER-EUSR may only be able to address your comments in a future revision of the standard. **To enable ETSI CYBER-EUSR to address your comments before the first publication of the standards, you should cumulatively submit to your NSO any comments submitted here from 16 April 2026 onwards.** If you don't know how to do this, email us at [cybersupport@etsi.org](mailto:cybersupport@etsi.org) and we will guide you in the process.

**Disclaimer:** The INTERIM DRAFTS available for download in this page are provided for information and are for future development work within the ETSI Technical Committee CYBER EUSR Working Group (CYBER-EUSR) only. ETSI and its Members accept no liability for any further use/implementation of these specifications. Approved and published specifications and reports shall be obtained exclusively via the ETSI Documentation Service at <http://www.etsi.org/standards-search>

**Commenting guidelines:** Your comments to improve this early draft are very welcomed. To ensure the effectiveness of your contribution, please make sure to include the following elements in your comment:

1. Clear identification of the section of the draft your comment is referring to.
2. Objective proposal to delete content, add content or substitute content.
3. Concrete contribution (exact content you suggest including in the draft as an addition to, or in substitution of, existing content).
4. Brief rationale to justify the proposal (e.g. why the suggested content should be added an/or the existing content should be deleted or substituted).

Please note that comments without concrete contributions will be noted but not acted upon by ETSI CYBER-EUSR. We trust and value your expertise and therefore expect you to take this opportunity to proactively contribute to solving the issues you find while analysing this early draft.

**Use of Artificial Intelligence (AI):** Please do not use large language models to generate your comments. Inclusion of language generated by “AI” creates a potential for intellectual property conflicts and liability for ETSI, which is not acceptable.

**How to comment:** To comment or provide feedback on the present interim draft please visit:

<https://labs.etsi.org/rep/stan4cra/en-304-635> and submit your comments as “issues” following the guidelines provided on this site. Alternatively, you may also send your comments to: [cybersupport@etsi.org](mailto:cybersupport@etsi.org) using the “[ETSI Commenting Format for Open Consultation.xlsx](#)” available in <https://docbox.etsi.org/CYBER/EUSR/Open>

**Feedback on your comments:** The resolutions made in **Consensus** by ETSI CYBER-EUSR members, on each comment received through these Open Consultations will be published at the ETSI Open Area and ETSI Lab, assuring full **Transparency**.

---

**Reference**

DEN/CYBER-EUS-0016

---

**Keywords**

CRA; Cybersecurity; Virtualization; Container

0

**ETSI**

---

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° w061004871

---

**Important notice**

The present document can be downloaded from the  
[ETSI Search & Browse Standards](#) application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed,  
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to  
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our  
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

---

**Notice of disclaimer & limitation of liability**

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

---

**Copyright Notification**

Reproduction is only permitted for the purpose of standardization work undertaken within ETSI.  
The copyright and the foregoing restrictions extend to reproduction in all media.

© ETSI 2026.  
All rights reserved.

1	<b>Contents</b>	
2	Intellectual Property Rights.....	9
3	Foreword.....	9
4	Modal verbs terminology .....	10
5	Introduction .....	10
6	1 Scope.....	11
7	2 References .....	11
8	2.1 Normative references.....	11
9	2.2 Informative references .....	12
10	3 Definition of terms, symbols and abbreviations.....	13
11	3.1 Terms and Definitions .....	13
12	3.2 Symbols .....	14
13	3.3 Abbreviations.....	14
14	4 Product Context.....	16
15	4.1 Product's intended purpose and reasonably foreseeable use .....	16
16	4.1.1 General .....	16
17	4.1.2 VES .....	16
18	4.1.2.1 Intended purpose and reasonably foreseeable use of the VES.....	16
19	4.1.2.2 Intended purpose and reasonably foreseeable use of the Hypervisor .....	17
20	4.1.2.3 Intended purpose and reasonably foreseeable use of the M&O system.....	17
21	4.1.3 CES .....	17
22	4.1.3.1 Intended purpose and reasonably foreseeable use of the CES.....	17
23	4.1.3.2 Intended purpose and reasonably foreseeable use of the CRS.....	18
24	4.1.3.3 Intended purpose and reasonably foreseeable use of the CE.....	18
25	4.1.3.4 Intended purpose and reasonably foreseeable use of the CO.....	18
26	4.2 Product architecture.....	18
27	4.2.1 General distinction of components.....	18
28	4.2.2 VES .....	19
29	4.2.2.1 Architectural types of VES.....	19
30	4.2.2.2 In-scope components of the VES .....	19
31	4.2.2.2.1 Hypervisor .....	19
32	4.2.2.2.2 Management and Orchestration System.....	20
33	4.2.2.3 Security-relevant environmental dependencies (out of scope) .....	21
34	4.2.3 CES .....	21
35	4.2.3.1 In-scope components .....	21
36	4.2.3.2 Security-relevant environmental dependencies (out of scope) .....	22
37	4.2.4 Deployment of M&O, CE, and CO outside the local execution environment .....	23
38	4.3 Product variants and conforming products .....	23
39	4.3.1 Definition of conforming products.....	23
40	4.3.2 Product variants.....	24
41	4.4 Operational environment .....	24
42	4.4.1 General Principles .....	24
43	4.4.2 Security Objectives for the Operational Environment with Requirement Traceability.....	24
44	4.4.3 Reference to Relevant CRA Harmonised Standards .....	26
45	4.5 Risk-Based requirement classes/categorization and application.....	26
46	4.5.1 Requirement Classes, Categorization and Security Profiles.....	26
47	4.5.1.1 Definition of Requirement Classes: Basic, Elevated, Advanced .....	26
48	4.5.1.2 Security Profile Definitions .....	27
49	4.5.1.3 Requirement Categorization .....	27
50	4.5.1.4 Guidance for Manufacturers.....	28
51	4.5.2 Requirement Application .....	29
52	4.5.2.1 Approaches for determining security requirements.....	29
53	4.5.2.2 SP-Based Requirement Application .....	29
54	4.6 Users.....	29
55	4.7 Use cases.....	30

56	4.7.1	Purpose .....	30
57	4.7.2	Use Cases for VES .....	30
58	4.7.3	Use Cases for CES .....	32
59	4.7.4	Use Case to Risk Mapping .....	34
60	5	Technical Requirements for the Products .....	35
61	5.1	Applicability of requirements .....	35
62	5.2	VES Security Requirements .....	35
63	5.2.1	Hypervisor Requirements.....	35
64	5.2.1.1	Isolation .....	35
65	5.2.1.1.1	General.....	35
66	5.2.1.1.2	VM Isolation .....	35
67	5.2.1.1.3	Control Plane Isolation .....	36
68	5.2.1.1.4	Network Plane Separation.....	36
69	5.2.1.2	Integrity Protection.....	37
70	5.2.1.2.1	Boot chain integrity verification .....	37
71	5.2.1.2.2	Guest VM image integrity verification .....	38
72	5.2.1.2.3	Runtime integrity protection .....	38
73	5.2.1.2.4	Remote attestation.....	39
74	5.2.1.3	Authentication .....	40
75	5.2.1.3.1	General.....	40
76	5.2.1.3.2	Administrative Authentication.....	40
77	5.2.1.3.3	Service Authentication.....	40
78	5.2.1.4	Authorization.....	41
79	5.2.1.4.1	General.....	41
80	5.2.1.4.2	Administrative Authorization .....	41
81	5.2.1.4.3	Service Authorization .....	42
82	5.2.1.5	Confidentiality Protection.....	42
83	5.2.1.6	Availability and Resilience.....	43
84	5.2.1.7	Logging .....	43
85	5.2.1.8	Secure Update.....	44
86	5.2.1.9	Secure Configuration and Default .....	45
87	5.2.1.10	Data Minimization.....	46
88	5.2.2	Applicability of Hypervisor Requirements .....	46
89	5.2.3	Management and Orchestration System Requirements.....	49
90	5.2.3.1	General applicability.....	49
91	5.2.3.2	Authentication .....	49
92	5.2.3.3	Authorization.....	50
93	5.2.3.4	Secure Configuration.....	50
94	5.2.3.5	Communication Security .....	51
95	5.2.3.6	Integrity Protection.....	51
96	5.2.3.7	Logging .....	51
97	5.2.3.8	Secure Update.....	52
98	5.3	CES Security Requirements.....	53
99	5.3.1	CRS Requirements .....	53
100	5.3.1.1	Isolation .....	53
101	5.3.1.1.1	Container Isolation.....	53
102	5.3.1.1.2	Control Plane Isolation .....	53
103	5.3.1.1.3	Network Plane Isolation.....	54
104	5.3.1.2	Integrity Protection.....	55
105	5.3.1.2.1	Boot chain integrity verification .....	55
106	5.3.1.2.2	Container image integrity verification .....	56
107	5.3.1.2.3	Runtime integrity protection .....	57
108	5.3.1.2.4	Remote attestation.....	57
109	5.3.1.3	Authentication .....	58
110	5.3.1.3.1	General.....	58
111	5.3.1.3.2	Administrative Authentication.....	58
112	5.3.1.3.3	Service Authentication.....	59
113	5.3.1.4	Authorization.....	59
114	5.3.1.4.1	General.....	59
115	5.3.1.4.2	Administrative Authorization .....	60
116	5.3.1.4.3	Service Authorization .....	60

117	5.3.1.5	Confidentiality Protection.....	61
118	5.3.1.6	Availability and Resilience.....	61
119	5.3.1.7	Logging .....	62
120	5.3.1.8	Secure Update.....	63
121	5.3.1.9	Secure Configuration and Default .....	64
122	5.3.1.10	Data Minimization.....	64
123	5.3.2	CE Requirements .....	65
124	5.3.2.1	Isolation .....	65
125	5.3.2.2	Integrity Protection.....	65
126	5.3.2.3	Authentication .....	66
127	5.3.2.4	Authorization.....	66
128	5.3.2.5	Confidentiality.....	66
129	5.3.2.6	Secure Update.....	67
130	5.3.3	CO Requirements.....	68
131	5.3.3.1	Isolation .....	68
132	5.3.3.2	Integrity Protection.....	68
133	5.3.3.3	Authentication .....	69
134	5.3.3.4	Authorization.....	69
135	5.3.3.5	Confidentiality.....	69
136	5.3.3.6	Secure Update.....	70
137	5.4	Assurance Security Requirements .....	71
138	5.4.1	Vulnerability Management.....	71
139	5.4.2	Software Bill of Materials (SBOM) .....	71
140	6	Conformance Assessment / Tests.....	73
141	6.1	Assessment Methodology .....	73
142	6.1.1	General Assessment Procedure .....	73
143	6.1.2	Advanced-class Equivalence and Technology-neutral Assessment .....	75
144	6.1.2.1	General .....	75
145	6.1.2.2	Equivalence of Hardware-based and Software-based Approaches.....	75
146	6.1.2.3	Technology-neutral Assessment Principles.....	76
147	6.2	Assessment Report Requirements.....	76
148	6.2.1	Mandatory Report Contents .....	76
149	6.2.2	Traceability Requirements .....	78
150	6.2.3	Evidence Package.....	78
151	6.3	VES.....	78
152	6.3.1	Hypervisor.....	78
153	6.3.1.1	Assessment for VM Isolation .....	78
154	6.3.1.1.1	Assessment Case AC-H-VM-ISO-001 .....	78
155	6.3.1.1.2	Assessment Case AC-H-VM-ISO-002 .....	80
156	6.3.1.1.3	Assessment Case AC-H-VM-ISO-003 .....	81
157	6.3.1.1.4	Assessment Case AC-H-VM-ISO-004 .....	83
158	6.3.1.2	Assessment for Control Plane Isolation.....	84
159	6.3.1.2.1	Assessment Case AC-H-CP-ISO-001.....	84
160	6.3.1.2.2	Assessment Case AC-H-CP-ISO-002.....	85
161	6.3.1.2.3	Assessment Case AC-H-CP-ISO-003.....	86
162	6.3.1.3	Assessment for Network Plane Separation.....	87
163	6.3.1.3.1	Assessment Case AC-H-NP-ISO-001.....	87
164	6.3.1.3.2	Assessment Case AC-H-NP-ISO-002.....	88
165	6.3.1.3.3	Assessment Case AC-H-NP-ISO-003.....	89
166	6.3.1.4	Assessment for Boot Chain Integrity Verification.....	91
167	6.3.1.4.1	Assessment Case AC-H-B-INT-001.....	91
168	6.3.1.4.2	Assessment Case AC-H-B-INT-002.....	92
169	6.3.1.4.3	Assessment Case AC-H-B-INT-003.....	93
170	6.3.1.5	Assessment for Guest VM Image Integrity Verification .....	94
171	6.3.1.5.1	Assessment Case AC-H-IMG-INT-001.....	94
172	6.3.1.5.2	Assessment Case AC-H-IMG-INT-002.....	96
173	6.3.1.5.3	Assessment Case AC-H-IMG-INT-003.....	97
174	6.3.1.6	Assessment for Runtime Integrity Protection.....	98
175	6.3.1.6.1	Assessment Case AC-H-RP-INT-002.....	98
176	6.3.1.6.2	Assessment Case AC-H-RP-INT-003.....	100
177	6.3.1.7	Assessment for Remote Attestation.....	101

178	6.3.1.7.1	Assessment Case AC-H-RA-INT-003 .....	101
179	6.3.1.7.2	Assessment Case AC-H-RA-INT-004 .....	103
180	6.3.1.8	Assessment for Administrative Authentication .....	105
181	6.3.1.8.1	Assessment Case AC-H-ADMIN-AUTH-001 .....	105
182	6.3.1.8.2	Assessment Case AC-H-ADMIN-AUTH-002 .....	106
183	6.3.1.8.3	Assessment Case AC-H-ADMIN-AUTH-003 .....	107
184	6.3.1.9	Assessment for Service Authentication .....	108
185	6.3.1.9.1	Assessment Case AC-H-SERV-AUTH-001 .....	108
186	6.3.1.9.2	Assessment Case AC-H-SERV-AUTH-002 .....	109
187	6.3.1.9.3	Assessment Case AC-H-SERV-AUTH-003 .....	110
188	6.3.1.10	Assessment for Administrative Authorization .....	111
189	6.3.1.10.1	Assessment Case AC-H-ADMIN-AUTHZ-001 .....	111
190	6.3.1.10.2	Assessment Case AC-H-ADMIN-AUTHZ-002 .....	112
191	6.3.1.10.3	Assessment Case AC-H-ADMIN-AUTHZ-003 .....	113
192	6.3.1.11	Assessment for Service Authorization .....	114
193	6.3.1.11.1	Assessment Case AC-H-SERV-AUTHZ-001 .....	114
194	6.3.1.11.2	Assessment Case AC-H-SERV-AUTHZ-002 .....	114
195	6.3.1.11.3	Assessment Case AC-H-SERV-AUTHZ-003 .....	115
196	6.3.1.12	Assessment for Confidentiality Protection .....	116
197	6.3.1.12.1	Assessment Case AC-H-CONF-001 .....	116
198	6.3.1.12.2	Assessment Case AC-H-CONF-002 .....	118
199	6.3.1.12.3	Assessment Case AC-H-CONF-003 .....	119
200	6.3.1.12.4	Assessment Case AC-H-CONF-004 .....	121
201	6.3.1.13	Assessment for Availability and Resilience .....	123
202	6.3.1.13.1	Assessment Case AC-H-AVAIL-001 .....	123
203	6.3.1.13.2	Assessment Case AC-H-AVAIL-002 .....	123
204	6.3.1.13.3	Assessment Case AC-H-AVAIL-003 .....	124
205	6.3.1.13.4	Assessment Case AC-H-AVAIL-004 .....	125
206	6.3.1.14	Assessment for Logging .....	126
207	6.3.1.14.1	Assessment Case AC-H-LOG-001 .....	126
208	6.3.1.14.2	Assessment Case AC-H-LOG-002 .....	127
209	6.3.1.14.3	Assessment Case AC-H-LOG-003 .....	128
210	6.3.1.15	Assessment for Secure Update .....	129
211	6.3.1.15.1	Assessment Case AC-H-UPD-001 .....	129
212	6.3.1.15.2	Assessment Case AC-H-UPD-002 .....	131
213	6.3.1.15.3	Assessment Case AC-H-UPD-003 .....	132
214	6.3.1.16	Assessment for Secure Configuration and Default .....	133
215	6.3.1.16.1	Assessment Case AC-H-CFG-001 .....	133
216	6.3.1.16.2	Assessment Case AC-H-CFG-002 .....	134
217	6.3.1.16.3	Assessment Case AC-H-CFG-003 .....	135
218	6.3.1.17	Assessment for Data Minimization .....	136
219	6.3.1.17.1	Assessment Case AC-H-DM-001 .....	136
220	6.3.1.17.2	Assessment Case AC-H-DM-002 .....	137
221	6.3.2	M&O System .....	138
222	6.4	CES .....	138
223	6.4.1	CRS .....	138
224	6.4.1.1	Assessment for Container Isolation .....	138
225	6.4.1.1.1	Assessment Case AC-CRS-CN-ISO-001 .....	138
226	6.4.1.1.2	Assessment Case AC-CRS-CN-ISO-002 .....	140
227	6.4.1.1.3	Assessment Case AC-CRS-CN-ISO-003 .....	141
228	6.4.1.2	Assessment for Control Plane Isolation .....	142
229	6.4.1.2.1	Assessment Case AC-CRS-CP-ISO-001 .....	142
230	6.4.1.2.2	Assessment Case AC-CRS-CP-ISO-002 .....	144
231	6.4.1.2.3	Assessment Case AC-CRS-CP-ISO-003 .....	145
232	6.4.1.3	Assessment for Network Plane Isolation .....	146
233	6.4.1.3.1	Assessment Case AC-CRS-NP-ISO-001 .....	146
234	6.4.1.3.2	Assessment Case AC-CRS-NP-ISO-002 .....	147
235	6.4.1.3.3	Assessment Case AC-CRS-NP-ISO-003 .....	149
236	6.4.1.4	Assessment for Boot Chain Integrity Verification .....	150
237	6.4.1.4.1	Assessment Case AC-CRS-B-INT-001 .....	150
238	6.4.1.4.2	Assessment Case AC-CRS-B-INT-002 .....	151
239	6.4.1.4.3	Assessment Case AC-CRS-B-INT-003 .....	152

240	6.4.1.5	Assessment for Container Image Integrity Verification .....	154
241	6.4.1.5.1	Assessment Case AC-CRS-IMG-INT-001 .....	154
242	6.4.1.5.2	Assessment Case AC-CRS-IMG-INT-002 .....	155
243	6.4.1.5.3	Assessment Case AC-CRS-IMG-INT-003 .....	156
244	6.4.1.6	Assessment for Runtime Integrity Protection .....	158
245	6.4.1.6.1	Assessment Case AC-CRS-RP-INT-002 .....	158
246	6.4.1.6.2	Assessment Case AC-CRS-RP-INT-003 .....	159
247	6.4.1.7	Assessment for Remote Attestation .....	161
248	6.4.1.7.1	Assessment Case AC-CRS-RA-INT-003 .....	161
249	6.4.1.7.2	Assessment Case AC-CRS-RA-INT-004 .....	163
250	6.4.1.8	Assessment for Administrative Authentication .....	165
251	6.4.1.8.1	Assessment Case AC-CRS-ADMIN-AUTH-001 .....	165
252	6.4.1.8.2	Assessment Case AC-CRS-ADMIN-AUTH-002 .....	166
253	6.4.1.8.3	Assessment Case AC-CRS-ADMIN-AUTH-003 .....	167
254	6.4.1.9	Assessment for Service Authentication .....	168
255	6.4.1.9.1	Assessment Case AC-CRS-SERV-AUTH-001 .....	168
256	6.4.1.9.2	Assessment Case AC-CRS-SERV-AUTH-002 .....	169
257	6.4.1.9.3	Assessment Case AC-CRS-SERV-AUTH-003 .....	170
258	6.4.1.10	Assessment for Administrative Authorization .....	172
259	6.4.1.10.1	Assessment Case AC-CRS-ADMIN-AUTHZ-001 .....	172
260	6.4.1.10.2	Assessment Case AC-CRS-ADMIN-AUTHZ-002 .....	172
261	6.4.1.10.3	Assessment Case AC-CRS-ADMIN-AUTHZ-003 .....	173
262	6.4.1.11	Assessment for Service Authorization .....	174
263	6.4.1.11.1	Assessment Case AC-CRS-SERV-AUTHZ-001 .....	174
264	6.4.1.11.2	Assessment Case AC-CRS-SERV-AUTHZ-002 .....	175
265	6.4.1.11.3	Assessment Case AC-CRS-SERV-AUTHZ-003 .....	176
266	6.4.1.12	Assessment for Confidentiality Protection .....	177
267	6.4.1.12.1	Assessment Case AC-CRS-CONF-001 .....	177
268	6.4.1.12.2	Assessment Case AC-CRS-CONF-002 .....	179
269	6.4.1.12.3	Assessment Case AC-CRS-CONF-003 .....	180
270	6.4.1.12.4	Assessment Case AC-CRS-CONF-004 .....	182
271	6.4.1.13	Assessment for Availability and Resilience .....	184
272	6.4.1.13.1	Assessment Case AC-CRS-AVAIL-001 .....	184
273	6.4.1.13.2	Assessment Case AC-CRS-AVAIL-002 .....	185
274	6.4.1.13.3	Assessment Case AC-CRS-AVAIL-003 .....	186
275	6.4.1.13.4	Assessment Case AC-CRS-AVAIL-004 .....	187
276	6.4.1.14	Assessment for Logging .....	188
277	6.4.1.14.1	Assessment Case AC-CRS-LOG-001 .....	188
278	6.4.1.14.2	Assessment Case AC-CRS-LOG-002 .....	189
279	6.4.1.14.3	Assessment Case AC-CRS-LOG-003 .....	190
280	6.4.1.15	Assessment for Secure Update .....	191
281	6.4.1.15.1	Assessment Case AC-CRS-UPD-001 .....	191
282	6.4.1.15.2	Assessment Case AC-CRS-UPD-002 .....	192
283	6.4.1.15.3	Assessment Case AC-CRS-UPD-003 .....	193
284	6.4.1.16	Assessment for Secure Configuration and Default .....	194
285	6.4.1.16.1	Assessment Case AC-CRS-CFG-001 .....	194
286	6.4.1.16.2	Assessment Case AC-CRS-CFG-002 .....	195
287	6.4.1.16.3	Assessment Case AC-CRS-CFG-003 .....	196
288	6.4.1.17	Assessment for Data Minimization .....	197
289	6.4.1.17.1	Assessment Case AC-CRS-DM-001 .....	197
290	6.4.1.17.2	Assessment Case AC-CRS-DM-002 .....	199
291	6.4.2	CE .....	200
292	6.4.3	CO .....	200
293	Annex A (informative): Relationship between the present document and the requirements of EU		
294	Regulation (EU) 2024/2847 - the Cyber Resilience Act .....		200
295	Annex B (informative): Security Analysis .....		204
296	B.1	Overview .....	204
297	B.2	Threats .....	204
298	B.2.1	Overview .....	204
299	B.2.2	VES .....	205

300	B.2.2.1	Hyper Type I.....	205
301	B.2.2.2	Hyper Type II .....	206
302	B.2.2.3	Orchestration and Management system.....	207
303	B.2.3	CES .....	208
304	B.2.3.1	Container Orchestrator .....	208
305	B.2.3.2	Container Engine.....	209
306	B.2.3.3	Container Runtime System.....	210
307	<b>B.2.4</b>	<b>Hardware-level common threat.....</b>	<b>211</b>
308	B.3	Risk Assessment Process .....	212
309	B.4	Risk Factor Scoring .....	213
310	B.5	Calculation and Mapping to Likelihood/Impact Levels.....	216
311	B.5.1	General .....	216
312	B.5.2	Likelihood Score .....	216
313	B.5.3	Impact Score.....	216
314	B.6	Risk Matrix .....	217
315	B.7	Risk Methodology Applied to Use Cases .....	217
316	B.7.1	General .....	217
317	B.7.2	Risk Evaluation of VES use cases.....	217
318	B.7.2.1	Hypervisor Type I.....	217
319	B.7.2.2	Hypervisor Type II .....	223
320	B.7.2.3	M&O System.....	227
321	B.7.3	Risk Evaluation of CES use cases .....	228
322	B.7.3.1	CRS .....	228
323	B.7.3.2	CE .....	233
324	B.7.3.3	CO .....	233
325	B.7.4	Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS .....	233
326	B.8	Threat-to-Requirement Traceability Matrix.....	236
327	B.8.1	VES .....	236
328	B.8.1.1	Hypervisor.....	236
329	B.8.1.2	M&O System.....	237
330	B.8.2	CES .....	238
331	B.8.2.1	CRS .....	238
332	B.8.2.2	CE.....	239
333	B.8.2.3	CO .....	239
334	B.8.3	Hardware-level common threat.....	239
335		Annex K: Cryptographic mechanisms.....	240
336	K.1	General.....	240
337	K.2	Mapping of Hypervisor requirements to agreed cryptographic mechanisms (ACM).....	240
338	K.3	Mapping of CRS requirements to agreed cryptographic mechanisms (ACM) .....	241
339		Annex R: Additional provisions for products relying on remote data processing solutions (RDPS).....	242
340		Annex (informative): Bibliography .....	242
341		Annex (informative): Change history.....	243
342		History .....	246
343			
344			

## 345 Intellectual Property Rights

### 346 Essential patents

347 IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations  
 348 pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be  
 349 found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to*  
 350 *ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the  
 351 [ETSI IPR online database](#).

352 Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs,  
 353 including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not  
 354 referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become,  
 355 essential to the present document.

### 356 Trademarks

357 The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners.  
 358 ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no  
 359 right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does  
 360 not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

361 **DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its  
 362 Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the  
 363 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of  
 364 the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

## 365 Foreword

366 This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber Security  
 367 (CYBER) and is now submitted for the combined Public Enquiry and Vote phase of the ETSI EN Approval Procedure.

368 The present document has been prepared under the Commission's standardisation request C(2025)618 [i.3] to provide  
 369 one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and  
 370 of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and  
 371 amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act)  
 372 [i.1].

373 Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance  
 374 with the normative clauses of the present document given in table A.1 confers, within the limits of the scope of the  
 375 present document, a presumption of conformity with the corresponding requirements of that Regulation and associated  
 376 EFTA regulations.

377

<b>Proposed national transposition dates</b>	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	18 months after doa

378

379

---

## Modal verbs terminology

380

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

381

382

383

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

384

---

## Introduction

385

The present document defines cybersecurity requirements applicable to Virtualization Execution Stack (VES) and Container Execution Stack (CES), including hypervisors and container runtime systems. It applies to products with digital elements designed to host, manage, and isolate virtualized workloads and execution environments, using:

386

387

388

- Virtual machines (VMs): full system emulation with independent operating systems; or

389

- Containers: process isolation using OS-level mechanisms.

390

The present document supports compliance with Regulation (EU) 2024/2847, the Cyber Resilience Act, specifically addressing the essential cybersecurity requirements defined in Annex I, Parts I and II.

391

---

## 392 1 Scope

393 The present document is based on the definitions provided in the Cyber Resilience Act (CRA) [i.2] for hypervisors and  
394 container runtime systems as core components.

395 While the CRA definition identifies the core execution components, actual market products typically include additional  
396 elements beyond the hypervisor kernel or container runtime binary. These additional components provide essential  
397 management, orchestration, and operational capabilities that are necessary for real-world deployment and are therefore  
398 included within the scope of the present document.

399 The present document focuses on:

- 400 • Virtualization Execution Stack (VES) for hypervisor-based environments; and
- 401 • Container Execution Stack (CES) for container-based environments.

402 The corresponding terms and definitions are provided in clause 3. The architectural decomposition, in-scope  
403 components, and security-relevant environmental dependencies are specified in clause 4.

404 Accordingly, the present document defines security requirements not only for the core execution systems identified in  
405 the CRA but also for the broader product context in which these systems are deployed, ensuring alignment with market  
406 reality and comprehensive coverage of security risks. The Management and Orchestration (M&O) System, Container  
407 Engine (CE), and Container Orchestrator (CO) are covered by the present document and are in scope only where they  
408 are developed or provided by the manufacturer, or under the responsibility of the manufacturer, as part of the declared  
409 product.

410 Any usage of AI agents is out of scope of the present document.

411 The present document is intended to be applied as follows:

- 412 1. the manufacturer identifies whether the product is a VES or a CES, and determines which components are  
413 included within the product boundary in accordance with clauses 4.1 and 4.2;
- 414 2. the manufacturer identifies the relevant product use case in accordance with clause 4.7 and associated risk  
415 level in accordance with clause 4.3 and, where needed, clause B;
- 416 3. the manufacturer determines the applicable Security Profile (SP) and the corresponding requirement  
417 application logic in accordance with clause 4.5 and clause 5;
- 418 4. the manufacturer applies only those requirements specified in clause 5 that are relevant to the components  
419 included in the product, its intended use, and the applicable SP; and
- 420 5. conformity assessment is performed against the corresponding assessment provisions in clause 6.

421 Where the product includes or depends on components that are outside the scope of the present document, the  
422 applicable requirements are to be addressed through the relevant operational-environment provisions or other relevant  
423 harmonised standards, as identified in clauses 4.4.

---

## 424 2 References

### 425 2.1 Normative references

426 References are either specific (identified by date of publication and/or edition number or version number) or non-  
427 specific. For specific references, only the cited version applies. For non-specific references, the latest version of the  
428 referenced document (including any amendments) applies.

429 Referenced documents which are not found to be publicly available in the expected location might be found in the  
430 [ETSI docbox](#).

431 NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee  
432 their long-term validity.

433 The following referenced documents are necessary for the application of the present document.

434 [1] [prEN 40000-1-3](#): "Cybersecurity requirements for products with digital elements - Vulnerability  
435 Handling".

436 [2] [ENISA-ECCG](#): "Agreed Cryptographic Mechanisms", Version 2.0, April 2025.

## 437 2.2 Informative references

438 References are either specific (identified by date of publication and/or edition number or version number) or  
439 non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the  
440 referenced document (including any amendments) applies.

441 NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee  
442 their long-term validity.

443 The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's  
444 understanding but are not required for conformance to the present document.

445 [i.1] [Regulation \(EU\) 2024/2847](#) of the European Parliament and of the Council of 23 October 2024 on  
446 horizontal cybersecurity requirements for products with digital elements and amending  
447 Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber  
448 Resilience Act).

449 [i.2] Commission Implementing [Regulation \(EU\) 2025/2392](#) of 28 November 2025 on the technical  
450 description of the categories of important and critical products with digital elements pursuant to  
451 Regulation (EU) 2024/2847 of the European Parliament and of the Council.

452 [i.3] [C\(2025\)618 - Standardisation request M/606](#): Commission Implementing decision of 3.2.2025 on  
453 a standardisation request to the European Committee for Standardisation (CEN), the European  
454 Committee for Electrotechnical Standardisation (Cenelec) and the European Telecommunications  
455 Standards Institute (ETSI) as regards products with digital elements in support of Regulation (EU)  
456 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal  
457 cybersecurity requirements for products with digital elements and amending Regulations (EU) No  
458 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) .

459 [i.4] ETSI EN 304 623: "CYBER; CRA; Cybersecurity requirements for boot managers".

460 [i.5] ETSI EN 304 626: "CYBER; CRA; Cybersecurity requirements for operating systems".

461 [i.6] ETSI EN 304 625: "CYBER; CRA; Cybersecurity requirements for physical and virtual network  
462 interfaces".

463 [i.7] ETSI EN 304 624: "CYBER; CRA; Cybersecurity requirements for Public key infrastructure and  
464 digital certificate issuance software".

465 [i.8] ETSI EN 304 622: "CYBER; CRA; Cybersecurity requirements for Security information and  
466 event management (SIEM) systems".

467 [i.9] ETSI EN 304 636: "CYBER; CRA; Cybersecurity requirements for firewalls, intrusion detection  
468 and/or prevention systems".

469 [i.10] [prEN 40000-1-1](#): "Cybersecurity requirements for products with digital elements - Vocabulary".

470 [i.11] [prEN 40000-1-2](#): "Cybersecurity requirements for products with digital elements - Part 1-2:  
471 Principles for cyber resilience".

## 472 3 Definition of terms, symbols and abbreviations

### 473 3.1 Terms and Definitions

474 For the purposes of the present document, the terms given in CRA Regulation [i.1], prEN 40000-1-1 [i.10] and the  
475 following apply.

476 **confidential computing:** security capability that provides data-in-use protection through hardware-based isolated  
477 execution

478 **container:** software-based isolated execution environment that packages an application with its dependencies and runs  
479 using the host operating system kernel

480 **Container Engine (CE):** software component that manages the lifecycle of containers on a host, including image  
481 handling, container creation, execution, logging, storage, and related management interfaces

482 **Container Execution Stack (CES):** layered set of software components enabling the deployment, management, and  
483 execution of containerized applications

484 **Container Orchestrator (CO):** software component that automates the deployment, scheduling, scaling, networking,  
485 availability, and lifecycle management of containerized applications across one or more machines

486 **dedicated management network / interface:** A network interface or network segment that is exclusively used for  
487 management plane traffic and is not shared with guest workload traffic, storage traffic, data-plane traffic, VM/container  
488 traffic, or tenant-accessible networks.

489 NOTE: A dedicated management interface may be implemented as:

- 490 • a physically separate network interface assigned solely to management traffic; or
- 491 • a logically isolated network (e.g. VLAN, VXLAN, VRF) where isolation is enforced by the hypervisor /  
492 CRS / host networking layer and cannot be modified or bypassed by tenant workloads.

493 **hardware-mediated execution enclave:** execution environment in which hardware mechanisms provide protected  
494 isolation for code and data during execution and control access to memory and execution state from software outside  
495 that environment.

496 NOTE: Examples may include trusted execution environments or equivalent platform mechanisms providing  
497 isolated execution.

498 **hardware-based root of trust:** hardware-based trust anchor providing security functions used to establish or maintain  
499 trust in the product, such as protected key storage, measured boot, verified boot, attestation support, or cryptographic  
500 operations.

501 NOTE: Examples may include Trusted Platform Modules (TPM), secure elements, or equivalent platform trust  
502 anchors.

503 **hardware security module:** hardware or hardware-assisted component providing dedicated security functions, such as  
504 protected storage of keys or credentials, cryptographic processing, attestation support, or resistance against  
505 unauthorized access or modification.

506 NOTE: Examples may include Hardware Security Modules (HSM), Trusted Platform Modules (TPM), secure  
507 elements, or equivalent hardware security components.

508 **hypervisor:** software that enables the creation and management of virtual machines by abstracting, allocating and  
509 controlling access to a host's hardware resources, providing isolation between virtual machines

510 **impact:** severity of harm if a threat event occurs

511 **likelihood:** probability of a threat event occurring

512 **microVM:** lightweight virtual machine designed for minimal resource overhead and rapid startup, providing an isolated  
513 execution environment with a reduced attack surface compared to traditional virtual machines

- 514 **multi-tenancy:** architectural property of a virtualized or containerized environment in which multiple tenants share the  
515 same underlying infrastructure while maintaining isolation of their workloads and data
- 516 **strong authentication:** authentication that provides high-assurance identity verification for access to administrative  
517 interfaces, control functions or sensitive data, implemented using one or more mechanisms such as multi-factor  
518 authentication (MFA), certificate-based authentication, cryptographic key-based authentication using protected private  
519 keys, hardware-backed authentication mechanisms, or equivalent mechanisms providing the same assurance
- 520 **tenant:** isolated unit of use and management within a virtualized or containerized environment, representing a single  
521 user or organization sharing the underlying infrastructure
- 522 **type I hypervisor (Bare-Metal / Native):** hypervisor that runs directly on physical hardware to create and manage  
523 virtual machines
- 524 **type II hypervisor (Hosted):** hypervisor that runs as an application on top of a general-purpose host operating system  
525 to create and manage virtual machines
- 526 **Virtualization Execution Stack (VES):** set of software components responsible for deploying, running, and managing  
527 Virtual Machines (VMs) on shared physical infrastructure
- 528 **Virtual Machine (VM):** software-based virtualized computing environment that runs a guest operating system on  
529 abstracted hardware resources provided by a hypervisor
- 530 **Paravirtualization:** a virtualization approach where the guest OS is modified to make direct API calls (hypercalls) to  
531 the hypervisor instead of issuing hardware instructions that must be intercepted and emulated.

## 532 3.2 Symbols

533 Void.

## 534 3.3 Abbreviations

535 For the purposes of the present document, the following abbreviations apply:

536	3DES	Triple Data Encryption Standard
537	ABAC	Attribute-Based Access Control
538	ACL	Access Control List
539	ACM	Agreed Cryptographic Mechanisms
540	AES	Advanced Encryption Standard
541	API	Application Programming Interface
542	ASLR	Address Space Layout Redundancy
543	ASSUR	Assurance
544	BIOS	Basic Input/Output System
545	CE	Container Engine
546	CES	Container Execution Stack
547	CFG	Configuration
548	CI/CD	Continuous Integration / Continuous Delivery
549	CIS	Centre for Internet Security
550	CLI	Command Line Interface
551	CO	Container Orchestrator
552	CONF	Confidentiality
553	CRA	Cyber Resilience Act
554	CRI-O	Container Runtime Interface - Orchestrator (runtime)
555	CRS	Container Runtime System
556	CPU	Central Processing Unit
557	CTL	Control
558	DEF	Default
559	DMA	Direct Memory Access
560	DoS	Denial-of-Service
561	DRTM	Dynamic Root of Trust Measurement
562	ECDSA	Elliptic Curve Digital Signature Algorithm

563	ENISA	European Union Agency for Cybersecurity
564	ESXi	Elastic Sky X Integrated (VMware hypervisor name)
565	EUCC	EU Cybersecurity Certification Scheme on Common Criteria
566	FaaS	Function as a Service
567	GCM	Galois/Counter Mode (for encryption)
568	gRPC	gRPC Remote Procedure Call
569	HRoT	Hardware Root of Trust
570	HSM	Hardware Security Module
571	IaaS	Infrastructure as a Service
572	I/O	Input/Output
573	ID	Identification
574	IMA	Integrity Measurement Architecture
575	IMG	Image
576	IOMMU	Input/Output Memory Management Unit
577	ISO	International Organization for Standardization
578	IT	Information Technology
579	JIT	Just-In-Time
580	JSON	JavaScript Object Notation
581	JWT	JSON Web Token
582	KVM	Kernel-based Virtual Machine
583	LAN	Local Area Network
584	LOG	Logging
585	M&O	Management and Orchestration
586	MFA	Multi-Factor Authentication
587	mTLS	mutual Transport Layer Security
588	NIC	Network Interface Card
589	NTP	Network Time Protocol
590	OE	Operational Environment
591	OPA	Open Policy Agent
592	OS	Operating System
593	OSS	Open Source Software
594	PC	Personal Computer
595	PCR	Platform Configuration Register (related to TPM)
596	PDP	Policy Decision Point
597	PKI	Public Key Infrastructure
598	PTP	Precision Time Protocol
599	QEMU	Quick Emulator
600	RAM	Random Access Memory
601	RBAC	Role-Based Access Control
602	RC4	Rivest Cipher 4
603	REQ	Requirement
604	RPO	Recovery Point Objective
605	RSA	Rivest-Shamir-Adleman (Cryptographic algorithm)
606	SBOM	Software Bill of Materials
607	SDN	Software-Defined Networking
608	SEV	Secure Encrypted Virtualization
609	SEV-ES	Secure Encrypted Virtualization-Encrypted State
610	SHA	Secure Hash Algorithm
611	SIEM	Security Information and Event Management
612	SP	Security Profile
613	SPIFFE	Secure Production Identity Framework for Everyone
614	SR-IOV	Single Root Input/Output Virtualization
615	SSO	Single Sign-On
616	SUP	Support
617	TLS	Transport Layer Security
618	TPM	Trusted Platform Module
619	TS	Technical Specification
620	U/C	Use Case
621	UEFI	Unified Extensible Firmware Interface
622	UI	User Interface
623	UPD	Update
624	VES	Virtualization Execution Stack

625	VLAN	Virtual Local Area Network
626	VM	Virtual Machine
627	VMM	Virtual Machine Monitor (Hypervisor Core)
628	VPN	Virtual Private Network
629	VRF	Virtual Routing and Forwarding
630	vTPM	Virtual Trusted Platform Module
631	VT	Virtualization Technology
632	VULM	Vulnerability Management
633	VXLAN	Virtual Extensible LAN
634	XACML	eXtensible Access Control Markup Language

---

## 635 4 Product Context

### 636 4.1 Product's intended purpose and reasonably foreseeable 637 use

#### 638 4.1.1 General

639 The present document applies to products whose intended purpose is to provide virtualization execution or container  
640 execution functions, including workload execution, workload isolation, lifecycle management, configuration,  
641 administration, and, where applicable, orchestration and policy enforcement.

642 Reasonably foreseeable use includes operation in local, remote, clustered, distributed, or externally managed  
643 environments consistent with the product architecture and declared product functionality. It also includes interaction  
644 with operating systems, firmware, registries, identity services, logging systems, attestation systems, and other  
645 environmental components, even where such components remain outside the scope of the present document.

646 The present document addresses the intended purpose and reasonably foreseeable use of the following product  
647 categories:

- 648 • Virtualization Execution Stack (VES); and
- 649 • Container Execution Stack (CES).

650 The detailed product architecture, in-scope components, and security-relevant environmental dependencies are specified  
651 in clause 4.2.

#### 652 4.1.2 VES

##### 653 4.1.2.1 Intended purpose and reasonably foreseeable use of the VES

654 The intended purpose of a VES is to enable the creation, execution, isolation, administration, and management of  
655 virtualized workloads on one or more physical hosts.

656 A VES may include:

- 657 • a Hypervisor, which provides workload isolation, resource partitioning, and execution control; and
- 658 • a Management and Orchestration (M&O) System, which provides provisioning, configuration, administration,  
659 scheduling, and lifecycle management functions for virtualized resources and hosts.

660 Reasonably foreseeable use of a VES includes:

- 661 • deployment in Type I, Type II, or Hybrid virtualization architectures;
- 662 • administration through local or remote management interfaces;
- 663 • operation with one or more management systems coordinating one or more hypervisors;

- 664       • operation in environments where virtualized workloads share compute, memory, storage, and network  
665 resources under Hypervisor control.

#### 666 4.1.2.2 Intended purpose and reasonably foreseeable use of the Hypervisor

667 The intended purpose of the Hypervisor is to provide the execution environment for virtual machines and to enforce  
668 isolation, execution control, and resource partitioning between virtualized workloads.

669 This includes, as applicable:

- 670       • controlling execution of guest virtual machines;
- 671       • mediating access to CPU, memory, I/O, storage, and interrupt-handling resources;
- 672       • enforcing isolation between guest workloads and between guest workloads and privileged Hypervisor  
673 functions;
- 674       • managing virtual-machine lifecycle operations; and
- 675       • supporting security functions native to the Hypervisor, such as logging, integrity verification, update handling,  
676 or virtualized trust services.

#### 677 4.1.2.3 Intended purpose and reasonably foreseeable use of the M&O system

678 The intended purpose of the M&O System is to administer, configure, coordinate, and manage one or more Hypervisors  
679 and the virtualized infrastructure under their control.

680 This includes, as applicable:

- 681       • provisioning and configuration of virtualized resources and hosts;
- 682       • workload placement, scheduling, automation, and policy enforcement;
- 683       • lifecycle management of hosts and virtual machines;
- 684       • administration through interfaces used by operators or management agents; and
- 685       • orchestration of platform-wide functions such as updates, logging, monitoring, or trust-policy distribution.

### 686 4.1.3 CES

#### 687 4.1.3.1 Intended purpose and reasonably foreseeable use of the CES

688 The intended purpose of a CES is to enable the execution, isolation, management, and orchestration of containerized  
689 workloads.

690 A CES may include:

- 691       • a Container Runtime System (CRS), which provides the execution and isolation environment for containers;
- 692       • a Container Engine (CE), which provides image handling and container lifecycle management functions; and
- 693       • a Container Orchestrator (CO), which provides deployment, scheduling, scaling, and policy enforcement  
694 across one or more nodes.

695 Reasonably foreseeable use of a CES includes:

- 696       • deployment on a single host or across multiple hosts or cluster nodes;
- 697       • use of local or remote administrative and control interfaces;
- 698       • operation with registries, orchestration systems, logging systems, monitoring systems, secret-management  
699 systems, or identity services;

- 700       • operation in environments where multiple containerized workloads share compute, memory, storage, and  
701       network resources.

#### 702 4.1.3.2 Intended purpose and reasonably foreseeable use of the CRS

703 The intended purpose of the CRS is to execute containers and enforce isolation and resource control for containerized  
704 workloads on a host.

705 This includes, as applicable:

- 706       • creating, starting, stopping, and deleting containers;
- 707       • enforcing workload isolation and resource-control mechanisms;
- 708       • exposing runtime interfaces used by higher-level management components;
- 709       • protecting runtime configuration, control logic, and security-relevant state; and
- 710       • supporting native security functions such as logging, integrity verification, and update handling.

#### 711 4.1.3.3 Intended purpose and reasonably foreseeable use of the CE

712 The intended purpose of the CE is to manage container images and container lifecycle operations above the CRS.

713 This includes, as applicable:

- 714       • building, pulling, pushing, and storing container images;
- 715       • invoking and configuring CRS operations for container creation, execution, and termination;
- 716       • managing host-level container networking and storage integration; and
- 717       • managing engine-supported plugins, extensions, or other CE-native integration functions.

#### 718 4.1.3.4 Intended purpose and reasonably foreseeable use of the CO

719 The intended purpose of the CO is to coordinate deployment and lifecycle management of containerized workloads  
720 across one or more nodes.

721 This includes, as applicable:

- 722       • workload scheduling, scaling, and placement;
- 723       • admission control, policy enforcement, and cluster-level configuration management;
- 724       • management of service exposure, network policies, and workload recovery behaviour; and
- 725       • orchestration of updates, secrets handling, or cluster lifecycle functions where such functions are part of the  
726       product.

## 727 4.2 Product architecture

### 728 4.2.1 General distinction of components

729 The present document distinguishes between product components (within the scope of the present document) and  
730 environmental dependencies (outside the scope of the present document but potentially security-relevant).

- 731       • In-scope components represent the functional building blocks that together define the product boundary as  
732       considered in the present document. The requirements specified in the present document apply directly to these  
733       components.

- 734       • Environmental dependencies, by contrast, are not part of the product itself but belong to its operational  
735 environment (e.g. host operating system, firmware, or external services) (see clause 4.4).

## 736 4.2.2 VES

### 737 4.2.2.1 Architectural types of VES

738 Depending on the deployment model and integration with the host operating environment, the VES can be implemented  
739 in one of the following forms:

- 740       • Type I (Bare-metal hypervisor): A hypervisor in which the virtualisation layer holds the highest privilege level  
741 on the physical hardware, with all other software components — including any privileged management or  
742 device-driver domains — operating under its control and supervision regardless of whether a privileged  
743 management domain is present or not. Type I includes the following architectures:
  - 744           ○ Full virtualization: a technique where a hypervisor fully emulates the underlying hardware so that  
745 unmodified guest operating systems can run as if on real physical machines.
  - 746           ○ Paravirtualization: a technique where the guest operating system is modified to interact directly with  
747 the hypervisor through defined interfaces instead of relying on fully emulated hardware.
  - 748           ○ Hybrid system with primarily full virtualization with paravirtualized components.
- 749       • Type II (Hosted hypervisor):  
750 The hypervisor executes as an application on top of a general-purpose host operating system, relying on the  
751 host OS for device drivers and resource management.
- 752       • Hybrid (Mixed architecture):  
753 Combines characteristics of both Type I and Type II architectures. The hypervisor runs close to the hardware  
754 but leverages a general-purpose OS or service domain for device drivers, management, or orchestration  
755 functions.

### 756 4.2.2.2 In-scope components of the VES

#### 757 4.2.2.2.1 Hypervisor

758 The following Hypervisor components are within the scope of the present document:

- 759       • Hypervisor core kernel or microkernel: This is the heart of the hypervisor, responsible for core functions like  
760 memory and process management.
- 761       • Virtual Machine Monitor (VMM) and guest VM lifecycle controller: The VMM is the component that  
762 executes the guest operating system's instructions and manages the VM's operational state (creation, start, stop,  
763 deletion).
- 764       • Resource scheduling components (CPU, memory, I/O): These modules are responsible for deciding which VM  
765 gets access to which physical resource at any given time.
- 766       • Device Virtualization modules (e.g. disk controllers, SR-IOV support, paravirtualized device drivers, including  
767 any driver components running inside guest operating systems that directly interact with the host hypervisor):  
768 These components abstract the physical hardware and present virtual devices to the guest VMs.
- 769       • Paravirtualization drivers (when applicable): When the hypervisor employs Paravirtualization mechanisms, the  
770 following components are considered in-scope regardless of execution context:
  - 771           - Backend Paravirtualization drivers and services running in the host or privileged domain.
  - 772           - Frontend Paravirtualization drivers provided by the hypervisor manufacturer for installation in guest  
773 domains.
  - 774           - Inter-domain communication mechanisms used for paravirtual device operations (e.g. shared memory  
775 channels, ring buffers, grant tables, virtio queues).

776 - The privileged domain (e.g. Xen Dom0) only to the extent that it hosts hypervisor-critical management  
777 or device backend services.

778 NOTE: Guest-OS-native or third-party Paravirtualization drivers not supplied by the hypervisor manufacturer  
779 remain out of scope. The security boundaries and domain isolation over all inter-domain communication  
780 channels are expected to be enforced by the hypervisor, irrespective of the driver's origin hypervisor-

781 • Logging and auditing subsystems native to the hypervisor: The systems that record events and actions directly  
782 within the hypervisor layer for monitoring and security.

783 • Security update enforcement mechanisms: Tools or processes integrated into the hypervisor to ensure and  
784 enforce security updates.

#### 785 4.2.2.2.2 Management and Orchestration System

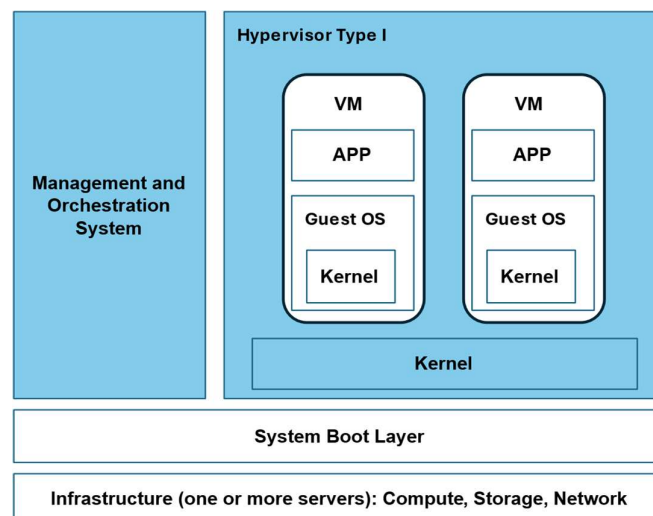
786 The following Management and Orchestration (M&O) components are within the scope of the present document:

- 787 • Management and Orchestration components: This is the system itself, including the main logic for  
788 infrastructure automation and management.
- 789 • Policy enforcement, scheduling, and automation engines used for orchestration
- 790 • Hypervisor management interface or CLI/API (local or remote): This is the external interface through which  
791 administrators or the Orchestration system interact with the Hypervisor. While it provides access to the  
792 hypervisor, it is generally considered part of the overarching M&O or management layer that enables control.

793 Deployment models of M&O system may include:

- 794 • Dedicated management host: The M&O system runs on separate physical hardware used exclusively for  
795 management and administration functions.
- 796 • Management cluster: The M&O system runs as one or more virtual machines on a dedicated cluster reserved  
797 for management and platform services.
- 798 • Virtualized M&O node: The M&O system runs as a virtual machine on the same virtualized infrastructure that  
799 it manages, placed on a host reserved for platform and management services.

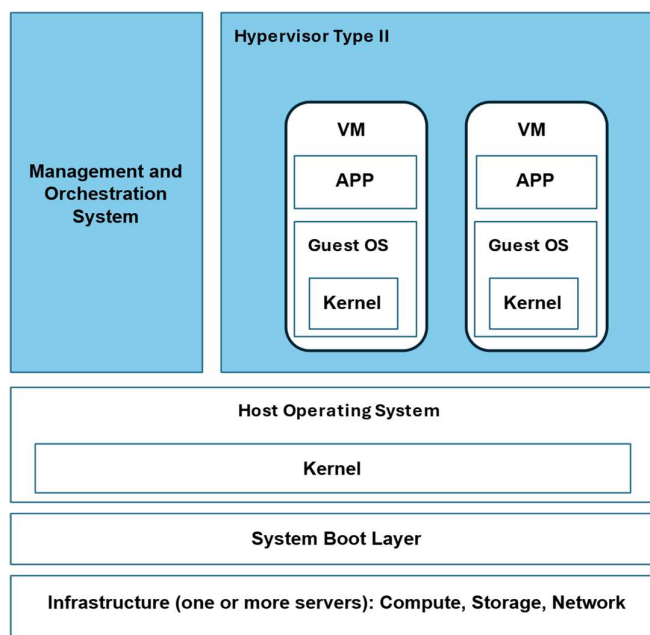
800 Figure 4.1.1.2.2-1 and Figure 4.1.1.2.2-2 highlight in blue the components that are within the scope of the present  
801 document. Components not highlighted in blue are considered out of scope.



802

803

**Figure 4.1.1.2.2-1: VES - Type I Hypervisor components**



804

805

**Figure 4.1.1.2.2-2: VES - Type I Hypervisor components**

#### 806 4.2.2.3 Security-relevant environmental dependencies (out of scope)

807 The following are considered outside the scope of the present document but may be security-relevant environmental  
808 components:

- 809 • The Host Firmware and BIOS/UEFI.
- 810 • The Host Operating System, when it is not bundled as part of the hypervisor.
- 811 • Guest Operating Systems and virtualized application layers, with the exception of paravirtualized drivers that  
812 directly interface with the hypervisor.
- 813 • Virtual network infrastructure beyond the hypervisor switch (e.g. SDN controllers).
- 814 • External logging platforms (e.g. SIEMs).
- 815 • External attestation systems.
- 816 • Hardware-level security controls (TPM, HSM) not embedded in the hypervisor or used to create a virtual  
817 equivalent (vTPM).
- 818 • Identity management systems, secrets vaults, and PKI.
- 819 • Processes, tools, environments, or development/packaging formats whose primary purpose is to develop,  
820 build, or package guest operating system artifacts, including but not limited to OS installation media, virtual  
821 disk images, VM templates, build pipelines, compilers, and related image-building utilities.
- 822 • Management and Orchestration (M&O) System provided independently by a third party.

### 823 4.2.3 CES

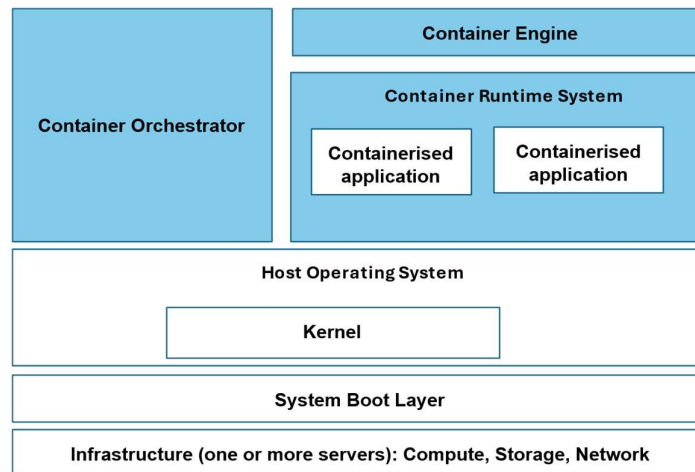
#### 824 4.2.3.1 In-scope components

825 The following components are in the scope of the present document:

- 826 • CRS: it is the lowest-level layer responsible for the execution and isolation of containers:

- 827 - The core binary (e.g. runc, crun).
- 828 - Enforcement modules that manage container isolation, resource control, and sandboxing through OS or  
829 application-level mechanisms (e.g. namespaces, microVMs, seccomp, cgroups).
- 830 - The API and services that manage the runtime and its containers on a host (e.g. containerd, CRI-O).
- 831 • Container Engine: it is the mid-level layer that manage the container lifecycle:
- 832 - The engine and its interfaces for building, pulling, pushing, and running container images (e.g. Docker,  
833 Podman, containerd).
- 834 - Management of container images including the verification a signature at the host level.
- 835 - Management of container networking at the host level (e.g. creation of network interfaces, bridge  
836 networks, and network namespace assignment).
- 837 - Management of volumes for persistent or shared storage.
- 838 - Management of plugins as supported by the engine.
- 839 • Container Orchestrator: it is the top-level layer that manages the entire lifecycle of applications and services  
840 across a cluster of machines (e.g. Kubernetes®):
- 841 - Features that ensure containers are deployed securely.
- 842 - Image signature verification at the cluster level, secure scheduling to trusted nodes, and secrets  
843 management.
- 844 - Policy enforcement that enforces security and isolation policies across the entire cluster (e.g. pod-level  
845 security policies and network isolation rules).
- 846 - Lifecycle management the handles the scaling, networking, load balancing, and self-healing of  
847 containerized applications.

848 Figure 4.1.2.1-1 highlights in blue the components that are within the scope of the present document. Components not  
849 highlighted in blue are considered out of scope.



850

851

**Figure 4.1.2.1-1: CES components**

#### 852 4.2.3.2 Security-relevant environmental dependencies (out of scope)

853 The following are considered outside the scope of the present document but may be security-relevant environmental  
854 components:

- 855 • The Host Operating System on which the CRS executes (OS kernel, drivers).

- 856 • Container registries (public/private).
- 857 • Container image build pipelines (CI/CD Pipelines).
- 858 • Third-party monitoring or security agents installed at runtime.
- 859 • Identity and Access Management systems used for authentication/authorization.
- 860 • Underlying hardware platform (TPM, HSM, hardware root-of-trust).
- 861 • Processes, tools, environments, or development/packaging formats whose primary purpose is to develop,  
862 build, or package container artifacts, including but not limited to build pipelines, CI/CD systems, compilers,  
863 language-specific package managers, container image builders, image layers, and image manifests.
- 864 • Container Engine (CE) and Container Orchestrator (CO) provided independently by a third party.

#### 865 4.2.4 Deployment of M&O, CE, and CO outside the local execution 866 environment

867 The in-scope components defined in the present document may, in some product architectures, be provided outside the  
868 same local execution environment as the rest of the product.

869 For a Virtualization Execution Stack (VES), this applies to the Management and Orchestration (M&O) System.

870 For a Container Execution Stack (CES), this applies to the Container Engine (CE) and the Container Orchestrator (CO).

871 Where such M&O, CE, or CO functionality is part of the declared product functionality and is developed by the  
872 manufacturer, or under the responsibility of the manufacturer, it is considered in scope under the present document.

873 Where such M&O, CE, or CO functionality is provided independently by a third party and is not part of the declared  
874 product functionality under the responsibility of the manufacturer, it is considered an environmental dependency and is  
875 out of scope under the present document.

876 Accordingly, a conformance claim under the present document applies only to M&O, CE, or CO functionality that is  
877 part of the declared product functionality and is developed by the manufacturer, or under the responsibility of the  
878 manufacturer. No conformance claim under the present document is made for independently provided third-party  
879 services.

### 880 4.3 Product variants and conforming products

#### 881 4.3.1 Definition of conforming products

882 The present document covers products implementing Virtualization or container execution functionalities as described  
883 in clauses 4.2.2 and 4.2.3.

884 A conforming product includes at least one of the following core components:

- 885 • the Hypervisor, for products implementing a Virtualization Execution Stack (VES); or
- 886 • the Container Runtime System (CRS), for products implementing a Container Execution Stack (CES).

887 The Hypervisor and CRS represent the essential execution environments that enable the isolation, management, and  
888 execution of virtualized or containerized workloads. They constitute the core mandatory components covered by the  
889 present document.

890 Other components described in clauses 4.2.2 and 4.2.3 such as the Management and Orchestration (M&O) System,  
891 Container Engine (CE), or Container Orchestrator (CO), are within the scope of the present document if they are part of  
892 the product.

893 Products that do not include at least a Hypervisor or a Container Runtime System (CRS) are not considered  
894 Virtualization Execution Stacks or Container Execution Stacks within the meaning of the present document and  
895 therefore fall outside its scope.

## 896 4.3.2 Product variants

897 Products conforming to the present document may therefore be implemented in one or more of the following variants:

- 898 • Hypervisor only: A product implementing solely the hypervisor component.
- 899 • Hypervisor + M&O System: A product integrating both the hypervisor and its management and orchestration  
900 components.
- 901 • Container Runtime System (CRS) only: A product implementing the container execution environment without  
902 including the container engine or orchestrator.
- 903 • CRS + Container Engine (CE): A product combining the low-level runtime with the mid-level engine  
904 responsible for image handling and lifecycle management.
- 905 • CRS + Container Engine + Container Orchestrator (CO): A product integrating all three layers of the container  
906 execution stack.
- 907 • CRS + Container Orchestrator (CO): a product integrating the runtime directly with an orchestrator that  
908 manages container scheduling, scaling and policy enforcement.
- 909 • CRS + integrated CO with CE: a product integrating the runtime with an orchestrator that natively provides  
910 container-lifecycle and image-management functions.

911 Products that do not include at least a Hypervisor or a Container Runtime System (CRS) are not considered  
912 Virtualization or Container Execution Stacks within the meaning of the present document and therefore fall outside its  
913 scope.

## 914 4.4 Operational environment

### 915 4.4.1 General Principles

916 Based on the out-of-scope elements identified for both the VES and the CES in clauses 4.2.2.3 and 4.2.3.2, the  
917 Operational Environment (OE) comprises components that are not part of the VES/CES but are essential to maintain the  
918 security of the in-scope components.

919 The manufacturer's responsibility under the present document is limited to implementing the security requirements  
920 applicable to the in-scope components defined in clauses 4.2.2.2 and 4.2.3.1.

921 However, for the VES/CES to operate securely and as intended, the OE components should function in a manner that  
922 supports and preserves the security of the product.

923 The following security objectives for the Operational Environment specify the expected security properties of each OE  
924 component.

925 Ensuring that these objectives are met is the responsibility of the product or system integrator.

926 Compliance of the OE components with the CRA for example using the relevant harmonised standards, is considered a  
927 precondition for secure operation.

### 928 4.4.2 Security Objectives for the Operational Environment with 929 Requirement Traceability

#### 930 1. Boot Manager

- 931 • The OE should verify the integrity and authenticity of all firmware and boot components before transferring  
932 control to the hypervisor or CRS.
- 933 • The OE should enforce a chain of trust.
- 934 • The OE should block boot if signature or hash verification fails.

**935 2. Host Operating System**

- 936 • The OE should enforce strict process, memory, and I/O isolation between all processes.
- 937 • The OE should support software-assisted isolation.
- 938 • The OE should provide syscall filtering, namespace isolation, and capability bounding to reduce kernel attack  
939 surface.
- 940 • The OE should expose a secure entropy source to the CRS and Hypervisor.

**941 3. Virtual or Physical Network Infrastructure (including SDN)**

- 942 • The OE should segregate management, control, and data traffic using logical or physical separation.
- 943 • The OE should protect management channels with encryption and mutual authentication.
- 944 • The OE should enforce ACLs and rate limits to block Unauthorized cross-plane traffic and DoS attempts.
- 945 • The OE should prevent MAC/IP spoofing and enforce integrity of network identities.
- 946 • The OE should provide accurate, authenticated time synchronization for distributed components.

**947 4. Identity and Access Management / PKI Services**

- 948 • The OE should issue, rotate, and revoke credentials for administrators and services.
- 949 • The OE should validate certificates against a trusted chain and enforce revocation checks.
- 950 • The OE should store private keys in protected hardware or secure software vaults.
- 951 • The OE should record certificate lifecycle events for auditing.

**952 5. Image Registry or Artifact Repository**

- 953 • The OE should verify digital signatures of all stored or pulled images.
- 954 • The OE should restrict registry access to authenticated and authorized users.

**955 6. External Logging and SIEM Systems**

- 956 • The OE should collect and store logs through authenticated and encrypted channels.
- 957 • The OE should preserve log integrity using signing or hashing mechanisms.
- 958 • The OE should restrict log access to authorized entities.
- 959 • The OE should correlate and alert on critical security events.
- 960 • The OE should synchronize event timestamps with a trusted time source.

**961 7. External Attestation System**

- 962 • The OE should validate attestation evidence and verify trust measurements.
- 963 • The OE should protect verifier credentials and trust anchors from compromise.
- 964 • The OE should detect replayed or stale attestation data.
- 965 • The OE should maintain logs of attestation requests and results.

**966 8. Trust Anchors**

- 967 • The OE should generate and protect cryptographic keys.
- 968 • The OE should expose secure APIs for key usage and measurement retrieval.

- 969       • The OE should provide measurement and reporting for trusted boot sequences.

## 970 **9. Firewall / Intrusion Detection or Prevention System**

- 971       • The OE should restrict external access to management and control interfaces.
- 972       • The OE should enforce segmentation between management, control, and data planes.
- 973       • The OE should detect and block anomalous or malicious network traffic.
- 974       • The OE should log all blocked events.

## 975 **10. Storage Systems**

- 976       • The OE should encrypt data at rest including VM or container images, configuration files, and logs.
- 977       • The OE should protect encryption keys in secure key management systems.
- 978       • The OE should securely erase deleted data to prevent recovery.

## 979 **4.4.3 Reference to Relevant CRA Harmonised Standards**

980 Compliance of the Operational Environment components with the CRA essential requirements may be presumed when  
981 the corresponding harmonised standards listed below are applied.

982 Where no harmonised standard yet exists, the system or product integrator should ensure equivalent security measures.

983 **Table 4.4.3-1: Relevant CRA harmonised standards for OE components**

OE Component	Relevant CRA Reference
Boot Manager	ETSI EN 304 623 [i.4] (CRA for Boot Manager)
Host Operating System	ETSI EN 304 626 [i.5] (CRA for Operating System)
Network Infrastructure (Virtual / Physical / SDN)	ETSI EN 304 625 [i.6] (CRA Network Interfaces)
Identity & Access Management / PKI	ETSI EN 304 624 [i.7] (CRA PKI)
External Logging / SIEM	ETSI EN 304 622 [i.8] (CRA SIEM)
Firewall / Intrusion Prevention System	ETSI EN 304 636 [i.9] (CRA Network Security Appliances)

## 984 **4.5 Risk-Based requirement classes/categorization and** 985 **application**

### 986 **4.5.1 Requirement Classes, Categorization and Security Profiles**

#### 987 **4.5.1.1 Definition of Requirement Classes: Basic, Elevated, Advanced**

988 To enable proportional application of security requirements under the CRA, requirements in the present document are  
989 expressed with requirement Classes: Basic, Elevated and Advanced. These levels indicate the depth or strength of the  
990 requirement.

#### 991 **Basic (B)**

- 992       • Definition: The minimum class of requirement, providing fundamental protection.
- 993       • Characteristics:
- 994       - Typically software-based or configuration safeguards.
- 995       - Establishes a baseline protection expected in all products.
- 996       - Suitable for low-risk environments.

#### 997 **Elevated (E)**

- 998       • Definition: Intermediate class of protection, addressing increased exposure.

- 999 • Characteristics:
- 1000 - Builds upon or replaces Basic with stronger mechanisms.
- 1001 - Leverages hardware-assisted features or stronger cryptographic protections.
- 1002 - Suitable for medium-risk environments.

#### 1003 **Advanced (A)**

- 1004 • Definition: The highest class of protection, incorporating the strongest available safeguards.
- 1005 • Characteristics:
- 1006 - Provides resilience against advanced threats.
- 1007 - Incorporates hardware roots of trust and runtime protections.
- 1008 - Suitable for high-risk environments.

### 1009 4.5.1.2 Security Profile Definitions

1010 To support risk-proportional application of security requirements under the CRA, the present document defines three  
 1011 Security Profiles (SP) for security requirements. The defined SPs support manufacturers in applying a risk-based  
 1012 approach, helping to determine applicable requirements aligned with the intended use case and proportionality  
 1013 principles under the CRA.

1014

**Table 4.5.1.2-1: Security profile definitions**

Level	Description
<b>SP 1</b>	A minimum set of baseline cybersecurity requirements applicable to all products. These requirements are essential for fundamental security posture, particularly for use cases where compromise results in low risk (e.g. UC-V1, UC-C1). This level corresponds to products assessed as Low Risk according to the methodology in clause B.
<b>SP 2</b>	Requirements for medium-risk environments. Depending on the requirement categories defined in clause 4.5.1.3: <ul style="list-style-type: none"> <li>i) cumulative requirements include all Basic + Elevated elements;</li> <li>ii) exclusive requirements apply at the Elevated level only;</li> <li>iii) single-class requirements at Elevated are tied to this SP. This corresponds to products assessed as Medium Risk in clause B.</li> </ul>
<b>SP 3</b>	Requirements for high-risk environments. Depending on the requirement categories defined in clause 4.5.1.3: <ul style="list-style-type: none"> <li>i) cumulative requirements include Basic + Elevated + Advanced;</li> <li>ii) exclusive requirements apply at the Advanced level only;</li> <li>iii) single-class requirements at Advanced are tied to this SP. This corresponds to products assessed as High Risk in clause B.</li> </ul>

1015

### 1016 4.5.1.3 Requirement Categorization

1017 The requirements in the present document are defined using three categories: Cumulative, Exclusive, and Single-Class.  
 1018 Each category specifies how a requirement applies across the different classes (Basic, Elevated, Advanced). The  
 1019 following subclauses describe these categories, providing guidance to implementers on how to apply them.

#### 1020 **(1) Cumulative Requirements (Additive)**

- 1021 • Definition: Higher classes include all lower classes. Each class adds more rigor on top of the previous one.
- 1022 • Rationale: This model applies when protections build in layers and depend on the foundation provided by  
 1023 lower classes.
- 1024 • Application: If a manufacturer is assigned SP 3, then all requirements at Basic, Elevated, and Advanced are  
 1025 applicable.
- 1026 • Example - VM Isolation:

- 1027 - Basic: CPU and memory isolation using software-based controls.
- 1028 - Elevated: Basic + hardware-assisted Virtualization and IOMMU separation.
- 1029 - Advanced: Elevated + mitigations for side-channel attacks.

### 1030 (2) Exclusive Requirements (Non-Additive)

- 1031 • Definition: Only one class applies. Classes represent alternative approaches, not cumulative layers.
- 1032 • Rationale: This model applies when requirements define mutually exclusive mechanisms, where implementing one makes the others unnecessary.
- 1033
- 1034 • Application: The manufacturer implements only the requirement at the class corresponding to the product's
- 1035 assigned SP.
- 1036 • Example - Image Integrity Verification:
- 1037 - Basic: Hash verification of VM/container images before execution.
- 1038 - Elevated: Signature verification of images against a trusted public key.
- 1039 - Advanced: Signature verification with keys stored in a hardware root of trust.

### 1040 (3) Single-Class Requirements

- 1041 • Definition: A requirement is defined at only one class (Basic, Elevated, or Advanced). No variations exist at
- 1042 the other classes.
- 1043 • Rationale: Some controls cannot be meaningfully scaled across classes and are best expressed at a single
- 1044 strength level.
- 1045 • Application: A Single-Class requirement is tied to the SP corresponding to its class:
- 1046 - Basic: SP 1.
- 1047 - Elevated: SP 2.
- 1048 - Advanced: SP 3.
- 1049 The manufacturer implements the requirement only when the product is assigned to the SP that matches the
- 1050 class.
- 1051 • Example:
- 1052 - Advanced-only (SP 3): Remote attestation of the hypervisor's runtime state.

### 1053 4.5.1.4 Guidance for Manufacturers

1054 To identify applicable requirements:

- 1055 1) Determine product use case and risk level (see clauses 4.7 and B).
- 1056 2) Identify requirement categorization:
- 1057 - If cumulative, implement all lower classes up to the assigned SP.
- 1058 - If exclusive, implement only the class that matches the assigned SP.
- 1059 - If single-class, implement the one requirement defined at that class.

## 1060 4.5.2 Requirement Application

### 1061 4.5.2.1 Approaches for determining security requirements

1062 The present document defines an SP-based approach to identify the applicable security requirements. This approach is  
1063 grounded in risk assessment and aligned with the CRA [i.1] proportionality principle.

1064 The SP-based approach is applicable where the product aligns with one of the representative use cases defined in clause  
1065 4.7.

1066 The representative use cases defined in clause 4.7 are representative and cover the most common product context  
1067 scenarios addressed by the present document. A product context that is not explicitly defined in clause 4.7 may be  
1068 assigned to the closest representative use case, provided that:

- 1069 • the product context has an equivalent or higher security risk profile than the representative use case to which it  
1070 is assigned;
- 1071 • the assignment is justified by the manufacturer in the conformity assessment documentation; and
- 1072 • the assignment does not reduce the applicable security requirements or assessment expectations under the  
1073 present document.

### 1074 4.5.2.2 SP-Based Requirement Application

1075 This approach provides manufacturers with predefined use cases mapped to Security Profiles (SPs). Each SP bundles  
1076 the appropriate requirements for the assessed risk level (Low, Medium, High).

1077 The representative use cases defined in clause 4.7 are representative product contexts used to support risk-based  
1078 alignment of requirements.

#### 1079 **Step 1 - Determine product use case and associated risk level**

1080 Manufacturers begin by identifying the use case for their product (see reference cases in clause 4.7). The risk level for  
1081 these use cases is pre-determined based on the application of the methodology in clause B:

- 1082 • If the product aligns with UC-V1 or UC-C1 → corresponds to a Low-risk use case.
- 1083 • If the product aligns with UC-V2 or UC-C2 → corresponds to a Medium-risk use case.
- 1084 • If the product aligns with UC-V3 or UC-C3 → corresponds to a High-risk use case.

#### 1085 **Step 2 - Map use case and determined risk level to SP and apply requirements**

1086 **Table 4.5.2.2-1: Security profile definitions**

Use Case	Risk Level	SP 1	SP 2	SP 3	Applicable requirements
UC-V1 / UC-C1	Low	x			Implement all requirements marked Basic, including cumulative, exclusive, and single-class Basic.
UC-V2 / UC-C2	Medium		x		Implement all requirements marked Basic and Elevated, including cumulative and exclusive Elevated, and single-class Elevated.
UC-V3 / UC-C3	High			x	Implement all requirements marked Basic, Elevated, and Advanced, including cumulative and exclusive Advanced, and single-class Advanced.

1087

## 1088 4.6 Users

### 1089 **Manufacturer**

1090 The organization responsible for design, development, and delivery of the product, including the implementation of its  
1091 security functions. The manufacturer provides the product documentation, security claims, and evidence required for  
1092 assessment.

1093 **Product Integrator**

1094 The organization responsible for deploying, configuring, or combining the product into a larger system or operational  
1095 environment. The product integrator ensures that:

- 1096 • Deployment meets the manufacturer's documented security configuration requirements.
- 1097 • Required platform capabilities, hardware features, and external trust anchors are present.
- 1098 • The product is operated in accordance with its secure deployment guidance.

1099 **4.7 Use cases**

1100 **4.7.1 Purpose**

1101 This clause defines representative use cases for VESs and CESs. These use cases describe typical deployment contexts  
1102 that are used in the present document to determine the applicability of security requirements.

1103 The use cases defined in this clause provide a structured basis for applying the requirements of the present document to  
1104 common deployment scenarios.

1105 The use cases defined in the present document are representative and cover the most common product context scenarios  
1106 addressed by the present document. A product context that is not explicitly defined may be assigned to the closest  
1107 representative use case, provided that:

- 1108 • the assignment is duly justified,
- 1109 • the assigned use case reflects an equivalent or higher security risk profile, and
- 1110 • the assignment does not result in a reduction of the applicable security requirements or assessment  
1111 expectations.

1112

1113 NOTE: Closest representative use case means the representative use case that most closely matches the product  
1114 context in terms of threat exposure, risk assessment, and associated security profile

1115 **4.7.2 Use Cases for VES**

1116 The following VES use cases define representative deployment contexts used in the present document to structure  
1117 requirement applicability.

1118 **UC-V1: VES for isolated, non-critical functions**

- 1119 • Description: A VES deployed on a dedicated host in an isolated environment. The host hosts a very small  
1120 number of static VMs dedicated to non-critical functions. In this context, non-critical functions refer to  
1121 operations whose compromise would not result in harm to human safety, disruption of critical infrastructure  
1122 services, or exposure of sensitive data belonging to multiple users.
- 1123 • Example environments:
  - 1124 - Standalone servers or industrial PCs deployed in physically isolated networks.
  - 1125 - Edge devices or gateways with highly restricted links.
  - 1126 - Personal computers or laptops running a local hypervisor in a non-production environment.
- 1127 • Illustrative end user business applications:
  - 1128 - Isolated diagnostic data collection points in a factory.
  - 1129 - Environmental monitoring systems for non-critical building zones.
  - 1130 - Local telemetry units for non-essential parameters in a grid.
  - 1131 - Read-only public information displays in industrial facilities (e.g. current temperature, non-operational  
1132 metrics).

- 1133 - Consumer devices such as personal computers or laptops running local virtual machines for hobbyist  
1134 testing, educational purposes, or personal experimentation.
- 1135 • Risk level: Low (see clause B.7.2).
- 1136 • Main identified risks: These include malware exploiting unpatched systems in isolated industrial networks;  
1137 Unauthorized physical access to devices resulting in data corruption; and misconfigured hypervisors with  
1138 default administrative credentials discovered in industrial telemetry units.
- 1139 **UC-V2: VES for general business workloads**
- 1140 • Description: A VES deployed within a shared multi-tenant environment (private, public, or hybrid cloud),  
1141 hosting a heterogeneous mix of virtualized workloads (VMs) for various customers or independent internal  
1142 units. The platform's primary goal is achieving resource efficiency and providing standard logical separation.  
1143 This use case also includes deployments where the VES runs on connected edge hardware hosting multiple  
1144 VMs (including legacy or specialized workloads) that require stronger isolation.
- 1145 • Example environments:
- 1146 - Enterprise private cloud platforms shared across multiple departments.
- 1147 - Public Cloud multi-tenant IaaS/PaaS used for non-critical services.
- 1148 - On-premises Virtualization clusters for internal workloads.
- 1149 - Internal development and test environments.
- 1150 - Industrial PCs or edge gateways hosting multiple VM workloads (e.g. legacy HMI + modern data  
1151 gateway).
- 1152 - uCPE (universal Customer Premises Equipment) devices running virtualized network appliances (SD-  
1153 WAN, firewall, router).
- 1154 - Edge servers in transportation or building systems.
- 1155 • Illustrative end user business applications:
- 1156 - Corporate IT service platforms.
- 1157 - Enterprise middleware supporting internal operations (e.g. ERP, HR, finance systems).
- 1158 - Internal microservices for business applications.
- 1159 - Educational and research platforms.
- 1160 - Private banking or insurance back-end systems running on controlled virtualized infrastructures.
- 1161 - Virtualized network functions at branch/edge sites (firewall, router, SD-WAN).
- 1162 • Risk level: Medium (see clause B.7.2).
- 1163 • Main identified risks: These include VM escapes exploiting vulnerabilities in Virtualization platforms;  
1164 privilege escalation attacks enabling host-level access; Unauthorized access to sensitive corporate data through  
1165 misconfiguration or credential compromise; and exploitation of known vulnerabilities in virtualized  
1166 management interfaces causing service outages or data breaches. Edge environments face higher exposure and  
1167 less continuous supervision.

### 1168 UC-V3: VES for critical workloads

- 1169 • Description: A high-assurance VES forming the foundation of infrastructures that support critical or regulated  
 1170 workloads requiring strong isolation, resource separation, and continuous security assurance. Such  
 1171 environments may host sensitive applications, essential services, or externally sourced workloads whose  
 1172 compromise could lead to major operational disruption, data breaches, or safety impacts. This scenario  
 1173 demands maximum assurance isolation and strong separation of concerns, typically involving untrusted,  
 1174 heterogeneous tenants (distinct organizations or independent units). In certain deployments, these platforms  
 1175 may also operate at connected edge locations, where strong VM isolation is required to separate safety-  
 1176 relevant and non-safety workloads on the same hardware.
- 1177 • Example environments:
- 1178 - Public IaaS/PaaS clouds where tenants are considered untrusted relative to one another.
  - 1179 - Private clouds supporting multiple departments or projects with strong isolation needs.
  - 1180 - Regulated private clouds (e.g. dedicated to a critical sector like energy).
  - 1181 - High-density Virtualization or micro-VM platforms supporting critical or externally validated workloads.
  - 1182 - Connected edge systems requiring high-assurance separation between operational and support  
 1183 workloads.
- 1184 • Illustrative end user business applications:
- 1185 - Telecom core clouds hosting critical network functions.
  - 1186 - Government or defence virtualized infrastructures processing sensitive or classified data.
  - 1187 - Financial services platforms hosted in public multi-tenant clouds.
  - 1188 - eHealth Platforms handling patient records (PHI).
  - 1189 - Safety-related edge processing workloads where VM isolation protects operational integrity.
- 1190 • Risk level: High (see clause B.7.2).
- 1191 • Main identified risks: These include exploitation of hypervisor vulnerabilities enabling cross-workload or  
 1192 cross-domain compromise; hardware-level side-channel attacks (e.g. Meltdown, Spectre, Foreshadow) leaking  
 1193 sensitive data; denial-of-service attacks targeting shared compute, storage, or network planes; and  
 1194 Unauthorized access through compromised management or orchestration components leading to system-wide  
 1195 outages or data exposure. When deployed at the edge, additional risks include semi-trusted network exposure  
 1196 and limited continuous monitoring.

### 1197 4.7.3 Use Cases for CES

1198 The following CES use cases define representative deployment contexts used in the present document to structure  
 1199 requirement applicability.

#### 1200 UC-C1: CES for isolated, non-critical functions

- 1201 • Description: A CES deployed on a device with limited connectivity. This device typically performs non-  
 1202 critical functions, often in an isolated environment. These applications perform non-critical functions, meaning  
 1203 their compromise would not result in harm to human safety, disruption of critical infrastructure services, or  
 1204 exposure of sensitive data belonging to multiple users.
- 1205 • Example environments:
- 1206 - Isolated test/dev environments.
  - 1207 - Basic controllers for non-critical functions.

- 1208 - Consumer devices such as personal smartphones running isolated app containers for non-critical  
1209 applications or laptops using local container engines (e.g. Docker Desktop, Podman) for personal  
1210 projects.
- 1211 • Illustrative end user business applications:
- 1212 - Educational or hobbyist containerized workloads.
- 1213 - Local telemetry or monitoring for non-critical parameters.
- 1214 - Entertainment or personal productivity apps packaged in containers.
- 1215 • Risk level: Low (see clause B.7.3).
- 1216 • Main identified risks: These include container escape vulnerabilities in container runtimes (e.g. runc);  
1217 incidents of exposed container management daemons without TLS protection (e.g. Docker daemon) in isolated  
1218 environments; and misconfigurations or Unauthorized physical access leading to minor service disruption or  
1219 data corruption.
- 1220 **UC-C2: CES for general business workloads**
- 1221 • Description: A CES deployed within a shared multi-tenant environment (private, public, or hybrid cloud),  
1222 hosting a heterogeneous mix of containerized workloads (e.g. microservices, CI/CD pipelines). The tenants  
1223 may belong to different organizations (e.g. a standard multi-client SaaS platform) or independent units within  
1224 the same organization. The platform's primary goal is resource efficiency, agility, and standard logical  
1225 separation. This use case also includes CES deployments at the network edge where the system runs  
1226 manufacturer-defined or business-relevant workloads, may store or process sensitive data, and is reachable  
1227 over external or semi-trusted networks, but where the consequence of compromise is operational or business  
1228 disruption rather than critical safety failure.
- 1229 • Example environments:
- 1230 - Enterprise private cloud infrastructures shared across multiple departments/teams (multi-tenancy at the  
1231 organizational level).
- 1232 - Public Cloud containerized clusters used for non-critical, general business applications.
- 1233 - On-premises container platforms.
- 1234 - Internal microservices architectures.
- 1235 - Connected edge devices and gateways operating in semi-trusted networks (e.g. industrial IoT gateways,  
1236 smart building controllers, retail analytics devices, agricultural monitoring and control systems).
- 1237 • Illustrative end user business applications:
- 1238 - Corporate IT shared services.
- 1239 - Educational and research platforms.
- 1240 - Retail and e-commerce frontends.
- 1241 - Enterprise middleware platforms.
- 1242 - Back-office systems (HR, general accounting).
- 1243 - Local AI inference and analytics (e.g. video analytics, quality monitoring, sensor fusion) performed at  
1244 the edge before forwarding to cloud services.
- 1245 • Risk level: Medium (see clause B.7.3).

- 1246 • Main identified risks: These include configuration and access-control weaknesses in container orchestration  
 1247 platforms (e.g. Kubernetes); container breakouts through insecure host mount privileges or misconfigured  
 1248 namespaces; privilege escalation within containers or across pods; Unauthorized lateral movement in internal  
 1249 networks; and exploitation of exposed container registries, management APIs, or CI/CD integration points.  
 1250 Edge deployments add risks from semi-trusted networks, weak monitoring, and securing local  
 1251 credentials/updates.

#### 1252 UC-C3: CES for critical workloads

- 1253 • Description: A CES forming the foundation of platforms that support critical or regulated workloads with a  
 1254 consequence of failure classified as high impact. This platform demands high assurance isolation, typically  
 1255 involving dynamic, untrusted containers from heterogeneous tenants (distinct organizations or independent  
 1256 units). This also includes deployments at the edge where the containerized workloads perform safety-relevant,  
 1257 regulatory, or critical operational functions.
- 1258 • Example environments:
- 1259 - Public container platforms hosting high-risk, multi-tenant applications.
  - 1260 - Regulated private container platforms supporting essential services (e.g. energy, telecommunications).
  - 1261 - Federated or cross-organization infrastructures hosting workloads from distinct trust domains.
  - 1262 - Zero-trust environments where container-level isolation is required to be the strongest security boundary.
  - 1263 - Edge deployments supporting critical services (e.g. industrial control gateways, medical data processing  
 1264 appliances, public infrastructure monitoring nodes).
- 1265 • Illustrative end user business applications:
- 1266 - Government or telecom workloads with elevated assurance or compliance requirements.
  - 1267 - Financial and e-identity services operated on high-assurance container infrastructures.
  - 1268 - eHealth services handling patient records (PHI).
  - 1269 - Critical data-analytics or control applications running in shared but tightly isolated environments.
  - 1270 - Real-time safety, monitoring, or control workloads deployed on edge.
- 1271 • Risk level: High (see clause B.7.3).
- 1272 • Main identified risks: These include API server exposures in container orchestration platforms (e.g.  
 1273 Kubernetes®) enabling attackers to take control of clusters; container escape exploits abused in public cloud  
 1274 environments; supply chain attacks targeting public container images; and Unauthorized data access between  
 1275 tenant workloads caused by misconfigured networking or weak isolation. Edge deployments add risks from  
 1276 semi-trusted networks, limited monitoring, and ensuring update/image integrity.

### 1277 4.7.4 Use Case to Risk Mapping

1278 **Table 4.7.4-1: Use case to risk mapping**

Product	Use Case	Description	Risk Level	Clause
VES	UC-V1	VES for isolated, non-critical functions	Low	B.7.2
VES	UC-V2	VES for general business workloads	Medium	B.7.2
VES	UC-V3	VES for critical workloads	High	B.7.2
CES	UC-C1	CES for isolated, non-critical functions	Low	B.7.3
CES	UC-C2	CES for general business workloads	Medium	B.7.3
CES	UC-C3	CES for critical workloads	High	B.7.3

1279

## 1280 5 Technical Requirements for the Products

### 1281 5.1 Applicability of requirements

1282 For each component described in clauses 4.2.2.2 and 4.2.3.1, the corresponding security requirements are defined in the  
1283 subsequent clauses of the present document.

1284 For a product implementing a Virtualization Execution Stack (VES), the product shall include a Hypervisor. The  
1285 requirements and Assessment activities identified for the Hypervisor shall apply. Where the product also includes a  
1286 Management and Orchestration (M&O) System, the corresponding requirements and Assessment activities shall also  
1287 apply.

1288 For a product implementing a Container Execution Stack (CES), the product shall include a Container Runtime System  
1289 (CRS). The requirements and Assessment activities identified for the Container Runtime System (CRS) shall apply.  
1290 Where the product also includes a Container Engine (CE) and/or a Container Orchestrator (CO), the corresponding  
1291 requirements and Assessment activities shall also apply.

1292 Where requirement applicability depends on a use case defined in clause 4.7, the corresponding requirements and  
1293 Assessment activities shall apply for the applicable use case or use cases.

### 1294 5.2 VES Security Requirements

#### 1295 5.2.1 Hypervisor Requirements

##### 1296 5.2.1.1 Isolation

###### 1297 5.2.1.1.1 General

1298 This clause defines isolation requirements across three dimensions: (a) isolation between virtual machines, (b) isolation  
1299 of hypervisor control/administrative planes, and (c) separation of network planes (management, guest, host).

###### 1300 5.2.1.1.2 VM Isolation

1301 <b>Classes are cumulative</b>
------------------------------------

#### 1302 **Basic**

1303 REQ-H-VM-ISO-001: The Hypervisor shall enforce isolation between guest virtual machines (workloads) at CPU,  
1304 memory, I/O, and interrupt-handling levels, ensuring that one guest virtual machine cannot read or modify resources of  
1305 another guest VM, and cannot degrade the availability of another guest VM beyond what is permitted by the  
1306 Hypervisor's configured scheduling and resource-sharing policies.

1307 NOTE: Performance variability caused by normal scheduling and resource sharing (e.g., reduced CPU time under load)  
1308 is acceptable when it remains within the Hypervisor's configured scheduling and resource-sharing policies. Temporary  
1309 depletion of shared or limited resources is addressed under REQ-H-VM-ISO-002.

#### 1310 **Elevated**

1311 REQ-H-VM-ISO-002: Where the Hypervisor exposes shared or limited resources to multiple guest virtual machines  
1312 (e.g., entropy sources, shared device backends, shared I/O services), the Hypervisor shall provide a mechanism to  
1313 prevent a guest virtual machine from causing sustained denial of service to other guest virtual machines through  
1314 resource exhaustion beyond the configured policy, for example through quotas, rate limiting, prioritization, admission  
1315 control, or controlled degraded behaviour.

#### 1316 **Advanced**

1317 REQ-H-VM-ISO-003: The Hypervisor shall strengthen isolation with mitigations against known side-channel attacks to  
1318 ensure robust separation between workloads sharing hardware resources.

1319 REQ-H-VM-ISO-004 (Guest–Host Isolation): The Hypervisor shall enforce that all access by privileged host-  
 1320 environment components to guest memory, guest CPU state, or other security-relevant guest execution state is mediated  
 1321 through defined, auditable guest-host interfaces, and that access outside those interfaces is prevented or detected by the  
 1322 Hypervisor's isolation mechanisms.

1323 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-VM-ISO-001	x	x	x
REQ-H-VM-ISO-002		x	x
REQ-H-VM-ISO-003			x
REQ-H-VM-ISO-004			x

1324 5.2.1.1.3 Control Plane Isolation

1325 **Classes are cumulative**

1326 **Basic**

1327 REQ-H-CP-ISO-001: The Hypervisor shall enforce access-control mechanisms for its administrative functions and  
 1328 interfaces. These mechanisms shall support configurations in which guest virtual machines are prevented from directly  
 1329 accessing or invoking administrative functions and interfaces, and in which only authenticated and authorized entities  
 1330 can perform management actions.

1331 **Elevated**

1332 REQ-H-CP-ISO-002: The Hypervisor shall enforce the use of cryptographically protected communication for network-  
 1333 accessible administrative interfaces, providing confidentiality and integrity protection, and shall support the use of  
 1334 distinct communication channels, ports, or virtual networks to enable isolation of administrative traffic from guest  
 1335 workloads and guest or data-plane traffic.

1336 **Advanced**

1337 REQ-H-CP-ISO-003: The Hypervisor shall enforce the use of cryptographically protected and mutually authenticated  
 1338 communication channels for network-accessible administrative interfaces, in order to reduce the risk of unauthorized  
 1339 access or lateral movement across tenants or system domains.

1340 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CP-ISO-001	x	x	x
REQ-H-CP-ISO-002		x	x
REQ-H-CP-ISO-003			x

1341 5.2.1.1.4 Network Plane Separation

1342 **Classes are cumulative**

1343 **Basic**

1344 REQ-H-NP-ISO-001: The Hypervisor shall maintain logical segregation between the management, guest, and host  
 1345 network planes. Virtual switching, routing, or equivalent network mechanisms shall ensure that traffic from one plane  
 1346 cannot traverse or address another plane by default. These mechanisms shall enforce explicit configuration of separate  
 1347 network contexts for each plane.

1348 **Elevated**

1349 REQ-H-NP-ISO-002: The Hypervisor shall support configuration of explicit traffic filtering and access-control rules  
 1350 between network planes to prevent Unauthorized or unintended cross-plane communication. These rules shall include  
 1351 filtering of packets at the Hypervisor's virtual network layer or at other enforcement points under the Hypervisor's  
 1352 control, ensuring that only management or orchestration traffic explicitly permitted by the configured policy is allowed  
 1353 to cross plane boundaries.

1354 **Advanced**

1355 REQ-H-NP-ISO-003: The Hypervisor shall support isolation of management traffic using dedicated physical interfaces,  
 1356 where such interfaces are available as part of the platform, or cryptographically protected network channels, preventing  
 1357 interception or modification by guest or host workloads. Where dedicated physical interfaces are not used, the  
 1358 cryptographic protection shall provide confidentiality, integrity, and authentication equivalent to a dedicated  
 1359 management network.

1360 NOTE 1: Cryptographically protected network channels may be implemented over logically separated networks  
 1361 (e.g. VLAN, VXLAN, or equivalent), provided that confidentiality, integrity, and authentication are  
 1362 ensured by the cryptographic protection mechanism.

1363 NOTE 2: This requirement does not imply that the Hypervisor itself includes hardware components or cannot be  
 1364 implemented purely in software. It applies to the Hypervisor's capability to use dedicated physical  
 1365 interfaces where such interfaces are available as part of the platform. Where such interfaces are not  
 1366 available, the requirement is addressed through support for cryptographically protected network channels.

#### 1367 Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-NP-ISO-001	x	x	x
REQ-H-NP-ISO-002		x	x
REQ-H-NP-ISO-003			x

### 1368 5.2.1.2 Integrity Protection

#### 1369 5.2.1.2.1 Boot chain integrity verification

##### 1370 **Classes are exclusive**

##### 1371 **Basic**

1372 REQ-H-B-INT-001: The Hypervisor shall implement an integrity verification mechanism for its executable core (kernel  
 1373 or microkernel) based on a trusted reference value.

1374 The Hypervisor shall invoke this integrity verification before initializing any guest workloads or exposing management  
 1375 interfaces. If the integrity verification of the executable core fails, the Hypervisor shall prevent initialization of guest  
 1376 workloads and activation of management interfaces.

##### 1377 **Elevated**

1378 REQ-H-B-INT-002: The Hypervisor shall implement mechanisms to participate in a verifiable chain of trust for the  
 1379 early boot components that load the Hypervisor. Each boot stage delivered as part of the Hypervisor product, including  
 1380 at least any bootloader delivered with the Hypervisor and the Hypervisor executable core, shall validate the integrity  
 1381 and authenticity of the next stage before transferring control.

1382 If this validation fails, the Hypervisor shall prevent initialization of guest workloads and activation of management  
 1383 interfaces.

1384 NOTE 1: For the purposes of REQ-H-B-INT-002, "participate in a verifiable chain of trust" means that the  
 1385 Hypervisor components involved in the boot process are capable of integrating into a platform-provided  
 1386 secure boot chain and enforce integrity and authenticity verification of subsequent stages.

##### 1387 **Advanced**

1388 REQ-H-B-INT-003: The Hypervisor shall rely on integrity and authenticity information for the boot chain on which it  
 1389 depends, as provided by boot mechanisms using hardware or software roots of trust. This information shall cover  
 1390 components used to load the Hypervisor, including the bootloader, the Hypervisor kernel or microkernel, and  
 1391 management services. Where the Hypervisor is provided with the verification status of components in this boot chain, it  
 1392 shall prevent initialization of guest workloads and management services if the verification status of any such component  
 1393 cannot be established or is reported as failed.

1394 NOTE 2: Hardware- or software-based roots of trust, such as verified boot mechanisms or dedicated trust anchors,  
 1395 may be used to implement the validation required by REQ-H-B-INT-003. This requirement does not  
 1396 imply that the Hypervisor itself performs verification of earlier boot phases after startup; it may rely on  
 1397 verification status information made available by the underlying boot mechanism.

1398 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-B-INT-001	x		
REQ-H-B-INT-002		x	
REQ-H-B-INT-003			x

1399 **5.2.1.2.2 Guest VM image integrity verification**1400 **Classes are exclusive**1401 **Basic**

1402 REQ-H-IMG-INT-001: The Hypervisor shall implement an integrity verification mechanism for guest VM images  
1403 based on a trusted reference value.

1404 The Hypervisor shall invoke this integrity verification automatically before executing or provisioning any guest VM. If  
1405 the integrity verification of a guest VM image fails, the Hypervisor shall prevent execution or instantiation of that guest  
1406 VM.

1407 **Elevated**

1408 REQ-H-IMG-INT-002: The Hypervisor shall implement mechanisms to verify both the integrity and the authenticity of  
1409 guest VM images before execution or provisioning. This verification shall establish that the image presented for  
1410 execution or provisioning originates from a trusted source and has not been altered since it was approved for  
1411 deployment.

1412 The Hypervisor shall prevent execution or instantiation of any guest VM image whose integrity or authenticity cannot  
1413 be successfully verified.

1414 NOTE 1: Authenticity verification may be implemented using public-key digital signatures validated through  
1415 certificate chains rooted in trusted authorities, or other cryptographic schemes that provide a verifiable  
1416 binding between the guest VM image and a trusted source and that prevent undetected modification of the  
1417 image, provided that equivalent assurance is achieved.

1418 NOTE 2: This requirement applies to the guest VM image as presented at the time of execution or provisioning,  
1419 including any redeployment or re-instantiation of the image.

1420 **Advanced**

1421 REQ-H-IMG-INT-003: The Hypervisor shall use integrity and authenticity verification for all guest VM images before  
1422 execution or provisioning, based on trust anchors stored as protected trust material. The protected trust material shall  
1423 not be modifiable during normal operational state.

1424 If verification of the integrity or authenticity of a guest VM image using the protected trust material fails, or if the  
1425 verification status of the image cannot be established, the Hypervisor shall prevent execution or instantiation of that  
1426 guest VM.

1427 NOTE 3: Protected trust material refers to keys, certificates or other trust anchors that are stored and managed so  
1428 that they cannot be altered through Unauthorized administrative or guest-facing interfaces and can only  
1429 be updated by authenticated and authorized administrators with appropriate privileges. Such updates are  
1430 recorded in security-relevant logs sufficient to support security auditing.

1431 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-IMG-INT-001	x		
REQ-H-IMG-INT-002		x	
REQ-H-IMG-INT-003			x

1432 **5.2.1.2.3 Runtime integrity protection**1433 **Classes are cumulative**1434 **Elevated**

1435 REQ-H-RP-INT-002: The Hypervisor shall enforce integrity protection for its configuration and control plane by  
 1436 ensuring that only authenticated and authorized modifications can be applied. The Hypervisor shall generate audit  
 1437 events for any security-relevant changes to configuration or internal control logic.

1438 **Advanced**

1439 REQ-H-RP-INT-003: The Hypervisor shall maintain the integrity of its security-critical runtime components and detect  
 1440 unauthorized modification of configuration state or control logic used to enforce security policies or isolation.

1441 The Hypervisor shall maintain a trusted baseline of the integrity state of these security-critical runtime components and  
 1442 shall perform periodic or continuous integrity validation against this baseline.

1443 Upon detecting a deviation from the trusted baseline, the Hypervisor shall automatically trigger protective actions, such  
 1444 as isolating the affected component, disabling affected functions, or restoring the component from a trusted state, and  
 1445 shall generate a security alert that cannot be suppressed by the affected component.

1446 NOTE: Runtime integrity validation may be implemented using hardware-assisted mechanisms, software-based  
 1447 mechanisms, or a combination of both, provided that the mechanisms are able to detect unauthorized  
 1448 modification of the security-critical runtime components defined by this requirement.

1449 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-RP-INT-002		X	X
REQ-H-RP-INT-003			X

1450 5.2.1.2.4 Remote attestation

1451 **Single Class**

1452 **Advanced**

1453 REQ-H-RA-INT-003: The Hypervisor may support measured boot and generation of attestation evidence describing its  
 1454 bootloader, kernel or microkernel, security-critical configuration, and essential runtime state.

1455 This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the Hypervisor  
 1456 instance for which the measurements were produced.

1457 The Hypervisor shall protect attestation keys and measurement data against Unauthorized access, disclosure, or replay.

1458 The attestation evidence should be limited to the minimum information necessary to verify the integrity and  
 1459 configuration state of the Hypervisor.

1460 NOTE 1: This requirement covers the Hypervisor capability to collect measurements, bind them cryptographically,  
 1461 and expose them as attestation evidence. The design and operation of local or remote verifiers, attestation  
 1462 protocols, and relying parties is outside the scope of the present document.

1463 REQ-H-RA-INT-004: Where attestation evidence produced by the Hypervisor is intended to be used in remote  
 1464 verification scenarios, the Hypervisor should support configuration of privacy-preserving attestation options that reduce  
 1465 linkability between attestation events while preserving the ability of authorized verifiers to assess the integrity and  
 1466 configuration state of the Hypervisor.

1467 NOTE 2: Privacy-preserving options may include, for example, use of pseudonymous identifiers, separation of  
 1468 identity and integrity information, or mechanisms for controlling the frequency and granularity of  
 1469 attestation.

1470 NOTE 3: Remote attestation is optional in the Advanced class. A product may conform to Advanced without  
 1471 supporting remote attestation. If remote attestation is not supported, this limitation shall be documented in  
 1472 the product security documentation.

1473 **Requirement applicability by use case and security profile**

1474 NOTE 4: In the following table, “o” indicates an optional capability within UC-V3 / SP3. Absence of support does  
 1475 not preclude Advanced-class conformance.

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
--------------	-------------	-------------	-------------

REQ-H-RA-INT-003			o
REQ-H-RA-INT-004			o

### 1476 5.2.1.3 Authentication

#### 1477 5.2.1.3.1 General

1478 This clause distinguishes between authentication of administrative users and authentication of external services that  
1479 access Hypervisor management functions or data.

#### 1480 5.2.1.3.2 Administrative Authentication

##### 1481 **Classes are exclusive**

##### 1482 **Basic**

1483 REQ-H-ADMIN-AUTH-001: Where the Hypervisor exposes administrative interfaces (including local command-line  
1484 interfaces, remote application programming interfaces, and virtual machine management consoles), the Hypervisor shall  
1485 require authentication for access to those interfaces.

1486 The Hypervisor shall enforce the use of unique, non-default credentials for administrative accounts. Where password-  
1487 based authentication is supported, the Hypervisor shall enforce password complexity requirements for administrative  
1488 authentication credentials and a mechanism limiting repeated failed authentication attempts, with configurable  
1489 parameters.

##### 1490 **Elevated**

1491 REQ-H-ADMIN-AUTH-002: Where the Hypervisor exposes network-accessible administrative interfaces, the  
1492 Hypervisor shall support cryptographic key-based authentication for administrative access and shall allow password-  
1493 based login to be disabled for those interfaces.

- 1494 • Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].
- 1495 • Private keys used for administrative authentication shall be stored and managed in a manner that prevents  
1496 Unauthorized extraction or reuse by guest workloads or other unprivileged components.

##### 1497 **Advanced**

1498 REQ-H-ADMIN-AUTH-003: Where the Hypervisor exposes administrative interfaces, the Hypervisor shall require  
1499 strong authentication (as defined in clause 3.1) for all administrative interfaces.

1500 Where certificate-based administrative authentication is used, certificate validity, expiration, and revocation status shall  
1501 be verified before administrative access is granted.

##### 1502 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-ADMIN-AUTH-001	x		
REQ-H-ADMIN-AUTH-002		x	
REQ-H-ADMIN-AUTH-003			x

### 1503 5.2.1.3.3 Service Authentication

##### 1504 **Classes are exclusive**

##### 1505 **Basic**

1506 REQ-H-SERV-AUTH-001: Where the Hypervisor exposes interfaces for external services to access management  
1507 functions or data, the Hypervisor shall require authentication for all external services, including orchestration systems,  
1508 monitoring tools, and backup agents, before granting access to management functions or data.

1509 The Hypervisor shall enforce the use of distinct credentials for each service account and shall not allow built-in shared  
1510 or hard-coded service credentials.

1511 The Hypervisor shall implement an account lockout mechanism, or an equivalent mechanism that limits repeated failed  
 1512 authentication attempts for service accounts, after a configurable number of consecutive failed authentication attempts.  
 1513 Where an account lockout mechanism is used, the Hypervisor shall support either administrative unlock or automatic  
 1514 unlock after a configurable delay.

1515 **Elevated**

1516 REQ-H-SERV-AUTH-002: Where the Hypervisor exposes interfaces for external services to access management  
 1517 functions or data, the Hypervisor shall require authenticated access for all external services and shall support certificate-  
 1518 based mutual authentication for services accessing management functions or data.

1519 Certificates used for service authentication shall be verified for validity and expiration before access is granted.

1520 

- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

1521 

- Private keys used for service authentication shall be stored and managed in a manner that prevents  
 1522 Unauthorized extraction or reuse by guest workloads or other unprivileged components.

1523 **Advanced**

1524 REQ-H-SERV-AUTH-003: Where the Hypervisor exposes interfaces for external services to access management  
 1525 functions or data, the Hypervisor shall enforce certificate-based mutual authentication for all service interactions that  
 1526 access management functions or data, with certificate paths validated against a trusted public key infrastructure.

1527 Certificates used for service authentication shall be bound to specific service identities and shall be verified for validity,  
 1528 expiration, and revocation status before access is granted.

1529 

- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

1530 

- Private keys used for service authentication shall be stored and managed in a manner that prevents  
 1531 Unauthorized extraction or reuse by guest workloads or other unprivileged components.

1532 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-SERV-AUTH-001	x		
REQ-H-SERV-AUTH-002		x	
REQ-H-SERV-AUTH-003			x

1533 **5.2.1.4 Authorization**

1534 **5.2.1.4.1 General**

1535 This clause distinguishes between authorization of administrative users and authorization of external services that  
 1536 access Hypervisor management functions or data.

1537 **5.2.1.4.2 Administrative Authorization**

1538 **Classes are exclusive**

1539 **Basic**

1540 REQ-H-ADMIN-AUTHZ-001: Where the Hypervisor exposes administrative interfaces, the hypervisor shall restrict all  
 1541 administrative actions to authenticated accounts and enforce a default-deny policy for access to management functions.

1542 **Elevated**

1543 REQ-H-ADMIN-AUTHZ-002: Where the Hypervisor exposes administrative interfaces, the hypervisor shall enforce  
 1544 granular, role-based access control (RBAC) or attribute-based access control (ABAC) for administrative accounts,  
 1545 ensuring that accounts are granted only the minimum default privileges required for their role.

1546 **Advanced**

1547 REQ-H-ADMIN-AUTHZ-003: Where the Hypervisor exposes administrative interfaces, the hypervisor shall enforce  
 1548 fine-grained authorization policies for administrative accounts, including separation of duties and support for time-

1549 bound or just-in-time privilege elevation, with the capability to integrate with external identity and policy decision  
1550 systems where available.

1551 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-ADMIN-AUTHZ-001	x		
REQ-H-ADMIN-AUTHZ-002		x	
REQ-H-ADMIN-AUTHZ-003			x

1552 **5.2.1.4.3 Service Authorization**

1553 **Classes are exclusive**

1554 **Basic**

1555 REQ-H-SERV-AUTHZ-001: Where the Hypervisor exposes interfaces for external services to access management  
1556 functions or data, the hypervisor shall enforce a default-deny policy for all service accounts, ensuring that service  
1557 accounts are only permitted to perform actions explicitly authorized.

1558 **Elevated**

1559 REQ-H-SERV-AUTHZ-002: Where the Hypervisor exposes interfaces for external services to access management  
1560 functions or data, the hypervisor shall enforce granular authorization for service accounts, binding allowed actions to  
1561 their authenticated identity. Each service account shall be configured with least-privilege permissions by default.

1562 **Advanced**

1563 REQ-H-SERV-AUTHZ-003: Where the Hypervisor exposes interfaces for external services to access management  
1564 functions or data, the hypervisor shall enforce fine-grained, policy-driven authorization for service accounts, binding  
1565 permissions to cryptographic service identities, supporting context-aware enforcement, with the capability to integrate  
1566 with external authorization systems where available.

1567 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-SERV-AUTHZ-001	x		
REQ-H-SERV-AUTHZ-002		x	
REQ-H-SERV-AUTHZ-003			x

1568 **5.2.1.5 Confidentiality Protection**

1569 **Classes are cumulative**

1570 **Basic**

1571 REQ-H-CONF-001 (Access control for sensitive data): The Hypervisor shall provide access-control mechanisms to  
1572 restrict access, through Hypervisor functions and interfaces, to sensitive data associated with the Hypervisor product,  
1573 including guest VM images, snapshot data, configuration files, credentials, and security-relevant logs. Where such data  
1574 is stored, managed, or exposed as part of the product, access shall be permitted only to authenticated and authorized  
1575 entities through controlled interfaces.

1576 NOTE: This requirement applies to access through Hypervisor functions and interfaces to sensitive data that is  
1577 stored, managed, or exposed as part of the product. It does not apply to copies of such data held  
1578 exclusively by out-of-scope environmental components, such as a host operating system or external  
1579 storage or network services.

1580 **Elevated**

1581 REQ-H-CONF-002 (Data at rest): The Hypervisor shall enforce encryption of sensitive data at rest for data that it stores  
1582 or manages, including VM images, snapshot data, configuration files, credentials, and security-relevant logs.

1583 Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by guest  
1584 workloads and other unprivileged components.

1585 REQ-H-CONF-003 (Data in transit): The Hypervisor shall support the use of encrypted communication for  
 1586 management interfaces, for VM migration traffic, and for inter-VM traffic where such traffic is mediated by  
 1587 Hypervisor-controlled virtual networking components.

1588 When these communications are configured to use encrypted transport, the Hypervisor shall prevent fallback to  
 1589 unencrypted or weakly protected channels.

#### 1590 **Advanced**

1591 REQ-H-CONF-004 (Data in use - confidential execution): The Hypervisor shall support provisioning and management  
 1592 of execution environments for guest workloads in which confidential data associated with those workloads is protected,  
 1593 during execution, from observation, inspection, or access through the Hypervisor, the host operating system, and  
 1594 administrative interfaces.

1595 On platforms that provide mechanisms for confidential execution, the Hypervisor shall be able to use these mechanisms  
 1596 to enforce the confidentiality of guest memory and associated processor state.

#### 1597 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CONF-001	x	x	x
REQ-H-CONF-002		x	x
REQ-H-CONF-003		x	x
REQ-H-CONF-004			x

### 1598 5.2.1.6 Availability and Resilience

#### 1599 **Classes are cumulative**

#### 1600 **Basic**

1601 REQ-H-AVAIL-001: The Hypervisor shall detect failures of internal management and control-plane components and  
 1602 shall support recovery of these components without requiring a full host reboot where feasible, in order to maintain  
 1603 availability of running guest virtual machines.

#### 1604 **Elevated**

1605 REQ-H-AVAIL-002: The Hypervisor shall provide an automated recovery mechanism for guest virtual machines  
 1606 affected by host or Hypervisor service failure, such that affected guest virtual machines are automatically returned to an  
 1607 operational state.

1608 NOTE 1: This may be achieved by automatic restart initiated by the Hypervisor, or by integration with an external  
 1609 supervisor/lifecycle manager that performs the restart or recovery action using Hypervisor-provided  
 1610 interfaces.

#### 1611 **Advanced**

1612 REQ-H-AVAIL-003: For selected guest virtual machines, the Hypervisor shall support a continuity mechanism that  
 1613 reduces service interruption compared to a full restart.

1614 NOTE 2: Continuity mechanisms may include checkpoint/restore, warm standby, live migration, or runtime-state  
 1615 replication where supported by the product context.

#### 1616 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-AVAIL-001	x	x	x
REQ-H-AVAIL-002		x	x
REQ-H-AVAIL-003			x

### 1617 5.2.1.7 Logging

#### 1618 **Classes are cumulative**

#### 1619 **Basic**

1620 REQ-H-LOG-001: The Hypervisor shall generate time-stamped audit logs for security-relevant administrative actions  
1621 and security events, including at least authentication attempts, configuration changes, and virtual machine lifecycle  
1622 operations.

1623 Each audit record shall include, as applicable to the event type, an event type identifier, the time of the event, the  
1624 outcome of the event, and the identity of the associated user, service, or process.

1625 NOTE: The set of administrative actions and security events that are considered security-relevant may vary  
1626 depending on the Hypervisor implementation and its exposed management interfaces. It is expected that  
1627 the manufacturer identifies, in the technical documentation, which actions/events are treated as security-  
1628 relevant for audit logging, and explains any product-specific additions beyond the minimum set listed  
1629 above.

### 1630 **Elevated**

1631 REQ-H-LOG-002: The Hypervisor shall protect audit logs from unauthorized access, modification, or deletion. Access  
1632 to audit logs shall be restricted to authorized administrative roles or processes.

1633 The Hypervisor shall support secure export of audit logs to external log management or security information and event  
1634 management systems using authenticated and encrypted communication channels.

1635 Where the Hypervisor exposes tenant-visible logs or log-derived information in multi-tenant environments, it shall  
1636 maintain separation between operator audit logs and tenant-visible logs or telemetry.

### 1637 **Advanced**

1638 REQ-H-LOG-003: The Hypervisor shall support cryptographic protection, trusted time-stamping, and append-only or  
1639 equivalent tamper-evident protection of audit logs to enable verification of the origin and integrity of audit records  
1640 during forensic analysis.

### 1641 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-LOG-001	x	x	x
REQ-H-LOG-002		x	x
REQ-H-LOG-003			x

## 1642 5.2.1.8 Secure Update

### 1643 **Classes are cumulative**

#### 1644 **Applicability:**

1645 These requirements apply whenever the Hypervisor component is updated. The update mechanisms and components  
1646 used to deliver and install updates for the Hypervisor, including host operating system update services, management  
1647 appliances, or orchestration-based update pipelines that are part of the evaluated solution, shall collectively satisfy the  
1648 requirements defined in this clause.

1649 Where the Hypervisor is delivered only as part of a larger product image or appliance, the product update mechanism  
1650 shall ensure that Hypervisor binaries and related components are updated in accordance with REQ-H-UPD-001 to REQ-  
1651 H-UPD-003, as applicable.

#### 1652 **Basic**

1653 REQ-H-UPD-001: The Hypervisor shall provide a mechanism to apply security updates to the Hypervisor without  
1654 requiring a full reinstallation of the Hypervisor or host system.

1655 The Hypervisor shall verify the authenticity and integrity of all Hypervisor updates before installation, using digital  
1656 signatures validated against trusted keys or certificates configured for the Hypervisor.

1657 If authenticity or integrity verification of a Hypervisor update fails, the Hypervisor shall prevent installation of that  
1658 update and shall record a security-relevant event.

1659 Keys or certificates used for Hypervisor update verification shall not be modifiable by guest workloads or other  
1660 unprivileged components during normal operation.

1661 **Elevated**

1662 REQ-H-UPD-002: The Hypervisor shall implement rollback protection to prevent unauthorized installation of outdated,  
1663 revoked, or replayed Hypervisor updates.

1664 Authorized rollback to a previous trusted Hypervisor version shall only be permitted when all of the following  
1665 conditions are met:

- 1666 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 1667 • The rollback action is explicitly authorized by an administrator with elevated privileges.
- 1668 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted  
1669 from and the version reverted to.

1670 **Advanced**

1671 REQ-H-UPD-003: The Hypervisor shall support live patching of critical Hypervisor components, including the kernel  
1672 or microkernel, device drivers, and security modules, in order to remediate vulnerabilities without requiring shutdown  
1673 of guest virtual machines and with only minimal interruption to workload execution.

1674 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-UPD-001	x	x	x
REQ-H-UPD-002		x	x
REQ-H-UPD-003			x

1675 **5.2.1.9 Secure Configuration and Default**1676 **Classes are cumulative**1677 **Basic**

1678 REQ-H-CFG-001: The Hypervisor shall, by default, disable remote administrative access pathways and insecure  
1679 management interfaces that are not required for initial provisioning of the Hypervisor product. This includes remote  
1680 shell access without strong authentication, legacy or unencrypted management APIs, and vendor-specific diagnostic  
1681 ports that are not required for secure operation.

1682 The Hypervisor shall provide configuration mechanisms that allow administrators to explicitly enable only those  
1683 interfaces and services that are necessary for the intended Virtualization, orchestration, and core host management  
1684 operations.

1685 **Elevated**

1686 REQ-H-CFG-002: The Hypervisor shall validate configuration parameters before applying them and shall implement  
1687 configuration validation logic that detects and prevents configurations which would disable or bypass security  
1688 mechanisms or violate guarantees for virtual machine isolation, control-plane protection, or minimum resource  
1689 allocations for security and management functions.

1690 When a configuration change is rejected due to a security-related validation failure, the Hypervisor shall provide clear  
1691 feedback to the administrator indicating the reason for rejection.

1692 **Advanced**

1693 REQ-H-CFG-003: The Hypervisor shall support the definition and use of security configuration baselines that specify  
1694 secure settings for critical Hypervisor components, including at least I/O memory management unit configuration,  
1695 virtual switch policies, and host firewall rules, as implemented within the Hypervisor product.

1696 The Hypervisor shall provide capabilities to automatically validate current configuration settings against a selected  
1697 security baseline and shall generate alerts or audit events when non-compliant settings are detected.

1698 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CFG-001	x	x	x

REQ-H-CFG-002		x	x
REQ-H-CFG-003			x

## 1699 5.2.1.10 Data Minimization

1700 **Single Class**1701 **Basic**

1702 REQ-H-DM-001: The Hypervisor shall limit log content, telemetry, and diagnostic data to information necessary for  
1703 operation, troubleshooting, and security monitoring. Logs, telemetry, and diagnostic outputs produced by the  
1704 Hypervisor shall not include plaintext authentication credentials, full cryptographic keys, complete virtual machine  
1705 memory contents, or guest workload payload data unless such inclusion is explicitly enabled for a specific debugging  
1706 purpose by an administrator.

1707 Payload or application data originating from guest workloads shall not be included in Hypervisor logs, telemetry, or  
1708 diagnostic outputs by default. Where the Hypervisor supports inclusion of such payload content for debugging  
1709 purposes, this behaviour shall only be enabled through explicit administrative configuration and shall be clearly  
1710 indicated to the administrator.

1711 REQ-H-DM-002: The Hypervisor shall provide administrative controls to restrict and, where necessary, disable the  
1712 collection of specific categories of log, telemetry, and diagnostic data, including the ability to disable collection of  
1713 particular log types, telemetry streams, metrics, or debug payload capture.

1714 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-DM-001	x	x	x
REQ-H-DM-002	x	x	x

## 1715 5.2.2 Applicability of Hypervisor Requirements

1716 The applicability of the Hypervisor requirements to Type I, Hybrid, and Type II Hypervisor architectures is specified in  
1717 Table 5.2.2-1.

1718 For the purposes of Table 5.2.2-1:

- 1719 1) A (Applicable): the requirement applies to the Hypervisor architecture within the scope of the present  
1720 document.
- 1721 2) C (Conditionally Applicable): the requirement applies where the Hypervisor architecture includes or relies on  
1722 the corresponding function, interface, mechanism, or supporting platform capability.
- 1723 3) N/A (Not Applicable): the requirement does not apply to the Hypervisor architecture within the scope of the  
1724 present document.

1725 Where a requirement is conditionally applicable, applicability shall be determined according to the implemented  
1726 product architecture and declared product configuration. The corresponding conformity assessment shall verify the  
1727 basis on which conditional applicability is claimed.

1728 **Table 5.2.2-1: Applicability of hypervisor requirements**

Requirement ID	Type I	Hybrid	Type II	Rationale
<b>Isolation</b>				
REQ-H-VM-ISO-001	A	A	A	Guest-to-guest separation for CPU, memory, I/O, and interrupt handling is a core virtualization property across Type I, Hybrid, and Type II architectures. Availability effects are interpreted relative to configured scheduling/resource-sharing policies.
REQ-H-VM-ISO-002	A	A	A	Shared/limited resource exhaustion controls apply when such resources are exposed to multiple guests; this situation can arise in all architectures (e.g., shared device backends or entropy sources).
REQ-H-VM-ISO-003	A	A	C	Side-channel mitigations are primarily implemented in the hypervisor/VMM. In Type II deployments, effectiveness can depend on Host OS scheduling and CPU feature exposure, so enforcement is architecture-dependent.
REQ-H-VM-ISO-004	C	C	C	Guest protection from a privileged host environment depends strongly on architecture and platform mechanisms and is typically associated with higher-

				assurance contexts; feasibility and evidence vary across architectures and product contexts.
REQ-H-CP-ISO-001	A	A	A	In all architectures, the management plane of the hypervisor is logically distinct from guest workloads, and access control for administrative interfaces is implemented within the hypervisor's management stack.
REQ-H-CP-ISO-002	A	A	C	Type I and Hybrid hypervisors control the Virtualization layer and associated virtual switching, so they can directly define separate logical pathways for administrative and guest/data traffic. In Type II designs, the hypervisor defines its management endpoints, while separation of traffic often relies on Host OS networking configuration, which leads to conditional applicability.
REQ-H-CP-ISO-003	A	A	A	Cryptographically protected and mutually authenticated communication channels for administrative interfaces are part of the hypervisor's management interface design in Type I, Hybrid, and Type II deployments, even if cryptographic libraries come from the Host OS or platform.
REQ-H-NP-ISO-001	A	A	A	Logical segregation between management, guest, and host network planes is implemented through virtual switches, routing constructs, or equivalent mechanisms in the hypervisor solution across all three architectures.
REQ-H-NP-ISO-002	A	A	A	Filtering and access-control rules between network planes are realized at the virtual switch or equivalent enforcement points that are part of the hypervisor solution in Type I, Hybrid, and Type II designs.
REQ-H-NP-ISO-003	A	A	A	Protection of management traffic using dedicated physical interfaces or cryptographically protected channels is part of the hypervisor's networking and configuration model in all architectures; Type II implementations typically rely on Host OS interfaces but still configure their management traffic accordingly.
<b>Integrity Protection</b>				
REQ-H-B-INT-001	A	C	N/A	In Type I designs, the hypervisor product can perform integrity verification of its executable core before management interfaces and guest workloads become available. In Hybrid designs, the minimal host kernel or equivalent component performs verification anchored in platform firmware, so enforcement depends partly on firmware trust. In Type II architectures, integrity of the executable core is handled by the Host OS boot and code-integrity mechanisms, and the hypervisor process inherits this trust.
REQ-H-B-INT-002	A	C	N/A	For Type I hypervisors, the early boot stages delivered with the product participate in a chain of trust from bootloader to hypervisor core and services. Hybrid designs participate from the bootloader/host-kernel handoff upward while relying on platform firmware for the initial stage. In Type II architectures, the chain of trust from firmware to Host OS kernel is outside the hypervisor product; the hypervisor runs as an application on top of the already established OS trust chain.
REQ-H-B-INT-003	A	C	C	Type I and Hybrid hypervisors can consume verified-boot or measured-boot status from platform roots of trust and make use of this information when deciding whether to start management services and guests. In Type II setups, the hypervisor can rely on verified boot information exposed by the Host OS or platform and, in advanced deployments, can gate VM operations on that status, so this behaviour depends partly on OS/platform capabilities.
REQ-H-IMG-INT-001	A	A	A	Integrity checking of guest VM images (for example, hash comparison against a manifest) before provisioning or execution is implemented within the hypervisor or its management components in all architectures.
REQ-H-IMG-INT-002	A	A	A	Verification of both integrity and authenticity of guest VM images using digital signatures or equivalent cryptographic schemes is performed by the hypervisor layer prior to VM launch, independently of the host type.
REQ-H-IMG-INT-003	A	A	C	For Type I and Hybrid hypervisors, trust anchors for image verification can be stored and protected within the hypervisor or management domain using platform trust services. In Type II environments, the hypervisor typically relies on Host OS keystores, hardware security modules, or OS abstractions to protect trust material, so support depends on the Host OS and platform.
REQ-H-RP-INT-002	A	A	A	Protection of configuration and control-plane state through authenticated, authorized modifications and the generation of audit events for security-relevant changes is handled within the hypervisor management plane and is not tied to a specific hypervisor architecture.
REQ-H-RP-INT-003	A	A	C	Type I and Hybrid hypervisors can maintain a baseline of security-critical runtime components and perform periodic or continuous integrity validation, with direct control over the execution environment. Type II hypervisors can apply similar mechanisms to their own processes and configuration but rely on Host OS facilities for deeper runtime integrity monitoring and some enforcement actions.
REQ-H-RA-INT-003	A	A	C	Type I and Hybrid hypervisors can observe their bootloader, kernel or microkernel, security-critical configuration, and key runtime elements and can generate attestation evidence that reflects this state. In Type II designs, the hypervisor can expose evidence about its own configuration and some runtime state but depends on Host OS and platform attestation frameworks to cover deeper layers, so attestation capability is influenced by the underlying environment.
REQ-H-RA-INT-004	A	A	C	Privacy-preserving choices for attestation (for example, pseudonymous identifiers or separation of identity and integrity information) can be implemented directly in Type I and Hybrid hypervisor attestation components. In Type II environments, the hypervisor influences the privacy properties of its own evidence but often builds on

				Host OS or platform attestation and identity mechanisms, so support depends partly on external components.
<b>Authentication</b>				
REQ-H-ADMIN-AUTH-001	A	A	A	Applies where administrative interfaces are exposed by the hypervisor solution; the management model and exposed interfaces vary across architectures and products.
REQ-H-ADMIN-AUTH-002	A	A	A	Applies where network-accessible administrative interfaces are exposed; cryptographic key-based mechanisms and password-disabling are relevant to those interfaces when present.
REQ-H-ADMIN-AUTH-003	A	A	A	Applies where administrative interfaces are exposed. Certificate-based administrative authentication and certificate validation against a trusted PKI are functions of the hypervisor management plane in all three architectural models, even when the PKI integration uses Host OS services.
REQ-H-SERV-AUTH-001	A	A	A	Applies where interfaces for external services to access management functions/data are exposed; service identity models vary across products.
REQ-H-SERV-AUTH-002	A	A	A	Applies where service access interfaces are exposed and certificate-based mutual authentication is used for those interfaces.
REQ-H-SERV-AUTH-003	A	A	A	Applies where service access interfaces are exposed and certificate-based mutual authentication is used for those interactions.
<b>Authorization</b>				
REQ-H-ADMIN-AUTHZ-001	A	A	A	Applies where administrative interfaces are exposed; default-deny authorization is relevant to those interfaces when present.
REQ-H-ADMIN-AUTHZ-002	A	A	A	Applies where an administrative identity/role model exists for administrative interfaces.
REQ-H-ADMIN-AUTHZ-003	A	A	A	Applies where administrative interfaces and policy enforcement exist; integration capabilities vary across products.
REQ-H-SERV-AUTHZ-001	A	A	A	Applies where service identities exist for management access; default-deny authorization is relevant to those identities when present.
REQ-H-SERV-AUTHZ-002	A	A	A	Applies where service identities are used to authorize management actions; implementations vary by product.
REQ-H-SERV-AUTHZ-003	A	A	A	Applies where cryptographic service identities and policy-driven authorization are used for management access.
<b>Confidentiality Protection</b>				
REQ-H-CONF-001	A	A	A	Access control for sensitive data under the hypervisor's control (such as VM images, snapshots, configuration, credentials, and security logs) is implemented within the management and storage logic of the hypervisor solution, regardless of whether underlying filesystems or storage ACLs are provided by a Host OS.
REQ-H-CONF-002	A	A	C	Protection of sensitive data at rest through encryption is part of the hypervisor solution in all architectures; Type II deployments often rely on Host OS or storage-layer capabilities, but the hypervisor still orchestrates or integrates those mechanisms for its managed data.
REQ-H-CONF-003	A	A	A	Encryption of management, migration, and hypervisor-mediated inter-VM traffic is handled by the hypervisor's networking and session-management design for Type I, Hybrid, and Type II, while the specific transport mechanisms may be provided by the Host OS or bundled components.
REQ-H-CONF-004	A	A	C	Execution environments that protect guest memory and processor state from observation by the hypervisor, host OS, and administrative interfaces rely on platform features such as confidential-computing extensions. Type I and Hybrid hypervisors interact more directly with these platform mechanisms, while Type II hypervisors depend on how the Host OS exposes and manages such capabilities.
<b>Availability and Resilience</b>				
REQ-H-AVAIL-001	A	A	A	Failure detection and recovery for internal management/control-plane components are relevant across all hypervisor architectures. In Type II designs, some recovery behaviour typically depends on Host OS services, but the requirement remains within the hypervisor solution scope through its management/control components and operational design.
REQ-H-AVAIL-002	A	A	A	Automated recovery of affected guest VMs after host or hypervisor-service failure is applicable across architectures. Type I and Hybrid deployments commonly provide this via integrated hypervisor platform mechanisms; Type II deployments commonly achieve it via lifecycle management performed through host/platform services using hypervisor interfaces and signals.
REQ-H-AVAIL-003	A	A	A	Continuity beyond a full restart is an Advanced availability objective within scope for Type I, Hybrid, and Type II hypervisor architectures.
<b>Logging</b>				
REQ-H-LOG-001	A	A	A	Generation of audit logs for security-relevant administrative actions and VM lifecycle operations is implemented by the hypervisor's logging subsystem across all of the architectures.
REQ-H-LOG-002	A	A	A	Protection of audit logs against Unauthorized access, modification, or deletion, and secure export to external log or SIEM systems over authenticated and encrypted channels, is implemented in the hypervisor solution; underlying transport mechanisms may come from the Host OS or platform.

REQ-H-LOG-003	A	A	A	Cryptographic protection and trusted time-stamping of audit logs, enabling verification of origin and integrity during forensic analysis, are capabilities of the hypervisor logging stack in Type I, Hybrid, and Type II deployments.
<b>Secure Update</b>				
REQ-H-UPD-001	A	A	A	Mechanisms for applying security updates to the hypervisor without re-installing the entire host environment form part of the product's lifecycle management for Type I, Hybrid, and Type II solutions; Type II often uses Host OS package managers or application update frameworks.
REQ-H-UPD-002	A	A	A	Authenticated and integrity-checked patch and update workflows are used across all hypervisor architectures, even when the underlying signature verification routines are provided by the Host OS or external tooling.
REQ-H-UPD-003	A	A	C	Protections against Unauthorized rollback to outdated, revoked, or replayed updates are implemented in the update logic of the hypervisor solution in Type I, Hybrid, and Type II deployments, sometimes in combination with Host OS package-management features.
<b>Secure Configuration and Default</b>				
REQ-H-CFG-001	A	A	A	Default configuration that leaves non-essential remote access paths and insecure management interfaces disabled is part of the initial setup profile for hypervisor products in all architectures.
REQ-H-CFG-002	A	A	A	Validation of configuration parameters, with detection of settings that would weaken isolation, control-plane protection, or resource allocations for management and security functions, is performed within the hypervisor configuration logic across Type I, Hybrid, and Type II.
REQ-H-CFG-003	A	A	A	Definition of security configuration baselines and comparison of current settings against those baselines (for example, IOMMU configuration, virtual switch policies, host firewall rules as exposed by the solution) are part of the hypervisor's configuration management capabilities in all three architectures.
<b>Data Minimization</b>				
REQ-H-DM-001	A	A	A	Limiting log content, telemetry, and diagnostic data to what is needed for operation, troubleshooting, and security monitoring, and avoiding inclusion of highly sensitive material, is controlled by the hypervisor's logging and telemetry implementation for Type I, Hybrid, and Type II.
REQ-H-DM-002	A	A	A	Administrative controls for narrowing or disabling specific categories of log, telemetry, and diagnostic collection, including debug payload capture, are implemented as part of the hypervisor's management and observability features for all architectures.

1729

## 1730 5.2.3 Management and Orchestration System Requirements

### 1731 5.2.3.1 General applicability

1732 These requirements apply to the Management and Orchestration (M&O) system regardless of its deployment model.  
 1733 The M&O system may be deployed on dedicated physical hardware, on a dedicated management cluster, or as a virtual  
 1734 machine running on the same virtualized infrastructure that it manages. When deployed as a virtual machine, it is hosted  
 1735 on a management-designated host or cluster segment, and separation from tenant or workload domains is maintained  
 1736 through logical isolation mechanisms such as network segmentation, access control policies, and the CPU, memory,  
 1737 I/O, and integrity isolation mechanisms provided by the hypervisor.

### 1738 5.2.3.2 Authentication

#### 1739 **Classes are exclusive**

##### 1740 **Basic**

1741 REQ-M&O-AUTH-001: The M&O system shall ensure that all administrative interfaces require authentication using  
 1742 unique, non-default credentials.

##### 1743 **Elevated**

1744 REQ-M&O-AUTH-002: The M&O system shall support multi-factor authentication (MFA) for administrative access  
 1745 and shall enforce the use of revocable authentication mechanisms for administrative and service API access. The M&O  
 1746 system shall also ensure that internal component communication is authenticated.

1747 **Advanced**

1748 REQ-M&O-AUTH-003: The M&O system shall support integration with external identity and trust services to validate  
1749 user and service identities and shall support automated credential rotation and revocation where available.

1750 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-AUTH-001	x		
REQ-M&O-AUTH-002		x	
REQ-M&O-AUTH-003			x

1751 **5.2.3.3 Authorization**1752 **Classes are exclusive**1753 **Basic**

1754 REQ-M&O-AUTHZ-001: The M&O system shall restrict all administrative actions to authenticated accounts and  
1755 enforce a default-deny policy for all resource access and control functions.

1756 **Elevated**

1757 REQ-M&O-AUTHZ-002: The M&O system shall enforce Role-Based Access Control (RBAC), ensuring that each  
1758 authenticated user or service account is assigned only the minimum privileges necessary for its defined role.

1759 **Advanced**

1760 REQ-M&O-AUTHZ-003: The M&O system shall support integration with an external identity and authorization  
1761 service to enforce authorization policies and shall generate audit events for privilege changes and authorization  
1762 decisions.

1763 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-AUTHZ-001	x		
REQ-M&O-AUTHZ-002		x	
REQ-M&O-AUTHZ-003			x

1764 **5.2.3.4 Secure Configuration**1765 **Classes are cumulative**1766 **Basic**

1767 REQ-M&O-CFG-001: The M&O system shall be delivered with secure default settings, including least-privilege access  
1768 roles, disabled unused services or ports, and enforced secure communication protocols.

1769 **Elevated**

1770 REQ-M&O-CFG-002: The M&O system shall require explicit authorization before applying any configuration changes  
1771 to the hypervisor or managed workloads. It shall also require explicit authorization before applying any changes to its  
1772 own internal configuration.

1773 **Advanced**

1774 REQ-M&O-CFG-003: The M&O system shall enforce security configuration baselines for all managed hypervisors and  
1775 its own components and shall automatically detect and alert on any deviation from these baselines.

1776 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-CFG-001	x	x	x
REQ-M&O-CFG-002		x	x
REQ-M&O-CFG-003			x

## 1777 5.2.3.5 Communication Security

1778 **Classes are cumulative**1779 **Basic**

1780 REQ-M&O-COM-001: The M&O system shall protect the confidentiality and integrity of management and  
 1781 orchestration communications exchanged through M&O interfaces and communication paths used to control or manage  
 1782 hypervisors.

1783 **Elevated**

1784 REQ-M&O-COM-002: The M&O system shall enforce certificate-based mutual authentication for communications  
 1785 used by the M&O system to exchange management or orchestration data with managed hypervisors.

1786 **Advanced**

1787 REQ-M&O-COM-003: The M&O system shall ensure that control operations are performed only over  
 1788 cryptographically protected and mutually authenticated communication channels and only toward managed hypervisors  
 1789 whose identity has been successfully verified and authorized by the M&O system. The M&O system shall reject control  
 1790 operations when the identity of the addressed managed hypervisor cannot be established or is not authorized.

1791 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-COM-001	x	x	x
REQ-M&O-COM-002		x	x
REQ-M&O-COM-003			x

## 1792 5.2.3.6 Integrity Protection

1793 **Classes are cumulative**1794 **Basic**

1795 REQ-M&O-INT-001: The M&O system shall verify the integrity of orchestration artifacts (including templates,  
 1796 playbooks, and scripts) at import or before execution by validating a cryptographic hash against a trusted reference  
 1797 value.

1798 **Elevated**

1799 REQ-M&O-INT-002: The M&O system shall verify the integrity and authenticity of all orchestration artifacts using  
 1800 digital signatures validated against a trusted keyring before execution.

1801 **Advanced**

1802 REQ-M&O-INT-003: The M&O system shall validate the integrity and authenticity of all third-party plugins,  
 1803 extensions, or integration modules before activation, and shall support administrative functions to register and revoke  
 1804 trusted signing keys.

1805 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-INT-001	x	x	x
REQ-M&O-INT-002		x	x
REQ-M&O-INT-003			x

## 1806 5.2.3.7 Logging

1807 **Classes are cumulative**1808 **Basic**

1809 REQ-M&O-LOG-001: The M&O system shall log all administrative actions, including user logins, configuration  
 1810 changes, VM lifecycle events, orchestration workflows, and security-related alerts.

1811 **Elevated**

1812 REQ-M&O-LOG-002: The M&O system shall protect stored logs against tampering and support secure export to  
1813 external logging or SIEM systems using encrypted, authenticated protocols.

1814 **Advanced**

1815 REQ-M&O-LOG-003: The M&O system shall support cryptographic signing and time-stamping of audit logs to ensure  
1816 verification of origin and integrity during forensic analysis.

1817 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-LOG-001	x	x	x
REQ-M&O-LOG-002		x	x
REQ-M&O-LOG-003			x

1818 **5.2.3.8 Secure Update**1819 **Classes are cumulative**1820 **Applicability:**

1821 These requirements apply whenever the Management and Orchestration (M&O) System is updated. The mechanisms  
1822 and components used to deliver and install updates for the M&O system, where they are part of the declared product,  
1823 shall collectively satisfy the requirements defined in this clause.

1824 Where the M&O system includes functionality to distribute, install, or orchestrate updates for managed hypervisors as  
1825 part of the declared product, the corresponding update actions shall also satisfy the requirements defined in this clause.

1826 **Basic**

1827 REQ-M&O-UPD-001: The M&O system shall provide a mechanism to securely obtain and apply updates to itself  
1828 without requiring a full system reinstallation.

1829 If update distribution or installation is delegated to an external system that is part of the declared product, the M&O  
1830 system shall verify and record update provenance and shall record update installation state for updates applied to itself  
1831 or orchestrated by it for managed hypervisors, based on information exposed by that external system.

1832 The M&O system shall verify the authenticity and integrity of all updates before installation or orchestration, using  
1833 digital signatures validated against trusted keys or certificates configured for the M&O system.

1834 If authenticity or integrity verification of an update fails, the M&O system shall prevent installation or orchestration of  
1835 that update and shall record a security-relevant event.

1836 **Elevated**

1837 REQ-M&O-UPD-002: The M&O system shall implement rollback protection to prevent unauthorized installation of  
1838 outdated, revoked, or replayed update versions.

1839 Authorized rollback to a previously trusted version shall only be permitted when all of the following conditions are met:

- 1840 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 1841 • The rollback action is explicitly authorized by an administrator with elevated privileges or an equivalent  
1842 authorized control function.
- 1843 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted  
1844 from and the version reverted to.

1845 **Advanced**

1846 REQ-M&O-UPD-003: The M&O system shall support execution of update actions across managed targets in a manner  
1847 that provides, for each target, identification of the update action, verification of update outcome, and detection of failed  
1848 or partial rollout states.

1849 **Requirement applicability by use case and security profile**

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-UPD-001	x	x	x
REQ-M&O-UPD-002		x	x
REQ-M&O-UPD-003			x

## 1850 5.3 CES Security Requirements

### 1851 5.3.1 CRS Requirements

#### 1852 5.3.1.1 Isolation

##### 1853 5.3.1.1.1 Container Isolation

1854 **Classes are cumulative**

#### 1855 **Basic**

1856 REQ-CRS-CN-ISO-001: The CRS shall enforce isolation between containers (workloads) using operating-system  
1857 isolation mechanisms under its control, including at least process, filesystem, network, and resource management  
1858 layers.

1859 The CRS shall support configurations in which one container cannot read, write, or otherwise interfere with another  
1860 container's processes, filesystem contents, network endpoints, or allocated resources.

#### 1861 **Elevated**

1862 REQ-CRS-CN-ISO-002: The CRS shall support configuration of strong separation between containers and the host  
1863 operating system by restricting container access to kernel interfaces, privileged operations, and system resources outside  
1864 their assigned execution context.

1865 This shall include the capability to:

- 1866 • Limit container privileges through reduction of kernel capabilities and prevention of direct access to host-level  
1867 device nodes and sensitive filesystems, and
- 1868 • Restrict container access to kernel interfaces and system calls that are not required for the declared workload,  
1869 using mechanisms such as syscall filtering, mandatory access control policies, or equivalent controls.

#### 1870 **Advanced**

1871 REQ-CRS-CN-ISO-003: The CRS shall support execution of containers within hardened isolation environments that  
1872 strengthen separation between container workloads and the host kernel, so that compromise of a container in such an  
1873 environment does not directly provide an attacker with equivalent access to the host kernel or to other containers.

1874 NOTE 1: Hardened isolation environments may be implemented using, for example, micro-VM based container  
1875 runtimes, user-space kernel sandboxes, or additional Virtualization or sandboxing layers.

1876 NOTE 2: REQ-CRS-CN-ISO-003 does not require all containers to use hardened isolation by default. It requires  
1877 that the CRS be capable of providing such isolation for workloads where it is needed.

#### 1878 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CN-ISO-001	x	x	x
REQ-CRS-CN-ISO-002		x	x
REQ-CRS-CN-ISO-003			x

##### 1879 5.3.1.1.2 Control Plane Isolation

1880 **Classes are cumulative**

#### 1881 **Basic**

1882 REQ-CRS-CP-ISO-001: The CRS shall provide access-control mechanisms for its management and control interfaces.  
 1883 These mechanisms shall support configurations in which containerized workloads are prevented from directly accessing  
 1884 or invoking CRS management or control functions, and in which only authenticated and authorized entities can perform  
 1885 control or configuration actions.

1886 CRS management sockets, control APIs, and management daemons shall be deployable in a manner where they are not  
 1887 reachable from within container namespaces.

1888 **Elevated**

1889 REQ-CRS-CP-ISO-002: The CRS shall support configuration of dedicated logical separation between CRS  
 1890 management and control traffic and container data-plane traffic. The CRS shall support the use of host-only  
 1891 communication endpoints, distinct network namespaces, or dedicated management networks for its management and  
 1892 control interfaces, enabling isolation of CRS control traffic from container workloads.

1893 **Advanced**

1894 REQ-CRS-CP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated  
 1895 communication channels for all remote or network-exposed CRS management and control interfaces, in order to reduce  
 1896 the risk of Unauthorized access, interception, tampering, or lateral movement through the control plane.

1897 Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over  
 1898 communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and  
 1899 authorized management entities.

1900 NOTE: For CRS management interfaces that are restricted to local host-only communication mechanisms with  
 1901 appropriate operating-system access controls, cryptographic protection at the transport layer may not be  
 1902 required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved  
 1903 through the underlying platform mechanisms.

1904 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CP-ISO-001	x	x	x
REQ-CRS-CP-ISO-002		x	x
REQ-CRS-CP-ISO-003			x

1905 5.3.1.1.3 Network Plane Isolation

1906 **Classes are cumulative**

1907 **Basic**

1908 REQ-CRS-NP-ISO-001: The CRS shall maintain logical separation between container network traffic and host or CRS  
 1909 management network traffic, using virtual networking mechanisms under its control.

1910 The CRS shall support configurations in which containers cannot directly access host or CRS management interfaces or  
 1911 services by default, and in which separate network contexts can be defined for management traffic and container  
 1912 workload traffic.

1913 **Elevated**

1914 REQ-CRS-NP-ISO-002: The CRS shall support configuration of explicit traffic filtering and access-control rules for  
 1915 network flows between containers, between container networks, and between container networks and host or  
 1916 management networks.

1917 These rules shall be enforced at the CRS virtual networking layer or at other enforcement points under the control of the  
 1918 CRS, ensuring that only network flows explicitly permitted by the configured policy are allowed to cross these  
 1919 boundaries.

1920 **Advanced**

1921 REQ-CRS-NP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated  
 1922 communication channels for management and control traffic that traverses shared or untrusted networks, in order to  
 1923 maintain separation between management traffic and container data-plane traffic.

1924 Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over  
 1925 communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and  
 1926 authorized management entities.

1927 NOTE: For management interfaces that are restricted to local host-only communication mechanisms with  
 1928 appropriate operating-system access controls, cryptographic protection at the transport layer may not be  
 1929 required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved  
 1930 through the underlying platform mechanisms.

### 1931 Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-NP-ISO-001	x	x	x
REQ-CRS-NP-ISO-002		x	x
REQ-CRS-NP-ISO-003			x

## 1932 5.3.1.2 Integrity Protection

### 1933 5.3.1.2.1 Boot chain integrity verification

#### 1934 **Classes are exclusive**

#### 1935 **Basic**

1936 REQ-CRS-B-INT-001: The CRS shall implement an integrity verification mechanism for its core runtime engine and  
 1937 other CRS executables that are required to start container execution services.

1938 The CRS shall be able to invoke this integrity verification before enabling container scheduling, container lifecycle  
 1939 operations, or other container execution interfaces.

1940 If the integrity verification of any CRS executable covered by this mechanism fails, the CRS shall prevent startup of  
 1941 container execution services and shall generate a security-relevant event.

#### 1942 **Elevated**

1943 REQ-CRS-B-INT-002: The CRS shall implement mechanisms to participate in a verifiable chain of trust for CRS  
 1944 startup components that are delivered as part of the CRS product and executed before container execution services are  
 1945 enabled.

1946 Each CRS startup stage delivered with the product and executed prior to exposing container execution interfaces shall  
 1947 validate the integrity and authenticity of the subsequent CRS stage before transferring control.

1948 If this validation fails for any CRS startup stage, the CRS shall prevent startup of container execution services and shall  
 1949 generate a security-relevant event.

1950 NOTE 1: For the purposes of REQ-CRS-B-INT-002, "participate in a verifiable chain of trust" means that CRS  
 1951 components involved in the startup sequence are capable of integrating into a platform-provided or CRS-  
 1952 defined chain of trust and enforcing integrity and authenticity verification of subsequent CRS stages.

#### 1953 **Advanced**

1954 REQ-CRS-B-INT-003: Where the underlying platform exposes boot integrity and authenticity status through a  
 1955 documented interface, the CRS shall support consuming this integrity and authenticity information for the operating  
 1956 system components and CRS components that shall be loaded and initialized before container execution services are  
 1957 enabled.

1958 The CRS shall support policies that prevent initialization of container execution services when the reported verification  
 1959 status for any required component in this chain indicates failure or when such status cannot be obtained.

1960 NOTE 2: REQ-CRS-B-INT-003 assumes that boot integrity and authenticity status mechanisms are provided and  
 1961 managed by the underlying platform or environment. The CRS responsibility is limited to consuming the  
 1962 exposed integrity and authenticity information and enforcing policies for container startup based on that  
 1963 information. The mechanism producing this status may be software-based cryptographic verification,  
 1964 hardware-assisted measurement/verification, or a combination thereof.

1965 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-B-INT-001	x		
REQ-CRS-B-INT-002		x	
REQ-CRS-B-INT-003			x

1966 **5.3.1.2.2 Container image integrity verification**1967 **Classes are exclusive**1968 **Basic**

1969 REQ-CRS-IMG-INT-001: The CRS shall implement an integrity verification mechanism for container images by  
 1970 validating that the image manifest and all referenced layers match their declared cryptographic digests before  
 1971 instantiation.

1972 The CRS shall invoke this integrity verification automatically before creating or starting any container from an image.  
 1973 If any mismatch, corruption, or removal of a referenced layer or manifest entry is detected, the CRS shall reject the  
 1974 image, shall prevent container instantiation from that image, and shall record a security-relevant event.

1975 NOTE 1: REQ-CRS-IMG-INT-001 focuses on integrity verification of container images using trusted reference  
 1976 values, such as cryptographic digests declared in image manifests or registries.

1977 **Elevated**

1978 REQ-CRS-IMG-INT-002: The CRS shall implement mechanisms to verify both the integrity and the authenticity of  
 1979 container images before execution. This verification shall establish that each image originates from a trusted publisher  
 1980 or approval authority and has not been altered after it was approved for deployment.

1981 Authenticity verification shall rely on cryptographic mechanisms that provide a verifiable binding between the image  
 1982 and the trusted publisher or approval authority and that prevent undetected modification of the image.

1983 The CRS shall prevent execution or instantiation of any container image whose integrity or authenticity cannot be  
 1984 successfully verified according to the configured trust policy.

1985 NOTE 2: Authenticity verification may be implemented using public-key digital signatures validated via certificate  
 1986 chains rooted in trusted authorities, keyless signing mechanisms bound to identified publishers, or  
 1987 equivalent cryptographic schemes that provide verifiable image provenance and tamper detection.

1988 **Advanced**

1989 REQ-CRS-IMG-INT-003: The CRS shall perform integrity and authenticity verification of container images using trust  
 1990 anchors stored as protected trust material that is not modifiable during normal operational state.

1991 The protected trust material shall not be modifiable by container workloads, unprivileged users, or routine runtime  
 1992 operations, and shall only be updated or replaced by authenticated and authorized administrators with appropriate  
 1993 privileges and recorded in security-relevant logs sufficient to support security auditing.

1994 If verification of the integrity or authenticity of a container image using this protected trust material fails, or if the  
 1995 verification status of the image cannot be established, the CRS shall prevent execution or instantiation of that container  
 1996 image and shall record a security relevant event.

1997 NOTE 3: Protected trust material refers to cryptographic keys, certificates, or other trust anchors used for container  
 1998 image integrity and authenticity verification, stored and managed so that they cannot be altered through  
 1999 unauthorized administrative or workload facing interfaces. Updates to protected trust material are  
 2000 permitted only via authenticated and authorized administrative actions and are logged for security audit  
 2001 purposes.

2002 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-IMG-INT-001	x		
REQ-CRS-IMG-INT-002		x	
REQ-CRS-IMG-INT-003			x

## 2003 5.3.1.2.3 Runtime integrity protection

2004 **Classes are cumulative**2005 **Elevated**

2006 REQ-CRS-RP-INT-002: The CRS shall enforce integrity protection for its configuration and control plane by ensuring  
2007 that only authenticated and authorized modifications can be applied. The CRS shall generate audit events for any  
2008 security relevant changes to configuration or internal control logic.

2009 **Advanced**

2010 REQ-CRS-RP-INT-003: The CRS shall maintain the integrity of its security critical runtime components and detect  
2011 Unauthorized modification of configuration state or control logic used to enforce security policies or isolation.

2012 The CRS shall maintain a trusted baseline of the integrity state of these security critical runtime components and shall  
2013 perform periodic or continuous integrity validation against this baseline.

2014 Upon detecting a deviation from the trusted baseline, the CRS shall automatically trigger protective actions, such as  
2015 isolating the affected component, disabling affected functions, or restoring the component from a trusted state, and shall  
2016 generate a security alert that cannot be suppressed by the affected component.

2017 NOTE 1: Security critical runtime components include, for example, processes or modules that enforce container  
2018 isolation, access control, security policies, or integrity checks, and configuration data that directly  
2019 influences these security functions.

2020 NOTE 2: Runtime integrity validation may be implemented using hardware assisted mechanisms, software based  
2021 mechanisms, or a combination of both, provided that the mechanisms are able to detect Unauthorized  
2022 modification of the security critical runtime components defined by this requirement.

2023 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-RP-INT-002		X	X
REQ-CRS-RP-INT-003			X

## 2024 5.3.1.2.4 Remote attestation

2025 **Single Class**2026 **Advanced**

2027 REQ-CRS-RA-INT-003: The CRS may support measured launch and generation of attestation evidence describing its  
2028 core runtime engine, security-critical configuration, and other runtime state that is relevant to the integrity of container  
2029 execution.

2030 This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the CRS instance  
2031 for which the measurements were produced.

2032 The CRS shall protect attestation keys and measurement data against Unauthorized access, disclosure, or replay.

2033 The attestation evidence should be limited to the minimum information necessary to verify the integrity and  
2034 configuration state of the CRS.

2035 NOTE 1: This requirement covers the CRS capability to collect measurements, bind them cryptographically to the  
2036 CRS instance, and expose them as attestation evidence. The design and operation of local or remote  
2037 verifiers, attestation protocols, and relying parties is outside the scope of the present document.

2038 NOTE 2: Where supported by the underlying platform or environment, the CRS may leverage platform trust  
2039 services or externally protected trust material as trust anchors for attestation keys and measurements.

2040 REQ-CRS-RA-INT-004: Where attestation evidence produced by the CRS is intended to be used in remote verification  
2041 scenarios, the CRS should support configuration of privacy-preserving attestation options that reduce linkability and  
2042 correlation between attestation events while preserving the ability of authorized verifiers to assess the integrity and  
2043 configuration state of the CRS.

2044 Such options should include, as appropriate for the deployment model:

- 2045 • Limiting attested information to what is necessary for integrity and configuration verification (data  
2046 minimization);
- 2047 • Configuration of the conditions and intervals under which attestation can be triggered or repeated, in order to  
2048 prevent excessive or Unauthorized monitoring;
- 2049 • Support for privacy-preserving techniques, such as the use of pseudonymous identifiers or separation of  
2050 identity and integrity information, where technically feasible.

2051 NOTE 3: REQ-CRS-RA-INT-004 does not require specific attestation protocols or identity systems. It requires  
2052 that, where remote attestation is used, the CRS provides configuration options that allow the operator to  
2053 balance integrity assurance and privacy considerations.

2054 NOTE 4: Remote attestation is optional in the Advanced class. A product may conform to Advanced without  
2055 supporting remote attestation. If remote attestation is not supported, this limitation shall be documented in  
2056 the product security documentation.

### 2057 Requirement applicability by use case and security profile

2058 NOTE 5: In the following table, “o” indicates an optional capability within UC-C3 / SP3. Absence of support does  
2059 not preclude Advanced-class conformance.

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-RA-INT-003			o
REQ-CRS-RA-INT-004			o

## 2060 5.3.1.3 Authentication

### 2061 5.3.1.3.1 General

2062 This clause distinguishes between authentication of administrative users and authentication of external services that  
2063 access CRS management functions or data.

### 2064 5.3.1.3.2 Administrative Authentication

#### 2065 **Classes are exclusive**

#### 2066 **Basic**

2067 REQ-CRS-ADMIN-AUTH-001: The CRS shall require authentication for access to all administrative interfaces,  
2068 including local command-line interfaces, remote application programming interfaces, and administrative dashboards or  
2069 consoles.

2070 The CRS shall enforce the use of unique, non-default credentials for administrative accounts. Where password-based  
2071 authentication is supported, the CRS shall enforce password complexity requirements and a mechanism limiting  
2072 repeated failed authentication attempts, with configurable parameters to mitigate brute-force attempts.

2073 Where password-based authentication is used, the CRS shall store administrative passwords only using non-reversible,  
2074 salted hashing functions. Default credentials, if present, shall be disabled or shall be changed before administrative  
2075 access is granted.

#### 2076 **Elevated**

2077 REQ-CRS-ADMIN-AUTH-002: The CRS shall require authenticated access to administrative interfaces and shall  
2078 support cryptographic key-based authentication for administrative accounts. The CRS shall allow administrative  
2079 password-based login to be disabled for administrative interfaces.

2080 • The CRS shall ensure that private keys used for administrative authentication are stored and managed in a  
2081 manner that prevents Unauthorized extraction or reuse by container workloads or other unprivileged  
2082 components.

2083 • Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

#### 2084 **Advanced**

2085 REQ-CRS-ADMIN-AUTH-003: The CRS shall require strong authentication (as defined in clause 3.1) for all  
2086 administrative interfaces.

2087 Where certificates are used for administrative authentication, their validity, expiration, and revocation status shall be  
2088 verified before administrative access is granted.

#### 2089 Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-ADMIN-AUTH-001	x		
REQ-CRS-ADMIN-AUTH-002		x	
REQ-CRS-ADMIN-AUTH-003			x

### 2090 5.3.1.3.3 Service Authentication

#### 2091 Classes are exclusive

#### 2092 Basic

2093 REQ-CRS-SERV-AUTH-001: The CRS shall require authentication for external services accessing CRS application  
2094 programming interfaces or control interfaces, including orchestration systems, monitoring tools, and backup agents.

2095 The CRS shall enforce the use of distinct credentials for each service account and shall not allow built-in shared or  
2096 hard-coded service credentials.

2097 The CRS shall provide configurable lockout or rate-limit mechanisms for service authentication attempts and shall store  
2098 service credentials in a manner that prevents exposure through configuration files, container images, or logs.

#### 2099 Elevated

2100 REQ-CRS-SERV-AUTH-002: The CRS shall support mutual cryptographic authentication for service-to-CRS  
2101 interactions that access management functions or data, allowing both the CRS and the external service to verify each  
2102 other's identity before exchanging management or control information.

2103 Cryptographic materials used for service authentication, including keys, certificates, or tokens, shall be protected from  
2104 Unauthorized extraction or reuse by container workloads or other unprivileged components.

2105 Algorithms and key lengths shall conform to current cryptographic best practices, aligned with [2].

#### 2106 Advanced

2107 REQ-CRS-SERV-AUTH-003: The CRS shall enforce verifiable binding of service identities to cryptographic  
2108 credentials for all service interactions that access management functions or data. Service credentials shall be issued,  
2109 rotated, and revoked through trusted credential or identity management systems, and the CRS shall automatically reject  
2110 revoked or expired credentials during authentication.

2111 The CRS shall support integration with external identity or credential management systems in order to provide  
2112 traceability of service identities across deployments and to ensure that only services with valid, managed identities can  
2113 invoke CRS management or control functions.

#### 2114 Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-SERV-AUTH-001	x		
REQ-CRS-SERV-AUTH-002		x	
REQ-CRS-SERV-AUTH-003			x

### 2115 5.3.1.4 Authorization

#### 2116 5.3.1.4.1 General

2117 This clause distinguishes between authorization of administrative users and authorization of external services that  
2118 access CRS management functions or data.

## 2119 5.3.1.4.2 Administrative Authorization

2120 **Classes are exclusive**2121 **Basic**

2122 REQ-CRS-ADMIN-AUTHZ-001: The CRS shall restrict all administrative actions to authenticated accounts and shall  
 2123 enforce a default-deny policy for access to management functions, permitting only those actions that are explicitly  
 2124 authorized.

2125 **Elevated**

2126 REQ-CRS-ADMIN-AUTHZ-002: The CRS shall enforce granular role-based access control or attribute-based access  
 2127 control for administrative accounts, ensuring that each administrative account is granted only the minimum default  
 2128 privileges required for its operational role.

2129 **Advanced**

2130 REQ-CRS-ADMIN-AUTHZ-003: The CRS shall enforce fine-grained authorization policies for administrative  
 2131 accounts, including support for separation of duties.

2132 Where the operational environment requires time-bound or just-in-time administrative privilege elevation, the CRS  
 2133 shall provide means to enforce time-limited elevation and revocation either through native capabilities or through  
 2134 integration with external identity/access management or policy decision systems.

2135 The CRS shall support integration with external identity and policy decision systems to enable externally defined  
 2136 administrative authorization policies to be enforced within the CRS administrative scope.

2137 NOTE: Time-bound or just-in-time privilege elevation may be implemented using external access-management  
 2138 controls integrated with the CRS; dedicated native mechanisms are not required.

2139 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-ADMIN-AUTHZ-001	x		
REQ-CRS-ADMIN-AUTHZ-002		x	
REQ-CRS-ADMIN-AUTHZ-003			x

## 2140 5.3.1.4.3 Service Authorization

2141 **Classes are exclusive**2142 **Basic**

2143 REQ-CRS-SERV-AUTHZ-001: The CRS shall enforce a default-deny policy for all service accounts, ensuring that  
 2144 service accounts are only permitted to perform actions that are explicitly authorized by CRS configuration or policy.

2145 **Elevated**

2146 REQ-CRS-SERV-AUTHZ-002: The CRS shall enforce granular authorization for service accounts, binding permitted  
 2147 actions to the authenticated service identity and enforcing least-privilege permissions for each service account.

2148 **Advanced**

2149 REQ-CRS-SERV-AUTHZ-003: The CRS shall enforce fine-grained, policy-driven authorization for service accounts,  
 2150 binding permissions to cryptographic service identities and supporting context-aware enforcement.

2151 The CRS shall support integration with external authorization systems, where available, to enable centralized  
 2152 management and evaluation of authorization policies for service accounts.

2153 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-SERV-AUTHZ-001	x		
REQ-CRS-SERV-AUTHZ-002		x	
REQ-CRS-SERV-AUTHZ-003			x

### 2154 5.3.1.5 Confidentiality Protection

2155 **Classes are cumulative**

#### 2156 **Basic**

2157 REQ-CRS-CONF-001 (Access control for sensitive data): The CRS shall provide access-control mechanisms to restrict  
2158 access to sensitive data under its control, including container images, configuration data, stored secrets, and security-  
2159 relevant logs.

2160 Access to such sensitive data shall be permitted only to authenticated and authorized entities through CRS-controlled  
2161 interfaces and APIs.

2162 NOTE 1: Sensitive data includes, for example, container images stored in registries under CRS control,  
2163 configuration objects that affect workload behaviour or security, secrets such as credentials and keys  
2164 managed via the CRS, and logs that may contain security-relevant information.

#### 2165 **Elevated**

2166 REQ-CRS-CONF-002 (Data at rest): The CRS shall enforce encryption of sensitive data at rest for data that it stores or  
2167 manages, including container images, configuration data, stored secrets, and security-relevant logs, either through  
2168 native capabilities or through use of encrypted storage, registries, or secret-management services.

2169 Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by  
2170 container workloads and other unprivileged components.

2171 REQ-CRS-CONF-003 (Data in transit): The CRS shall support the use of encrypted communication for CRS  
2172 management and control interfaces, for communication with image registries, and for other CRS APIs that handle  
2173 sensitive data or control operations.

2174 When these communications are configured to use encrypted transport, the CRS shall prevent fallback or downgrade to  
2175 unencrypted or weakly protected channels.

2176 NOTE 2: Encrypted communication may be implemented using TLS or equivalent cryptographic protocols,  
2177 including integration with service meshes or platform-provided secure communication mechanisms,  
2178 provided that confidentiality, integrity, and peer authentication are ensured.

#### 2179 **Advanced**

2180 REQ-CRS-CONF-004 (Data in use - confidential execution): Where the underlying platform provides mechanisms for  
2181 isolated or protected execution environments, the CRS shall support provisioning and management of execution  
2182 environments for containerized workloads in which confidential data associated with those workloads is protected,  
2183 during execution, from observation, inspection, or access through the CRS, the host operating system, and other  
2184 workloads.

2185 The CRS shall provide configuration and API interfaces to deploy workloads into such protected execution  
2186 environments and shall support secure provisioning of secrets and key material to those workloads such that the secrets  
2187 and key material are not accessible to the CRS itself, the host operating system, or other workloads.

2188 NOTE 3: Platform mechanisms for confidential execution may include hardware-mediated execution enclaves,  
2189 confidential containers, or other hardware or software mechanisms that provide protected memory and  
2190 processor-state isolation beyond standard container separation.

#### 2191 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CONF-001	x	x	x
REQ-CRS-CONF-002		x	x
REQ-CRS-CONF-003		x	x
REQ-CRS-CONF-004			x

### 2192 5.3.1.6 Availability and Resilience

2193 **Classes are cumulative**

#### 2194 **Basic**

2195 REQ-CRS-AVAIL-001: The CRS shall maintain resource isolation between container workloads and shall protect CRS  
2196 control-plane processes from resource exhaustion caused by container workloads. Resource isolation shall cover at least  
2197 CPU, memory, and storage resources.

2198 The CRS shall provide mechanisms such that failure or abnormal termination of an individual container is contained  
2199 within its execution context and does not, under normal operating conditions and within configured resource limits,  
2200 compromise operation of the CRS control plane or cause other containers to terminate unexpectedly.

#### 2201 **Elevated**

2202 REQ-CRS-AVAIL-002: The CRS shall provide automatic recovery mechanisms such that failed containers are restarted  
2203 based on defined policies without requiring manual intervention. Where supported by the deployment environment,  
2204 these mechanisms may be used in conjunction with orchestration or clustering systems to restore workloads on the same  
2205 or an alternative host.

2206 REQ-CRS-AVAIL-003: The CRS shall implement resource scheduling or reservation mechanisms that support  
2207 configuration of minimum resource allocations for CRS security and management functions, covering at least CPU,  
2208 memory, and I/O resources.

2209 These mechanisms shall protect the CRS control plane from resource starvation caused by container workloads or  
2210 external resource contention, ensuring that security and management functions remain able to operate under load.

#### 2211 **Advanced**

2212 REQ-CRS-AVAIL-004: Where the deployment environment supports workload relocation, the CRS shall support  
2213 checkpoint and restore of the runtime state of selected container workloads, including process state and relevant in-  
2214 memory execution context, in order to mitigate control-plane or host-level failures.

2215 The CRS shall support the use of this checkpoint and restore capability to enable failover with minimal disruption to  
2216 workload execution when combined with orchestration or clustering mechanisms that coordinate relocation to a standby  
2217 instance or alternative host.

#### 2218 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-AVAIL-001	x	x	x
REQ-CRS-AVAIL-002		x	x
REQ-CRS-AVAIL-003		x	x
REQ-CRS-AVAIL-004			x

#### 2219 5.3.1.7 Logging

##### 2220 **Classes are cumulative**

#### 2221 **Basic**

2222 REQ-CRS-LOG-001: The CRS shall generate time-stamped audit logs for security-relevant administrative actions and  
2223 security events, including at least authentication attempts to CRS administrative or control interfaces, configuration  
2224 changes affecting CRS behaviour or security posture, and container lifecycle operations (creation, start, stop, deletion,  
2225 and restart) initiated through the CRS.

2226 Each audit record shall include, as applicable to the event type, an event type identifier, the time of the event, the  
2227 outcome of the event, and the identity of the associated user, service, or process.

2228 The CRS shall ensure that audit logs are not writable or directly modifiable by container workloads or other  
2229 unprivileged processes running within container namespaces.

2230 NOTE: The set of administrative actions and security events that are considered security-relevant may vary  
2231 depending on the CRS implementation and its exposed interfaces. It is expected that the manufacturer  
2232 identifies, in the technical documentation, which actions/events are treated as security-relevant for audit  
2233 logging, and explains any product-specific additions beyond the minimum set listed above.

#### 2234 **Elevated**

2235 REQ-CRS-LOG-002: The CRS shall protect audit logs from unauthorized access, modification, or deletion. Access to  
2236 CRS audit logs shall be restricted to authorized administrative roles or trusted system processes under the control of the  
2237 CRS or host operating system.

2238 The CRS shall support secure export of audit logs to external log management or security information and event  
2239 management systems using authenticated and encrypted communication channels.

2240 Where the CRS exposes tenant-visible logs or log-derived information in multi-tenant environments, it shall maintain  
2241 separation between operator audit logs and tenant-visible logs or telemetry.

#### 2242 **Advanced**

2243 REQ-CRS-LOG-003: The CRS shall support cryptographic protection, trusted time-stamping, and tamper-evident  
2244 protection of audit logs to enable verification of the origin and integrity of audit records during forensic analysis. Where  
2245 the CRS relies on external time or signing services, audit records shall include sufficient metadata to allow verification  
2246 of the trust source used for time-stamping and integrity protection.

#### 2247 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-LOG-001	x	x	x
REQ-CRS-LOG-002		x	x
REQ-CRS-LOG-003			x

### 2248 5.3.1.8 Secure Update

#### 2249 **Classes are cumulative**

#### 2250 **Applicability:**

2251 These requirements apply whenever the CRS components are updated. The mechanisms and components used to  
2252 deliver and install updates for the CRS, including host operating system package managers, management appliances, or  
2253 orchestration-based update pipelines that are part of the evaluated solution, shall collectively satisfy the requirements  
2254 defined in this clause.

2255 Where the CRS is delivered only as part of a larger product image or appliance, the product update mechanism shall  
2256 ensure that CRS binaries and related components are updated in accordance with REQ-CRS-UPD-001 to REQ-CRS-  
2257 UPD-002, as applicable.

#### 2258 **Basic**

2259 REQ-CRS-UPD-001: The CRS shall support applying security updates to CRS components without requiring a full  
2260 reinstallation of the CRS or the host system.

2261 The CRS shall verify the authenticity and integrity of all CRS updates before installation, using digital signatures  
2262 validated against trusted keys or certificates configured for the CRS.

2263 If authenticity or integrity verification of a CRS update fails, the CRS shall prevent installation of that update and shall  
2264 record a security-relevant event.

2265 Keys or certificates used for CRS update verification shall not be modifiable by container workloads or other  
2266 unprivileged components during normal operation.

#### 2267 **Elevated**

2268 REQ-CRS-UPD-002: The CRS shall implement rollback protection to prevent unauthorized installation of outdated,  
2269 revoked, or replayed CRS updates.

2270 Authorized rollback to a previously CRS version shall only be permitted when all of the following conditions are met:

- 2271 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 2272 • The rollback action is explicitly authorized by an administrator with elevated privileges or an equivalent  
2273 authorized control function.

- 2274 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted  
2275 from and the version reverted to.

2276 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-UPD-001	x	x	x
REQ-CRS-UPD-002		x	x

2277 **5.3.1.9 Secure Configuration and Default**

2278 **Classes are cumulative**

2279 **Basic**

2280 REQ-CRS-CFG-001: The CRS shall, by default, disable interfaces and features that are not required for container  
2281 execution and CRS administration, including remote debugging endpoints, unauthenticated control sockets, and  
2282 developer interfaces that expose management or control capabilities.

2283 Only administrators shall be able to explicitly enable such interfaces and features through CRS configuration.

2284 **Elevated**

2285 REQ-CRS-CFG-002: The CRS shall validate configuration parameters under its control before applying them and shall  
2286 reject settings that would disable or bypass security mechanisms or that would compromise container-to-container  
2287 isolation or controlled access to host resources.

2288 When a configuration change is rejected for security reasons, the CRS shall provide clear feedback to the administrator  
2289 indicating the reason for rejection.

2290 **Advanced**

2291 REQ-CRS-CFG-003: The CRS shall support the definition and use of security configuration baselines that define secure  
2292 settings for CRS daemon configuration and default container security profiles, including at least isolation-related  
2293 options and default access to host resources as implemented within the CRS product.

2294 The CRS shall provide capabilities to automatically validate current CRS configuration and default container profile  
2295 settings against a selected security baseline and shall generate alerts or audit events when non-compliant settings are  
2296 detected.

2297 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CFG-001	x	x	x
REQ-CRS-CFG-002		x	x
REQ-CRS-CFG-003			x

2298 **5.3.1.10 Data Minimization**

2299 **Single Class**

2300 **Basic**

2301 REQ-CRS-DM-001: The CRS shall limit logging, telemetry, and diagnostic data to information necessary for runtime  
2302 operation, troubleshooting, and security monitoring. Logs, telemetry, and diagnostic outputs produced by the CRS shall  
2303 not include plaintext credentials, full cryptographic keys, complete container memory contents, or container workload  
2304 payload data unless such inclusion is explicitly enabled for a specific debugging purpose by an administrator.

2305 Payload content from container workloads shall not be included in CRS logs, telemetry, or diagnostic outputs by  
2306 default. Where the CRS supports inclusion of payload content for debugging purposes, this behaviour shall only be  
2307 enabled through explicit administrative configuration and shall be clearly indicated to the administrator.

2308 REQ-CRS-DM-002: The CRS shall provide administrative controls to disable or restrict the collection of specific  
2309 categories of log, telemetry, and runtime metric data. These controls shall allow administrators to disable or limit the  
2310 collection of particular log types, telemetry streams, metrics, or debug payload capture, in order to align data collection  
2311 with operational and regulatory requirements.

2312 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-DM-001	x	x	x
REQ-CRS-DM-002	x	x	x

2313 **5.3.2 CE Requirements**2314 **5.3.2.1 Isolation**2315 **Classes are cumulative**2316 **Basic**

2317 REQ-CE-ISO-001: The CE shall provide a secure interface to configure and invoke CRS operations for the creation,  
 2318 execution, and termination of containers, ensuring that the underlying CRS enforces strong logical separation  
 2319 consistently across all containers. The CE shall prevent container workloads from accessing host administrative  
 2320 functions through interfaces or configuration options exposed by the CE.

2321 **Elevated**

2322 REQ-CE-ISO-002: The CE shall enforce this by restricting use of CE options and configuration parameters that would  
 2323 weaken container isolation, including privileged execution, host namespace sharing, and unrestricted host mounts,  
 2324 unless explicitly enabled by an authorized administrator.

2325 **Advanced**

2326 REQ-CE-ISO-003: The CE shall protect its administrative interface against unauthorized access and shall, where such  
 2327 interface is network-accessible, support separation of that interface from container workload traffic and restriction of  
 2328 access to authorized host processes only.

2329 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-ISO-001	x	x	x
REQ-CE-ISO-002		x	x
REQ-CE-ISO-003			x

2330 **5.3.2.2 Integrity Protection**2331 **Classes are exclusive**2332 **Basic**

2333 REQ-CE-INTEG-001: The CE shall verify the integrity of container images and associated metadata, including  
 2334 manifests, before execution. Images that fail verification shall not be made available to the CRS.

2335 **Elevated**

2336 REQ-CE-INTEG-002: The CE shall verify the authenticity and integrity of container images using digital signatures  
 2337 validated against trusted keys or authorities.

2338 **Advanced**

2339 REQ-CE-INTEG-003: The CE shall verify the authenticity and integrity of container images using trust anchors stored  
 2340 as protected trust material that is not modifiable during normal operational state. If verification of the integrity or  
 2341 authenticity of a container image using this protected trust material fails, or if the verification status of the image cannot  
 2342 be established, the CE shall prevent that image from being made available to the CRS.

2343 NOTE: Protected trust material refers to keys, certificates, or other trust anchors used for container image integrity  
 2344 and authenticity verification, stored and managed so that they cannot be altered through unauthorized  
 2345 administrative or workload-facing interfaces.

2346 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
--------------	-------------	-------------	-------------

REQ-CE-INTEG-001	x		
REQ-CE-INTEG-002		x	
REQ-CE-INTEG-003			x

## 2347 5.3.2.3 Authentication

2348 **Classes are exclusive**2349 **Basic**

2350 REQ-CE-AUTH-001: The CE shall enforce strong authentication (as defined in clause 3.1) on all interfaces used by  
2351 orchestration components or external agents to manage containers, ensuring that only authenticated actors can execute  
2352 workload lifecycle or configuration operations.

2353 **Elevated**

2354 REQ-CE-AUTH-002: The CE shall enforce cryptographic key-based mutual authentication for communication with the  
2355 orchestrator or authorized agents. Private keys shall be protected from Unauthorized extraction or reuse.

2356 **Advanced**

2357 REQ-CE-AUTH-003: The CE shall enforce certificate-based mutual authentication for all management interfaces.  
2358 Certificates used for authentication shall be validated against a trusted public key infrastructure, including verification  
2359 of certificate validity, expiration, and revocation status, before access is granted.

2360 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-AUTH-001	x		
REQ-CE-AUTH-002		x	
REQ-CE-AUTH-003			x

## 2361 5.3.2.4 Authorization

2362 **Classes are exclusive**2363 **Basic**

2364 REQ-CE-AUTHZ-001: The CE shall enforce access control on all operations that affect container lifecycle or resource  
2365 usage, using a default-deny policy and evaluating the authenticated identity of the caller.

2366 **Elevated**

2367 REQ-CE-AUTHZ-002: The CE shall enforce access control using a Role-Based Access Control (RBAC) model. It shall  
2368 support the configuration of workload-level security policies that restrict access to sensitive host resources (devices,  
2369 filesystem paths, syscalls) and translate these policies into runtime enforcement parameters for the CRS.

2370 **Advanced**

2371 REQ-CE-AUTHZ-003: The CE shall enforce fine-grained, Attribute-Based Access Control (ABAC) or context-aware  
2372 authorization for control operations, supporting complex policy decision points and integration with external  
2373 authorization systems.

2374 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-AUTHZ-001	x		
REQ-CE-AUTHZ-002		x	
REQ-CE-AUTHZ-003			x

## 2375 5.3.2.5 Confidentiality

2376 **Classes are cumulative**2377 **Basic**

2378 REQ-CE-CONF-001 (Protection of stored data): The CE shall prevent Unauthorized access to container images,  
 2379 configuration data, and secrets stored or managed by the CE by enforcing CE-controlled access boundaries and ensuring  
 2380 that such data is not exposed to workloads or users unless explicitly authorized through CE configuration or APIs.

2381 **Elevated**

2382 REQ-CE-CONF-002 (Data at rest and in transit): The CE shall support encryption at rest for container images,  
 2383 configuration data, and secrets under its control, and shall ensure that CE administrative interfaces and communications  
 2384 with external services are protected using encrypted channels.

2385 **Advanced**

2386 REQ-CE-CONF-003 (Data in use): Where the underlying platform exposes protected execution contexts that provide  
 2387 confidentiality of workload data in use against privileged software (confidential execution), the CE shall provide  
 2388 configuration and API interfaces to enable workload execution in such protected contexts and shall, where the user opts  
 2389 to use these capabilities, enforce placement and associated security policies such that declared confidential workloads  
 2390 run only in such protected contexts. The CE shall support secure provisioning of secrets and key material into the  
 2391 protected execution context such that they are not accessible to the CE, host operating system, or other workloads.

2392 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-CONF-001	x	x	x
REQ-CE-CONF-002		x	x
REQ-CE-CONF-003			x

2393 **5.3.2.6 Secure Update**

2394 **Classes are cumulative**

2395 **Applicability:**

2396 These requirements apply whenever the Container Engine (CE) is updated. The mechanisms and components used to  
 2397 deliver and install updates for the CE, where they are part of the declared product, shall collectively satisfy the  
 2398 requirements defined in this clause.

2399 Where the CE is delivered only as part of a larger product image or appliance, the product update mechanism shall  
 2400 ensure that CE binaries and related components are updated in accordance with REQ-CE-UPD-001 to REQ-CE-UPD-  
 2401 002, as applicable.

2402 **Basic**

2403 REQ-CE-UPD-001: The CE shall support applying security updates to CE components without requiring a full  
 2404 reinstallation of the CE or the host system.

2405 The CE shall verify the authenticity and integrity of all CE updates before installation, using digital signatures validated  
 2406 against trusted keys or certificates configured for the CE.

2407 If authenticity or integrity verification of a CE update fails, the CE shall prevent installation of that update and shall  
 2408 record a security-relevant event.

2409 Keys or certificates used for CE update verification shall not be modifiable by container workloads or other  
 2410 unprivileged components during normal operation.

2411 **Elevated**

2412 REQ-CE-UPD-002: The CE shall implement rollback protection to prevent Unauthorized installation of outdated,  
 2413 revoked, or replayed CE updates.

2414 Authorized rollback to a previously CE version shall only be permitted when all of the following conditions are met:

- 2415 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 2416 • The rollback action is explicitly authorized by an administrator with elevated privileges or an equivalent  
 2417 authorized control function.

- 2418 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted  
2419 from and the version reverted to.

2420 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-UPD-001	x	x	x
REQ-CE-UPD-002		x	x

2421 **5.3.3 CO Requirements**

2422 **5.3.3.1 Isolation**

2423 **Classes are cumulative**

2424 **Basic**

2425 REQ-CO-ISO-001: The CO shall enforce strong logical network isolation between workloads scheduled in different  
2426 namespaces or tenancy domains, using network isolation policies or equivalent orchestration mechanisms.

2427 **Elevated**

2428 REQ-CO-ISO-002: The CO shall ensure its management and orchestration interfaces (API server) are logically isolated  
2429 from the container data plane to prevent Unauthorized pivot attempts from compromised workloads.

2430 **Advanced**

2431 REQ-CO-ISO-003: The CO shall enforce cryptographic protection on control plane communications and ensure that  
2432 control plane interfaces cannot be accessed from workload or data plane networks.

2433 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-ISO-001	x	x	x
REQ-CO-ISO-002		x	x
REQ-CO-ISO-003			x

2434 **5.3.3.2 Integrity Protection**

2435 **Classes are cumulative**

2436 **Basic**

2437 REQ-CO-INTEG-001: The CO shall protect the integrity of its configuration and policy state, ensuring that only  
2438 authorized components and roles can modify control plane configuration, workload specifications, or cluster policies.

2439 **Elevated**

2440 REQ-CO-INTEG-002: The CO shall verify the integrity of its control plane component binaries (including API server  
2441 and scheduler) before operation, using cryptographic hashes or signatures validated against a trusted reference.

2442 **Advanced**

2443 REQ-CO-INTEG-003: The CO shall enforce integrity of workload deployments at admission time by applying  
2444 admission control policies and preventing deployment of workloads that violate integrity, authorization, or  
2445 configuration constraints.

2446 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-INTEG-001	x	x	x
REQ-CO-INTEG-002		x	x
REQ-CO-INTEG-003			x

## 2447 5.3.3.3 Authentication

2448 **Classes are exclusive**2449 **Basic**

2450 REQ-CO-AUTH-001: The CO shall enforce authenticated access for all administrative and user interactions with the  
 2451 control plane APIs and management interfaces. Where password-based authentication is used, password strength and  
 2452 lockout policies shall be enforced to mitigate brute-force attempts.

2453 **Elevated**

2454 REQ-CO-AUTH-002: The CO shall enforce cryptographically verifiable authentication mechanisms for service  
 2455 accounts and administrative identities and shall require mutual authentication for internal control plane communication.

2456 NOTE: This may be achieved through mechanisms such as mutual TLS (mTLS), token-based identity,  
 2457 SPIFFE/SPIRE identities, or equivalent cryptographically authenticated identity systems.

2458 **Advanced**

2459 REQ-CO-AUTH-003: The CO shall enforce certificate-based authentication across all control plane and administrative  
 2460 interfaces and shall support integration with a trusted Public Key Infrastructure (PKI) for issuing and managing these  
 2461 certificates.

2462 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-AUTH-001	x		
REQ-CO-AUTH-002		x	
REQ-CO-AUTH-003			x

## 2463 5.3.3.4 Authorization

2464 **Classes are exclusive**2465 **Basic**

2466 REQ-CO-AUTHZ-001: The CO shall enforce a default-deny authorization policy for all control plane actions and  
 2467 resource access, restricting actions to authenticated entities.

2468 **Elevated**

2469 REQ-CO-AUTHZ-002: The CO shall enforce fine-grained access policies on workload and resource control operations  
 2470 through Role-Based Access Control (RBAC), ensuring the principle of least privilege is applied to all users and service  
 2471 accounts.

2472 **Advanced**

2473 REQ-CO-AUTHZ-003: The CO shall support Attribute-Based Access Control (ABAC) or context-aware authorization  
 2474 for control plane actions, enabling policy enforcement based on runtime context.

2475 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-AUTHZ-001	x		
REQ-CO-AUTHZ-002		x	
REQ-CO-AUTHZ-003			x

## 2476 5.3.3.5 Confidentiality

2477 **Classes are cumulative**2478 **Basic**

2479 REQ-CO-CONF-001 (Secrets Protection): The CO shall ensure that control plane secrets, including authentication  
 2480 tokens, private keys, and service credentials, are not stored in plaintext and are accessible only through a controlled  
 2481 interface. The CO shall support either internal secrets store or integration with a Key Management System (KMS).

2482 **Elevated**

2483 REQ-CO-CONF-002 (Data in transit): The CO shall ensure that control-plane traffic and orchestrator-managed  
 2484 communication channels that it configures or enforces are encrypted. The CO shall enforce configuration policies that  
 2485 prevent the use of unencrypted communication paths for control-plane traffic and for orchestrator-managed  
 2486 communication channels under its control.

2487 **Advanced**

2488 REQ-CO-CONF-003 (Data in use): The CO shall provide support for scheduling and managing container workloads in  
 2489 platform-supported confidential execution environments. The CO shall enforce workload placement, isolation  
 2490 constraints, and associated security policies such that confidential workload data cannot be observed by the CO, host  
 2491 operating system, or other workloads.

2492 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-CONF-001	x	x	x
REQ-CO-CONF-002		x	x
REQ-CO-CONF-003			x

2493 **5.3.3.6 Secure Update**

2494 **Classes are cumulative**

2495 **Applicability:**

2496 These requirements apply whenever the Container Orchestrator (CO) is updated. The mechanisms and components used  
 2497 to deliver and install updates for the CO, where they are part of the declared product, shall collectively satisfy the  
 2498 requirements defined in this clause.

2499 Where the CO is delivered only as part of a larger product image or appliance, the product update mechanism shall  
 2500 ensure that CO binaries and related components are updated in accordance with REQ-CO-UPD-001 to REQ-CO-UPD-  
 2501 003, as applicable.

2502 **Basic**

2503 REQ-CO-UPD-001: The CO shall support applying security updates to CO components without requiring a full  
 2504 reinstallation of the CO or the host system.

2505 The CO shall verify the authenticity and integrity of all CO updates before installation, using digital signatures  
 2506 validated against trusted keys or certificates configured for the CO.

2507 If authenticity or integrity verification of a CO update fails, the CO shall prevent installation of that update and shall  
 2508 record a security-relevant event.

2509 Trusted keys or certificates used for CO update verification shall not be modifiable by workloads or other unprivileged  
 2510 components during normal operation.

2511 **Elevated**

2512 REQ-CO-UPD-002: Where rollback of CO components is supported by the declared product, the CO shall implement  
 2513 rollback protection to prevent Unauthorized installation of outdated, revoked, or replayed CO updates.

2514 Authorized rollback to a previously trusted CO version shall only be permitted when all of the following conditions are  
 2515 met:

- 2516 • The rollback image or package is verified for integrity and authenticity against a trusted key or certificate.
- 2517 • The rollback action is explicitly authorized by an administrator with elevated privileges or an equivalent  
 2518 authorized control function.

- 2519 • The rollback operation is recorded as a security-relevant event in the audit logs, including the version reverted  
2520 from and the version reverted to.

2521 **Advanced**

2522 REQ-CO-UPD-003: The CO shall support execution of update actions across managed cluster components in a manner  
2523 that provides, for each managed component, identification of the update action, verification of update outcome, and  
2524 detection of failed or partial rollout states.

2525 **Requirement applicability by use case and security profile**

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-UPD-001	x	x	x
REQ-CO-UPD-002		x	x
REQ-CO-UPD-003			x

2526 **5.4 Assurance Security Requirements**

2527 **5.4.1 Vulnerability Management**

2528 REQ-ASSUR-VULM-001: Vulnerability handling requirements for VES and CES products shall comply with the  
2529 vulnerability handling requirements defined in prEN 40000-1-3 [1].

2530 REQ-ASSUR-VULM-002: In addition to REQ-ASSUR-VULM-001, for products incorporating open-source software  
2531 (OSS), the manufacturer shall establish processes to leverage OSS community vulnerability reporting and resolution  
2532 mechanisms. This includes timely monitoring of upstream advisories and security patches. The manufacturer shall  
2533 ensure that applied updates are regression tested, covering both Long-Term Support (LTS) releases and relevant interim  
2534 point releases, to maintain product stability and security.

2535 NOTE 1: Open-source components form a significant part of VES and CES implementations. Effective use of  
2536 community-driven security alerts and patches is essential to reduce exposure time. Regression testing  
2537 ensures that rapid security fixes do not introduce unintended disruptions in operational environments.

2538 REQ-ASSUR-VULM-003: The product shall not contain unmitigated known exploitable vulnerabilities.

2539 NOTE 2: For the purposes of the present requirement, a known exploitable vulnerability is considered mitigated  
2540 where appropriate technical or operational measures, including measures described in the product  
2541 guidance where relevant, sufficiently prevent or reduce the feasibility of exploitation in the product  
2542 context, taking into account the intended purpose and reasonably foreseeable use of the product.

2543 **Requirement applicability by use case and security profile**

2544 For VES

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-ASSUR-VULM-001	x	x	x
REQ-ASSUR-VULM-002	x*	x*	x*
REQ-ASSUR-VULM-003	x	x	x

2545

2546 For CES

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-ASSUR-VULM-001	x	x	x
REQ-ASSUR-VULM-002	x*	x*	x*
REQ-ASSUR-VULM-003	x	x	x

2547

2548 NOTE 3: x\* indicates applicability where the product incorporates open-source software (OSS).

2549 **5.4.2 Software Bill of Materials (SBOM)**

2550 REQ-ASSUR-SBOM-001: SBOM requirements for VES and CES products shall comply with the SBOM requirements  
2551 defined in prEN 40000-1-3 [1].

2552 REQ-ASSUR-SBOM-002: In addition to REQ-ASSUR-SBOM-001, the manufacturer shall provide a Software Bill of  
 2553 Materials (SBOM) for VES and CES products (see in-scope components of the VES and CES as described in clauses  
 2554 4.2.2.2 and 4.2.3.1), identifying all software components, top-level dependencies, and their versions.

2555 REQ-ASSUR-SBOM-003: For VES, and in addition to REQ-ASSUR-SBOM-001, the SBOM shall include (at  
 2556 minimum):

- 2557 • Hypervisor kernel or microkernel.
- 2558 • Hypervisor management interfaces (CLI/API).
- 2559 • Virtual Machine Monitor (VMM) and lifecycle controllers.
- 2560 • Resource scheduling components (CPU, memory, I/O).
- 2561 • Device Virtualization modules.
- 2562 • Hardware-related dependencies bundled with the VES product.

2563 REQ-ASSUR-SBOM-004: For CES, and in addition to REQ-ASSUR-SBOM-001, the SBOM shall include (at  
 2564 minimum):

- 2565 • CRS: runtime binaries, enforcement modules, APIs/services.
- 2566 • Container Engine: engine binaries, image management tools, container networking modules (bridge,  
 2567 namespace managers), storage/volume plugins, supported extensions.
- 2568 • Container Orchestrator: orchestrator binaries, admission controllers, scheduling modules, secrets management,  
 2569 network policy enforcement, cluster lifecycle management components.
- 2570 • Hardware-related dependencies bundled with the CES product.

2571 NOTE: In this clause examples are provided to illustrate typical components and formats that may fall under the  
 2572 SBOM requirements. These examples are non-exhaustive and non-normative. For instance:

- 2573 • Device Virtualization modules may include SR-IOV drivers, paravirtualized device drivers, or  
 2574 host-guest interface modules.
- 2575 • CRS runtime binaries may include runc or crun.
- 2576 • CRS enforcement modules may include namespaces, seccomp, cgroups, microVMs.
- 2577 • CRS APIs/services may include containerd, CRI-O.
- 2578 • Container engine binaries may include Docker or Podman.
- 2579 • Container orchestrator binaries may include Kubernetes.
- 2580 • Machine-readable SBOM formats may include SPDX or CycloneDX.
- 2581 • Hardware-related dependencies may include vTPM firmware, drivers for SR-IOV NICs, storage  
 2582 controllers, or other hardware-specific modules.

### 2583 Requirement applicability by use case and security profile

2584 For VES

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-ASSUR-SBOM-001	x	x	x
REQ-ASSUR-SBOM-002	x	x	x
REQ-ASSUR-SBOM-003	x	x	x
REQ-ASSUR-SBOM-004			

2585

2586 For CES

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
--------------	-------------	-------------	-------------

REQ-ASSUR-SBOM-001	x	x	x
REQ-ASSUR-SBOM-002	x	x	x
REQ-ASSUR-SBOM-003			
REQ-ASSUR-SBOM-004	x	x	x

2587

## 2588 6 Conformance Assessment / Tests

### 2589 6.1 Assessment Methodology

#### 2590 6.1.1 General Assessment Procedure

2591 This clause defines the methodology for assessing conformity with the VES and CES security requirements specified in  
2592 the present document.

2593 The assessment shall be performed according to:

- 2594 1. the product's declared configuration and included components as defined in clause 4.3;
- 2595 2. the representative use case applicable to the product context;
- 2596 3. the threats applicable to the declared deployment and use case; and
- 2597 4. the applicable SP 1/2/3 corresponding to the assessed risk level (Low/Medium/High).

2598 Assessment activities shall verify that the implemented requirements mitigate the threats associated with the declared  
2599 deployment and use case.

2600 Each requirement is associated with an assessment description that follows a structured format:

- 2601 1) **Assessment reference:** Identifies the link to the exact requirement ID(s).
- 2602 2) **Assessment objective:** Defines the security property or capability that shall be verified, ensuring that the  
2603 assessment remains focused on the intent of the requirement.
- 2604 3) **Assessment preparation:** Describes the environment, setup, and preconditions required before executing the  
2605 test. It includes the following elements as applicable:
  - 2606 - Test environment: Describe the hardware, software, and network setup used for the assessment,  
2607 including versions, topology, and any relevant dependencies.
  - 2608 - Preconditions: Specify any configurations, credentials, or operational states that should be established  
2609 before the test (e.g. product initialized, certificates loaded, user roles created).
  - 2610 - Required tools: Identify the tools or software necessary to perform the assessment (e.g. vulnerability  
2611 scanners, protocol fuzzers, traffic Analyzers, static code Analyzers, cryptographic test suites).
  - 2612 - Reference any vendor-provided setup guides, configuration instructions, or operational manuals, as well  
2613 as any relevant standards or technical notes, that define how the product shall be configured or operated  
2614 for the assessment.
  - 2615 - Product configuration documentation: Confirm that the assessment environment reflects the product  
2616 composition declared by the manufacturer as defined in clause 4.3, including which components are  
2617 included or excluded. For example:
    - 2618 ▪ If the product includes only the Hypervisor, M&O components are not assessed.
    - 2619 ▪ If the product includes both the Hypervisor and the M&O system, each component is assessed  
2620 against its respective requirements, and the assessment shall additionally verify the security-  
2621 relevant interactions between them.

- 2622           ▪    If optional components are excluded, the assessment shall verify that their absence does not affect  
2623           requirement fulfilment.
- 2624       4)   **Assessment activities:** Defines the assessment steps to be performed for the requirement. The specific  
2625           assessment activities shall be defined in the corresponding requirement-specific assessment case and may  
2626           include documentation review, configuration inspection, security testing, code or binary analysis, or  
2627           observation of runtime behaviour, as relevant to the assessed requirement.
- 2628       5)   **Assessment verdict:** Defines the pass/fail criteria.
- 2629           -    **Pass:**  
2630           The assessment is considered passed if the product demonstrably fulfils the requirement and meets the  
2631           defined security thresholds.  
2632           Examples of such thresholds include:
- 2633           ▪    Minimum cryptographic strength (e.g. AES-128 or higher);
- 2634           ▪    Password policy limits (e.g. minimum of 12 characters);
- 2635           ▪    Login protection mechanisms (e.g. account lockout after five consecutive failed attempts);
- 2636           ▪    Resistance to a specified attack potential (e.g. equivalent to CSA High/AVA\_VAN.3 or higher).
- 2637           -    **Fail:**  
2638           The assessment is considered failed if the requirement is not fulfilled, or if the defined security  
2639           thresholds are not achieved (e.g. insufficient key length, missing authentication enforcement, or  
2640           inadequate resistance to the required attack potential).
- 2641       6)   **Assessment evidence:** Defines the artefacts and documentation that shall be collected to demonstrate that the  
2642           requirement has been assessed and fulfilled. The evidence shall be sufficient to enable independent verification  
2643           of the assessment results and to demonstrate compliance with the relevant CRA essential requirements. The  
2644           Assessment evidence includes, where applicable:
- 2645           -    Test or assessment reports showing the steps performed and results obtained;
- 2646           -    Logs, configuration files, or audit traces demonstrating the implementation of the requirement;
- 2647           -    Screenshots, captures, or console outputs confirming the correct execution or protection behaviour;
- 2648           -    Relevant vendor or design documentation describing the applied security measures;
- 2649           -    Evidence of the declared product composition used to scope the assessment (as per clause 4.3)
- 2650           -    Use-case description and risk level justification applied to determine the applicable SP
- 2651           -    Threat applicability analysis demonstrating which threats from clause B.2 were considered for the  
2652           assessed configuration
- 2653           -    Configuration-specific test results where feature or deployment variations affect implementation of the  
2654           requirement
- 2655           -    Evidence of security-relevant interactions between in-scope components when multiple components (e.g.  
2656           Hypervisor and M&O system) are included in the product
- 2657           The assessment is considered incomplete if the assessment evidence is missing, inconsistent, or insufficient to  
2658           verify fulfilment of the requirement.
- 2659       The conformity assessment defined in the present document is evidence-based. Assessment activities are performed in  
2660       order to obtain, examine, or confirm the evidence necessary to demonstrate that the product, in its declared  
2661       configuration, fulfils the applicable requirements.

## 2662 6.1.2 Advanced-class Equivalence and Technology-neutral Assessment

### 2663 6.1.2.1 General

2664 This clause complements the general assessment procedure defined in clause 6.1.1. It provides additional guidance for  
2665 assessing Advanced-class requirements, in particular where such requirements refer to platform-anchored, hardware-  
2666 assisted, or externally protected trust mechanisms.

### 2667 6.1.2.2 Equivalence of Hardware-based and Software-based Approaches

2668 This clause defines general equivalence principles for Advanced-class assessment. The detailed assessment activities  
2669 and verdict criteria remain specified in the corresponding requirement-specific assessment cases.

2670 For Advanced-class requirements that refer to platform-anchored, hardware-assisted, or externally protected trust  
2671 mechanisms, products may fulfil the requirement by implementing:

- 2672 • Hardware-based trust services; or
- 2673 • Software-based or hybrid mechanisms that provide demonstrably equivalent security outcomes and verifiable  
2674 assurance.

2675 Where relevant to the requirement being assessed, equivalence shall be demonstrated by showing that the  
2676 implementation achieves the corresponding security properties listed below:

- 2677 1) **Cryptographic chain of trust**  
2678 A verifiable cryptographic chain of trust is established from boot through runtime, where each stage validates  
2679 the integrity and authenticity of the next stage using digital signatures or equivalent integrity verification of  
2680 authenticated components.
- 2681 2) **Protected key storage**  
2682 Cryptographic keys and credentials used to enforce trust are managed and stored under strict access-control  
2683 policies that prevent Unauthorized reading, modification, or substitution, including by privileged software  
2684 components.
- 2685 3) **Runtime integrity verification**  
2686 Continuous or periodic validation of security-critical components, such as configuration data, binaries, and  
2687 privileged memory regions, is implemented, with automated detection and documented response to integrity  
2688 violations.
- 2689 4) **Attestation capability**  
2690 The product can generate cryptographically verifiable evidence of its configuration and integrity state that can  
2691 be validated by an authorized verifier.
- 2692 5) **Transparency of trust anchors and residual assumptions**  
2693 Where the implementation anchors trust in hardware roots of trust, hardware-mediated execution enclaves,  
2694 hardware security modules, or in protected software-based trust material, the product documentation shall  
2695 describe the trust anchors and protected trust material on which the implementation depends for boot-chain  
2696 integrity, image verification, runtime integrity, attestation, or data-in-use protection. Examples of such  
2697 hardware components may include CPU security features, firmware trust stores, trusted platform modules,  
2698 trusted execution environments, hardware security modules, secure elements, or equivalent platform trust  
2699 anchors. The documentation shall identify at least:
  - 2700 - The hardware and software components that act as trust anchors or provide protected trust material;
  - 2701 - The security properties that are assumed for these components; and
  - 2702 - Residual risks if those assumptions do not hold, including the limits of what software-based mechanisms  
2703 can detect or prevent and potential impacts of hardware-level or supply-chain compromise consistent  
2704 with the threat scenarios defined in clause B.2 (such as T-ALL-HW-ACCESS).

2705 The assessment shall verify achievement of the intended security outcomes of the relevant Advanced-class  
2706 requirements, rather than requiring the use of specific hardware technologies. Hardware-based and software-based  
2707 implementations shall be considered equivalent when they demonstrably provide the same level of protection against

2708 the applicable threats and offer verifiable assurance of their security properties, taking into account the documented  
 2709 trust assumptions and residual risks. The absence of hardware-based trust mechanisms shall not by itself be treated as a  
 2710 deficiency or residual risk. Software-based implementations without hardware dependencies may be used to meet  
 2711 Advanced-class requirements where they demonstrably achieve comparable protection against the mapped threats.

### 2712 6.1.2.3 Technology-neutral Assessment Principles

2713 The general assessment methodology defined in clause 6.1.1 applies to all assessments under the present document. The  
 2714 principles in this clause provide additional guidance for assessment of Advanced-class requirements, in particular where  
 2715 technology-neutral equivalence between hardware-based, software-based, or hybrid approaches is relevant.

2716 When applying the methodology in clause 6.1.1 to any class requirements, the assessment shall:

- 2717 • Evaluate the implementation against the applicable threats identified in clause B.2 and the declared use case  
 2718 and risk level, rather than against a preferred reference architecture;
- 2719 • Focus on security outcomes and effectiveness of threat mitigation, not on the presence of particular hardware  
 2720 features or vendor technologies;
- 2721 • Not classify an implementation as deficient solely because it does not use specific hardware mechanisms, such  
 2722 as hardware roots of trust, hardware-mediated execution enclaves, hardware security modules, IOMMU, or  
 2723 equivalent mechanisms, if the product demonstrates equivalent security properties through alternative  
 2724 architectural means;
- 2725 • Consider compensating controls such as transparency, auditability, rapid update capability, formal verification  
 2726 or extensive testing results when hardware immutability or hardware trust anchors are not used; and
- 2727 • Verify that documentation of trust anchors, trust assumptions, and residual risks is available and consistent  
 2728 with the implemented architecture and the applicable Advanced-class requirements.

2729 These principles apply in particular to Advanced-class requirements that reference platform-anchored trust mechanisms  
 2730 for boot integrity, image verification, runtime integrity, remote attestation, or data-in-use protection.

## 2731 6.2 Assessment Report Requirements

### 2732 6.2.1 Mandatory Report Contents

2733 The assessment report shall provide sufficient information to demonstrate that the assessment was performed in  
 2734 accordance with the present document, that the scope of the assessment corresponds to the declared product  
 2735 configuration, and that the assessed requirements mitigate the applicable threats.

2736 The report shall include the following clauses:

#### 2737 (a) Product Identification and Configuration

- 2738 • Product name, version, manufacturer, and build or release identifier.
- 2739 • Declared product composition as defined in clause 4.3, clearly identifying:
  - 2740 - Components included in the assessed product.
  - 2741 - Components excluded from the product and therefore not assessed.
- 2742 • Architectural classification (e.g. Hypervisor Type 1 / Type 2 / Hybrid, Virtual Execution Stack configuration).
- 2743 • Any declared platform or hardware dependencies relevant to security enforcement.

#### 2744 (b) Use Case, Risk, and Requirement Application

- 2745 • Declared use case(s) and deployment context.
- 2746 • Risk level (Low / Medium / High) determined in accordance with clause B, and corresponding Security Profile  
 2747 (SP 1 / 2 / 3).

- 2748 • Applicable requirement set derived from the representative use case applicable to the product context.

2749 **(c) Threat Applicability**

- 2750 • List of applicable threats from clause B.2 relevant to the declared configuration and use case.

2751 **(d) Assessment activities and Results**

2752 For each assessed requirement, the report shall provide:

- 2753 • Requirement ID and full requirement text.
- 2754 • Requirement class (Basic / Elevated / Advanced).
- 2755 • Linked threat(s) from clause B.2 and rationale for applicability.
- 2756 • Assessment activities performed, including any class-specific or configuration-specific procedures.
- 2757 • Test environment and configuration details (e.g. hardware features, firmware level, cluster topology,  
2758 VM/container layout).
- 2759 • Observations and results.
- 2760 • Verdict (Pass / Fail) with justification.
- 2761 • Assessment evidence Reference (using categories defined in clause 6.2.3).

2762 Where alternative assessment paths are permitted (e.g. due to unavailable platform features), the report shall document:

- 2763 • The unavailable capability and reason
- 2764 • The alternative procedure used
- 2765 • Justification for equivalence of assurance

2766 **(e) Component-Level and System-Level Findings**

- 2767 • Results for each individual component in the product composition.
- 2768 • System-level interaction findings, including isolation, trust boundaries, and interface mediation, where  
2769 multiple components are present.
- 2770 • Identification of any constraints, operational assumptions, or observed limitations.

2771 **(f) Overall Conformity Statement**

- 2772 • Statement of conformity or non-conformity with the present document.
- 2773 • Assigned SP (1 / 2 / 3) and the representative use case applicable to the product context.
- 2774 • Residual risks, limitations, known deviations, or conditional security dependencies.

2775 **(g) Operational Environment Dependencies for Advanced Class Assessments**

2776 When an assessment at the Advanced class relies on Operational Environment (OE) hardware components to achieve  
2777 the required assurance level, such as hardware roots of trust, hardware-mediated execution enclaves, hardware security  
2778 modules, or other platform trust anchors, the assessment report shall explicitly document:

- 2779 • Where applicable, the specific hardware security components including their version and their security  
2780 properties required for conformity (e.g. secure key storage, measured boot, attestation capability).
- 2781 • Where applicable, the specific software security components including their version and their security  
2782 properties required for conformity (e.g. secure key storage, measured boot, attestation capability).
- 2783 • The manner in which these components contribute to the enforcement of the assessed requirements.
- 2784 • Any platform prerequisites or firmware dependencies necessary for correct operation.

2785 The report shall reference corresponding evidence under category [PLT-CAP] (Platform Capability Evidence).

## 2786 6.2.2 Traceability Requirements

2787 The assessment report shall provide complete traceability demonstrating:

2788 Threat → Risk Level → Requirement → Assessment Activity → Evidence → Verdict

2789 This traceability shall be sufficient to enable an independent reviewer to:

- 2790 • Confirm that all applicable threats were addressed.
- 2791 • Verify that the rigor of assessment is consistent with the declared SP.
- 2792 • Reconstruct assessment decisions without direct access to the product.

## 2793 6.2.3 Evidence Package

2794 The assessment shall be supported by an evidence package referenced by the report. Evidence should be organized  
2795 using the following evidence categories:

Tag	Evidence Category	Purpose	Examples
[LOG]	Execution Logs & Test Outputs	Shows what was done and what happened during testing	Console logs, CLI output, packet captures, fuzz results
[CONFIG]	System & Configuration State	Captures how the system was configured at test time	Config files, ACLs, hypervisor settings, routing tables
[SCOPE]	Product Composition & Scope Evidence	Shows which components are included/excluded per clause 4.3	Product composition statement, SBOM, component list
[RA]	Risk & Threat Applicability Evidence	Shows link to clause B scoring & threat selection	Risk scoring tables, threat mapping justification
[DESIGN]	Architecture & Implementation Documentation	Explains how the security mechanism works	Security architecture documentation, mitigation mechanism descriptions, trust-model descriptions, component-boundary descriptions, formal verification evidence, security analysis, or equivalent architectural evidence for diverse implementation approaches
[OBSERVATION]	Captured Traces & Measurement Evidence	Instrumentation proving enforcement or behaviour	Screenshots, Wireshark traces, leakage graphs, audit views
[PLT-CAP]	Platform Capability Evidence	Documents feature availability or dependency	CPU/IOMMU features, trust anchor presence, firmware versions, platform-exposed integrity status interfaces, or other platform capability or dependency evidence relevant to the assessed requirement

2796

## 2797 6.3 VES

### 2798 6.3.1 Hypervisor

#### 2799 6.3.1.1 Assessment for VM Isolation

##### 2800 6.3.1.1.1 Assessment Case AC-H-VM-ISO-001

##### 2801 1. Assessment reference

2802 Requirement: REQ-H-VM-ISO-001 (Basic)

## 2803 2. Assessment objective

2804 Verify that the Hypervisor enforces isolation between guest virtual machines at CPU, memory, I/O, and interrupt-  
2805 handling levels, ensuring that one guest virtual machine cannot read or modify resources of another guest virtual  
2806 machine, and cannot degrade the availability of another guest virtual machine beyond what is permitted by the  
2807 Hypervisor's configured scheduling and resource-sharing policies.

## 2808 3. Assessment preparation

2809 The assessment shall have access to:

- 2810 • Documentation of the Hypervisor's isolation model (CPU scheduling, memory virtualization, IOMMU/device  
2811 assignment, virtual switching, interrupt virtualization).
- 2812 • Documentation identifying which resources are shared vs dedicated across VMs.
- 2813 • Documentation of the test environment, including the host platform, Hypervisor version, guest VM operating  
2814 systems, and relevant network and storage configuration used for the assessment.
- 2815 • A test configuration with at least two VMs on the same host, under different administrative domains or with  
2816 clearly distinct identities and data.
- 2817 • Ability to run test code inside each VM (e.g. to probe memory, devices, CPU load, and interrupt behaviour).
- 2818 • Preconditions confirming that the Hypervisor is initialized, the VMs are deployed and running, and the  
2819 relevant isolation features are enabled in accordance with the product guidance.
- 2820 • The tools necessary to perform the assessment, as applicable, including tools for configuration inspection,  
2821 monitoring, logging, and probing memory, device access, CPU load, or interrupt behaviour.
- 2822 • Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to  
2823 establish the assessed environment.

## 2824 4. Assessment activities

2825 The assessment shall at least:

### 2826 1. Memory isolation

- 2827 • Review the Hypervisor memory-assignment and mapping configuration to confirm that memory allocated  
2828 to VM B is not mapped into VM A.
- 2829 • Where the product supports Hypervisor-mediated shared-memory, device-mapping, or passthrough  
2830 mechanisms, attempt from VM A to use such mechanisms without authorization to obtain access to  
2831 memory assigned to VM B.
- 2832 • Verify that unauthorized access is not possible and that VM A cannot read or modify memory assigned to  
2833 VM B.

### 2834 2. I/O isolation

- 2835 • Identify virtual and, where applicable, passed-through physical devices assigned to each VM.
- 2836 • From VM A, attempt to access devices or storage explicitly assigned to VM B.
- 2837 • Verify that VM A can only access its own assigned devices and cannot read or modify data belonging to  
2838 VM B's devices.

### 2839 3. CPU and interrupt isolation

- 2840 • Review documentation and configuration to confirm that CPU state and interrupts are virtualized per VM.

- 2841           • Under normal operation and during stress from VM A, verify that VM B continues to operate within the  
2842 Hypervisor's configured scheduling and resource-sharing policies and does not receive spurious interrupts  
2843 or exhibit behaviour indicating direct interference beyond those policies.

#### 2844 5. Assessment verdict

- 2845           • Pass: tests and documentation confirm that CPU, memory, I/O, and interrupt-handling isolation is enforced;  
2846 one VM cannot directly read or modify another VM's resources or interfere with its execution beyond the  
2847 Hypervisor's configured scheduling and resource-sharing policies.
- 2848           • Fail: a VM can access another VM's memory or devices, or otherwise bypass the isolation guarantees or  
2849 degrade the availability of another VM beyond the Hypervisor's configured scheduling and resource-sharing  
2850 policies.

#### 2851 6. Assessment evidence

- 2852           • [SCOPE] Description of the VMs, host, and resources used in the isolation assessment.
- 2853           • [CONFIG] Hypervisor configuration for CPU scheduling, memory/IOMMU, device assignment, interrupt  
2854 virtualization, and VM definitions.
- 2855           • [DESIGN] Documentation of the Hypervisor isolation mechanisms across CPU, memory, I/O, and interrupts.
- 2856           • [OBSERVATION] / [LOG] Results of memory/I-O/CPU/interrupt tests showing blocked access to other VMs'  
2857 resources and operation within configured scheduling and resource-sharing policies under load.

### 2858 6.3.1.1.2 Assessment Case AC-H-VM-ISO-002

#### 2859 1. Assessment reference

2860 Requirement: REQ-H-VM-ISO-002 (Elevated)

#### 2861 2. Assessment objective

2862 Verify that, where the Hypervisor exposes shared or limited resources to multiple guest virtual machines, it provides a  
2863 mechanism to prevent a guest virtual machine from causing sustained denial of service to other guest virtual machines  
2864 through resource exhaustion beyond the configured policy.

#### 2865 3. Assessment preparation

2866 The assessment shall have access to:

- 2867           • Documentation identifying the shared or limited resources exposed to multiple VMs (e.g. entropy sources,  
2868 shared device backends, shared I/O services, queues, backend worker pools, or other limited shared services).
- 2869           • Documentation describing the protection mechanisms used to prevent sustained denial of service through  
2870 resource exhaustion (e.g. quotas, rate limiting, prioritization, admission control, or controlled degraded  
2871 behaviour).
- 2872           • Documentation of the relevant policy and threshold settings for the shared or limited resources under  
2873 assessment.
- 2874           • A test configuration with at least two VMs using one or more shared or limited resources on the same host.
- 2875           • Preconditions confirming that the relevant resource-sharing functions are enabled and configured in  
2876 accordance with product guidance.
- 2877           • The tools necessary to generate sustained load or exhaustion attempts from one VM and to observe resource  
2878 availability and service continuity for the other VM.
- 2879           • Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to  
2880 establish the assessed environment.

#### 2881 4. Assessment activities

2882 The assessment shall at least:

2883 1. Identification of shared or limited resources

- 2884 • Review documentation and configuration to identify the shared or limited resources exposed to multiple  
2885 VMs and the configured protection mechanisms.

2886 2. Exhaustion attempt from VM A

- 2887 • From VM A, attempt to exhaust or monopolize the identified shared or limited resource using sustained  
2888 requests, load generation, or repeated service access beyond normal usage.

2889 3. Observation of effect on VM B

- 2890 • Verify whether VM B continues to obtain access to the shared or limited resource in accordance with the  
2891 configured policy.
- 2892 • Verify that any degradation observed is limited to the behaviour defined by the configured policy and does  
2893 not result in sustained denial of service beyond that policy.

2894 4. Policy enforcement visibility

- 2895 • Verify, through configuration, status views, logs, or monitoring outputs, that the relevant protection  
2896 mechanism was active and applied during the exhaustion attempt.

2897 **5. Assessment verdict**

- 2898 • Pass: for shared or limited resources exposed to multiple VMs, the Hypervisor implements and applies  
2899 protection mechanisms that prevent sustained denial of service through resource exhaustion beyond the  
2900 configured policy.
- 2901 • Fail: a VM can monopolize a shared or limited resource in a way that causes sustained denial of service to  
2902 other VMs beyond the configured policy, or the documented protection mechanisms are absent or ineffective.

2903 **6. Assessment evidence**

- 2904 • [SCOPE] Description of the shared or limited resources and VMs used in the assessment.
- 2905 • [CONFIG] Configuration of quotas, rate limits, prioritization, admission control, or other relevant policy  
2906 settings.
- 2907 • [DESIGN] Documentation of the shared-resource protection model and enforcement mechanisms.
- 2908 • [OBSERVATION] / [LOG] Test results, logs, or monitoring outputs showing behaviour of VM A and VM B  
2909 during the exhaustion attempt and enforcement of the configured policy.

2910 **6.3.1.1.3 Assessment Case AC-H-VM-ISO-003**

2911 **1. Assessment reference**

2912 Requirement: REQ-H-VM-ISO-003 (Advanced)

2913 **2. Assessment objective**

2914 Verify that the Hypervisor strengthens isolation between workloads sharing hardware resources by implementing  
2915 mitigations against known side-channel attacks, and that these mitigations are effective in accordance with the  
2916 documented security model.

2917 **3. Assessment preparation**

2918 The assessment shall have access to:

- 2919 • Documentation describing the side-channel threat model considered (e.g. cache, branch prediction, speculative  
2920 execution, shared functional units, timing channels) and the corresponding mitigations implemented by the  
2921 Hypervisor.
- 2922 • Documentation of the test environment, including the host platform, CPU features relevant to the claimed  
2923 mitigations, Hypervisor version, and the co-residency configuration of the VMs used in the assessment.
- 2924 • Documentation of any configuration options associated with the claimed mitigations, where such options are  
2925 provided by the product.
- 2926 • A test configuration with at least two co-resident VMs on the same host.
- 2927 • The tools necessary to perform side-channel-relevant measurement workloads and to observe mitigation state  
2928 or configuration, where applicable.
- 2929 • Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to  
2930 establish the assessed environment.

#### 2931 4. Assessment activities

2932 The assessment shall at least:

- 2933 1. Mitigation identification
  - 2934 • Review documentation to identify the concrete mitigations provided against known side-channel attacks.
  - 2935 • Verify, where applicable, that any documented mitigation-related configuration settings are set in  
2936 accordance with the claimed security model.
- 2937 2. Behaviour with the implemented mitigation set
  - 2938 • Run side-channel-relevant measurement workloads in two co-resident VMs and observe whether system  
2939 behaviour is consistent with the documented mitigation objectives.
  - 2940 • Where the product provides alternative mitigation modes or profiles, and where safely possible and  
2941 consistent with product guidance, compare behaviour under different documented settings.
- 2942 3. Administrative or operational visibility
  - 2943 • Verify, where the product provides such capability, that the implemented mitigation state or selected  
2944 profile can be determined through configuration views, status commands, or logs.

#### 2945 5. Assessment verdict

- 2946 • Pass: the Hypervisor implements mitigations against known side-channel attacks as documented, and the  
2947 assessment shows behaviour consistent with strengthened isolation between workloads sharing hardware  
2948 resources in accordance with the documented security model.
- 2949 • Fail: no meaningful side-channel mitigations are provided for the claimed threat model, or the observed  
2950 behaviour is inconsistent with the documented mitigation objectives.

#### 2951 6. Assessment evidence

- 2952 • [CONFIG] Hypervisor configuration showing side-channel mitigation settings, where applicable.
- 2953 • [DESIGN] Documentation of the side-channel threat model and implemented mitigations.
- 2954 • [PLT-CAP] Platform capability evidence where relevant (e.g. CPU features used for isolation,  
2955 cache/partitioning capabilities).
- 2956 • [OBSERVATION] Measurement results or traces from side-channel-relevant tests demonstrating behaviour  
2957 consistent with the documented mitigation objectives.

2958 **6.3.1.1.4 Assessment Case AC-H-VM-ISO-004**

2959 **1. Assessment reference**

2960 Requirement: REQ-H-VM-ISO-004 (Advanced)

2961 **2. Assessment objective**

2962 Verify that all access by privileged host-environment components to guest memory, guest CPU state, or other security-  
2963 relevant guest execution state is mediated through defined, auditable guest-host interfaces, and that access outside those  
2964 interfaces is prevented or detected by the Hypervisor's isolation mechanisms.

2965 **3. Assessment preparation**

2966 The assessment shall have access to:

- 2967 • Documentation identifying the privileged host-environment components that may access guest memory, guest  
2968 CPU state, or other security-relevant guest execution state.
- 2969 • Documentation of the defined guest-host interfaces through which such access is mediated, including any  
2970 authorization, logging, or audit mechanisms associated with those interfaces.
- 2971 • Documentation describing how unauthorized or out-of-path access is prevented or detected by the Hypervisor.
- 2972 • A test environment in which at least one privileged host-environment component is present and guest-state  
2973 access through the defined interface can be exercised.
- 2974 • Preconditions confirming that the relevant guest-host interface, logging, and audit mechanisms are enabled in  
2975 accordance with product guidance.
- 2976 • The tools necessary to observe interface use, logs, audit trails, and any prevention or detection behaviour.
- 2977 • Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to  
2978 establish the assessed environment.

2979 **4. Assessment activities**

2980 The assessment shall at least:

- 2981 1. Identification of mediated interfaces
  - 2982 • Review documentation to identify which guest-host interfaces are defined for privileged access to guest  
2983 memory, guest CPU state, or other security-relevant guest execution state.
- 2984 2. Use of defined interfaces
  - 2985 • Exercise one or more defined guest-host interfaces from the privileged host environment in accordance  
2986 with the documented administrative context.
  - 2987 • Verify that access occurs through the documented interface and produces the expected audit or log  
2988 evidence.
- 2989 3. Attempted access outside defined interfaces
  - 2990 • Attempt, where feasible and safe, to access guest memory, guest CPU state, or other security-relevant  
2991 guest execution state from the privileged host environment outside the defined guest-host interfaces.
  - 2992 • Verify that such access is prevented or detected by the Hypervisor's isolation mechanisms.
- 2993 4. Auditability
  - 2994 • Verify that access through the defined interface is auditable and that attempts outside the defined interface,  
2995 where detected, generate the expected evidence or alerts.

2996 **5. Assessment verdict**

2997 • Pass: privileged host-environment access to guest memory, guest CPU state, or other security-relevant guest  
 2998 execution state is mediated through defined, auditable guest-host interfaces, and access outside those interfaces  
 2999 is prevented or detected by the Hypervisor's isolation mechanisms.

3000 • Fail: privileged host-environment components can access guest security-relevant state outside defined  
 3001 interfaces without prevention or detection, or the documented mediated interfaces are not auditable as required.

## 3002 6. Assessment evidence

3003 • [SCOPE] Description of the privileged host-environment components and guest-host interfaces in scope for the  
 3004 assessment.

3005 • [CONFIG] Configuration of guest-host interface controls, authorization settings, and audit/logging settings.

3006 • [DESIGN] Documentation of the guest-host mediation model and Hypervisor isolation mechanisms for  
 3007 privileged access.

3008 • [OBSERVATION] / [LOG] Evidence showing authorized mediated access through defined interfaces and  
 3009 prevention or detection of access attempts outside those interfaces.

## 3010 6.3.1.2 Assessment for Control Plane Isolation

### 3011 6.3.1.2.1 Assessment Case AC-H-CP-ISO-001

#### 3012 1. Assessment reference

3013 Requirement: REQ-H-CP-ISO-001 (Basic)

#### 3014 2. Assessment objective

3015 Verify that:

3016 1) The Hypervisor provides access-control mechanisms for its administrative functions and interfaces.

3017 2) The Hypervisor can be configured so that guest virtual machines cannot directly access or invoke  
 3018 administrative functions/interfaces.

3019 3) Only authenticated and authorized entities can perform management actions.

#### 3020 3. Assessment preparation

3021 The assessment shall have access to:

3022 • Documentation describing Hypervisor administrative interfaces (CLI, GUI, APIs, management agents) and  
 3023 access-control mechanisms (authentication, authorization, roles).

3024 • Documentation describing how guest VMs connect to the Hypervisor (management vs non-management  
 3025 channels).

3026 • A test configuration with:

3027 - at least one administrative account;

3028 - at least one guest VM with network connectivity;

3029 - the ability to attempt management actions from both an admin context and from within a guest VM.

#### 3030 4. Assessment activities

3031 The assessment shall at least:

3032 1) Identify all exposed administrative interfaces and confirm from configuration that authentication and  
 3033 authorization controls are configured (e.g. user accounts, roles, access policies).

3034 2) From a legitimate administrative context, perform management operations (e.g. VM create/start/stop,  
 3035 configuration changes) and verify that successful access requires authentication and that authorization rules  
 3036 (roles/permissions) are enforced.

3037 3) From within a guest VM, attempt to:

- 3038 - reach administrative endpoints (management IP/ports, APIs, management agents);
- 3039 - perform management actions using unauthenticated or guest-level credentials.

3040 Verify that:

- 3041 - direct access to administrative interfaces is blocked by configuration (e.g. routing/firewall, ACLs) or
- 3042 requires proper administrative authentication; and
- 3043 - guest VM principals are not authorized to perform management actions.

#### 3044 **5. Assessment verdict**

- 3045 • Pass: access-control mechanisms exist; Hypervisor can be configured so guest VMs cannot directly invoke
- 3046 administrative functions; and only authenticated, authorized entities can perform management actions.
- 3047 • Fail: guest VMs can directly access administrative interfaces and perform management actions; or
- 3048 administrative interfaces are accessible without proper authentication/authorization.

#### 3049 **6. Assessment evidence**

- 3050 • [CONFIG] Access-control configuration for administrative interfaces (users, roles, policies, network ACLs).
- 3051 • [DESIGN] Documentation of administrative functions/interfaces and the access-control model.
- 3052 • [OBSERVATION] / [LOG] Evidence of successful authenticated/authorized admin actions and failed/blocked
- 3053 attempts from guest VMs.

### 3054 **6.3.1.2.2 Assessment Case AC-H-CP-ISO-002**

#### 3055 **1. Assessment reference**

3056 Requirement: REQ-H-CP-ISO-002 (Elevated)

#### 3057 **2. Assessment objective**

3058 Verify that:

- 3059 1) The Hypervisor supports configuration of dedicated logical separation between administrative traffic and
- 3060 guest/data-plane traffic.
- 3061 2) The Hypervisor supports the use of distinct communication channels, ports or virtual networks for
- 3062 administrative interfaces, enabling isolation of administrative traffic from guest workloads.

#### 3063 **3. Assessment preparation**

3064 The assessment shall have access to:

- 3065 • Documentation describing how administrative interfaces are exposed (management networks, dedicated
- 3066 interfaces, ports, VLANs, virtual switches).
- 3067 • Documentation describing recommended practices for separating management and data/guest traffic.
- 3068 • A test configuration with:
- 3069 - at least one management network/interface;
- 3070 - at least one guest network/interface used by VMs;
- 3071 - the ability to inspect routing/switching and firewalling rules.

#### 3072 **4. Assessment activities**

3073 The assessment shall at least:

- 3074 1) Configure the Hypervisor according to documentation so that administrative interfaces use dedicated logical
- 3075 constructs (e.g. separate VLAN/virtual network, dedicated IP/port set, separate interface).

- 3076 2) Verify via configuration and inspection that administrative endpoints (management IP/ports, APIs, consoles)  
3077 are bound only to the designated management network/ports and not exposed on guest/data-plane networks.
- 3078 3) From a guest VM network, attempt to connect to management endpoints that should be reachable only via the  
3079 management network and verify that such attempts are blocked.
- 3080 4) From an administrative host placed on the management network, verify that management interfaces are  
3081 reachable via the dedicated channel and that guest/data traffic does not traverse this path.

### 3082 5. Assessment verdict

- 3083 • Pass: the Hypervisor supports and can be configured with distinct communication channels/ports/virtual  
3084 networks for administrative interfaces; administrative traffic is logically separated from guest/data-plane  
3085 traffic; and guest networks cannot be used to reach management endpoints in the assessed configuration.
- 3086 • Fail: no effective mechanism to separate administrative and guest/data-plane traffic; or administrative  
3087 interfaces are reachable over guest networks despite intended separation.

### 3088 6. Assessment evidence

- 3089 • [CONFIG] Network/interface configuration showing distinct management vs guest/data networks, ports and  
3090 bindings.
- 3091 • [DESIGN] Documentation of control-plane vs data-plane separation mechanisms and recommended  
3092 topologies.
- 3093 • [OBSERVATION] / [LOG] Connection tests from guest and admin networks showing blocked access from  
3094 guest side and permitted access from management side.

## 3095 6.3.1.2.3 Assessment Case AC-H-CP-ISO-003

### 3096 1. Assessment reference

3097 Requirement: REQ-H-CP-ISO-003 (Advanced)

### 3098 2. Assessment objective

3099 Verify that:

- 3100 1) The Hypervisor supports cryptographically protected communication channels for all administrative interfaces.
- 3101 2) Administrative channels can be configured for mutual authentication between the Hypervisor and  
3102 administrative clients/systems.
- 3103 3) Use of such channels reduces the risk of Unauthorized access or lateral movement (e.g. no cleartext admin  
3104 protocols, no unauthenticated management channels).

### 3105 3. Assessment preparation

3106 The assessment shall have access to:

- 3107 • Documentation of supported cryptographic protocols for administrative interfaces, supported cipher suites and  
3108 mutual authentication options (client certificates, key-based methods).
- 3109 • Documentation of procedures for configuring certificates/keys and enabling mutual authentication.
- 3110 • A test configuration where administrative interfaces can be configured to use cryptographically protected,  
3111 mutually authenticated channels, and where protocol-level inspection (e.g. via test tools) is possible.

### 3112 4. Assessment activities

3113 The assessment shall at least:

- 3114 1) For each administrative interface type (e.g. web console, API endpoint, management agent), configure  
3115 cryptographic protection according to product documentation.
- 3116 2) Enable mutual authentication where supported and configure at least one administrative client with valid  
3117 credentials and one with invalid or missing credentials.

- 3118 3) Verify, using connection attempts and protocol inspection, that:
- 3119 - administrative traffic is encrypted and uses the configured cryptographic protocol;
- 3120 - servers authenticate to clients (e.g. certificate validation); and
- 3121 - clients present valid credentials (keys/certificates) to complete administrative sessions for interfaces  
3122 configured with mutual authentication.
- 3123 4) Attempt connections without valid credentials or using obsolete/weak protocol modes (if configurable) and  
3124 verify they are rejected.
- 3125 5) Confirm there are no remaining administrative interfaces exposed in cleartext or without authentication in the  
3126 assessed configuration.

### 3127 5. Assessment verdict

- 3128 • Pass: all administrative interfaces can use cryptographically protected channels; mutual authentication can be  
3129 enabled and enforced; and Unauthorized/unauthenticated access attempts over these channels are rejected.
- 3130 • Fail: some administrative interfaces cannot be protected cryptographically; mutual authentication cannot be  
3131 enabled where required; or cleartext/unauthenticated management channels remain exposed.

### 3132 6. Assessment evidence

- 3133 • [CONFIG] Cryptographic and authentication configuration for administrative interfaces (protocol settings,  
3134 certificates/keys, mutual-auth options).
- 3135 • [DESIGN] Documentation of supported protocols, mutual authentication mechanisms and security profiles for  
3136 management channels.
- 3137 • [LOG] / [OBSERVATION] Protocol traces and connection logs showing encrypted, mutually authenticated  
3138 sessions, and failed attempts without valid credentials or with disallowed protocol modes.

## 3139 6.3.1.3 Assessment for Network Plane Separation

### 3140 6.3.1.3.1 Assessment Case AC-H-NP-ISO-001

#### 3141 1. Assessment reference

3142 Requirement: REQ-H-NP-ISO-001 (Basic)

#### 3143 2. Assessment objective

3144 Verify that:

- 3145 1) The Hypervisor maintains logical segregation between management, guest and host network planes.
- 3146 2) Virtual switching, routing or equivalent mechanisms ensure that traffic from one plane cannot traverse or  
3147 address another plane by default.
- 3148 3) Separate network contexts can be explicitly configured for each plane.

#### 3149 3. Assessment preparation

3150 The assessment shall have access to:

- 3151 • Documentation of the Hypervisor's network model (e.g. virtual switches, port groups, VLANs, routing, host  
3152 network interfaces) and the definitions of "management", "guest" and "host" planes.
- 3153 • Documentation describing how to configure separate network contexts for each plane.
- 3154 • A test configuration with at least:
- 3155 - one management network/plane,
- 3156 - one guest/data network/plane used by VMs,

- 3157 - host-level services (e.g. backup/monitoring) that may represent a host network plane,  
 3158 - tools to generate and observe traffic between these networks.

#### 3159 4. Assessment activities

3160 The assessment shall at least:

- 3161 1) Configure the Hypervisor with distinct network contexts for management, guest and host traffic as per  
 3162 documentation (e.g. separate port groups, VLANs or virtual networks).
- 3163 2) Verify via configuration and inspection that management interfaces are bound only to the management plane,  
 3164 guest VM interfaces only to the guest plane, and host services only to host/appropriate plane.
- 3165 3) From each plane (management endpoint, guest VM, host service), attempt to send traffic to addresses in the  
 3166 other planes without adding any explicit cross-plane rules and verify that:
- 3167 - by default, traffic from one plane does not reach or address endpoints in another plane;  
 3168 - no routing or switching entries exist that would allow cross-plane traffic beyond what is documented as  
 3169 default.

#### 3170 5. Assessment verdict

- 3171 • Pass: management, guest and host planes are logically segregated; by default, traffic from one plane cannot  
 3172 traverse or address another; and separate network contexts for each plane can be configured and observed.
- 3173 • Fail: planes are not effectively segregated; or default configuration allows cross-plane traffic contrary to the  
 3174 requirement.

#### 3175 6. Assessment evidence

- 3176 • [CONFIG] Network configuration (e.g. virtual switches, port groups, VLAN settings, IP addressing and  
 3177 bindings of interfaces to planes).
- 3178 • [DESIGN] Documentation of network plane definitions and segregation mechanisms.
- 3179 • [OBSERVATION] / [LOG] Packet captures, routing/switching tables and connection tests showing lack of  
 3180 cross-plane reachability in the default/segregated configuration.

### 3181 6.3.1.3.2 Assessment Case AC-H-NP-ISO-002

#### 3182 1. Assessment reference

3183 Requirement: REQ-H-NP-ISO-002 (Elevated)

#### 3184 2. Assessment objective

3185 Verify that:

- 3186 1) The Hypervisor supports configuration of explicit traffic filtering and access-control rules between network  
 3187 planes.
- 3188 2) These rules can be enforced at the Hypervisor's virtual network layer or other enforcement points under its  
 3189 control.
- 3190 3) Only cross-plane traffic explicitly permitted by policy (e.g. management/orchestration flows) is allowed to  
 3191 cross plane boundaries.

#### 3192 3. Assessment preparation

3193 The assessment shall have access to:

- 3194 • Documentation of mechanisms for defining traffic filtering and access-control between planes (e.g. virtual  
 3195 switch ACLs, distributed firewalls, router filters, security groups).
- 3196 • Documentation of example policies permitting specific management/orchestration flows.
- 3197 • A test configuration with:

- 3198 - management, guest and, where applicable, host planes as in AC-H-NP-ISO-001;
- 3199 - at least one management/orchestration system that legitimately needs to communicate across planes.

#### 3200 4. Assessment activities

3201 The assessment shall at least:

- 3202 1) Define and apply explicit cross-plane filtering rules at the Hypervisor virtual network layer or equivalent  
3203 enforcement point to:
- 3204 - allow only documented management/orchestration flows (e.g. from management plane to guest VM  
3205 agents on specific ports); and
  - 3206 - deny all other cross-plane traffic by default.
- 3207 2) From each plane, generate test traffic to other planes:
- 3208 - traffic matching permitted flows;
  - 3209 - traffic that should be blocked (e.g. arbitrary guest-to-management or guest-to-host connections,  
3210 unexpected ports/protocols).
- 3211 3) Verify that:
- 3212 - permitted flows succeed and follow the configured policy;
  - 3213 - non-permitted cross-plane traffic is blocked at the configured enforcement point; and
  - 3214 - changes to the policy (e.g. adding/removing a rule) have the expected effect on cross-plane  
3215 communication.

#### 3216 5. Assessment verdict

- 3217 • Pass: the Hypervisor provides configurable filtering/access-control rules between planes; these rules are  
3218 enforced at the virtual network or equivalent layer; and only explicitly permitted cross-plane traffic is allowed  
3219 while other cross-plane traffic is blocked.
- 3220 • Fail: cross-plane filtering cannot be configured; or enforcement is ineffective; or cross-plane traffic bypasses  
3221 the configured policy.

#### 3222 6. Assessment evidence

- 3223 • [CONFIG] Defined cross-plane filtering/ACL rules and their association with management/guest/host  
3224 networks.
- 3225 • [DESIGN] Documentation of enforcement mechanisms and their role in plane separation.
- 3226 • [OBSERVATION] / [LOG] Connection attempts and packet traces showing success of permitted flows and  
3227 blocking of non-permitted cross-plane traffic.

#### 3228 6.3.1.3.3 Assessment Case AC-H-NP-ISO-003

##### 3229 1. Assessment reference

3230 Requirement: REQ-H-NP-ISO-003 (Advanced)

##### 3231 2. Assessment objective

3232 Verify that:

- 3233 1) The Hypervisor supports isolation of management traffic using either:
- 3234 - dedicated physical interfaces; or
  - 3235 - cryptographically protected network channels.
- 3236 2) Management traffic is protected against interception or modification by guest or host workloads.

- 3237 3) Where dedicated physical interfaces are not used, cryptographic protection provides confidentiality, integrity  
3238 and authentication equivalent to a dedicated management network.

### 3239 3. Assessment preparation

3240 The assessment shall have access to:

- 3241 • Documentation describing supported approaches for isolating management traffic (e.g. dedicated NICs/ports,  
3242 out-of-band networks, IPsec/TLS tunnels, overlay management networks).
- 3243 • Documentation of cryptographic protocols and authentication mechanisms used when dedicated physical  
3244 interfaces are not available.
- 3245 • A test configuration with:
  - 3246 - either a dedicated physical management interface/network or a cryptographically protected management  
3247 channel;
  - 3248 - guest workloads generating network traffic;
  - 3249 - the ability to capture and inspect traffic from guest-accessible paths.

### 3250 4. Assessment activities

3251 Depending on the isolation method, the assessment shall at least:

3252 Dedicated physical interface case

- 3253 1) Configure the Hypervisor so that all management interfaces use a dedicated physical NIC or physical  
3254 management network segment.
- 3255 2) Verify via configuration and inspection that:
  - 3256 - management traffic egresses/ingresses only via the dedicated physical interface/network;
  - 3257 - guest VMs and host non-management workloads do not have access to that interface/network (e.g. no  
3258 virtual NICs or routes pointing to it).
- 3259 3) From guest networks and host non-management interfaces, capture traffic and verify that management traffic  
3260 is not visible on those paths.

3261 Cryptographically protected channel case (when dedicated physical interfaces are not used)

- 3262 4) Configure management traffic over a cryptographically protected channel (e.g. IPsec, TLS VPN, mutually  
3263 authenticated TLS overlay) between the Hypervisor and management endpoints.
- 3264 5) Verify via protocol inspection and logs that:
  - 3265 - management traffic is encrypted and integrity-protected;
  - 3266 - mutual authentication between Hypervisor and management entities is in place.
- 3267 6) From guest networks and host non-management workloads, capture traffic and verify that:
  - 3268 - management traffic is only observable in encrypted form (no cleartext management protocols); and
  - 3269 - attempts to inject or modify management traffic are rejected or fail to affect the management channel.

### 3270 5. Assessment verdict

- 3271 • Pass: the Hypervisor supports either dedicated physical interfaces or cryptographically protected channels to  
3272 isolate management traffic; management traffic is not exposed in cleartext or modifiable by guest/host  
3273 workloads; and cryptographic protection, where used instead of dedicated interfaces, provides confidentiality,  
3274 integrity and authentication equivalent to a dedicated management network.
- 3275 • Fail: management traffic is visible or modifiable from guest or host workloads; dedicated management  
3276 interfaces or equivalent cryptographic protection cannot be configured; or cleartext management flows appear  
3277 on shared paths without adequate isolation.

3278 **6. Assessment evidence**

- 3279 • [CONFIG] Network and interface configuration showing use of dedicated management interfaces or  
3280 cryptographically protected management channels.
- 3281 • [DESIGN] Documentation of management traffic isolation model and, where applicable, cryptographic  
3282 protocols and authentication methods.
- 3283 • [PLT-CAP] Evidence of platform/network capabilities used (e.g. dedicated NICs, VPN/IPsec support).
- 3284 • [OBSERVATION] Packet captures and logs demonstrating that management traffic uses the isolated interface  
3285 or encrypted channel and is not exposed to guest or non-management workloads.

3286 **6.3.1.4 Assessment for Boot Chain Integrity Verification**3287 **6.3.1.4.1 Assessment Case AC-H-B-INT-001**3288 **1. Assessment reference**

3289 Requirement: REQ-H-B-INT-001 (Basic)

3290 **2. Assessment objective**

3291 Verify that:

- 3292 1) The Hypervisor implements an integrity verification mechanism for its executable core (kernel or microkernel)  
3293 based on a trusted reference value.
- 3294 2) Integrity verification of the executable core is invoked before any guest workloads are initialized or  
3295 management interfaces are exposed.
- 3296 3) If integrity verification of the executable core fails, the Hypervisor prevents initialization of guest workloads  
3297 and activation of management interfaces.

3298 **3. Assessment preparation**

3299 The assessment shall have access to:

- 3300 • Documentation describing the executable core integrity verification mechanism, including how trusted  
3301 reference values are established and stored.
- 3302 • Documentation of the boot sequence, indicating when integrity verification occurs relative to guest  
3303 initialization and management interface activation.
- 3304 • A test configuration where the Hypervisor can be booted with:
- 3305 - an intact, correctly signed/verified executable core; and
- 3306 - a deliberately altered or invalid executable core (or equivalent method) to trigger verification failure,  
3307 consistent with product guidance.

3308 **4. Assessment activities**

3309 The assessment shall at least:

- 3310 1) Review documentation and configuration to identify the integrity verification mechanism, the trusted reference  
3311 values, and the point in the boot sequence at which verification is invoked.
- 3312 2) Boot the Hypervisor with an intact executable core and verify that:
- 3313 - integrity verification is invoked as documented;
- 3314 - the Hypervisor completes initialization; and
- 3315 - guest workloads and management interfaces can be initialized/activated normally.
- 3316 3) Boot the Hypervisor with an executable core that fails integrity verification (e.g. corrupted or otherwise invalid  
3317 according to the documented mechanism) and verify that:

- 3318 - the integrity verification detects the failure; and
- 3319 - guest workloads are not initialized and management interfaces are not activated.

### 3320 5. Assessment verdict

- 3321 • Pass: executable-core integrity verification is based on a trusted reference value; it is invoked before guest and  
3322 management initialization; and failure of verification prevents initialization of guest workloads and activation  
3323 of management interfaces.
- 3324 • Fail: integrity verification is not implemented or not based on a trusted reference; it does not occur before  
3325 guest/management initialization; or verification failure does not block guest and management initialization.

### 3326 6. Assessment evidence

- 3327 • [DESIGN] Documentation of the executable core integrity verification mechanism and boot sequence.
- 3328 • [CONFIG] Boot and verification configuration, including references to trusted reference values.
- 3329 • [OBSERVATION] / [LOG] Boot logs or traces showing successful verification and normal startup, and  
3330 logs/traces showing detection of an integrity failure and prevention of guest/management initialization.

### 3331 6.3.1.4.2 Assessment Case AC-H-B-INT-002

#### 3332 1. Assessment reference

3333 Requirement: REQ-H-B-INT-002 (Elevated)

#### 3334 2. Assessment objective

3335 Verify that:

- 3336 1) The Hypervisor implements mechanisms to participate in a verifiable chain of trust for early boot components  
3337 that load the Hypervisor.
- 3338 2) Each boot stage delivered as part of the Hypervisor product, including at least any bootloader delivered with  
3339 the Hypervisor and the Hypervisor executable core, validates the integrity and authenticity of the next stage  
3340 before transferring control.
- 3341 3) If this validation fails at any such stage, the Hypervisor prevents initialization of guest workloads and  
3342 activation of management interfaces.

#### 3343 3. Assessment preparation

3344 The assessment shall have access to:

- 3345 • Documentation of the boot chain architecture for components delivered as part of the Hypervisor product (e.g.  
3346 bootloader(s), Hypervisor kernel/microkernel, early management services).
- 3347 • Documentation of how each boot stage validates integrity and authenticity of the next stage, including where  
3348 the trust anchors or reference values reside and how they integrate into a platform secure-boot or equivalent  
3349 verified boot chain.
- 3350 • A test configuration that allows:
- 3351 - a normal boot with all stages intact; and
- 3352 - a controlled failure in integrity/authenticity validation of one of the Hypervisor-delivered boot stages  
3353 (e.g. modified image, invalid signature), consistent with product guidance.

#### 3354 4. Assessment activities

3355 The assessment shall at least:

- 3356 1) Review documentation to identify all boot stages delivered as part of the Hypervisor product and the validation  
3357 performed at each stage (e.g. signature check, hash comparison).

- 3358 2) In a normal configuration, boot the system and verify from logs or diagnostic outputs that each Hypervisor-  
 3359 delivered boot stage validates the integrity/authenticity of the next stage before passing control, and that the  
 3360 overall boot completes with guest and management initialization available.
- 3361 3) Introduce a deliberate integrity/authenticity failure in one Hypervisor-delivered boot stage (e.g. corrupted or  
 3362 unsigned image, or image signed with an untrusted key) and boot the system.
- 3363 4) Verify that:
- 3364 - the failing validation is detected at the appropriate stage; and
- 3365 - guest workloads are not initialized and management interfaces are not activated following the failed  
 3366 validation.

### 3367 5. Assessment verdict

- 3368 • Pass: all Hypervisor-delivered boot stages validate the integrity and authenticity of the next stage before  
 3369 transferring control; validation failures at any such stage are detected; and, when such failures occur, the  
 3370 Hypervisor prevents initialization of guest workloads and activation of management interfaces.
- 3371 • Fail: some Hypervisor-delivered boot stages do not validate the next stage; validation failures are not detected;  
 3372 or validation failures do not prevent guest/management initialization.

### 3373 6. Assessment evidence

- 3374 • [DESIGN] Documentation of the Hypervisor-related boot chain and per-stage validation mechanisms,  
 3375 including integration into a platform secure/verified boot chain.
- 3376 • [CONFIG] Boot configuration and key/certificate or reference-value configuration for Hypervisor-delivered  
 3377 stages.
- 3378 • [OBSERVATION] / [LOG] Boot logs or traces showing successful chained validation of stages and separate  
 3379 logs/traces demonstrating detection of a validation failure and corresponding prevention of guest/management  
 3380 initialization.

#### 3381 6.3.1.4.3 Assessment Case AC-H-B-INT-003

##### 3382 1. Assessment reference

3383 Requirement: REQ-H-B-INT-003 (Advanced)

##### 3384 2. Assessment objective

3385 Verify that:

- 3386 1) The Hypervisor relies on integrity and authenticity information for the boot chain on which it depends, as  
 3387 provided by verified boot mechanisms using hardware or software roots of trust.
- 3388 2) This information covers components used to load the Hypervisor, including the bootloader, the Hypervisor  
 3389 kernel or microkernel, and management services.
- 3390 3) The Hypervisor prevents initialization of guest workloads and management services if the verification status of  
 3391 any component in this boot chain cannot be established or is reported as failed.

##### 3392 3. Assessment preparation

3393 The assessment shall have access to:

- 3394 • Documentation of the verified boot mechanism and roots of trust (hardware or software) used to provide  
 3395 integrity and authenticity information for the Hypervisor boot chain components.
- 3396 • Documentation describing how the Hypervisor consumes or relies on verification status (e.g. secure boot APIs,  
 3397 measurement/attestation interfaces, verified boot status flags) for the bootloader, Hypervisor core and  
 3398 management services.
- 3399 • A test configuration where:
- 3400 - the platform verified boot mechanism is enabled and correctly configured;

- 3401 - the Hypervisor is deployed with all boot chain components covered by the verified boot mechanism;
- 3402 - a failure or indeterminate verification status for at least one boot-chain component can be induced in a
- 3403 controlled manner, consistent with product guidance.

#### 3404 4. Assessment activities

3405 The assessment shall at least:

- 3406 1) Review documentation to confirm which boot-chain components (bootloader, Hypervisor kernel/microkernel,
- 3407 management services) are covered by verified boot and how verification status is made available to the
- 3408 Hypervisor.
- 3409 2) In a correctly configured environment, boot the system and verify that:
- 3410 - verification status for each relevant boot-chain component is successfully established by the verified boot
- 3411 mechanism;
- 3412 - the Hypervisor relies on this status when completing its initialization; and
- 3413 - guest workloads and management services are initialized only when all required components are in a
- 3414 verified-good state.
- 3415 3) Introduce a controlled failure or indeterminate status for one boot-chain component (e.g. a component failing
- 3416 verification according to the verified boot mechanism) and reboot.
- 3417 4) Verify that:
- 3418 - the Hypervisor detects that verification status for that component is failed or cannot be established; and
- 3419 - the Hypervisor prevents initialization of guest workloads and management services in this condition.

#### 3420 5. Assessment verdict

- 3421 • Pass: the Hypervisor relies on verified boot information from hardware or software roots of trust for the
- 3422 bootloader, Hypervisor core and management services; and when verification status for any such component is
- 3423 missing or failed, guest workloads and management services are not initialized.
- 3424 • Fail: verified boot information is not used for all required components; the Hypervisor initializes
- 3425 guest/management services despite failed or indeterminate verification status; or the Hypervisor cannot obtain
- 3426 or act on verification status for the boot chain.

#### 3427 6. Assessment evidence

- 3428 • [DESIGN] Documentation of the verified boot mechanism, roots of trust and the way verification status for
- 3429 boot-chain components is exposed to and used by the Hypervisor.
- 3430 • [PLT-CAP] Evidence of platform capabilities providing verified boot / root-of-trust support used by the
- 3431 Hypervisor.
- 3432 • [CONFIG] Verified boot configuration and Hypervisor integration settings.
- 3433 • [OBSERVATION] / [LOG] Boot logs or status outputs showing verified-good operation and, separately,
- 3434 logs/status showing detection of failed/unknown verification and prevention of guest/management
- 3435 initialization.

### 3436 6.3.1.5 Assessment for Guest VM Image Integrity Verification

#### 3437 6.3.1.5.1 Assessment Case AC-H-IMG-INT-001

##### 3438 1. Assessment reference

3439 Requirement: REQ-H-IMG-INT-001 (Basic)

##### 3440 2. Assessment objective

3441 Verify that:

- 3442 1) The Hypervisor implements an integrity verification mechanism for guest VM images based on trusted  
3443 reference values.
- 3444 2) Integrity verification is automatically invoked before executing or provisioning any guest VM.
- 3445 3) If integrity verification for a guest VM image fails, the Hypervisor prevents execution or instantiation of that  
3446 guest VM.

### 3447 3. Assessment preparation

3448 The assessment shall have access to:

- 3449 • Documentation describing the guest VM image integrity verification mechanism, including how trusted  
3450 reference values (e.g. hashes, manifests) are established and stored.
- 3451 • Documentation of the VM provisioning/startup sequence, indicating when image integrity verification is  
3452 invoked.
- 3453 • A test environment with:
  - 3454 - at least one repository or storage location for guest VM images;
  - 3455 - at least one valid guest VM image;
  - 3456 - the ability to create a deliberately modified (integrity-failing) copy of a guest VM image in a controlled  
3457 manner.

### 3458 4. Assessment activities

3459 The assessment shall at least:

- 3460 1) Review documentation and configuration to identify:
  - 3461 - which guest VM images are subject to integrity verification;
  - 3462 - the trusted reference values used;
  - 3463 - the point in the provisioning/startup process where verification occurs.
- 3464 2) With an unmodified guest VM image, request provisioning/startup of the VM and verify, using logs or status  
3465 outputs, that:
  - 3466 - integrity verification is invoked automatically;
  - 3467 - the VM is instantiated or started successfully when verification succeeds.
- 3468 3) Replace or alter a guest VM image so that it fails integrity verification according to the documented  
3469 mechanism, then request provisioning/startup of that VM. Verify that:
  - 3470 - the Hypervisor reports an integrity verification failure;
  - 3471 - execution or instantiation of that VM is prevented.

### 3472 5. Assessment verdict

- 3473 • Pass: guest VM image integrity verification based on trusted reference values is automatically invoked before  
3474 VM provisioning/execution; and a verification failure for an image prevents that VM from being instantiated  
3475 or executed.
- 3476 • Fail: integrity verification is not implemented or not based on trusted reference values; it is not invoked  
3477 automatically; or VMs can be instantiated despite integrity verification failures.

### 3478 6. Assessment evidence

- 3479 • [DESIGN] Documentation of the image integrity verification mechanism and its place in the VM lifecycle.
- 3480 • [CONFIG] Configuration showing enabled integrity verification and reference values for guest VM images.

- 3481       • [LOG] / [OBSERVATION] Records of successful verification and VM startup for a valid image, and records  
3482 showing verification failure and blocked startup for a modified image.

### 3483 6.3.1.5.2 Assessment Case AC-H-IMG-INT-002

#### 3484 1. Assessment reference

3485 Requirement: REQ-H-IMG-INT-002 (Elevated)

#### 3486 2. Assessment objective

3487 Verify that:

- 3488       1) The Hypervisor verifies both integrity and authenticity of guest VM images before execution or provisioning.  
3489       2) Authenticity verification establishes that the image originates from a trusted source and has not been altered  
3490 since approval.  
3491       3) The Hypervisor prevents execution or instantiation of any guest VM image whose integrity or authenticity  
3492 cannot be successfully verified.

#### 3493 3. Assessment preparation

3494 The assessment shall have access to:

- 3495       • Documentation describing integrity and authenticity verification for guest VM images, including trust model  
3496 (e.g. signatures, certificate chains, trust stores, or equivalent cryptographic schemes).  
3497       • Documentation of how trusted publishers/sources are represented and configured.  
3498       • A test environment with:  
3499       - one or more guest VM images created and "approved" using the trusted process (e.g. signed or otherwise  
3500 bound to a trusted source);  
3501       - images modified or created so that integrity or authenticity verification will fail (e.g. tampered content,  
3502 invalid or missing signature, signature under an untrusted key), prepared in a controlled way.

#### 3503 4. Assessment activities

3504 The assessment shall at least:

- 3505       1) Review documentation and configuration to identify:  
3506       - how integrity and authenticity checks are combined;  
3507       - what constitutes a "trusted source" and how trust anchors or approval data are configured.  
3508       2) For an image that is correctly produced and approved:  
3509       - request provisioning/startup of a VM from that image;  
3510       - verify that integrity and authenticity verification succeed;  
3511       - confirm that the VM is instantiated/executed.  
3512       3) For images that are invalid in different ways (e.g. modified after approval, missing or invalid signature, signed  
3513 by an untrusted source):  
3514       - request provisioning/startup;  
3515       - verify that the Hypervisor reports integrity/authenticity verification failure or inability to verify;  
3516       - confirm that execution/instantiation of such VMs is prevented.

#### 3517 5. Assessment verdict

- 3518       • Pass: integrity and authenticity of guest VM images are verified prior to provisioning/execution; only images  
3519 from trusted sources that have not been altered since approval can be instantiated; and images whose  
3520 integrity/authenticity cannot be verified are blocked.

- 3521 • Fail: authenticity is not verified; untrusted or modified images can be instantiated; or failure to verify  
3522 integrity/authenticity does not prevent VM instantiation.

## 3523 6. Assessment evidence

- 3524 • [DESIGN] Documentation of the guest VM image trust model, including integrity and authenticity verification  
3525 and trusted sources.
- 3526 • [CONFIG] Trust store or equivalent configuration defining trusted publishers/sources and related policies.
- 3527 • [LOG] / [OBSERVATION] Records showing successful verification and startup for approved images and  
3528 failure/blocked startup for images with invalid integrity or authenticity.

### 3529 6.3.1.5.3 Assessment Case AC-H-IMG-INT-003

#### 3530 1. Assessment reference

3531 Requirement: REQ-H-IMG-INT-003 (Advanced)

#### 3532 2. Assessment objective

3533 Verify that:

- 3534 1) The Hypervisor uses integrity and authenticity verification for all guest VM images before execution or  
3535 provisioning, based on trust anchors stored as protected trust material.
- 3536 2) Protected trust material (keys, certificates or other trust anchors) is stored and managed so that it is not  
3537 modifiable during normal operational state via standard administrative or guest-facing interfaces.
- 3538 3) If verification of a guest VM image using the protected trust material fails, or the verification status cannot be  
3539 established, the Hypervisor prevents execution or instantiation of that guest VM.

#### 3540 3. Assessment preparation

3541 The assessment shall have access to:

- 3542 • Documentation describing:
- 3543 - the nature and storage of protected trust material (e.g. trust stores, key vaults, secure configuration  
3544 objects);
- 3545 - the protections that prevent modification of trust material during normal operation;
- 3546 - the process for controlled update of trust material (e.g. dedicated security management procedure).
- 3547 • A test environment with:
- 3548 - at least one valid guest VM image verified against the protected trust material;
- 3549 - the ability to simulate conditions where verification fails or the trust anchors no longer match the image  
3550 (in a controlled manner);
- 3551 - an administrative interface reflective of "normal operational state".

#### 3552 4. Assessment activities

3553 The assessment shall at least:

- 3554 1) Review documentation and configuration to confirm:
- 3555 - which trust anchors are used to verify guest VM images;
- 3556 - where and how this trust material is stored;
- 3557 - which operations are permitted in normal operational state versus dedicated security update procedures.
- 3558 2) In normal operation, attempt to modify or replace trust material through standard administrative channels (e.g.  
3559 routine configuration interfaces) and verify that such modifications are prevented or require a controlled  
3560 security management process.

- 3561 3) With valid trust material and a correctly approved guest VM image, request provisioning/startup and verify  
3562 that integrity/authenticity verification succeeds and the VM is instantiated.
- 3563 4) Create a test condition where verification fails or becomes indeterminate for a guest VM image when checked  
3564 against the protected trust material (e.g. image altered after approval, trust anchor no longer matches image),  
3565 and request provisioning/startup. Verify that:
- 3566 - verification failure or inability to establish verification status is detected;
- 3567 - the Hypervisor prevents execution or instantiation of that VM.

### 3568 5. Assessment verdict

- 3569 • Pass: all guest VM images are verified for integrity and authenticity against protected trust material; trust  
3570 anchors are not modifiable during normal operation; and images that fail or cannot be verified against this  
3571 material are prevented from execution/instantiation.
- 3572 • Fail: trust material is modifiable in normal operation without a controlled process; images can be instantiated  
3573 without verification against protected trust anchors; or failed/indeterminate verification does not prevent VM  
3574 instantiation.

### 3575 6. Assessment evidence

- 3576 • [DESIGN] Documentation of protected trust material, its storage and update procedures, and its use in image  
3577 verification.
- 3578 • [CONFIG] Configuration of trust stores or equivalent structures, including permissions or controls that protect  
3579 them in normal operation.
- 3580 • [LOG] / [OBSERVATION] Evidence of successful verification and startup for valid images, and evidence of  
3581 blocked startup when verification against protected trust material fails or is not possible.

## 3582 6.3.1.6 Assessment for Runtime Integrity Protection

### 3583 6.3.1.6.1 Assessment Case AC-H-RP-INT-002

#### 3584 1. Assessment reference

3585 Requirement: REQ-H-RP-INT-002 (Elevated)

#### 3586 2. Assessment objective

3587 Verify that:

- 3588 1) The Hypervisor enforces integrity protection for its configuration and control plane so that only authenticated  
3589 and authorized modifications can be applied.
- 3590 2) The Hypervisor generates audit events for any security-relevant changes to configuration or internal control  
3591 logic.

#### 3592 3. Assessment preparation

3593 The assessment shall have access to:

- 3594 • Documentation describing:
- 3595 - the Hypervisor configuration and control-plane model;
- 3596 - which configuration items and control-plane elements are considered security-relevant;
- 3597 - which roles or identities are permitted to modify them;
- 3598 - the interfaces (local/remote) through which such modifications are performed.
- 3599 • Documentation of the auditing or logging mechanism for configuration and control-plane changes.
- 3600 • A test configuration with:

- 3601 - at least one administrative account/role with privileges to change security-relevant configuration or  
3602 control-plane logic;
- 3603 - at least one non-privileged or Unauthorized account/role;
- 3604 - access to configuration/control-plane interfaces under test;
- 3605 - audit/logging enabled and accessible for inspection.

#### 3606 4. Assessment activities

3607 The assessment shall at least:

- 3608 1) Review documentation and Hypervisor settings to identify:
- 3609 - the set of security-relevant configuration parameters and control-plane elements;
- 3610 - the authentication and authorization mechanisms governing their modification;
- 3611 - how audit events are generated and recorded for such changes.
- 3612 2) Using an authenticated and authorized administrative account:
- 3613 - perform representative security-relevant configuration changes and, where supported, changes to control-  
3614 plane logic (for example enabling/disabling a security feature, changing access-control rules, modifying  
3615 isolation-relevant parameters);
- 3616 - verify that the Hypervisor accepts these changes;
- 3617 - verify that corresponding audit events are generated, including at least information on the acting identity,  
3618 the nature of the change, and time of the event.
- 3619 3) Using an unauthenticated context, a non-privileged account, or an interface that should not permit these  
3620 changes:
- 3621 - attempt to perform the same or equivalent security-relevant modifications;
- 3622 - verify that these attempts are rejected and do not alter configuration or control-plane state;
- 3623 - verify that failed or denied modification attempts are logged as appropriate (if specified in the product  
3624 behaviour).

#### 3625 5. Assessment verdict

- 3626 • Pass: only authenticated and authorized entities can apply security-relevant configuration or control-plane  
3627 changes; attempts by unauthenticated or Unauthorized entities are rejected without altering state; and audit  
3628 events are generated for security-relevant changes to configuration or internal control logic.
- 3629 • Fail: security-relevant changes can be made without proper authentication or authorization; or such changes do  
3630 not result in audit events; or Unauthorized attempts are not prevented.

#### 3631 6. Assessment evidence

- 3632 • [DESIGN] Documentation of the configuration and control-plane model, including the definition of security-  
3633 relevant changes and the access-control model.
- 3634 • [CONFIG] Hypervisor configuration showing role/permission assignments for configuration and control-plane  
3635 interfaces, and logging/audit configuration.
- 3636 • [LOG] Audit or event logs capturing successful authorized changes and rejected Unauthorized attempts.
- 3637 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating  
3638 acceptance of authorized changes and rejection of Unauthorized ones.

## 3639 6.3.1.6.2 Assessment Case AC-H-RP-INT-003

## 3640 1. Assessment reference

3641 Requirement: REQ-H-RP-INT-003 (Advanced)

## 3642 2. Assessment objective

3643 Verify that:

- 3644 1) The Hypervisor maintains the integrity of its security-critical runtime components and detects Unauthorized  
3645 modification of configuration state or control logic used to enforce security policies or isolation.
- 3646 2) The Hypervisor maintains a trusted baseline of the integrity state of these security-critical runtime components  
3647 and performs periodic or continuous integrity validation against this baseline.
- 3648 3) Upon detecting a deviation from the trusted baseline, the Hypervisor automatically triggers protective actions  
3649 (such as isolating the affected component, disabling affected functions, or restoring from a trusted state) and  
3650 generates a security alert that cannot be suppressed by the affected component.

## 3651 3. Assessment preparation

3652 The assessment shall have access to:

- 3653 • Documentation that:
- 3654 - identifies which runtime components are considered security-critical for policy enforcement and isolation  
3655 (for example security modules, policy engines, core control-plane daemons);
  - 3656 - describes how the trusted baseline for these components is established, stored, and updated;
  - 3657 - explains whether integrity validation is periodic, continuous, or event-driven, and how it is configured;
  - 3658 - describes the set of protective actions that can be triggered when deviations are detected;
  - 3659 - describes the alerting mechanism and how alerts are exposed to management/monitoring systems.
- 3660 • A Hypervisor test deployment where runtime integrity validation is enabled and its status can be observed (for  
3661 example through logs, status commands, or telemetry).
- 3662 • A controlled means, consistent with vendor guidance, to simulate or induce an integrity deviation affecting at  
3663 least one security-critical runtime component or its configuration state (for example a supported test hook,  
3664 debug mode, test image, or vendor-provided integrity-test procedure).

## 3665 4. Assessment activities

3666 The assessment shall at least:

- 3667 1) Review documentation and configuration to confirm:
- 3668 - the list or classes of security-critical runtime components covered by integrity validation;
  - 3669 - the method used to create and protect the trusted baseline;
  - 3670 - the frequency or trigger conditions for integrity checks;
  - 3671 - the configured protective actions and alerting behaviour.
- 3672 2) In a normal (non-tampered) configuration:
- 3673 - operate the Hypervisor under representative load for a suitable period;
  - 3674 - verify from logs, status outputs, or telemetry that integrity validation is active (for example periodic  
3675 checks, continuous monitoring, or attestation events for the defined components);
  - 3676 - confirm that no integrity deviation alerts are present when the baseline is intact.
- 3677 3) Introduce a controlled integrity deviation affecting a security-critical runtime component or its configuration  
3678 state, using a method sanctioned for testing:
- 3679 - allow the Hypervisor to perform its runtime validation;

- 3680 - verify that the deviation from the trusted baseline is detected;
- 3681 - verify that the documented protective actions are automatically triggered (for example isolation or  
3682 disablement of the affected component, restoration from a trusted baseline, or equivalent measures);
- 3683 - verify that a security alert is generated and remains visible via Hypervisor or management/monitoring  
3684 interfaces, and that this alert cannot be cleared or suppressed solely by the affected component.

### 3685 5. Assessment verdict

- 3686 • Pass: security-critical runtime components are covered by a trusted baseline; the Hypervisor performs periodic  
3687 or continuous validation against this baseline; controlled deviations are detected; documented protective  
3688 actions are automatically applied; and a security alert is generated that the affected component cannot  
3689 suppress.
- 3690 • Fail: runtime integrity validation does not cover security-critical components; is not active or observable;  
3691 deviations are not detected; protective actions are not triggered; or alerts can be suppressed or removed by the  
3692 affected component.

### 3693 6. Assessment evidence

- 3694 • [DESIGN] Documentation of security-critical runtime components, trusted baseline creation and protection,  
3695 runtime integrity mechanisms, and protective/alerting behaviour.
- 3696 • [CONFIG] Configuration enabling runtime integrity validation, including monitored components, check  
3697 parameters, and alerting/protection settings.
- 3698 • [LOG] Integrity-check logs and security alerts from normal operation and from the controlled deviation  
3699 scenario, including timestamps and component identifiers.
- 3700 • [OBSERVATION] Test records or monitoring outputs showing integrity checks in operation, detection of the  
3701 induced deviation, execution of protective actions, and persistence of the generated alert.
- 3702 • [PLT-CAP] Where hardware- or platform-assisted runtime integrity mechanisms are used, evidence of the  
3703 underlying platform capabilities and their configuration.

## 3704 6.3.1.7 Assessment for Remote Attestation

### 3705 6.3.1.7.1 Assessment Case AC-H-RA-INT-003

#### 3706 1. Assessment reference

3707 Requirement: REQ-H-RA-INT-003 (Advanced)

#### 3708 2. Assessment objective

3709 Verify that:

- 3710 1) The Hypervisor supports measured boot and generates attestation evidence describing its bootloader, kernel or  
3711 microkernel, security-critical configuration, and essential runtime state.
- 3712 2) The attestation evidence is cryptographically verifiable by an authorized verifier and is bound to the specific  
3713 Hypervisor instance for which the measurements were produced.
- 3714 3) Attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay.
- 3715 4) The attestation evidence is limited to the minimum information necessary to verify the integrity and  
3716 configuration state of the Hypervisor.

#### 3717 3. Assessment preparation

3718 The assessment shall have access to:

- 3719 • Documentation describing:
- 3720 - the measured-boot and attestation design, including which components and configuration items are  
3721 measured (bootloader, kernel or microkernel, security-critical configuration, essential runtime state);

- 3722 - how measurements are collected, stored, and associated with a Hypervisor instance;
- 3723 - the format and export mechanism of the attestation evidence (for example data fields, encoding, transport  
3724 or API);
- 3725 - the cryptographic mechanisms used to protect the attestation evidence and bind it to a Hypervisor  
3726 instance (for example signature algorithms, keys, certificate chains);
- 3727 - the protection model for attestation keys and measurement data, including which entities are permitted to  
3728 read, manage, or update them.
- 3729 • A test configuration with:
  - 3730 - a Hypervisor instance with measured boot and attestation enabled according to product documentation;
  - 3731 - an interface to request attestation evidence (for example CLI, API, or management interface);
  - 3732 - a verification tool or reference verifier configured with the relevant trust anchors to validate the  
3733 attestation evidence;
  - 3734 - at least one guest or non-privileged context that can be used to attempt Unauthorized access to attestation  
3735 keys or raw measurement data.

#### 3736 4. Assessment activities

3737 The assessment shall at least:

- 3738 1) Review documentation and Hypervisor settings to identify:
  - 3739 - the set of elements covered by measurements (bootloader, kernel or microkernel, security-critical  
3740 configuration, essential runtime state);
  - 3741 - how these measurements are represented in the attestation evidence;
  - 3742 - which cryptographic algorithms and keys are used to protect and sign the evidence;
  - 3743 - how the attestation evidence is bound to a specific Hypervisor instance (for example instance identifiers,  
3744 platform identity, or key binding);
  - 3745 - how attestation keys and measurement data are stored and access-controlled.
- 3746 2) After a normal boot of the Hypervisor:
  - 3747 - request attestation evidence using the documented interface;
  - 3748 - use the verification tool to confirm that the evidence verifies successfully under the configured trust  
3749 anchors and corresponds to the expected Hypervisor instance;
  - 3750 - compare fields in the evidence with the deployed software and configuration (for example bootloader  
3751 and kernel versions, key security-critical settings) to confirm consistency;
  - 3752 - review the content of the evidence to confirm that it is limited to the information required to verify  
3753 integrity and configuration state and does not unnecessarily expose unrelated sensitive operational data.
- 3754 3) From a guest workload, non-privileged account, or other context that should not have direct access:
  - 3755 - attempt to read or export attestation private keys or raw measurement data using available interfaces;
  - 3756 - verify that such attempts are rejected or blocked and that keys and measurements cannot be obtained  
3757 through these paths.
- 3758 4) Where anti-replay behaviour is documented:
  - 3759 - capture a valid attestation response;
  - 3760 - attempt to reuse or replay this response in a context that expects fresh attestation;

- 3761 - verify, using the verifier or tool, that replayed evidence is detected or rejected in accordance with the  
3762 documented behaviour.

### 3763 5. Assessment verdict

- 3764 • Pass: the Hypervisor generates attestation evidence covering the bootloader, kernel or microkernel, security-  
3765 critical configuration, and essential runtime state; the evidence is cryptographically verifiable and bound to the  
3766 specific Hypervisor instance; attestation keys and measurement data are protected against Unauthorized  
3767 access, disclosure, or replay; and the evidence content is limited to the minimum necessary to verify the  
3768 integrity and configuration state of the Hypervisor.
- 3769 • Fail: attestation evidence is not generated as specified; cannot be cryptographically verified or is not clearly  
3770 bound to a specific instance; Unauthorized entities can access or export attestation keys or measurement data  
3771 or successfully replay stale evidence where freshness is required; or the evidence content includes unnecessary  
3772 sensitive information beyond what is required to verify integrity and configuration state.

### 3773 6. Assessment evidence

- 3774 • [DESIGN] Documentation of the measured-boot and attestation design, including measured components,  
3775 evidence format, cryptographic protection, instance binding, and the protection model for attestation keys and  
3776 measurements.
- 3777 • [CONFIG] Hypervisor and, where applicable, platform configuration enabling measured boot and attestation  
3778 and defining trust anchors used by verifiers.
- 3779 • [LOG] Records of attestation requests and responses and any logged events related to attestation key access  
3780 failures or replay handling.
- 3781 • [OBSERVATION] Captured attestation evidence samples and verifier outputs showing successful validation  
3782 for normal runs and rejection or detection of replayed or tampered evidence.
- 3783 • [PLT-CAP] Evidence of platform capabilities used to support measurement and attestation, where the  
3784 implementation relies on platform services.

#### 3785 6.3.1.7.2 Assessment Case AC-H-RA-INT-004

##### 3786 1. Assessment reference

3787 Requirement: REQ-H-RA-INT-004 (Advanced)

##### 3788 2. Assessment objective

3789 Verify that:

- 3790 1) The Hypervisor supports configuration of privacy-preserving attestation options for remote attestation use  
3791 cases that reduce linkability between attestation events.
- 3792 2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and  
3793 configuration state of the Hypervisor.

##### 3794 3. Assessment preparation

3795 The assessment shall have access to:

- 3796 • Documentation describing:
- 3797 - the Hypervisor remote attestation model, including how Hypervisor identity and instance information are  
3798 represented in attestation evidence;
- 3799 - the privacy-preserving options available (for example use of pseudonymous identifiers, separation of  
3800 identity and integrity information, control of attestation frequency or granularity);
- 3801 - how these options are configured;
- 3802 - the expected impact of each option on attestation evidence fields and verifier behaviour.
- 3803 • A test configuration with:

- 3804 - a Hypervisor deployment where remote attestation to one or more authorized verifiers is enabled;
- 3805 - the ability to configure and modify the documented privacy-preserving options;
- 3806 - tools or procedures to trigger multiple attestation events and capture the resulting attestation evidence
- 3807 under different privacy configurations;
- 3808 - access to an authorized verifier or verification tool capable of validating integrity and configuration state
- 3809 from the attestation evidence in both baseline and privacy-enhanced modes.

#### 3810 4. Assessment activities

3811 The assessment shall at least:

- 3812 1) Review documentation and Hypervisor settings to identify:
  - 3813 - the default behaviour of remote attestation when privacy-preserving options are not enabled;
  - 3814 - the specific privacy options available and their documented effects on identifiers, metadata, and
  - 3815 attestation frequency or granularity;
  - 3816 - any prerequisites or constraints for enabling these options.
- 3817 2) With privacy-preserving options disabled or in their default state:
  - 3818 - trigger multiple attestation events over time;
  - 3819 - collect the corresponding attestation evidence;
  - 3820 - Analyze the evidence to determine the stability and linkability of identifiers and other potentially
  - 3821 identifying fields across events;
  - 3822 - use the authorized verifier or tool to confirm that the integrity and configuration state of the Hypervisor
  - 3823 can be correctly assessed from this baseline evidence set.
- 3824 3) With one or more privacy-preserving options enabled as documented:
  - 3825 - trigger multiple attestation events under the privacy-enhanced configuration;
  - 3826 - collect the resulting attestation evidence;
  - 3827 - compare this evidence set with the baseline to verify that linkability between attestation events is reduced
  - 3828 in the documented manner (for example through pseudonymous or rotated identifiers, separation of
  - 3829 identity and integrity data, or reduced authorized identifying metadata);
  - 3830 - verify, using the authorized verifier or tool, that the integrity and configuration state of the Hypervisor
  - 3831 can still be reliably assessed using the privacy-enhanced evidence and any associated metadata or
  - 3832 policies.

#### 3833 5. Assessment verdict

- 3834 • Pass: when remote attestation is used, the Hypervisor provides configurable privacy-preserving attestation
- 3835 options that, when enabled, reduce linkability between attestation events as documented and still allow
- 3836 authorized verifiers to correctly assess the Hypervisor's integrity and configuration state.
- 3837 • Fail: privacy-preserving options do not meaningfully reduce linkability between attestation events; or enabling
- 3838 these options prevents authorized verifiers from correctly assessing the Hypervisor's integrity and
- 3839 configuration state.

#### 3840 6. Assessment evidence

- 3841 • [DESIGN] Documentation of the remote attestation model, identity representation, and privacy-preserving
- 3842 options, including their expected effects on attestation evidence and verifier behaviour.
- 3843 • [CONFIG] Hypervisor and verifier configuration showing remote attestation settings and selected privacy-
- 3844 preserving options.

- 3845       • [LOG] Records of attestation events and related configuration changes, including timestamps and any  
3846       metadata relevant to privacy options.
- 3847       • [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options,  
3848       with verifier outputs showing successful integrity and configuration assessment in both modes and reduced  
3849       linkability when privacy options are enabled.

### 3850   6.3.1.8    Assessment for Administrative Authentication

#### 3851   6.3.1.8.1   Assessment Case AC-H-ADMIN-AUTH-001

##### 3852   **1. Assessment reference**

3853   Requirement: REQ-H-ADMIN-AUTH-001 (Basic)

##### 3854   **2. Assessment objective**

3855   Verify that:

- 3856       1)   All interfaces that provide administrative functionality require successful authentication before any  
3857       administrative operation is performed.
- 3858       2)   Default administrative passwords, if present, cannot be used as persistent credentials without enforced change  
3859       or are disabled.
- 3860       3)   The Hypervisor supports configurable password complexity requirements for administrative accounts and  
3861       enforces the configured policy.

##### 3862   **3. Assessment preparation**

3863   The assessment shall have access to:

- 3864       •   The declared product composition and list of components exposing administrative interfaces.
- 3865       •   A list of all administrative interfaces (e.g. local CLI, remote management, APIs, VM management consoles).
- 3866       •   Documentation on administrative accounts (including any factory default administrative credentials) and  
3867       password policy options.
- 3868       •   At least one initialized configuration with one or more administrative accounts and access to the identified  
3869       interfaces.

##### 3870   **4. Assessment activities**

3871   The assessment shall at least:

- 3872       1)   Confirm from documentation and configuration which interfaces provide administrative functionality and  
3873       verify that, in the assessed configuration, each such interface requires authentication before administrative  
3874       operations.
- 3875       2)   Review installation and hardening documentation for default administrative credentials and verify that, in the  
3876       assessed configuration, any such credentials are either disabled or subject to enforced change before use for  
3877       administrative operations.
- 3878       3)   Confirm that password complexity requirements for administrative accounts can be configured and verify that  
3879       non-compliant passwords are not accepted for administrative accounts in the assessed configuration.

##### 3880   **5. Assessment verdict**

- 3881       •   Pass: all administrative interfaces require authentication; default administrative credentials, if present, cannot  
3882       be used as persistent credentials without enforced change or are disabled; and password complexity for  
3883       administrative accounts is configurable and enforced.
- 3884       •   Fail: at least one administrative interface allows unauthenticated administrative operations; or default  
3885       administrative credentials can be used for ongoing administrative access; or password complexity for  
3886       administrative accounts cannot be configured or is not enforced.

##### 3887   **6. Assessment evidence**

- 3888 • [SCOPE] Declared product composition and list of administrative interfaces considered.
- 3889 • [CONFIG] Administrative authentication and password policy configuration.
- 3890 • [DESIGN] Description of administrative access model, default accounts and password policy mechanism.
- 3891 • [LOG] / [OBSERVATION] Logs, screenshots or console outputs showing authentication enforcement and  
3892 rejection of non-compliant passwords.

### 3893 6.3.1.8.2 Assessment Case AC-H-ADMIN-AUTH-002

#### 3894 1. Assessment reference

3895 Requirement: REQ-H-ADMIN-AUTH-002 (Elevated)

#### 3896 2. Assessment objective

3897 Verify that:

- 3898 1) Administrative interfaces require authenticated access and support cryptographic key-based authentication for  
3899 administrative accounts.
- 3900 2) Password-based login for administrative interfaces can be disabled.
- 3901 3) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile  
3902 declared by the manufacturer, aligned with [2].
- 3903 4) Private keys used for administrative authentication are protected from access or reuse by guest workloads and  
3904 other unprivileged components.

#### 3905 3. Assessment preparation

3906 The assessment shall have access to:

- 3907 • Documentation on key-based administrative authentication for each administrative interface.
- 3908 • Documentation of applicable cryptographic requirements or profiles for administrative authentication.
- 3909 • Documentation of key storage locations and access-control mechanisms for administrative private keys.
- 3910 • At least one configuration where administrative accounts use key-based authentication on relevant interfaces  
3911 and password-based login can be disabled.

#### 3912 4. Assessment activities

3913 The assessment shall at least:

- 3914 1) Confirm from documentation and configuration that administrative accounts can use cryptographic key-based  
3915 authentication on relevant interfaces and verify that an administrative account can obtain administrative access  
3916 using the key-based mechanism.
- 3917 2) Confirm that the product provides configuration to disable password-based login for administrative interfaces  
3918 and verify that, where disabled, password-only administrative logins are rejected while key-based  
3919 authentication remains available.
- 3920 3) Review configuration or observable protocol characteristics to confirm that algorithms, key lengths and  
3921 protocol versions used for key-based administrative authentication comply with the declared cryptographic  
3922 profile and that weaker parameters identified as insufficient are not used.
- 3923 4) Review the design and configuration of key storage for administrative private keys and confirm that guest  
3924 workloads and other unprivileged components cannot read, export or otherwise misuse these keys.

#### 3925 5. Assessment verdict

- 3926 • Pass: administrative interfaces require authenticated access and support key-based administrative  
3927 authentication; password-based login for administrative interfaces can be disabled and is effectively blocked  
3928 where configured; cryptographic parameters comply with the declared profile; and administrative private keys  
3929 are protected from guest and unprivileged access.

- 3930       • Fail: key-based administrative authentication is not available or not functional; or password-based login cannot  
3931       be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared  
3932       profile; or administrative private keys are accessible to guest workloads or other unprivileged components.

3933       **6. Assessment evidence**

- 3934       • [CONFIG] Configuration for key-based administrative authentication, password-login disablement and  
3935       cryptographic settings.
- 3936       • [DESIGN] Documentation of key management and key storage architecture and cryptographic profile for  
3937       administrative authentication.
- 3938       • [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of  
3939       password-only attempts when disabled.
- 3940       • [PLT-CAP] Where applicable, evidence of platform capabilities used for key protection.

3941       **6.3.1.8.3 Assessment Case AC-H-ADMIN-AUTH-003**

3942       **1. Assessment reference**

3943       Requirement: REQ-H-ADMIN-AUTH-003 (Advanced)

3944       **2. Assessment objective**

3945       Verify that:

- 3946       1) Certificate-based strong authentication is supported and enforceable for administrative accounts on  
3947       administrative interfaces declared in scope.
- 3948       2) Certificates used for administrative authentication are validated against a trusted PKI, including chain of trust,  
3949       validity period and, where supported, revocation status.
- 3950       3) Administrative accounts configured for certificate-based strong authentication cannot use weaker  
3951       authentication methods, except for explicitly documented emergency mechanisms.

3952       **3. Assessment preparation**

3953       The assessment shall have access to:

- 3954       • Documentation describing certificate-based administrative authentication for each relevant interface (e.g.  
3955       mutual TLS, client certificates).
- 3956       • Documentation of PKI integration: trusted CAs, account/certificate mapping, revocation mechanisms (CRL,  
3957       OCSP or equivalent).
- 3958       • At least one configuration with administrative accounts using certificates issued by a trusted CA, and  
3959       untrusted, expired and, where supported, revoked certificates (or equivalent) for negative checks.

3960       **4. Assessment activities**

3961       The assessment shall at least:

- 3962       1) Confirm from documentation and configuration that administrative accounts can be configured for certificate-  
3963       based authentication on administrative interfaces declared in scope and verify that administrative access is  
3964       granted only when a valid certificate is presented in line with the configured policy.
- 3965       2) Confirm that the product validates administrative certificates against the configured trusted CAs and checks  
3966       validity period, and verify that untrusted, expired or not-yet-valid certificates are rejected; where revocation is  
3967       supported, verify that revoked certificates are rejected when revocation information is available.
- 3968       3) Review configuration options for enforcing certificate-based strong authentication and verify that  
3969       administrative accounts configured for such authentication cannot obtain administrative access using weaker  
3970       methods alone (e.g. password-only), except where an explicitly documented emergency mechanism applies  
3971       and its conditions are met.

3972       **5. Assessment verdict**

- 3973 • Pass: certificate-based authentication is available and enforceable for administrative accounts on  
 3974 administrative interfaces declared in scope; PKI validation (trusted CAs, validity, and, where supported,  
 3975 revocation) behaves as documented; downgrade to weaker authentication for accounts configured for  
 3976 certificate-based strong authentication is not permitted except under documented emergency mechanisms.
- 3977 • Fail: certificate-based administrative authentication cannot be configured or is not functioning; or PKI  
 3978 validation is incomplete or inconsistent with documentation; or downgrade to weaker methods is possible for  
 3979 accounts intended to use certificate-based strong authentication.

## 3980 6. Assessment evidence

- 3981 • [CONFIG] Configuration for certificate-based administrative authentication, trusted CA configuration,  
 3982 revocation settings and account-to-certificate mappings.
- 3983 • [DESIGN] Documentation of the architecture for certificate-based administrative authentication.
- 3984 • [LOG] / [OBSERVATION] Evidence of successful authentication with valid administrative certificates and  
 3985 rejection of untrusted, expired or revoked certificates, and of rejected downgrade attempts.
- 3986 • [PLT-CAP] Where relevant, evidence of platform capabilities used to support trust anchors or protect key and  
 3987 trust stores.

### 3988 6.3.1.9 Assessment for Service Authentication

#### 3989 6.3.1.9.1 Assessment Case AC-H-SERV-AUTH-001

##### 3990 1. Assessment reference

3991 Requirement: REQ-H-SERV-AUTH-001 (Basic)

##### 3992 2. Assessment objective

3993 Verify that:

- 3994 1) External services accessing Hypervisor management functions or data are authenticated.
- 3995 2) Each service account uses distinct, non-default credentials and no shared or hard-coded service credentials are  
 3996 relied upon.
- 3997 3) A configurable lockout mechanism exists and is enforced for service accounts after consecutive failed  
 3998 authentication attempts.

##### 3999 3. Assessment preparation

4000 The assessment shall have access to:

- 4001 • Declared product composition and list of external services/interfaces accessing management functions or data  
 4002 (e.g. orchestration, monitoring, backup).
- 4003 • Documentation on service accounts, default/built-in service credentials, credential policy and lockout options.
- 4004 • At least one configuration with representative external services integrated via service accounts.

##### 4005 4. Assessment activities

4006 The assessment shall at least:

- 4007 1) Identify all external services accessing management functions or data and verify that each is required to  
 4008 authenticate before access.
- 4009 2) Review documentation/configuration to confirm distinct credentials per service account and absence of shared  
 4010 or hard-coded service credentials in normal operation.
- 4011 3) Confirm that lockout thresholds for service accounts can be configured and verify that repeated failed  
 4012 authentication attempts trigger lockout according to the configured parameters.

##### 4013 5. Assessment verdict

- 4014 • Pass: external services accessing management functions or data require authentication; each service account  
4015 uses distinct, non-default credentials without reliance on shared/hard-coded credentials; and a configurable  
4016 lockout mechanism for service accounts exists and is enforced.
- 4017 • Fail: unauthenticated service access is possible; or shared/default/hard-coded service credentials are required  
4018 for normal use; or a lockout mechanism for service accounts is missing, not configurable, or not enforced.

4019 **6. Assessment evidence**

- 4020 • [SCOPE] List of external services and interfaces accessing management functions or data.
- 4021 • [CONFIG] Service authentication, service account and lockout configuration.
- 4022 • [DESIGN] Description of service account model, handling of default/built-in service credentials and lockout  
4023 mechanism.
- 4024 • [LOG] / [OBSERVATION] Evidence of enforced service authentication and lockout after failed attempts.

4025 **6.3.1.9.2 Assessment Case AC-H-SERV-AUTH-002**

4026 **1. Assessment reference**

4027 Requirement: REQ-H-SERV-AUTH-002 (Elevated)

4028 **2. Assessment objective**

4029 Verify that:

- 4030 1) External services accessing management functions or data require authenticated access and can use certificate-  
4031 based mutual authentication.
- 4032 2) Algorithms and key lengths for service mutual authentication conform to the declared cryptographic profile,  
4033 aligned with [2].
- 4034 3) Private keys used for service authentication are protected from extraction or reuse by guest workloads and  
4035 other unprivileged components.

4036 **3. Assessment preparation**

4037 The assessment shall have access to:

- 4038 • Documentation on certificate-based mutual authentication for external services.
- 4039 • Documentation on cryptographic profiles for service authentication (algorithms, key sizes, disallowed  
4040 options).
- 4041 • Documentation on key management and storage for service authentication keys, including separation from  
4042 guest workloads/unprivileged components.
- 4043 • At least one configuration where representative services access management functions or data using  
4044 certificate-based mutual authentication.

4045 **4. Assessment activities**

4046 The assessment shall at least:

- 4047 1) Confirm from documentation/configuration that services accessing management functions or data are  
4048 authenticated and that certificate-based mutual authentication is supported; verify that a service configured for  
4049 mutual authentication can establish an authenticated session and access management functions or data.
- 4050 2) Review configuration/protocol characteristics to confirm that algorithms, key lengths and protocol versions  
4051 used for service mutual authentication comply with the declared cryptographic profile and that weaker/non-  
4052 compliant options are not used.
- 4053 3) Review key storage design and configuration to confirm that private keys used for service authentication  
4054 reside in protected key stores and cannot be read, exported or misused by guest workloads or other  
4055 unprivileged components.

4056 **5. Assessment verdict**

- 4057 • Pass: authenticated access and certificate-based mutual authentication are available for services accessing  
 4058 management functions or data; cryptographic parameters conform to the declared profile; and service  
 4059 authentication keys are protected from guest and unprivileged access.
- 4060 • Fail: service access does not consistently require authentication; or mutual authentication is not  
 4061 available/functional where claimed; or cryptographic parameters are weaker than the declared profile; or  
 4062 service authentication keys are accessible to guest workloads or other unprivileged components.

4063 **6. Assessment evidence**

- 4064 • [CONFIG] Mutual authentication configuration for services, including certificates and crypto settings.
- 4065 • [DESIGN] Service authentication and key management architecture, including storage and protection of  
 4066 service keys and applied cryptographic profile.
- 4067 • [LOG] / [OBSERVATION] Evidence of successful certificate-based mutual authentication and rejection of  
 4068 non-compliant or unauthenticated connections.
- 4069 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication keys.

4070 **6.3.1.9.3 Assessment Case AC-H-SERV-AUTH-003**4071 **1. Assessment reference**

4072 Requirement: REQ-H-SERV-AUTH-003 (Advanced)

4073 **2. Assessment objective**

4074 Verify that:

- 4075 1) Certificate-based mutual authentication is enforced for service interactions that access Hypervisor management  
 4076 functions or data.
- 4077 2) Certificate paths for service authentication are validated against a trusted PKI.
- 4078 3) Certificates used for service authentication are bound to specific service identities.
- 4079 4) Certificates are checked for validity, expiration and revocation status before access is granted.

4080 **3. Assessment preparation**

4081 The assessment shall have access to:

- 4082 • Documentation describing enforcement of certificate-based mutual authentication for service interactions  
 4083 accessing management functions or data, including in-scope interfaces.
- 4084 • Documentation of the PKI trust model (trusted CAs, trust stores, path validation rules) used for service  
 4085 authentication.
- 4086 • Documentation on how service identities are bound to certificates and how validity and revocation information  
 4087 (CRL, OCSP or equivalent) is obtained and processed.
- 4088 • At least one configuration in which all such service interactions use certificate-based mutual authentication,  
 4089 with valid and invalid/expired/revoked service certificates (or equivalent conditions) available for testing.

4090 **4. Assessment activities**

4091 The assessment shall at least:

- 4092 1) Confirm from documentation/configuration that certificate-based mutual authentication is enforced for service  
 4093 interactions accessing management functions or data and verify that such interactions cannot be established  
 4094 without successful mutual certificate authentication.
- 4095 2) Review identity-binding rules and verify that certificates used for service authentication are associated with  
 4096 specific service identities as documented.

- 4097 3) Confirm that certificate paths are validated against a trusted PKI and verify that certificates from untrusted  
4098 CAs or with invalid chains are rejected.
- 4099 4) Verify that the Hypervisor checks certificate validity period and rejects expired or not-yet-valid certificates,  
4100 and, where revocation mechanisms are supported, that revoked (or otherwise invalidated) service certificates  
4101 are rejected before access is granted.

#### 4102 5. Assessment verdict

- 4103 • Pass: certificate-based mutual authentication is enforced for service interactions accessing management  
4104 functions or data; certificate paths are validated against a trusted PKI; certificates are bound to specific service  
4105 identities; and validity, expiration and revocation (or equivalent) are checked and enforced before access is  
4106 granted.
- 4107 • Fail: service interactions can occur without enforced certificate-based mutual authentication; or certificate  
4108 paths are not properly validated; or certificates are not effectively bound to specific service identities; or  
4109 validity/expiration/revocation checks are missing or ineffective.

#### 4110 6. Assessment evidence

- 4111 • [CONFIG] Configuration enforcing certificate-based mutual authentication for services, including trust stores,  
4112 CA configuration and identity bindings.
- 4113 • [DESIGN] Documentation of the PKI trust model for service authentication, identity binding and revocation  
4114 handling.
- 4115 • [LOG] / [OBSERVATION] Evidence of successful service interactions with valid certificates and rejection of  
4116 unauthenticated, untrusted, expired or revoked certificates.
- 4117 • [PLT-CAP] Where relevant, evidence of platform capabilities used to support certificate validation and PKI  
4118 trust.

### 4119 6.3.1.10 Assessment for Administrative Authorization

#### 4120 6.3.1.10.1 Assessment Case AC-H-ADMIN-AUTHZ-001

##### 4121 1. Assessment reference

4122 Requirement: REQ-H-ADMIN-AUTHZ-001 (Basic)

##### 4123 2. Assessment objective

4124 Verify that:

- 4125 1) All administrative actions are restricted to authenticated administrative accounts.
- 4126 2) A default-deny policy is enforced for access to management functions.

##### 4127 3. Assessment preparation

4128 The assessment shall have access to:

- 4129 • Documentation of administrative roles, accounts and management interfaces.
- 4130 • Documentation of the authorization model for administrative actions (including default behaviour).
- 4131 • At least one configuration with multiple administrative accounts and access to management functions.

##### 4132 4. Assessment activities

4133 The assessment shall at least:

- 4134 1) Verify that unauthenticated users and non-administrative accounts cannot perform administrative actions on  
4135 any management interface.
- 4136 2) Confirm from configuration and tests that unspecified administrative actions are denied by default, and that  
4137 only explicitly authorized administrative actions are permitted for authenticated accounts.

4138 **5. Assessment verdict**

- 4139 • Pass: administrative actions can only be performed by authenticated administrative accounts and any non-  
4140 authorized administrative action is denied by default.
- 4141 • Fail: unauthenticated or non-administrative users can perform administrative actions; or unspecified  
4142 administrative actions are allowed.

4143 **6. Assessment evidence**

- 4144 • [CONFIG] Administrative authorization configuration and default policies.
- 4145 • [DESIGN] Description of administrative authorization model and default-deny behaviour.
- 4146 • [LOG] / [OBSERVATION] Evidence of denied attempts by unauthenticated/non-admin users and permitted  
4147 actions only for authorized administrative accounts.

4148 **6.3.1.10.2 Assessment Case AC-H-ADMIN-AUTHZ-002**4149 **1. Assessment reference**

4150 Requirement: REQ-H-ADMIN-AUTHZ-002 (Elevated)

4151 **2. Assessment objective**

4152 Verify that:

- 4153 1) RBAC or ABAC is enforced for administrative accounts.
- 4154 2) Administrative accounts are granted only the minimum privileges required (least privilege).

4155 **3. Assessment preparation**

4156 The assessment shall have access to:

- 4157 • Documentation describing RBAC/ABAC for administrative accounts (roles, attributes, permissions).
- 4158 • At least one configuration with multiple administrative roles/attributes and clearly differentiated privilege  
4159 levels.

4160 **4. Assessment activities**

4161 The assessment shall at least:

- 4162 1) Verify that administrative permissions are defined and enforced via RBAC or ABAC (e.g. role-based or  
4163 attribute-based rules) rather than a single "full admin" profile.
- 4164 2) For selected administrative accounts, confirm that they can perform only the actions associated with their  
4165 defined role/attributes and that higher-privilege actions are denied, demonstrating least-privilege enforcement.

4166 **5. Assessment verdict**

- 4167 • Pass: RBAC or ABAC is used to control administrative actions and selected administrative accounts are  
4168 restricted to the minimum privileges defined for their role/attributes.
- 4169 • Fail: RBAC/ABAC is not effectively enforced; or administrative accounts have broader privileges than  
4170 defined; or least-privilege cannot be demonstrated.

4171 **6. Assessment evidence**

- 4172 • [CONFIG] RBAC/ABAC configuration for administrative accounts (roles/attributes and permissions).
- 4173 • [DESIGN] Documentation of the administrative authorization model and least-privilege approach.
- 4174 • [LOG] / [OBSERVATION] Evidence of allowed actions within a role's scope and denied actions outside that  
4175 scope.

### 4176 6.3.1.10.3 Assessment Case AC-H-ADMIN-AUTHZ-003

#### 4177 1. Assessment reference

4178 Requirement: REQ-H-ADMIN-AUTHZ-003 (Advanced)

#### 4179 2. Assessment objective

4180 Verify that:

- 4181 1) Fine-grained authorization policies are enforced for administrative accounts.
- 4182 2) Separation of Duties (SoD) is implemented for administrative functions where required.
- 4183 3) Time-bound or just-in-time (JIT) privilege elevation is supported and enforced.
- 4184 4) Integration with external identity and/or policy decision systems is supported where available.

#### 4185 3. Assessment preparation

4186 The assessment shall have access to:

- 4187 • Documentation of policy-based administrative authorization (policy language/rules, SoD, JIT).
- 4188 • Documentation describing how external identity or policy decision systems (if supported) are integrated.
- 4189 • At least one configuration with:
  - 4190 - separate administrative roles/functions demonstrating SoD; and
  - 4191 - an example of time-bound or JIT privilege elevation.

#### 4192 4. Assessment activities

4193 The assessment shall at least:

- 4194 1) Verify that administrative permissions are expressed as fine-grained policies (not only static roles), and that  
4195 different administrative functions can be assigned to different accounts to enforce SoD (e.g. configuration vs  
4196 approval).
- 4197 2) Verify that JIT or time-bound elevation can be configured and that elevated privileges are granted only for a  
4198 limited scope/time and then automatically revoked.
- 4199 3) Where integration with external identity/policy systems is supported, verify that the Hypervisor can use such  
4200 systems to make administrative authorization decisions or to retrieve administrative identity/attributes used in  
4201 policies.

#### 4202 5. Assessment verdict

- 4203 • Pass: administrative authorization is policy-driven and fine-grained; SoD is enforceable and demonstrable;  
4204 JIT/time-bound elevation is supported and correctly revoked; and, where applicable, integration with external  
4205 identity/policy systems works as documented.
- 4206 • Fail: authorization cannot be expressed as fine-grained policies; SoD cannot be enforced; JIT/time-bound  
4207 elevation is not available or not effective; or supported external integration cannot be used for administrative  
4208 authorization.

#### 4209 6. Assessment evidence

- 4210 • [CONFIG] Administrative policy configuration showing fine-grained rules, SoD and JIT/time-bound  
4211 elevation.
- 4212 • [DESIGN] Documentation of policy-driven authorization, SoD model and external identity/policy integration.
- 4213 • [LOG] / [OBSERVATION] Evidence of policy decisions enforcing SoD, temporary elevation and subsequent  
4214 revocation; and, where applicable, decisions influenced by external identity/policy systems.

### 4215 6.3.1.11 Assessment for Service Authorization

#### 4216 6.3.1.11.1 Assessment Case AC-H-SERV-AUTHZ-001

##### 4217 1. Assessment reference

4218 Requirement: REQ-H-SERV-AUTHZ-001 (Basic)

##### 4219 2. Assessment objective

4220 Verify that:

- 4221 1) A default-deny policy is enforced for service accounts.
- 4222 2) Service accounts can only perform actions explicitly authorized.

##### 4223 3. Assessment preparation

4224 The assessment shall have access to:

- 4225 • List of Hypervisor service accounts and service-facing interfaces.
- 4226 • Documentation of the authorization model for service accounts (policies, defaults).
- 4227 • At least one configuration with several service accounts, including one with no explicit permissions.

##### 4228 4. Assessment activities

4229 The assessment shall at least:

- 4230 1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny actions  
4231 not explicitly authorized.
- 4232 2) Verify that a service account with no explicit permissions cannot perform management or data actions.
- 4233 3) For a service account with a defined set of permissions, verify that allowed actions succeed while non-  
4234 authorized actions are denied.

##### 4235 5. Assessment verdict

- 4236 • Pass: default behaviour for service accounts is deny; an unprivileged service account cannot perform actions;  
4237 and only explicitly authorized actions are permitted.
- 4238 • Fail: unspecified actions for service accounts are allowed; or an unprivileged service account can perform  
4239 operations.

##### 4240 6. Assessment evidence

- 4241 • [SCOPE] List of service accounts and relevant interfaces.
- 4242 • [CONFIG] Service-account authorization and default policy configuration.
- 4243 • [DESIGN] Description of the service-account authorization model and default-deny behaviour.
- 4244 • [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and permitted  
4245 operations only when explicitly authorized.

#### 4246 6.3.1.11.2 Assessment Case AC-H-SERV-AUTHZ-002

##### 4247 1. Assessment reference

4248 Requirement: REQ-H-SERV-AUTHZ-002 (Elevated)

##### 4249 2. Assessment objective

4250 Verify that:

- 4251 1) Granular authorization is enforced for service accounts, binding allowed actions to their authenticated identity.
- 4252 2) Each service account is configured with least-privilege permissions.

4253 **3. Assessment preparation**

4254 The assessment shall have access to:

- 4255 • Documentation on how service identities are represented and bound to permissions.
- 4256 • Documentation on the permission/role model for service accounts.
- 4257 • At least one configuration with multiple service accounts having different, scoped permission sets.

4258 **4. Assessment activities**

4259 The assessment shall at least:

- 4260 1) Confirm that authorization decisions for service accounts are based on the authenticated service identity and  
4261 not only on generic properties (e.g. IP or network zone).
- 4262 2) For at least two service accounts with different permission sets, verify that each can only perform the actions  
4263 explicitly bound to its identity and that actions assigned to another service identity are denied.
- 4264 3) Verify that a service account configured with minimal permissions cannot perform operations outside that  
4265 minimal set (least privilege).

4266 **5. Assessment verdict**

- 4267 • Pass: authorization for service accounts is identity-bound and granular; and least-privilege permission sets can  
4268 be configured and are enforced.
- 4269 • Fail: actions are not clearly bound to service identity; or service accounts can perform operations outside their  
4270 defined permissions; or least-privilege cannot be demonstrated.

4271 **6. Assessment evidence**

- 4272 • [CONFIG] Service-account authorization configuration showing bindings between specific identities and  
4273 allowed actions, including minimal-permission accounts.
- 4274 • [DESIGN] Documentation of the service-account authorization model, identity binding and least-privilege  
4275 approach.
- 4276 • [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and  
4277 denied operations outside that scope.

4278 **6.3.1.11.3 Assessment Case AC-H-SERV-AUTHZ-003**

4279 **1. Assessment reference**

4280 Requirement: REQ-H-SERV-AUTHZ-003 (Advanced)

4281 **2. Assessment objective**

4282 Verify that:

- 4283 1) Fine-grained, policy-driven authorization is enforced for service accounts.
- 4284 2) Permissions are bound to cryptographic service identities.
- 4285 3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-  
4286 account authorization decisions.
- 4287 4) Integration with external authorization systems is supported where available.

4288 **3. Assessment preparation**

4289 The assessment shall have access to:

- 4290 • Documentation of policy-driven authorization for service accounts (policy language or rules, conditions,  
4291 effects).
- 4292 • Documentation on how cryptographic service identities (e.g. certificates, tokens) are bound to permissions.

- 4293 • Documentation on context attributes available for policy decisions.
- 4294 • Documentation on integration with external authorization systems (e.g. external Policy Decision Point (PDP)
- 4295 or policy engine), if supported.
- 4296 • At least one configuration where:
  - 4297 - service-account policies reference cryptographic identities and at least one context attribute; and
  - 4298 - where supported, an external authorization system is integrated.

#### 4299 **4. Assessment activities**

4300 The assessment shall at least:

- 4301 1) Verify that service-account authorization is expressed as policies (not just fixed roles) and that permissions for
- 4302 at least one service account are defined via such policies.
- 4303 2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed
- 4304 configuration, authorization decisions for at least one service depend on its cryptographic identity.
- 4305 3) Verify that at least one context attribute (e.g. namespace, cluster, workload label) can influence authorization
- 4306 decisions for service accounts and that access is allowed or denied based on that context.
- 4307 4) Where integration with external authorization is supported, verify that the Hypervisor can consult or delegate
- 4308 service-account authorization decisions to an external system and that policy changes in that system are
- 4309 reflected in authorization outcomes.

#### 4310 **5. Assessment verdict**

- 4311 • Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic
- 4312 service identities; context-aware enforcement is supported; and, where applicable, external authorization
- 4313 integration is effective.
- 4314 • Fail: policies cannot be used to express fine-grained authorization; or permissions are not bound to
- 4315 cryptographic identities; or context cannot meaningfully influence authorization decisions; or supported
- 4316 external authorization integration does not work as intended.

#### 4317 **6. Assessment evidence**

- 4318 • [CONFIG] Policy definitions and configuration for service-account authorization referencing cryptographic
- 4319 identities and context attributes, and, where applicable, external authorization integration settings.
- 4320 • [DESIGN] Documentation of the policy-driven service authorization architecture, including cryptographic
- 4321 identity binding, context handling and external authorization interfaces.
- 4322 • [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and
- 4323 context, and of changes resulting from external authorization policy updates where supported.

### 4324 **6.3.1.12 Assessment for Confidentiality Protection**

#### 4325 **6.3.1.12.1 Assessment Case AC-H-CONF-001**

##### 4326 **1. Assessment reference**

4327 Requirement: REQ-H-CONF-001 (Basic)

##### 4328 **2. Assessment objective**

4329 Verify that:

- 4330 1) The Hypervisor provides access-control mechanisms to restrict access to sensitive data under its control,
- 4331 including at least guest VM images, snapshot data, configuration files, credentials, and security-relevant logs.
- 4332 2) Access to such sensitive data is permitted only to authenticated and authorized entities through controlled
- 4333 interfaces, and cannot be obtained directly by unauthenticated or Unauthorized contexts.

### 4334 3. Assessment preparation

4335 The assessment shall have access to:

- 4336 • Documentation describing:
  - 4337 - the categories of sensitive data managed by the Hypervisor (VM images, snapshots, configuration files, credentials, security-relevant logs);
  - 4338
  - 4339 - the storage locations and interfaces through which this data can be accessed or managed (for example management UI, CLI, APIs, hypervisor file stores);
  - 4340
  - 4341 - the access-control model and roles/identities allowed to read or modify each category of sensitive data.
- 4342 • A test configuration with:
  - 4343 - at least one administrative account or role with permissions to manage sensitive data;
  - 4344 - at least one non-privileged or operational account with limited permissions;
  - 4345 - where relevant, one or more guest VMs whose images, snapshots, and logs can be created and manipulated;
  - 4346
  - 4347 - access to the Hypervisor management interfaces and, where applicable, to the underlying storage paths for sensitive data.
  - 4348

### 4349 4. Assessment activities

4350 The assessment shall at least:

- 4351 1) Review documentation and Hypervisor configuration to identify:
  - 4352 - which repositories or storage locations hold VM images, snapshots, configuration files, credentials, and security-relevant logs;
  - 4353
  - 4354 - which interfaces (UI, CLI, API) are intended for accessing or managing each category of sensitive data;
  - 4355 - which roles/accounts are authorized to use these interfaces.
- 4356 2) Using an authenticated administrative account:
  - 4357 - attempt to access and, where permitted, modify or export representative items of each sensitive data category via the documented controlled interfaces;
  - 4358
  - 4359 - verify that access is successful only through these interfaces and behaves as documented.
- 4360 3) Using a non-privileged account, guest VM context, or unauthenticated connection (for example direct filesystem or storage access if exposed, or calling management APIs without proper credentials):
  - 4361
  - 4362 - attempt to read or modify VM images, snapshots, configuration files, credentials, and security-relevant logs;
  - 4363
  - 4364 - verify that such attempts are denied by the Hypervisor access-control mechanisms and do not result in disclosure or modification of sensitive data.
  - 4365
- 4366 4) Where storage paths are visible at the host level, attempt to access sensitive data directly on disk or in configuration directories from a context that is not granted the corresponding management permissions, and verify that operating-system or Hypervisor controls prevent Unauthorized access.
- 4367
- 4368

### 4369 5. Assessment verdict

- 4370 • Pass: access to VM images, snapshots, configuration files, credentials, and security-relevant logs is only possible through controlled interfaces and only for authenticated and authorized entities; attempts by unauthenticated or Unauthorized contexts to access or modify such data are prevented.
- 4371
- 4372
- 4373 • Fail: sensitive data can be read or modified without using controlled interfaces, or by unauthenticated or Unauthorized entities; or the documented access-control model is not enforced in practice.
- 4374

### 4375 6. Assessment evidence

- 4376 • [DESIGN] Documentation of sensitive data types, storage locations, and the associated access-control model.
- 4377 • [CONFIG] Hypervisor configuration showing role/permission assignments and any access-control rules for  
4378 management interfaces and storage locations.
- 4379 • [LOG] Security or audit logs capturing successful authorized accesses and rejected Unauthorized attempts,  
4380 where such logging is implemented.
- 4381 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating  
4382 authorized access succeeding and Unauthorized access failing.
- 4383 • [SCOPE] Evidence identifying which data stores and components were considered in-scope for sensitive data.

#### 4384 6.3.1.12.2 Assessment Case AC-H-CONF-002

##### 4385 1. Assessment reference

4386 Requirement: REQ-H-CONF-002 (Elevated)

##### 4387 2. Assessment objective

4388 Verify that:

- 4389 1) The Hypervisor supports encryption of sensitive data at rest, including VM images, snapshot data,  
4390 configuration files, credentials, and security-relevant logs, as specified by the product.
- 4391 2) When data-at-rest protection is enabled, encryption is effectively applied to the configured sensitive data  
4392 repositories.
- 4393 3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by  
4394 guest workloads and other unprivileged components.

##### 4395 3. Assessment preparation

4396 The assessment shall have access to:

- 4397 • Documentation describing:
  - 4398 - the data-at-rest encryption design, including which types of data and storage locations can be encrypted  
4399 (VM images, snapshots, configuration files, credentials, security logs);
  - 4400 - supported key-management approaches (for example internal key store, external KMS, hardware-backed  
4401 keys);
  - 4402 - how encryption and key management are configured and monitored by the administrator;
  - 4403 - restrictions on access to encryption keys and key-management interfaces.
- 4404 • A test configuration with:
  - 4405 - one or more data stores configured to use the Hypervisor data-at-rest encryption mechanisms for the  
4406 relevant categories of sensitive data;
  - 4407 - at least one administrative account authorized to configure and monitor encryption and key management;
  - 4408 - at least one guest workload and one non-privileged host context that can be used to attempt access to  
4409 encrypted data and key material;
  - 4410 - where applicable, access to the underlying storage devices or files to inspect whether data is stored  
4411 encrypted.

##### 4412 4. Assessment activities

4413 The assessment shall at least:

- 4414 1) Review documentation and configuration to identify:
  - 4415 - which sensitive data categories and storage locations are covered by data-at-rest encryption;
  - 4416 - how encryption is enabled and managed (for example per datastore, per volume, per file);

- 4417 - where encryption keys are stored and how they are protected from guest and non-privileged access.
- 4418 2) With the Hypervisor configured according to product guidance:
- 4419 - create or update representative VM images, snapshots, configuration files, credentials, and security logs  
4420 under the encrypted configuration;
- 4421 - inspect the underlying storage (for example by viewing raw blocks, file contents from outside the  
4422 Hypervisor context, or using vendor-provided diagnostic tools) to confirm that these items are stored in  
4423 an encrypted form and not as plaintext;
- 4424 - verify that disabling or removing access to encryption keys (for example by unmounting the key store or  
4425 revoking KMS access) prevents normal decryption and access to the protected data.
- 4426 3) From guest workloads and other non-privileged contexts:
- 4427 - attempt to access encryption keys or key stores using available interfaces, filesystems, or APIs;
- 4428 - verify that such attempts are denied and that key material is not readable or exportable by these contexts.
- 4429 4) Where external or hardware-backed key management is used, verify that:
- 4430 - keys used for data-at-rest encryption are stored in the external or hardware-backed mechanism as  
4431 documented;
- 4432 - the Hypervisor only receives derived or wrapped keys as documented and does not expose raw key  
4433 material to guest or non-privileged components.

#### 4434 5. Assessment verdict

- 4435 • Pass: the Hypervisor supports and correctly applies encryption for the configured categories of sensitive data  
4436 at rest; encrypted data is not stored in plaintext on underlying storage; and encryption keys are managed and  
4437 protected so that guest workloads and other unprivileged components cannot access them.
- 4438 • Fail: sensitive data at rest configured to be protected is stored unencrypted or in recoverable plaintext;  
4439 encryption keys are accessible to guest workloads or other unprivileged contexts; or the implemented  
4440 behaviour differs from the documented encryption and key-management model.

#### 4441 6. Assessment evidence

- 4442 • [DESIGN] Documentation of the data-at-rest encryption model, including covered data types, key-  
4443 management design, and protection measures for key material.
- 4444 • [CONFIG] Hypervisor configuration showing enabled encryption for relevant datastores and the configured  
4445 key-management settings.
- 4446 • [LOG] Events or audit trails related to enabling/disabling encryption, key-management operations, and failed  
4447 access attempts to protected data or keys.
- 4448 • [OBSERVATION] Test records and storage inspections (for example hex dumps, tool outputs) demonstrating  
4449 that protected data is stored encrypted and that data becomes inaccessible when keys are revoked.
- 4450 • [PLT-CAP] Evidence of any platform or external key-management capabilities used to support data-at-rest  
4451 encryption.

#### 4452 6.3.1.12.3 Assessment Case AC-H-CONF-003

##### 4453 1. Assessment reference

4454 Requirement: REQ-H-CONF-003 (Elevated)

##### 4455 2. Assessment objective

4456 Verify that:

- 4457 1) The Hypervisor supports the use of encrypted communication for management interfaces, for VM migration  
4458 traffic, and for inter-VM traffic where such traffic is mediated by Hypervisor-controlled virtual networking  
4459 components.

- 4460 2) When encrypted transport is configured for these communications, traffic is actually protected with encryption  
4461 consistent with the configuration.
- 4462 3) The Hypervisor prevents fallback from configured encrypted transport to unencrypted or weakly protected  
4463 channels.

### 4464 3. Assessment preparation

4465 The assessment shall have access to:

- 4466 • Documentation describing:
  - 4467 - the Hypervisor management interfaces, VM migration protocols, and inter-VM communication paths that  
4468 can be protected by encrypted transport;
  - 4469 - supported encryption mechanisms and relevant configuration options (for example TLS profiles, SSH,  
4470 IPsec, or other secure tunnelling mechanisms);
  - 4471 - how encryption requirements and anti-fallback behaviour are enforced and monitored.
- 4472 • A test configuration with:
  - 4473 - at least one management interface accessible over a network;
  - 4474 - at least one VM migration path between Hypervisor hosts;
  - 4475 - at least two VMs whose inter-VM traffic can be mediated by the Hypervisor virtual networking layer;
  - 4476 - tools to capture and inspect network traffic (for example packet capture on relevant interfaces).

### 4477 4. Assessment activities

4478 The assessment shall at least:

- 4479 1) Review documentation and configuration to identify:
  - 4480 - which management, migration, and inter-VM traffic flows are in scope for encrypted transport;
  - 4481 - the expected encryption protocols, ciphers, and configuration settings;
  - 4482 - any configuration options that control whether unencrypted or weakly protected channels are allowed.
- 4483 2) Configure encrypted transport for:
  - 4484 - a representative management interface;
  - 4485 - a VM migration operation between two Hypervisor hosts;
  - 4486 - an inter-VM communication path mediated by the Hypervisor virtual network.

4487 For each case, initiate the relevant communication while capturing network traffic on appropriate interfaces,  
4488 and verify that:

- 4489 - payloads are encrypted and not readable as clear text;
  - 4490 - negotiated protocols and cipher suites match the documented configuration.
- 4491 3) Attempt to downgrade or misconfigure these communications by:
    - 4492 - disabling encryption in client tools or attempting to force use of plaintext protocols;
    - 4493 - proposing weak or unsupported cipher suites;
    - 4494 - modifying configuration to request unencrypted channels where encrypted transport has been mandated.

4495 Verify that, when encrypted transport is configured as required, the Hypervisor does not silently fall back to  
4496 unencrypted or weakly protected channels, and instead either maintains encrypted communication or fails the  
4497 connection as documented.

### 4498 5. Assessment verdict

- 4499 • Pass: the Hypervisor supports encrypted transport for management interfaces, VM migration traffic, and  
 4500 Hypervisor-mediated inter-VM traffic; when configured to use encrypted transport, captured traffic is  
 4501 effectively encrypted; and attempts to fall back to unencrypted or weakly protected channels are prevented or  
 4502 cause the connection to fail as documented.
- 4503 • Fail: encrypted transport cannot be enabled as documented; traffic that is configured to be encrypted is  
 4504 observed in plaintext or with weaker protection than configured; or the Hypervisor silently falls back to  
 4505 unencrypted or weakly protected channels when downgrade or misconfiguration is attempted.

#### 4506 6. Assessment evidence

- 4507 • [DESIGN] Documentation of supported secure transport mechanisms for management, migration, and inter-  
 4508 VM traffic, including anti-fallback behaviour.
- 4509 • [CONFIG] Hypervisor configuration showing enabled encryption settings for tested interfaces and flows.
- 4510 • [LOG] Events or audit records related to secure channel establishment, failures, and attempted downgrades.
- 4511 • [OBSERVATION] Packet captures and analysis outputs demonstrating encrypted traffic when protection is  
 4512 configured and showing connection failure or refusal when downgrade is attempted.
- 4513 • [PLT-CAP] Evidence of any platform networking or cryptographic capabilities used to support secure  
 4514 transport.

#### 4515 6.3.1.12.4 Assessment Case AC-H-CONF-004

##### 4516 1. Assessment reference

4517 Requirement: REQ-H-CONF-004 (Advanced)

##### 4518 2. Assessment objective

4519 Verify that:

- 4520 1) The Hypervisor supports provisioning and management of execution environments for guest workloads in  
 4521 which confidential data associated with those workloads is protected, during execution, from observation,  
 4522 inspection, or access through the Hypervisor, the host operating system, and administrative interfaces.
- 4523 2) On platforms that provide mechanisms for confidential execution, the Hypervisor can use these mechanisms to  
 4524 enforce confidentiality of guest memory and associated processor state for workloads designated as  
 4525 confidential.

##### 4526 3. Assessment preparation

4527 The assessment shall have access to:

- 4528 • Documentation describing:
- 4529 - the confidential-execution model supported by the Hypervisor, including which guest types or  
 4530 configurations can run with confidential execution;
  - 4531 - the platform mechanisms used (for example memory-encryption features, secure enclaves, or other  
 4532 confidential-computing capabilities);
  - 4533 - management workflows for provisioning, monitoring, and decommissioning confidential workloads;
  - 4534 - limitations on Hypervisor and host visibility into confidential guest memory and processor state.
- 4535 • A test configuration with:
- 4536 - a platform that provides the documented confidential-execution features;
  - 4537 - the Hypervisor configured to support confidential workloads according to product guidance;
  - 4538 - at least one guest VM that can be instantiated both in normal mode and in confidential-execution mode;
  - 4539 - tools or interfaces normally usable by Hypervisor administrators or host operators to inspect guest  
 4540 memory or processor state (for example debug APIs, snapshot tools, or memory-dump mechanisms).

4541 **4. Assessment activities**

4542 The assessment shall at least:

- 4543 1) Review documentation and configuration to identify:
- 4544 - which workloads or guest types can be run in confidential-execution mode;
- 4545 - which platform features are relied upon and how they are enabled;
- 4546 - which Hypervisor or host-level interfaces are restricted when a guest is running confidentially.
- 4547 2) Instantiate a guest VM in normal (non-confidential) mode and, using available Hypervisor or host tools:
- 4548 - confirm that it is possible to inspect or dump guest memory or processor state in the manner described by
- 4549 the product;
- 4550 - confirm that this behaviour matches documentation for non-confidential workloads.
- 4551 3) Instantiate an equivalent guest VM in confidential-execution mode and:
- 4552 - attempt to use the same inspection, debugging, or memory-dump tools from the Hypervisor and host;
- 4553 - verify that direct inspection of confidential guest memory and associated processor state is blocked or
- 4554 that retrieved data is unintelligible ciphertext, in accordance with the confidential-execution model;
- 4555 - verify that management interfaces clearly indicate the confidential status of the workload.
- 4556 4) Where platform-level reporting is available (for example platform capabilities or status registers relating to
- 4557 memory encryption or secure execution):
- 4558 - confirm that the relevant features are enabled for the confidential workload;
- 4559 - verify that disabling or misconfiguring these features (where possible) prevents confidential mode from
- 4560 being established or results in a documented error state.

4561 **5. Assessment verdict**

- 4562 • Pass: the Hypervisor can provision and manage confidential-execution environments for guest workloads;
- 4563 when such environments are used, confidential workload memory and processor state cannot be inspected or
- 4564 accessed via Hypervisor, host, or administrative interfaces beyond what is documented; and the
- 4565 implementation correctly relies on platform confidential-execution mechanisms where available.
- 4566 • Fail: workloads configured as confidential can still have their memory or processor state inspected in clear
- 4567 form via standard Hypervisor or host tools; platform confidential-execution features are not used or are
- 4568 misconfigured; or the behaviour diverges from the documented confidential-execution model.

4569 **6. Assessment evidence**

- 4570 • [DESIGN] Documentation of the confidential-execution architecture, including reliance on platform features
- 4571 and restrictions on Hypervisor/host access.
- 4572 • [CONFIG] Hypervisor and platform configuration showing how confidential execution is enabled and which
- 4573 guests are configured to use it.
- 4574 • [LOG] Events and audit records relating to creation, modification, and failure of confidential workloads or
- 4575 platform confidential-execution features.
- 4576 • [OBSERVATION] Test records showing successful inspection of non-confidential guests and blocked or
- 4577 unintelligible access attempts for confidential guests, plus management UI/CLI outputs indicating confidential
- 4578 status.
- 4579 • [PLT-CAP] Evidence of platform confidential-computing capabilities used by the Hypervisor.

### 4580 6.3.1.13 Assessment for Availability and Resilience

#### 4581 6.3.1.13.1 Assessment Case AC-H-AVAIL-001

##### 4582 1. Assessment reference

4583 Requirement: REQ-H-AVAIL-001 (Basic)

##### 4584 2. Assessment objective

4585 Verify that:

- 4586 1) The Hypervisor detects failures of internal management and control-plane components.
- 4587 2) The Hypervisor supports recovery of those components without requiring a full host reboot.
- 4588 3) Availability of running guest virtual machines is maintained during such failures and recovery.

##### 4589 3. Assessment preparation

4590 The assessment shall have access to:

- 4591 • Documentation describing Hypervisor management and control-plane components and associated health monitoring or watchdog mechanisms.
- 4592
- 4593 • Documentation of supported recovery methods for management/control components (e.g. process restart, service failover) and their impact on guest VMs.
- 4594
- 4595 • At least one configuration with running guest VMs and access to management/control interfaces.

##### 4596 4. Assessment activities

4597 The assessment shall at least:

- 4598 1) Identify key management and control-plane components (e.g. management daemons, schedulers, control services) and confirm that mechanisms exist to monitor their health and detect failures.
- 4599
- 4600 2) In the assessed configuration, induce or simulate a failure of at least one such component (e.g. stop a management service) and verify that the Hypervisor detects the failure.
- 4601
- 4602 3) Verify that the affected component(s) can be recovered (e.g. automatically restarted or restored) without rebooting the host, and that running guest VMs continue operating during the failure and recovery.
- 4603

##### 4604 5. Assessment verdict

- 4605 • Pass: failures of internal management/control-plane components are detected; recovery is possible without a full host reboot; and running guest VMs remain available during failure and recovery.
- 4606
- 4607 • Fail: failures are not reliably detected; recovery requires a full host reboot; or guest VMs lose availability as a result of management/control-plane component failure and recovery.
- 4608

##### 4609 6. Assessment evidence

- 4610 • [CONFIG] Configuration of management/control components and health monitoring/restart mechanisms.
- 4611 • [DESIGN] Documentation of management/control-plane architecture and failure/recovery behaviour.
- 4612 • [LOG] / [OBSERVATION] Evidence of failure detection, component recovery without host reboot, and continued availability of guest VMs during the procedure.
- 4613

#### 4614 6.3.1.13.2 Assessment Case AC-H-AVAIL-002

##### 4615 1. Assessment reference

4616 Requirement: REQ-H-AVAIL-002 (Elevated)

##### 4617 2. Assessment objective

4618 Verify that:

- 4619 1) The Hypervisor provides mechanisms to support high availability of guest virtual machines.

- 4620 2) Host or Hypervisor service failure is automatically detected.
- 4621 3) Affected guest VMs are automatically restarted or migrated to another available host using native clustering or  
4622 integration with external clustering/orchestration systems.

### 4623 3. Assessment preparation

4624 The assessment shall have access to:

- 4625 • Documentation of Hypervisor high-availability mechanisms for guest VMs, including clustering and  
4626 integrations with external orchestrators.
- 4627 • Documentation describing detection of host/Hypervisor service failure and the associated response.
- 4628 • A test environment with at least two Hypervisor hosts participating in a cluster or managed by an orchestrator,  
4629 hosting one or more test VMs.

### 4630 4. Assessment activities

4631 The assessment shall at least:

- 4632 1) Confirm that high-availability or clustering features are enabled/configured for selected guest VMs.
- 4633 2) Induce or simulate failure of a Hypervisor host or critical Hypervisor service (e.g. stopping the Hypervisor  
4634 service or isolating the host) and verify that the failure is automatically detected by the  
4635 HA/cluster/orchestration mechanism.
- 4636 3) Verify that affected guest VMs are automatically restarted or migrated to another available host, without  
4637 requiring manual intervention, and that the VMs become available again after the recovery action.

### 4638 5. Assessment verdict

- 4639 • Pass: host/Hypervisor service failures are automatically detected; guest VMs configured for HA are  
4640 automatically restarted or migrated to another host using native or integrated clustering/orchestration; and  
4641 availability of such VMs is restored automatically.
- 4642 • Fail: failures are not automatically detected; or affected guest VMs are not automatically restarted/migrated; or  
4643 manual intervention is required to restore availability in contradiction with the documented HA behaviour.

### 4644 6. Assessment evidence

- 4645 • [CONFIG] High-availability/clustering configuration for guest VMs and host group/cluster settings.
- 4646 • [DESIGN] Documentation of HA/cluster/orchestration architecture and failure-handling behaviour.
- 4647 • [LOG] / [OBSERVATION] Evidence of failure detection and subsequent automatic restart or migration of  
4648 guest VMs.

## 4649 6.3.1.13.3 Assessment Case AC-H-AVAIL-003

### 4650 1. Assessment reference

4651 Requirement: REQ-H-AVAIL-003 (Elevated)

### 4652 2. Assessment objective

4653 Verify that:

- 4654 1) The Hypervisor implements resource scheduling mechanisms that allow configuration of minimum resource  
4655 allocations for security and management functions (CPU, memory and I/O).
- 4656 2) These mechanisms protect the Hypervisor control plane from resource exhaustion caused by guest workloads  
4657 or external contention, ensuring that security and management functions remain able to operate under load.

### 4658 3. Assessment preparation

4659 The assessment shall have access to:

- 4660 • Documentation describing Hypervisor resource scheduling mechanisms (e.g. reservations, priorities, quotas,  
4661 cgroup-like mechanisms) and how they apply to security and management functions.

- 4662 • Documentation specifying how minimum CPU, memory and I/O resources can be configured for these  
4663 functions.
- 4664 • A test configuration with several guest VMs and the ability to generate high load from guest workloads.

#### 4665 4. Assessment activities

4666 The assessment shall at least:

- 4667 1) Confirm that resource scheduling or reservation mechanisms exist and that minimum CPU, memory and I/O  
4668 allocations can be configured for security and management components (e.g. management agents, control-  
4669 plane services, logging, security monitoring).
- 4670 2) Configure such minimum allocations in the assessed environment.
- 4671 3) Generate high resource load from guest workloads and/or external contention (e.g. CPU- and I/O-intensive  
4672 tasks) and verify that:
- 4673 - management and security interfaces remain responsive; and
- 4674 - critical control-plane functions (e.g. VM lifecycle operations, logging, access control) continue to  
4675 operate despite guest load.

#### 4676 5. Assessment verdict

- 4677 • Pass: resource scheduling mechanisms support configuration of minimum CPU, memory and I/O allocations  
4678 for security/management functions; and under high guest load or external contention, these functions remain  
4679 able to operate as intended.
- 4680 • Fail: minimum allocations for security/management functions cannot be configured; or under load, the control  
4681 plane becomes starved of resources, causing loss or severe degradation of security or management functions.

#### 4682 6. Assessment evidence

- 4683 • [CONFIG] Resource scheduling/reservation configuration for security and management components.
- 4684 • [DESIGN] Documentation of resource scheduling mechanisms and their application to control-plane  
4685 protection.
- 4686 • [LOG] / [OBSERVATION] Evidence that management and security functions remain operational under load  
4687 (e.g. successful management operations, logs, monitoring outputs during stress tests).

### 4688 6.3.1.13.4 Assessment Case AC-H-AVAIL-004

#### 4689 1. Assessment reference

4690 Requirement: REQ-H-AVAIL-004 (Advanced)

#### 4691 2. Assessment objective

4692 Verify that:

- 4693 1) The Hypervisor supports replication of the runtime state of selected guest VMs (including CPU, memory and  
4694 relevant I/O buffers) to standby instances.
- 4695 2) In the event of host or Hypervisor service failure, failover to the standby instance can be performed with  
4696 minimal interruption to workload execution.

#### 4697 3. Assessment preparation

4698 The assessment shall have access to:

- 4699 • Documentation describing live or near-live state replication/failover mechanisms for guest VMs (e.g.  
4700 continuous or periodic state replication, checkpointing) and supported limitations.
- 4701 • Documentation on configuration of primary and standby instances for selected guest VMs and any  
4702 requirements for networking/storage alignment.

- 4703 • A test environment with at least one pair of Hypervisor hosts supporting a primary/standby configuration for a  
4704 test VM.

4705 **4. Assessment activities**

4706 The assessment shall at least:

- 4707 1) Configure a test guest VM with runtime state replication to a standby instance as described in the  
4708 documentation.
- 4709 2) Verify that the runtime state (CPU and memory, and associated I/O buffers as applicable) of the primary VM  
4710 is replicated to the standby according to the Hypervisor's replication mechanism.
- 4711 3) Induce or simulate failure of the host or Hypervisor service hosting the primary VM and verify that the VM  
4712 fails over to the standby instance.
- 4713 4) Observe the impact on workload execution and confirm that service interruption is limited to the minimal  
4714 duration inherent to the replication/failover mechanism (for example, brief pause or reconnection), without  
4715 requiring full VM reboot from scratch.

4716 **5. Assessment verdict**

- 4717 • Pass: runtime state replication to a standby instance is supported and functional for selected guest VMs;  
4718 failover in response to host/Hypervisor failure brings the standby instance into service using replicated state;  
4719 and workload interruption is minimal compared with a full restart.
- 4720 • Fail: runtime state replication to a standby instance cannot be configured or does not function as documented;  
4721 or failover requires full VM restart without using replicated state; or workload interruption is not consistent  
4722 with "minimal interruption" as described in the documentation.

4723 **6. Assessment evidence**

- 4724 • [CONFIG] Primary/standby and state-replication configuration for selected guest VMs.
- 4725 • [DESIGN] Documentation of VM state replication and failover architecture, including scope of replicated state  
4726 and expected failover behaviour.
- 4727 • [LOG] / [OBSERVATION] Evidence of ongoing state replication, recorded failover events and observed  
4728 workload behaviour during and after failover.

4729 **6.3.1.14 Assessment for Logging**

4730 **6.3.1.14.1 Assessment Case AC-H-LOG-001**

4731 **1. Assessment reference**

4732 Requirement: REQ-H-LOG-001 (Basic)

4733 **2. Assessment objective**

4734 Verify that the Hypervisor:

- 4735 1) Generates audit logs for security-relevant administrative actions and security events.
- 4736 2) Logs at least: authentication attempts, configuration changes, and virtual machine lifecycle operations.

4737 **3. Assessment preparation**

4738 The assessment shall have access to:

- 4739 • Documentation of the Hypervisor logging/auditing capabilities and log formats.
- 4740 • Documentation describing which actions/events are considered security-relevant.
- 4741 • A configuration with administrative access enabled and at least one VM that can be created, modified,  
4742 started/stopped and deleted.

4743 **4. Assessment activities**

4744 The assessment shall at least:

- 4745 1) Review documentation/configuration to identify the audit log sources and where audit logs are stored.
- 4746 2) Perform, in the assessed configuration:
- 4747 - successful and failed administrative authentication attempts;
  - 4748 - administrative configuration changes (e.g. policy or settings updates);
  - 4749 - VM lifecycle operations (e.g. create, start, stop, delete).
- 4750 3) Verify that audit logs record these actions/events with sufficient detail (at least: type of action/event, affected
- 4751 object, timestamp, and responsible account or process where applicable).

4752 **5. Assessment verdict**

- 4753 • Pass: the Hypervisor generates audit logs for security-relevant administrative actions and security events,
- 4754 including authentication attempts, configuration changes, and VM lifecycle operations, with sufficient detail.
- 4755 • Fail: any of the required actions/events are not logged, or log records lack sufficient information to support
- 4756 accountability and analysis.

4757 **6. Assessment evidence**

- 4758 • [CONFIG] Logging/audit configuration for the Hypervisor.
- 4759 • [DESIGN] Documentation of audit logging behaviour and logged event categories.
- 4760 • [LOG] / [OBSERVATION] Sample audit log entries showing authentication attempts, configuration changes
- 4761 and VM lifecycle operations.

4762 **6.3.1.14.2 Assessment Case AC-H-LOG-002**4763 **1. Assessment reference**

4764 Requirement: REQ-H-LOG-002 (Elevated)

4765 **2. Assessment objective**

4766 Verify that:

- 4767 1) Audit logs are protected from Unauthorized access, modification or deletion.
- 4768 2) Access to audit logs is restricted to authorized administrative roles or processes.
- 4769 3) The Hypervisor supports secure export of audit logs to external log management/SIEM systems using
- 4770 authenticated and encrypted channels.

4771 **3. Assessment preparation**

4772 The assessment shall have access to:

- 4773 • Documentation on storage locations for audit logs and associated access controls (file permissions, roles,
- 4774 APIs).
- 4775 • Documentation on roles/processes allowed to access or manage audit logs.
- 4776 • Documentation of log export mechanisms and supported secure transport options (e.g. TLS-protected syslog,
- 4777 HTTPS, secure agents).
- 4778 • A configuration with audit logging enabled and at least one external log management or SIEM endpoint (test
- 4779 instance) reachable via a secure channel.

4780 **4. Assessment activities**

4781 The assessment shall at least:

- 4782 1) Review configuration and underlying platform permissions to confirm that audit log storage is restricted to  
4783 authorized administrative roles/processes and that modification/deletion by non-authorized users or workloads  
4784 is prevented.
- 4785 2) Attempt to read, modify or delete audit logs using a non-privileged account or process and verify that these  
4786 operations are denied.
- 4787 3) Configure secure export of audit logs to an external log management/SIEM endpoint using an authenticated,  
4788 encrypted channel as documented (e.g. TLS with server authentication and, where applicable, client  
4789 authentication).
- 4790 4) Verify, via configuration and protocol inspection, that the export occurs over authenticated and encrypted  
4791 channels and that logs received by the external system correspond to those generated locally.

#### 4792 5. Assessment verdict

- 4793 • Pass: audit logs are protected against Unauthorized access/modification/deletion; only authorized  
4794 administrative roles/processes can access them; and secure, authenticated and encrypted export to external  
4795 log/SIEM systems is supported and functional.
- 4796 • Fail: non-authorized access or tampering with audit logs is possible; or access controls on audit logs are  
4797 insufficient; or log export uses unauthenticated/unencrypted channels or cannot be securely configured.

#### 4798 6. Assessment evidence

- 4799 • [CONFIG] Audit log storage permissions, role mappings and export configuration (including security  
4800 settings).
- 4801 • [DESIGN] Documentation of audit log protection model and secure export mechanisms.
- 4802 • [LOG] / [OBSERVATION] Evidence of denied access attempts by non-privileged accounts and successful  
4803 secure export (e.g. protocol traces, external log records).

#### 4804 6.3.1.14.3 Assessment Case AC-H-LOG-003

##### 4805 1. Assessment reference

4806 Requirement: REQ-H-LOG-003 (Advanced)

##### 4807 2. Assessment objective

4808 Verify that:

- 4809 1) The Hypervisor supports cryptographic protection of audit logs (e.g. signatures, MACs, chained hashes) to  
4810 enable verification of the origin and integrity of audit records.
- 4811 2) Trusted time-stamping is applied to audit logs, enabling verification of the timing of events during forensic  
4812 analysis.
- 4813 3) The combination of cryptographic protection and time-stamping allows later verification of origin and  
4814 integrity of audit records.

##### 4815 3. Assessment preparation

4816 The assessment shall have access to:

- 4817 • Documentation of cryptographic mechanisms used to protect audit logs (e.g. per-record signatures, HMAC,  
4818 hash chains, log sealing) and how verification is performed.
- 4819 • Documentation of the time source and time-stamping method used for audit records (e.g. synchronized system  
4820 time, external time service, secure clock).
- 4821 • A configuration with cryptographic log protection and trusted time-stamping enabled, with the ability to export  
4822 or access raw audit log files/streams for verification.

##### 4823 4. Assessment activities

4824 The assessment shall at least:

- 4825 1) Enable cryptographic protection and trusted time-stamping for audit logs as described in the documentation.
- 4826 2) Generate a set of audit events (e.g. logins, configuration changes, VM lifecycle operations) and capture the  
4827 resulting audit logs.
- 4828 3) Using documented tools or procedures, verify that:
- 4829 - cryptographic checks (e.g. signatures/MACs/hash chains) over the audit records succeed, demonstrating  
4830 integrity and origin as per design; and
- 4831 - each audit record includes a trusted timestamp consistent with the configured time source.
- 4832 4) Attempt to alter or remove part of the audit log (in a test copy) and confirm that integrity verification fails or  
4833 clearly indicates tampering.

#### 4834 5. Assessment verdict

- 4835 • Pass: cryptographic mechanisms and trusted time-stamping are available and can be used to verify the origin  
4836 and integrity of audit records; and tampering with audit logs is detectable through the provided verification  
4837 process.
- 4838 • Fail: cryptographic protection or trusted time-stamping of audit logs cannot be enabled; or verification of  
4839 origin and integrity is not possible or ineffective; or modifications to audit logs cannot be reliably detected.

#### 4840 6. Assessment evidence

- 4841 • [CONFIG] Configuration enabling cryptographic protection and trusted time-stamping of audit logs.
- 4842 • [DESIGN] Documentation of log protection and time-stamping architecture, including verification procedures  
4843 and time sources.
- 4844 • [LOG] / [OBSERVATION] Audit logs and verification outputs showing successful integrity/origin checks for  
4845 unmodified logs and detection of tampering in modified logs.

### 4846 6.3.1.15 Assessment for Secure Update

#### 4847 6.3.1.15.1 Assessment Case AC-H-UPD-001

##### 4848 1. Assessment reference

4849 Requirement: REQ-H-UPD-001 (Basic)

##### 4850 2. Assessment objective

4851 Verify that:

- 4852 1) The Hypervisor supports applying security updates without a full reinstallation of the Hypervisor or host  
4853 system.
- 4854 2) The Hypervisor verifies the authenticity and integrity of all Hypervisor updates before installation, using  
4855 digital signatures validated against trusted keys or certificates configured for the Hypervisor.
- 4856 3) If authenticity or integrity verification of a Hypervisor update fails, the Hypervisor prevents installation of that  
4857 update and records a security-relevant event.
- 4858 4) Keys or certificates used for Hypervisor update verification are not modifiable by guest workloads or other  
4859 unprivileged components during normal operation.

##### 4860 3. Assessment preparation

4861 The assessment shall have access to:

- 4862 • Documentation of Hypervisor update mechanisms (direct Hypervisor updates, host OS updates, appliance  
4863 images, orchestration pipelines) used in the evaluated solution.
- 4864 • Documentation describing which components constitute the Hypervisor and related components in scope  
4865 (binaries, modules, drivers).

- 4866 • Documentation describing the update verification process, including signature format, trusted keys or  
4867 certificates, and the protection of such trust material.
- 4868 • A test configuration where a newer Hypervisor version or security patch is available for application.
- 4869 • Ability to obtain:
  - 4870 ○ a valid, properly signed update; and
  - 4871 ○ an invalid, revoked, missing-signature, or intentionally modified update package, or equivalent test  
4872 artefact, to simulate verification failure.

#### 4873 **4. Assessment activities**

4874 The assessment shall at least:

- 4875 1) Identify the update mechanism(s) used to deliver patches/updates to the Hypervisor in the evaluated solution.
- 4876 2) Apply a Hypervisor update or security patch using the documented mechanism(s) and verify that the update  
4877 completes without requiring a full reinstallation of the Hypervisor or host system.
- 4878 3) Using a valid update package, verify that authenticity and integrity verification is performed as documented  
4879 and that the update is accepted only when such verification succeeds.
- 4880 4) Attempt to apply an update package with invalid, missing, revoked, or intentionally modified  
4881 authenticity/integrity protection and verify that:
  - 4882 a. authenticity/integrity verification fails;
  - 4883 b. the Hypervisor prevents installation of the update; and
  - 4884 c. a security-relevant event is recorded.
- 4885 5) Verify after update that the Hypervisor version/build reflects the applied patch and that existing configuration  
4886 and guest VMs remain available according to the product's normal update behaviour.
- 4887 6) Confirm from documentation and configuration that the trusted keys/certificates used for update verification  
4888 are configured for the Hypervisor and are not modifiable by guest workloads or other unprivileged  
4889 components during normal operation.

#### 4890 **5. Assessment verdict**

- 4891 • Pass: the Hypervisor (as deployed in the evaluated solution) can be patched or updated using the documented  
4892 mechanism(s) without full reinstallation of the Hypervisor or host system; authenticity and integrity of  
4893 Hypervisor updates are verified before installation using trusted keys/certificates configured for the  
4894 Hypervisor; updates with failed verification are not installed; security-relevant failure events are recorded; and  
4895 the trust material used for update verification is protected against modification by guest workloads or other  
4896 unprivileged components.
- 4897 • Fail: patching or update of the Hypervisor requires full reinstallation of the Hypervisor or host system; or  
4898 Hypervisor updates can be installed without authenticity/integrity verification; or updates with failed  
4899 verification can still be installed; or security-relevant failure events are not recorded; or the trust material used  
4900 for update verification is modifiable by guest workloads or other unprivileged components.

#### 4901 **6. Assessment evidence**

- 4902 • [CONFIG] Update mechanism configuration (repositories, management appliance/orchestrator settings, image  
4903 update settings).
- 4904 • [CONFIG] Configuration of update verification, trust anchors (keys/certificates), and update sources.
- 4905 • [DESIGN] Documentation of the Hypervisor update model and relation to host OS/appliance/orchestrator.
- 4906 • [LOG] / [OBSERVATION] Evidence of a successful in-place Hypervisor update (before/after version, update  
4907 logs, continuity of configuration/VMs). Evidence showing successful installation of a valid signed update and  
4908 prevention of installation for an invalid, revoked, or tampered update, including the recorded security-relevant  
4909 event.

## 4910 6.3.1.15.2 Assessment Case AC-H-UPD-002

### 4911 1. Assessment reference

4912 Requirement: REQ-H-UPD-002 (Elevated)

### 4913 2. Assessment objective

4914 Verify that:

- 4915 1) The Hypervisor implements rollback protection to prevent unauthorized installation of outdated, revoked, or  
4916 replayed Hypervisor updates.
- 4917 2) Authorized rollback to a previously trusted version is only permitted when:
  - 4918 a. the rollback image is verified for integrity and authenticity against a trusted key/certificate;
  - 4919 b. the rollback action is explicitly authorized by an administrator with elevated privileges; and
  - 4920 c. the rollback event is logged.

### 4921 3. Assessment preparation

4922 The assessment shall have access to:

- 4923 • Documentation of rollback protection mechanisms (version tracking, anti-replay, revocation handling) and  
4924 rollback procedures.
- 4925 • Documentation of administrative roles/privileges required to initiate rollback and the logging of  
4926 update/rollback events.
- 4927 • A configuration that supports both forward updates and rollback to a previous version.

### 4928 4. Assessment activities

4929 The assessment shall at least:

- 4930 1) Confirm from documentation and configuration that mechanisms exist to detect and prevent installation of  
4931 outdated, revoked, or replayed updates (e.g. version counters, manifests, revocation lists).
- 4932 2) Attempt to install an older, revoked, or replayed update image without using the documented authorized  
4933 rollback procedure and verify that the installation is blocked by rollback protection.
- 4934 3) Perform an authorized rollback using the documented procedure, ensuring that:
  - 4935 a. the rollback image is verified for integrity and authenticity;
  - 4936 b. the rollback action requires and is performed by an administrator with elevated privileges; and
  - 4937 c. the rollback event is recorded in the audit/update logs.
- 4938 4) Verify that, after rollback, the Hypervisor runs the earlier trusted version and that the logs capture the rollback  
4939 with sufficient detail (who, when, from/to which version).

### 4940 5. Assessment verdict

- 4941 • Pass: unauthorized installation of outdated, revoked, or replayed updates is prevented; and authorized rollback  
4942 is only possible following integrity/authenticity verification, elevated administrative approval, and logging of  
4943 the rollback event.
- 4944 • Fail: outdated, revoked, or replayed updates can be installed without appropriate controls; or rollback does not  
4945 require elevated authorization; or rollback events are not logged.

### 4946 6. Assessment evidence

- 4947 • [CONFIG] Update/rollback configuration, including version/rollback controls and admin permission settings.
- 4948 • [DESIGN] Documentation of rollback protection mechanisms and authorized rollback process.

- 4949       • [LOG] / [OBSERVATION] Evidence of blocked unauthorized rollback attempts and logged authorized  
4950       rollback operations.

### 4951 6.3.1.15.3 Assessment Case AC-H-UPD-003

#### 4952 1. Assessment reference

4953 Requirement: REQ-H-UPD-003 (Advanced)

#### 4954 2. Assessment objective

4955 Verify that:

- 4956       1) The Hypervisor supports live patching of critical Hypervisor components, including the kernel or microkernel,  
4957       device drivers, and security modules, where such components are part of the declared product and claimed to  
4958       be live patchable.
- 4959       2) Live patching can be used to remediate vulnerabilities without requiring shutdown of guest virtual machines.
- 4960       3) Live patching causes only minimal interruption to workload execution, in accordance with the documented  
4961       product behaviour.

#### 4962 3. Assessment preparation

4963 The assessment shall have access to:

- 4964       • Documentation identifying the Hypervisor components that support live patching and any preconditions,  
4965       limitations, or exclusions associated with live patching.
- 4966       • Documentation describing the live patching process, including how patch application is initiated, how success  
4967       or failure is reported, and the expected effect on running guest virtual machines and workload execution.
- 4968       • A test configuration with one or more running guest virtual machines hosting representative workloads.
- 4969       • A live patch or equivalent update package applicable to a declared live-patchable Hypervisor component.
- 4970       • The tools necessary to observe patch application, guest virtual machine continuity, workload execution  
4971       continuity, and any interruption caused by the patching process.

#### 4972 4. Assessment activities

4973 The assessment shall at least:

- 4974       1) Review documentation and configuration to identify which critical Hypervisor components are claimed to  
4975       support live patching and the conditions under which live patching is supported.
- 4976       2) With one or more guest virtual machines running, apply a live patch to a declared live-patchable Hypervisor  
4977       component using the documented procedure.
- 4978       3) Verify that:
- 4979           a. the live patch is applied without requiring shutdown of the running guest virtual machines;
- 4980           b. the Hypervisor reports the patch application outcome; and
- 4981           c. the guest virtual machines and representative workloads continue to operate, subject only to the  
4982           minimal interruption described in the product documentation.
- 4983       4) Where the live patch application fails or is rejected, verify that the Hypervisor reports the failure and remains  
4984       in a consistent state.

#### 4985 5. Assessment verdict

- 4986       • Pass: the Hypervisor supports live patching for the claimed critical Hypervisor components; live patching can  
4987       be performed without requiring shutdown of guest virtual machines; and the observed effect on workload  
4988       execution is limited to the minimal interruption described in the product documentation.
- 4989       • Fail: the claimed live patching capability is absent; live patching requires shutdown of guest virtual machines;  
4990       patch application outcome is not reported; the Hypervisor does not remain in a consistent state after failure; or  
4991       interruption exceeds the documented and claimed product behaviour without justification.

4992 **6. Assessment evidence**

- 4993 • [CONFIG] Configuration identifying the components for which live patching is enabled or supported and the  
4994 settings relevant to live patch execution.
- 4995 • [DESIGN] Documentation of the live patching model, supported component scope, and expected effect on  
4996 running guest virtual machines and workloads.
- 4997 • [LOG] / [OBSERVATION] Evidence of live patch application, reported outcome, guest virtual machine  
4998 continuity, workload execution behaviour during patching, and consistent behaviour in the event of patch  
4999 failure or rejection.

5000 **6.3.1.16 Assessment for Secure Configuration and Default**5001 **6.3.1.16.1 Assessment Case AC-H-CFG-001**5002 **1. Assessment reference**

5003 Requirement: REQ-H-CFG-001 (Basic)

5004 **2. Assessment objective**

5005 Verify that:

- 5006 1) By default, the Hypervisor disables remote administrative access pathways and insecure management  
5007 interfaces that are not required for initial provisioning (e.g. unauthenticated/weak remote shell, legacy or  
5008 unencrypted management APIs, non-essential vendor diagnostic ports).
- 5009 2) The Hypervisor provides configuration mechanisms that allow administrators to explicitly enable only those  
5010 interfaces and services needed for the intended Virtualization, orchestration and core host management  
5011 operations.

5012 **3. Assessment preparation**

5013 The assessment shall have access to:

- 5014 • Documentation listing Hypervisor administrative/management interfaces (remote shell, APIs, diagnostic ports,  
5015 web consoles, etc.) and which are required for initial provisioning.
- 5016 • Documentation describing configuration options to enable/disable interfaces and services.
- 5017 • A fresh or default installation of the Hypervisor in its documented default configuration.

5018 **4. Assessment activities**

5019 The assessment shall at least:

- 5020 1) In the default configuration, attempt to access all documented administrative and management interfaces  
5021 (including remote shell, legacy APIs, diagnostic ports) and verify that only those required for initial  
5022 provisioning are enabled; insecure or unnecessary interfaces are disabled by default.
- 5023 2) Review configuration mechanisms (e.g. CLI, API, UI) and verify that administrators can explicitly enable  
5024 specific interfaces/services; then enable a subset of interfaces required for typical  
5025 Virtualization/orchestration/host management operations and confirm:
- 5026 - chosen interfaces become accessible; and
- 5027 - interfaces/services not explicitly enabled remain disabled.

5028 **5. Assessment verdict**

- 5029 • Pass: non-essential and insecure administrative/management interfaces are disabled by default; and  
5030 administrators can enable only those interfaces/services required for intended operations via explicit  
5031 configuration.
- 5032 • Fail: insecure or non-required administrative interfaces are enabled by default; or there is no practical way to  
5033 selectively enable only necessary interfaces/services.

5034 **6. Assessment evidence**

- 5035 • [CONFIG] Default and modified configuration showing interface/service enablement states.
- 5036 • [DESIGN] Documentation of management interfaces, default exposure, and configuration mechanisms.
- 5037 • [OBSERVATION] / [LOG] Connection attempts and screenshots/CLI output showing which interfaces are  
5038 reachable in default and explicitly enabled states.

### 5039 6.3.1.16.2 Assessment Case AC-H-CFG-002

#### 5040 1. Assessment reference

5041 Requirement: REQ-H-CFG-002 (Elevated)

#### 5042 2. Assessment objective

5043 Verify that:

- 5044 1) The Hypervisor validates configuration parameters before applying them.
- 5045 2) Configuration validation logic detects and prevents configurations that would:
  - 5046 - disable or bypass security requirements defined in the present document; or
  - 5047 - violate guarantees for VM isolation, control-plane protection, or minimum resource allocations for  
5048 security/management functions.
- 5049 3) When a configuration change is rejected for security reasons, the Hypervisor provides clear feedback  
5050 indicating the reason for rejection.

#### 5051 3. Assessment preparation

5052 The assessment shall have access to:

- 5053 • Documentation describing configuration validation, including examples of invalid or disallowed settings (e.g.  
5054 disabling mandatory controls, removing isolation, reducing reserved resources below minimums).
- 5055 • Documentation listing mandatory security controls and key guarantees (VM isolation, control-plane  
5056 protection, minimum security/management resources).
- 5057 • A test environment where security-relevant configuration parameters can be changed.

#### 5058 4. Assessment activities

5059 The assessment shall at least:

- 5060 1) Identify a set of configuration parameters whose insecure values would:
  - 5061 - disable/bypass mandatory security controls;
  - 5062 - break VM isolation;
  - 5063 - undermine control-plane protection; or
  - 5064 - reduce reserved resources for security/management below documented minima.
- 5065 2) Attempt to apply such insecure configurations and verify that the Hypervisor rejects or blocks them based on  
5066 configuration validation.
- 5067 3) Observe the feedback presented to the administrator for each rejected change and verify that it clearly indicates  
5068 the security-related reason (e.g. "would disable mandatory control X", "would violate minimum resource  
5069 reservation for control plane", "would compromise VM isolation").
- 5070 4) Apply valid configuration changes and verify that they are accepted and correctly enforced, demonstrating that  
5071 validation is specific to insecure cases, not a generic failure.

#### 5072 5. Assessment verdict

- 5073 • Pass: configuration parameters are validated before application; attempts to apply configurations that  
5074 disable/bypass mandatory security controls or violate isolation/control-plane/minimum-resource guarantees are  
5075 prevented; and clear feedback is provided for security-related rejections.

- 5076 • Fail: unsafe configurations can be applied; or configuration validation does not cover the specified guarantees;
- 5077 or security-related rejection feedback is missing or unclear.

## 5078 6. Assessment evidence

- 5079 • [CONFIG] Examples of attempted insecure configurations and resulting states (accepted/rejected).
- 5080 • [DESIGN] Documentation of configuration validation logic and mappings to mandatory controls and key
- 5081 guarantees.
- 5082 • [LOG] / [OBSERVATION] CLI/UI output and error messages showing rejection of insecure configurations
- 5083 and the reasons given; evidence of accepted secure configuration changes.

### 5084 6.3.1.16.3 Assessment Case AC-H-CFG-003

#### 5085 1. Assessment reference

5086 Requirement: REQ-H-CFG-003 (Advanced)

#### 5087 2. Assessment objective

5088 Verify that:

- 5089 1) The Hypervisor supports defining and using security configuration baselines specifying secure settings for
- 5090 critical components, including at least:
  - 5091 - I/O memory management unit (IOMMU) configuration;
  - 5092 - virtual switch policies;
  - 5093 - host firewall rules, as implemented within the Hypervisor product.
- 5094 2) The Hypervisor can automatically validate current configuration against a selected security baseline.
- 5095 3) The Hypervisor generates alerts or audit events when non-compliant settings are detected.

#### 5096 3. Assessment preparation

5097 The assessment shall have access to:

- 5098 • Documentation describing security configuration baselines (how they are defined, stored, selected and
- 5099 applied).
- 5100 • Documentation showing how baseline rules cover IOMMU configuration, virtual switch policies and host
- 5101 firewall rules within the Hypervisor product.
- 5102 • Documentation of automatic baseline validation mechanisms and alert/audit behaviour on non-compliance.
- 5103 • A test configuration where at least one security baseline can be defined/selected and where deviations from
- 5104 that baseline can be introduced.

#### 5105 4. Assessment activities

5106 The assessment shall at least:

- 5107 1) Define or select a security configuration baseline that includes explicit requirements for IOMMU settings,
- 5108 virtual switch policies and host firewall rules, according to product capabilities.
- 5109 2) Ensure the Hypervisor configuration is aligned with the baseline and run the baseline validation function;
- 5110 verify that the Hypervisor reports compliance and does not generate non-compliance alerts/events.
- 5111 3) Introduce one or more deliberate deviations from the baseline (e.g. relax a firewall rule, disable or weaken a
- 5112 virtual switch policy, misconfigure IOMMU-related settings) and run baseline validation again.
- 5113 4) Verify that:
  - 5114 - the Hypervisor detects the non-compliant settings; and
  - 5115 - alerts or audit events are generated indicating which configuration elements are non-compliant with the
  - 5116 selected baseline.

5117 **5. Assessment verdict**

5118 • Pass: security baselines can be defined/used for critical components (IOMMU, virtual switch, host firewall);  
 5119 baseline validation can be run automatically; and non-compliant settings trigger alerts or audit events  
 5120 identifying the deviations.

5121 • Fail: security baselines cannot be defined or do not cover the required components; automatic validation  
 5122 against a baseline is not available; or non-compliance does not result in clear alerts/audit events.

5123 **6. Assessment evidence**

5124 • [CONFIG] Example security baseline definition and corresponding Hypervisor configuration (before and after  
 5125 deviation).

5126 • [DESIGN] Documentation of security baseline management, validation logic and alert/audit integration.

5127 • [LOG] / [OBSERVATION] Outputs from baseline validation showing compliant and non-compliant states,  
 5128 including generated alerts or audit log entries identifying non-compliant settings.

5129 **6.3.1.17 Assessment for Data Minimization**5130 **6.3.1.17.1 Assessment Case AC-H-DM-001**5131 **1. Assessment reference**

5132 Requirement: REQ-H-DM-001 (Basic)

5133 **2. Assessment objective**

5134 Verify that:

- 5135 1) Hypervisor logs, telemetry and diagnostic outputs are limited to information required for operation,  
 5136 troubleshooting and security monitoring.
- 5137 2) Logs/telemetry/diagnostics do not contain sensitive security material such as plaintext authentication  
 5138 credentials, full cryptographic keys or complete VM memory contents.
- 5139 3) Payload or application data from guest workloads is not included in Hypervisor logs/telemetry/diagnostics by  
 5140 default.
- 5141 4) Where inclusion of payload content is supported for debugging, it is only enabled by explicit administrative  
 5142 configuration and this state is clearly indicated to the administrator.

5143 **3. Assessment preparation**

5144 The assessment shall have access to:

- 5145 • Documentation of Hypervisor logging/telemetry/diagnostic mechanisms, including examples of fields and any  
 5146 "debug" or "deep inspection" modes.
- 5147 • Documentation describing which information is explicitly excluded from logs/telemetry (credentials, keys,  
 5148 VM memory, guest payload).
- 5149 • A test configuration with:
- 5150 - administrative access to configure logging/telemetry/diagnostics;
- 5151 - at least one VM/guest workload that processes identifiable test payload data;
- 5152 - ability to generate authentication events and crypto operations with known test values.

5153 **4. Assessment activities**

5154 The assessment shall at least:

- 5155 1) Review documentation and configuration to identify:
- 5156 - what categories of events and data are logged/collected by default; and

- 5157 - any options for deep/verbose/debug logging that may include payload or detailed context.
- 5158 2) In default configuration, perform:
- 5159 - successful and failed administrative authentication attempts using known test usernames/password  
5160 patterns;
- 5161 - crypto operations (e.g. key generation/usage) with identifiable test keys;
- 5162 - normal VM workload execution with test payload data (e.g. specific strings or patterns in application  
5163 traffic).
- 5164 Then inspect logs, telemetry and diagnostics to verify that:
- 5165 - plaintext credentials are not present;
- 5166 - full cryptographic keys or complete VM memory dumps are not present; and
- 5167 - guest payload/application data is not present by default.
- 5168 3) Where the Hypervisor supports payload inclusion for debugging, enable the relevant option(s) using  
5169 administrative configuration and verify that:
- 5170 - only an administrator can enable this behaviour;
- 5171 - configuration/status clearly indicates that payload capture/debug logging is enabled; and
- 5172 - after enabling, test payload data from guest workloads appears in logs/diagnostics as documented.

#### 5173 5. Assessment verdict

- 5174 • Pass: by default, logs/telemetry/diagnostics are limited to necessary operational and security data; they do not  
5175 contain plaintext credentials, full keys or complete VM memory; guest payload data is excluded by default;  
5176 and any payload inclusion requires explicit administrative configuration and is clearly indicated.
- 5177 • Fail: sensitive security material or guest payload data is present in default logs/telemetry/diagnostics; or  
5178 payload/debug capture can be enabled without explicit administrative action or without clear indication to the  
5179 administrator.

#### 5180 6. Assessment evidence

- 5181 • [CONFIG] Logging/telemetry/diagnostic configuration (default and with debug/payload options, if any).
- 5182 • [DESIGN] Documentation of data minimization behaviour, including exclusions for credentials, keys, VM  
5183 memory and guest payload.
- 5184 • [LOG] / [OBSERVATION] Samples of logs/telemetry/diagnostics under default configuration and with  
5185 debug/payload options enabled, demonstrating absence/presence of test secrets and payload as appropriate.

### 5186 6.3.1.17.2 Assessment Case AC-H-DM-002

#### 5187 1. Assessment reference

5188 Requirement: REQ-H-DM-002 (Basic)

#### 5189 2. Assessment objective

5190 Verify that:

- 5191 1) The Hypervisor provides administrative controls to restrict and, where necessary, disable collection of specific  
5192 categories of log, telemetry and diagnostic data.
- 5193 2) These controls include the ability to disable collection of particular log types, telemetry streams, metrics, or  
5194 debug payload capture.

#### 5195 3. Assessment preparation

5196 The assessment shall have access to:

- 5197 • Documentation describing configurable categories of logs, telemetry streams, metrics and any debug/payload  
5198 capture options.
- 5199 • Documentation of administrative roles/permissions required to modify logging/telemetry configuration.
- 5200 • A test configuration where different log categories, telemetry streams and metrics can be enabled/disabled and  
5201 where events can be generated to exercise them.

#### 5202 4. Assessment activities

5203 The assessment shall at least:

- 5204 3) Identify, from documentation and configuration, the configurable categories (e.g. audit logs vs debug logs,  
5205 specific telemetry streams, metrics groups, payload capture options).
- 5206 4) Using administrative controls, configure a subset of categories to be disabled (e.g. disable a particular debug  
5207 log type, a telemetry stream and, if supported, debug payload capture).
- 5208 5) Generate events and workload activity that would normally produce entries in the disabled categories and  
5209 verify that:
  - 5210 - no new records are produced for the disabled log types/streams/metrics; and
  - 5211 - other, non-disabled categories continue to function as configured.
- 5212 6) Re-enable one or more previously disabled categories and verify that records are again produced for those  
5213 categories when new events occur.

#### 5214 5. Assessment verdict

- 5215 • Pass: administrators can restrict/disable specific log types, telemetry streams, metrics and any debug payload  
5216 capture; disabled categories no longer produce data while others continue to operate; and re-enabling restores  
5217 data collection as configured.
- 5218 • Fail: logging/telemetry categories cannot be selectively restricted or disabled; or documented categories  
5219 remain collected despite being configured as disabled.

#### 5220 6. Assessment evidence

- 5221 • [CONFIG] Logging/telemetry/metrics configuration before and after enabling/disabling categories, including  
5222 debug/payload options.
- 5223 • [DESIGN] Documentation of administrative controls for data collection and category granularity (log types,  
5224 streams, metrics, payload capture).
- 5225 • [LOG] / [OBSERVATION] Evidence showing presence/absence of data from specific categories when  
5226 enabled vs disabled, demonstrating effective restriction and disabling.

### 5227 6.3.2 M&O System

5228 [To be completed in a future version]

## 5229 6.4 CES

### 5230 6.4.1 CRS

#### 5231 6.4.1.1 Assessment for Container Isolation

##### 5232 6.4.1.1.1 Assessment Case AC-CRS-CN-ISO-001

#### 5233 1. Assessment reference

5234 Requirement: REQ-CRS-CN-ISO-001 (Basic)

## 5235 2. Assessment objective

5236 Verify that:

- 5237 1) The CRS enforces isolation between containers using OS isolation mechanisms under its control, including at  
5238 least process, filesystem, network and resource management layers.
- 5239 2) The CRS supports configurations in which one container cannot read, write or otherwise interfere with another  
5240 container's processes, filesystem contents, network endpoints or allocated resources.

## 5241 3. Assessment preparation

5242 The assessment shall have access to:

- 5243 • Documentation of the CRS container isolation model (e.g. namespaces, cgroups/quotas, virtual networking,  
5244 per-container filesystems).
- 5245 • Documentation describing how containers are configured to achieve strict isolation (e.g. non-shared PID  
5246 (Process ID) /IPC (Inter-Process Communication) /network/mount namespaces, independent root filesystems).
- 5247 • A test environment with at least two containers running on the same CRS-managed host, under different  
5248 identities/workloads, with the ability to execute commands inside each container.

## 5249 4. Assessment activities

5250 The assessment shall at least:

- 5251 1) Process isolation
- 5252 - From container A, list processes and attempt to inspect or signal processes belonging to container B.
- 5253 - Verify that only processes of container A are visible/controllable.
- 5254 2) Filesystem isolation
- 5255 - From container A, enumerate filesystem paths and attempt to access directories or files that are part of  
5256 container B's filesystem (other than intentionally shared volumes, if any).
- 5257 - Verify that container A cannot read or modify container B's filesystem contents.
- 5258 3) Network isolation
- 5259 - Identify network endpoints assigned to each container.
- 5260 - From container A, attempt to bind to IP addresses/ports assigned to container B or otherwise interfere  
5261 with B's network endpoints beyond normal, configured connectivity (e.g. service-level communication).
- 5262 - Verify that container A cannot hijack or directly interfere with container B's endpoints.
- 5263 4) Resource isolation
- 5264 - Review configured resource controls (e.g. CPU and memory limits) and generate load in container A.
- 5265 - Verify that container B continues to operate as expected and that its resource allocations behave  
5266 according to the configured limits, without arbitrary starvation caused by A beyond normal scheduling  
5267 behaviour.

## 5268 5. Assessment verdict

- 5269 • Pass: documentation and tests confirm that process, filesystem, network and resource isolation are enforced,  
5270 and that one container cannot read, write or otherwise interfere with another container's execution, data or  
5271 configuration state in the assessed configuration.
- 5272 • Fail: a container can see or control another container's processes, access its filesystem unexpectedly, interfere  
5273 with its network endpoints or consume resources in a way that bypasses configured isolation.

## 5274 6. Assessment evidence

- 5275 • [SCOPE] Description of CRS deployment, host and containers used for the isolation assessment.

- 5276 • [CONFIG] CRS configuration for namespaces, networking, storage and resource limits for the test containers.
- 5277 • [DESIGN] Documentation of CRS container isolation mechanisms across process, filesystem, network and  
5278 resource layers.
- 5279 • [OBSERVATION] / [LOG] Command outputs/logs showing isolation behaviour and blocked cross-container  
5280 access attempts.

#### 5281 6.4.1.1.2 Assessment Case AC-CRS-CN-ISO-002

##### 5282 1. Assessment reference

5283 Requirement: REQ-CRS-CN-ISO-002 (Elevated)

##### 5284 2. Assessment objective

5285 Verify that:

- 5286 1) The CRS supports configuration of strong separation between containers and the host OS by restricting  
5287 container access to kernel interfaces, privileged operations and system resources outside their assigned  
5288 execution context.
- 5289 2) The CRS can:
  - 5290 - limit container privileges (e.g. reduced kernel capabilities, prevention of access to host-level device  
5291 nodes and sensitive filesystems); and
  - 5292 - restrict container access to kernel interfaces and system calls not required for the declared workload,  
5293 using mechanisms such as syscall filtering, mandatory access control (MAC) policies or equivalent  
5294 controls.

##### 5295 3. Assessment preparation

5296 The assessment shall have access to:

- 5297 • Documentation of CRS mechanisms for strong container-host separation (e.g. capabilities model, device-node  
5298 access controls, mount policies, syscall filtering, MAC policies).
- 5299 • Documentation of typical secure configurations (profiles, templates or policies) for non-privileged containers.
- 5300 • A test environment where containers can be created with differing privilege profiles (e.g. reduced capabilities,  
5301 syscall filters, MAC labels).

##### 5302 4. Assessment activities

5303 The assessment shall at least:

- 5304 1) Privilege reduction
  - 5305 - Configure a container with a restricted privilege profile as recommended (e.g. reduced capabilities, non-  
5306 privileged user, no direct device-node access).
  - 5307 - Inside the container, attempt operations that require elevated privileges or kernel capabilities (e.g.  
5308 loading kernel modules, raw socket creation, mounting host filesystems, accessing /dev nodes that should  
5309 be restricted).
  - 5310 - Verify that such operations are denied.
- 5311 2) Host resource protection
  - 5312 - From the restricted container, attempt to access sensitive host filesystems or directories (e.g. host root,  
5313 system configuration directories, host-level logs) that are not explicitly mounted into the container.
  - 5314 - Verify that these locations are not accessible.
- 5315 3) Kernel interface and syscall restriction
  - 5316 - Configure syscall filtering or equivalent kernel-interface restriction (e.g. seccomp profiles, MAC rules)  
5317 for the container according to documentation for the declared workload.

- 5318 - From within the container, attempt to invoke known disallowed system calls or interact with kernel  
5319 interfaces that the profile is intended to block.
- 5320 - Verify that such calls are blocked (e.g. denied, terminated) while permitted system calls required for  
5321 normal workload operation continue to function.

## 5322 5. Assessment verdict

- 5323 • Pass: the CRS can configure strong separation between containers and the host; privilege reduction  
5324 mechanisms are effective; container access to device nodes and sensitive filesystems is restricted; and  
5325 syscall/filtering/MAC-equivalent controls prevent access to non-required kernel interfaces while allowing  
5326 necessary workload functions.
- 5327 • Fail: containers can retain broad privileges, directly access sensitive host resources, or bypass syscall/interface  
5328 restrictions in the assessed secure configuration.

## 5329 6. Assessment evidence

- 5330 • [CONFIG] Container profiles showing reduced capabilities, device/mount restrictions and syscall/MAC  
5331 policies.
- 5332 • [DESIGN] Documentation of CRS mechanisms for strong container-host separation and recommended secure  
5333 profiles.
- 5334 • [OBSERVATION] / [LOG] Outputs showing denied privileged operations, blocked access to host resources  
5335 and enforcement of syscall/MAC restrictions.

### 5336 6.4.1.1.3 Assessment Case AC-CRS-CN-ISO-003

#### 5337 1. Assessment reference

5338 Requirement: REQ-CRS-CN-ISO-003 (Advanced)

#### 5339 2. Assessment objective

5340 Verify that:

- 5341 1) The CRS supports execution of containers within hardened isolation environments that strengthen separation  
5342 between container workloads and the host kernel (e.g. micro-VM-based runtimes, user-space kernel  
5343 sandboxes, additional Virtualization/sandbox layers or equivalent).
- 5344 2) In such hardened environments, compromise of a container does not directly grant an attacker equivalent  
5345 access to the host kernel or to other containers.

#### 5346 3. Assessment preparation

5347 The assessment shall have access to:

- 5348 • Documentation describing hardened isolation environments supported by the CRS for containers (e.g.  
5349 architecture diagrams and isolation properties of micro-VM runtimes, sandbox layers).
- 5350 • Documentation explaining configuration and operational conditions for enabling hardened isolation for  
5351 selected workloads.
- 5352 • A test configuration where at least one container can be deployed in a hardened isolation mode and another  
5353 container in a standard mode on the same host.

#### 5354 4. Assessment activities

5355 The assessment shall at least:

- 5356 1) Hardened environment configuration
- 5357 - Configure a container to run using the CRS hardened isolation mechanism (e.g. micro-VM-backed  
5358 container, user-space kernel sandbox).
- 5359 - Verify from configuration/status that the container is indeed running with the hardened isolation mode  
5360 enabled as documented.

- 5361 2) Containment of compromise within hardened environment
- 5362 - Within the hardened-isolation container, attempt operations that would normally be indicative of host  
5363 compromise if successful (e.g. direct interaction with host kernel interfaces, access to host filesystems,  
5364 access to other containers' namespaces or storage, escalation to host-level privileges).
- 5365 - Verify that such operations are confined or blocked by the hardened isolation layer and do not result in  
5366 direct access to the host kernel or other containers.
- 5367 3) Separation from other containers
- 5368 - With at least one additional container running (standard or hardened), attempt from the hardened  
5369 container to interact with or control the other container beyond normal, configured communication (e.g.  
5370 attach to its processes, access its filesystems, inject code).
- 5371 - Verify that these attempts fail and that cross-container compromise via the hardened environment is  
5372 prevented.
- 5373 4) Administrative visibility
- 5374 - Verify that administrators can identify containers running with hardened isolation (e.g. via configuration,  
5375 labels, status commands or logs), so that use of this mode is auditable and can be targeted to high-risk  
5376 workloads.

## 5377 5. Assessment verdict

- 5378 • Pass: the CRS supports hardened isolation environments for containers; such environments can be configured  
5379 for selected workloads; and in the assessed configuration, compromise of a container within a hardened  
5380 environment does not directly provide equivalent access to the host kernel or other containers.
- 5381 • Fail: hardened isolation mode is not available or cannot be configured; or a compromised container in a  
5382 hardened environment can directly access the host kernel or other containers in ways inconsistent with the  
5383 strengthened isolation objective.

## 5384 6. Assessment evidence

- 5385 • [CONFIG] CRS configuration showing containers deployed in hardened isolation mode and any associated  
5386 policies.
- 5387 • [DESIGN] Documentation of hardened isolation architecture, including threat model and separation properties  
5388 with respect to the host kernel and other containers.
- 5389 • [PLT-CAP] Where applicable, evidence of platform capabilities used to implement hardened environments  
5390 (e.g. hardware or software Virtualization features).
- 5391 • [OBSERVATION] Logs/traces and test results demonstrating that host/kernel and cross-container access  
5392 attempts from a hardened container are blocked or confined as designed.

### 5393 6.4.1.2 Assessment for Control Plane Isolation

#### 5394 6.4.1.2.1 Assessment Case AC-CRS-CP-ISO-001

##### 5395 1. Assessment reference

5396 Requirement: REQ-CRS-CP-ISO-001 (Basic)

##### 5397 2. Assessment objective

5398 Verify that:

- 5399 1) The CRS provides access-control mechanisms for its management and control interfaces.
- 5400 2) The CRS can be configured such that containerized workloads cannot directly access or invoke CRS  
5401 management or control functions.
- 5402 3) Only authenticated and authorized entities can perform control or configuration actions.

- 5403 4) CRS management sockets, control APIs and management daemons can be deployed so they are not reachable  
5404 from within container namespaces.

### 5405 3. Assessment preparation

5406 The assessment shall have access to:

- 5407 • Documentation of CRS management and control interfaces (local sockets, REST APIs, CLIs, management  
5408 daemons) and their access-control mechanisms.
- 5409 • Documentation of how these interfaces are bound/deployed (e.g. host network namespace, Unix domain  
5410 sockets, host-only endpoints).
- 5411 • A test environment with:
  - 5412 - at least one administrative account or management client;
  - 5413 - at least one container with its own namespaces;
  - 5414 - the ability to run commands inside containers and on the host.

### 5415 4. Assessment activities

5416 The assessment shall at least:

- 5417 1) Identify all CRS management/control interfaces and confirm that authentication and authorization controls are  
5418 configured (e.g. accounts, roles, policies).
- 5419 2) From an authorized management context (host or management workstation), perform control actions (e.g.  
5420 create/delete container, change policy) and verify that successful access requires authentication and that  
5421 authorization (role/permission) checks are enforced.
- 5422 3) From within a container:
  - 5423 - enumerate reachable network endpoints and local sockets;
  - 5424 - attempt to connect to CRS management sockets, control APIs or management daemons using no  
5425 credentials and container-level credentials;
  - 5426 - attempt to perform control/configuration actions.

5427 Verify that:

- 5428 - CRS management endpoints are not reachable from within container namespaces (e.g. bound only to host  
5429 network namespace or host-only sockets); and
- 5430 - container principals cannot perform management or control actions.

### 5431 5. Assessment verdict

- 5432 • Pass: access-control mechanisms exist; CRS can be deployed so that management/control interfaces are not  
5433 reachable from container namespaces; and only authenticated, authorized entities can perform  
5434 control/configuration actions.
- 5435 • Fail: containers can reach and invoke CRS management/control functions; or management interfaces do not  
5436 enforce proper authentication/authorization.

### 5437 6. Assessment evidence

- 5438 • [CONFIG] CRS management/control interface configuration, including bindings (host-only vs container-  
5439 visible) and access-control settings.
- 5440 • [DESIGN] Documentation of CRS management/control model and access-control mechanisms.
- 5441 • [OBSERVATION] / [LOG] Evidence of successful authorized management actions and failed/blocked  
5442 attempts from within containers to reach management endpoints or perform control actions.

## 5443 6.4.1.2.2 Assessment Case AC-CRS-CP-ISO-002

5444 **1. Assessment reference**

5445 Requirement: REQ-CRS-CP-ISO-002 (Elevated)

5446 **2. Assessment objective**

5447 Verify that:

- 5448 1) The CRS supports configuration of dedicated logical separation between CRS management and control traffic  
5449 and container data-plane traffic.
- 5450 2) The CRS supports the use of host-only communication endpoints, distinct network namespaces or dedicated  
5451 management networks for its management/control interfaces, enabling isolation of CRS control traffic from  
5452 container workloads.

5453 **3. Assessment preparation**

5454 The assessment shall have access to:

- 5455 • Documentation describing how CRS management/control interfaces can be exposed over host-only endpoints,  
5456 distinct network namespaces or dedicated management networks.
- 5457 • Documentation of recommended topologies for separating management/control traffic from container data-  
5458 plane traffic.
- 5459 • A test configuration with:
- 5460 - at least one management/control endpoint exposed via host-only or dedicated management connectivity;
- 5461 - at least one container data-plane network/namespace;
- 5462 - tools to inspect network namespaces, interfaces, routes and firewall rules.

5463 **4. Assessment activities**

5464 The assessment shall at least:

- 5465 1) Configure the CRS so that management/control interfaces use one of the supported separation mechanisms  
5466 (e.g. host-only socket, host network namespace only, dedicated management network or VLAN).
- 5467 2) Verify via configuration and inspection that:
- 5468 - CRS management/control endpoints are bound only to the designated management/host-only  
5469 mechanisms; and
- 5470 - they are not exposed on container data-plane networks or in container network namespaces.
- 5471 3) From a container attached only to the data-plane network/namespace, attempt to connect to CRS  
5472 management/control endpoints; verify that such attempts are blocked or cannot reach the endpoints.
- 5473 4) From an administrative context on the host or on the dedicated management network, verify that  
5474 management/control interfaces are reachable via the designated paths, and that container data traffic does not  
5475 traverse those paths.

5476 **5. Assessment verdict**

- 5477 • Pass: CRS management/control traffic can be logically separated from container data-plane traffic using host-  
5478 only endpoints, distinct network namespaces or dedicated management networks; and container data-plane  
5479 networks cannot be used to reach CRS management/control interfaces in the assessed configuration.
- 5480 • Fail: no effective mechanism to separate CRS management/control and container data-plane traffic; or  
5481 management/control interfaces remain reachable over container data-plane networks despite intended  
5482 separation.

5483 **6. Assessment evidence**

- 5484 • [CONFIG] Network and namespace configuration showing bindings of CRS management/control interfaces
- 5485 and separation from container data-plane networks.
- 5486 • [DESIGN] Documentation of supported mechanisms for control-plane vs data-plane separation.
- 5487 • [OBSERVATION] / [LOG] Connection tests from container and management contexts demonstrating blocked
- 5488 access from data-plane and permitted access via the management/host-only paths.

### 5489 6.4.1.2.3 Assessment Case AC-CRS-CP-ISO-003

#### 5490 1. Assessment reference

5491 Requirement: REQ-CRS-CP-ISO-003 (Advanced)

#### 5492 2. Assessment objective

5493 Verify that:

- 5494 1) The CRS supports cryptographically protected and mutually authenticated communication channels for all
- 5495 remote or network-exposed CRS management and control interfaces.
- 5496 2) When such channels are used, management/control traffic is transmitted only over communication paths that
- 5497 provide confidentiality, integrity protection and mutual authentication between the CRS and authorized
- 5498 management entities.
- 5499 3) For CRS management interfaces restricted to local host-only mechanisms, equivalent protection is provided by
- 5500 underlying platform access controls, consistent with the NOTE.

#### 5501 3. Assessment preparation

5502 The assessment shall have access to:

- 5503 • Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic
- 5504 protocols and mutual authentication options (client certificates, key-based methods).
- 5505 • Documentation for configuring certificates/keys and enabling mutually authenticated channels.
- 5506 • Documentation of host-only management mechanisms (e.g. Unix sockets, loopback-only endpoints) and the
- 5507 associated OS access-control model.
- 5508 • A test configuration with:
  - 5509 - at least one remote/network-exposed management/control interface;
  - 5510 - representative host-only management mechanisms, if provided;
  - 5511 - one management client with valid credentials and another with invalid/missing credentials.

#### 5512 4. Assessment activities

5513 The assessment shall at least:

5514 For remote/network-exposed management/control interfaces:

- 5515 1) Configure each such interface to use a cryptographically protected transport with mutual authentication as
- 5516 documented.
- 5517 2) Establish a management session using a client with valid credentials (certificates/keys) and verify via protocol
- 5518 inspection and logs that:
  - 5519 - the session is encrypted;
  - 5520 - the CRS authenticates the client; and
  - 5521 - the client authenticates the CRS (e.g. server certificate validation).
- 5522 3) Attempt connections from clients without valid credentials or using disallowed protocol/cipher configurations
- 5523 and verify that these are rejected and cannot perform management/control actions.

- 5524 4) Verify that no alternative cleartext or unauthenticated network paths exist for CRS management/control in the  
5525 assessed configuration (e.g. no parallel HTTP port without TLS).

5526 For host-only management mechanisms (where present):

- 5527 1) Verify that such interfaces are bound only to host-local mechanisms (e.g. Unix domain sockets, loopback-only  
5528 addresses) and not reachable from remote networks or container namespaces.
- 5529 2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these  
5530 host-only interfaces to authorized local principals, providing confidentiality, integrity and authenticity  
5531 comparable to the cryptographic protections required on remote interfaces.

## 5532 5. Assessment verdict

- 5533 • Pass: all remote/network-exposed CRS management/control interfaces can use cryptographically protected,  
5534 mutually authenticated channels; management/control traffic over these interfaces is confined to  
5535 communication paths providing confidentiality, integrity and mutual authentication; and host-only  
5536 management mechanisms rely on OS access controls that provide equivalent protection without exposing  
5537 network-reachable cleartext admin paths.
- 5538 • Fail: some remote management/control interfaces cannot be protected cryptographically or cannot use mutual  
5539 authentication; cleartext or unauthenticated admin channels remain reachable; or host-only mechanisms are  
5540 exposed or insufficiently protected.

## 5541 6. Assessment evidence

- 5542 • [CONFIG] Cryptographic and authentication configuration for remote management/control interfaces, and  
5543 configuration for host-only management mechanisms (bindings, permissions).
- 5544 • [DESIGN] Documentation of supported management/control protocols, mutual authentication mechanisms  
5545 and security model for host-only interfaces.
- 5546 • [LOG] / [OBSERVATION] Protocol traces and logs showing encrypted, mutually authenticated sessions,  
5547 failed attempts without valid credentials, absence of cleartext admin paths, and enforcement of OS access  
5548 controls on host-only interfaces.

### 5549 6.4.1.3 Assessment for Network Plane Isolation

#### 5550 6.4.1.3.1 Assessment Case AC-CRS-NP-ISO-001

##### 5551 1. Assessment reference

5552 Requirement: REQ-CRS-NP-ISO-001 (Basic)

##### 5553 2. Assessment objective

5554 Verify that:

- 5555 1) The CRS maintains logical separation between container network traffic and host/CRS management network  
5556 traffic using virtual networking mechanisms under its control.
- 5557 2) The CRS supports configurations in which containers cannot directly access host or CRS management  
5558 interfaces/services by default.
- 5559 3) Separate network contexts can be defined for management traffic and container workload traffic.

##### 5560 3. Assessment preparation

5561 The assessment shall have access to:

- 5562 • Documentation of the CRS networking model (container networks, host network, management endpoints,  
5563 virtual switches, CNI/plugin, etc.).
- 5564 • Documentation describing how management endpoints (APIs, daemons, sockets) are bound to  
5565 host/management networks versus container networks.
- 5566 • A test configuration with:

- 5567 - at least one container network with one or more containers;
- 5568 - CRS/host management interfaces bound to host/management network context;
- 5569 - tools to list network interfaces/namespaces and test connectivity from containers and host.

#### 5570 4. Assessment activities

5571 The assessment shall at least:

- 5572 1) Configure the CRS so that:
  - 5573 - containers are attached to one or more container networks; and
  - 5574 - CRS/host management interfaces are exposed only on host or dedicated management network contexts,
  - 5575 as documented.
- 5576 2) From within a container, enumerate network interfaces, routing tables and reachable IP/ports, and attempt to connect to:
  - 5577
  - 5578 - CRS management APIs;
  - 5579 - CRS management daemons;
  - 5580 - host administrative services.

5581 Verify that these interfaces/services are not reachable by default.
- 5582 3) From a host/management context, verify that management interfaces are reachable via the intended
  - 5583 management/host network and that container data-plane traffic uses separate network contexts.

#### 5584 5. Assessment verdict

- 5585 • Pass: virtual networking under CRS control provides logical separation; containers cannot directly access host
  - 5586 or CRS management interfaces/services by default; and separate network contexts for management and
  - 5587 container workload traffic can be configured and observed.
- 5588 • Fail: container traffic can reach host/CRS management interfaces by default; or management and container
  - 5589 traffic share the same network context without separation options.

#### 5590 6. Assessment evidence

- 5591 • [SCOPE] Description of the CRS deployment, container networks, and management/host networks used in the
  - 5592 assessment.
- 5593 • [CONFIG] CRS networking configuration (CNI or equivalent), interface bindings for management endpoints,
  - 5594 container network definitions.
- 5595 • [DESIGN] Documentation of CRS network plane separation model and default exposure of management vs
  - 5596 container traffic.
- 5597 • [OBSERVATION] / [LOG] Connectivity tests and command outputs from containers and host showing lack
  - 5598 of default access from containers to management/host interfaces.

#### 5599 6.4.1.3.2 Assessment Case AC-CRS-NP-ISO-002

##### 5600 1. Assessment reference

5601 Requirement: REQ-CRS-NP-ISO-002 (Elevated)

##### 5602 2. Assessment objective

5603 Verify that:

- 5604 1) The CRS supports configuration of explicit traffic filtering and access-control rules for:
  - 5605 - flows between containers;
  - 5606 - flows between container networks;

- 5607 - flows between container networks and host/management networks.
- 5608 2) These rules are enforced at the CRS virtual networking layer or other enforcement points under CRS control.
- 5609 3) Only network flows explicitly permitted by policy are allowed to cross these boundaries.

### 5610 3. Assessment preparation

5611 The assessment shall have access to:

- 5612 • Documentation of CRS mechanisms for defining and enforcing network policies (e.g. network policies,  
5613 security groups, virtual firewall rules, CNI plug-in ACLs).
- 5614 • Documentation with examples of permitted and denied flows.
- 5615 • A test configuration with:
  - 5616 - at least two container networks and multiple containers;
  - 5617 - host/management network reachable from the CRS;
  - 5618 - the ability to define and modify network policies and to generate traffic between these endpoints.

### 5619 4. Assessment activities

5620 The assessment shall at least:

- 5621 1) Define and apply network policies or equivalent rules at the CRS virtual networking layer (or other CRS-  
5622 controlled enforcement point) to:
  - 5623 - allow specific flows (e.g. container A in network X to container B in network Y on a given port;  
5624 management endpoint to specific container agents); and
  - 5625 - deny all other cross-container, cross-network and container-to-host/management flows by default.
- 5626 2) Generate test traffic:
  - 5627 - flows that match permitted rules;
  - 5628 - flows that should be blocked (e.g. arbitrary container-to-container connections across networks,  
5629 container-to-management attempts outside permitted scope).
- 5630 3) Verify that:
  - 5631 - permitted flows succeed as configured;
  - 5632 - non-permitted flows are blocked at the CRS-controlled enforcement points; and
  - 5633 - adjusting rules (add/remove) changes allowed/denied flows as expected.

### 5634 5. Assessment verdict

- 5635 • Pass: the CRS can define and enforce explicit traffic filtering/access-control rules between containers,  
5636 container networks and host/management networks, and only flows explicitly permitted by policy can cross  
5637 these boundaries.
- 5638 • Fail: policies cannot be defined at the CRS-controlled layer; enforcement is ineffective; or cross-boundary  
5639 traffic bypasses the configured rules.

### 5640 6. Assessment evidence

- 5641 • [CONFIG] Network policy and access-control rule definitions and their association with container networks  
5642 and host/management networks.
- 5643 • [DESIGN] Documentation of CRS virtual networking enforcement mechanisms for inter-container and cross-  
5644 plane traffic.

- 5645 • [OBSERVATION] / [LOG] Packet captures, policy logs or connectivity test results showing success of  
5646 permitted flows and blocking of non-permitted flows.

### 5647 6.4.1.3.3 Assessment Case AC-CRS-NP-ISO-003

#### 5648 1. Assessment reference

5649 Requirement: REQ-CRS-NP-ISO-003 (Advanced)

#### 5650 2. Assessment objective

5651 Verify that:

- 5652 1) The CRS supports cryptographically protected and mutually authenticated communication channels for  
5653 management and control traffic that traverses shared or untrusted networks.
- 5654 2) When such channels are used, CRS management/control traffic is transmitted only over paths that provide  
5655 confidentiality, integrity protection and mutual authentication between the CRS and authorized management  
5656 entities.
- 5657 3) For management interfaces restricted to local host-only mechanisms, equivalent protection is provided by OS  
5658 platform access controls, consistent with the NOTE.

#### 5659 3. Assessment preparation

5660 The assessment shall have access to:

- 5661 • Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic  
5662 protocols (e.g. TLS, mTLS, SSH, VPN/IPsec) and mutual authentication methods (client certificates, keys,  
5663 tokens).
- 5664 • Documentation explaining how to configure these protected channels and how they are bound to  
5665 management/control traffic.
- 5666 • Documentation of host-only management interfaces (e.g. Unix domain sockets, loopback-only endpoints) and  
5667 their OS-level access-control model.
- 5668 • A test configuration with:
- 5669 - at least one CRS management/control interface exposed over a shared or untrusted network;
- 5670 - at least one representative host-only management mechanism;
- 5671 - management clients with valid and invalid credentials.

#### 5672 4. Assessment activities

5673 For management/control interfaces over shared or untrusted networks, the assessment shall at least:

- 5674 1) Configure each such interface to use cryptographically protected transport with mutual authentication as  
5675 documented (e.g. mTLS for control APIs, SSH with key-based admin access, VPN/IPsec tunnels for remote  
5676 management).
- 5677 2) From a client with valid credentials, establish a management/control session and verify via protocol inspection  
5678 and logs that:
- 5679 - the session is encrypted;
- 5680 - the CRS authenticates the client;
- 5681 - the client authenticates the CRS.
- 5682 3) Attempt connections from clients without valid credentials or using disallowed protocol or cipher  
5683 configurations and verify that these are rejected and cannot perform management or control actions.
- 5684 4) Verify that no cleartext or unauthenticated alternative management/control paths are reachable over the  
5685 shared/untrusted network in the assessed configuration.

5686 For host-only management interfaces, the assessment shall at least:

- 5687 1) Verify that such interfaces are bound only to local mechanisms (e.g. Unix domain sockets, loopback-only  
5688 addresses) and are not reachable from container namespaces or external networks.
- 5689 2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these  
5690 interfaces to authorized local principals, providing confidentiality, integrity and authenticity comparable to  
5691 what cryptographic protection would provide on a network path.

## 5692 5. Assessment verdict

- 5693 • Pass: the CRS supports cryptographically protected, mutually authenticated channels for management/control  
5694 traffic over shared/untrusted networks; such traffic is carried only over paths providing confidentiality,  
5695 integrity and mutual authentication; and host-only management mechanisms rely on appropriate OS access  
5696 controls without exposing unprotected network-reachable management paths.
- 5697 • Fail: remote CRS management/control interfaces cannot be protected cryptographically or cannot use mutual  
5698 authentication; cleartext or unauthenticated management/control paths exist over shared/untrusted networks; or  
5699 host-only mechanisms are exposed beyond the host or lack sufficient OS-level protection.

## 5700 6. Assessment evidence

- 5701 • [CONFIG] Cryptographic and authentication configuration for remote CRS management/control interfaces;  
5702 configuration for host-only interfaces (bindings, permissions).
- 5703 • [DESIGN] Documentation of CRS control-plane security model, supported protocols, mutual authentication  
5704 mechanisms and host-only protection model.
- 5705 • [LOG] / [OBSERVATION] Protocol traces and logs demonstrating encrypted, mutually authenticated  
5706 management sessions, rejected invalid attempts, absence of cleartext management traffic on shared networks,  
5707 and enforcement of OS access controls for host-only mechanisms.

### 5708 6.4.1.4 Assessment for Boot Chain Integrity Verification

#### 5709 6.4.1.4.1 Assessment Case AC-CRS-B-INT-001

##### 5710 1. Assessment reference

5711 Requirement: REQ-CRS-B-INT-001 (Basic)

##### 5712 2. Assessment objective

5713 Verify that:

- 5714 1) The CRS implements an integrity verification mechanism for its core runtime engine and other CRS  
5715 executables required to start container execution services, based on trusted reference values.
- 5716 2) The CRS can invoke this integrity verification before enabling container scheduling, container lifecycle  
5717 operations, or other container execution interfaces.
- 5718 3) If integrity verification of any covered CRS executable fails, the CRS prevents startup of container execution  
5719 services and generates a security-relevant event.

##### 5720 3. Assessment preparation

5721 The assessment shall have access to:

- 5722 • Documentation describing the integrity verification mechanism for CRS executables, including which  
5723 executables are covered and how reference values are established and stored.
- 5724 • Documentation of the CRS startup sequence, showing when integrity verification is invoked relative to  
5725 enabling container execution services.
- 5726 • A test configuration that allows:
- 5727 - a normal startup with unmodified CRS executables; and
- 5728 - a controlled integrity-verification failure for at least one covered executable (e.g. tampered binary,  
5729 invalid signature, test mode), consistent with vendor guidance.

5730 **4. Assessment activities**

5731 The assessment shall at least:

- 5732 1) Review documentation and configuration to identify:
- 5733 - the set of CRS executables covered by integrity checking;
- 5734 - the verification mechanism and trusted reference values;
- 5735 - the point in startup where the check is invoked.
- 5736 2) Perform a normal CRS startup and verify that:
- 5737 - integrity verification is invoked as documented;
- 5738 - container execution services (scheduling, lifecycle operations, APIs) become available.
- 5739 3) Induce a failure in the integrity verification of one covered executable and restart the CRS. Verify that:
- 5740 - the verification failure is detected;
- 5741 - container execution services do not start;
- 5742 - a security-relevant event is generated and recorded.

5743 **5. Assessment verdict**

- 5744 • Pass: integrity verification is implemented for the core runtime engine and required CRS executables; it can be  
5745 invoked before container execution services are enabled; and a verification failure prevents startup and  
5746 produces a security-relevant event.
- 5747 • Fail: there is no effective integrity verification; integrity checks are not invoked before container execution  
5748 services; or verification failures do not block startup and are not logged as security-relevant events.

5749 **6. Assessment evidence**

- 5750 • [DESIGN] Description of the CRS executable integrity verification mechanism and startup sequence.
- 5751 • [CONFIG] Configuration showing enabled integrity checks and list of covered executables.
- 5752 • [LOG] / [OBSERVATION] Startup logs and traces for normal and failed verification scenarios, including the  
5753 recorded security-relevant event.

5754 **6.4.1.4.2 Assessment Case AC-CRS-B-INT-002**

5755 **1. Assessment reference**

5756 Requirement: REQ-CRS-B-INT-002 (Elevated)

5757 **2. Assessment objective**

5758 Verify that:

- 5759 1) The CRS implements mechanisms to participate in a verifiable chain of trust for CRS startup components  
5760 delivered as part of the CRS product and executed before container execution services are enabled.
- 5761 2) Each CRS startup stage delivered with the product and executed prior to exposing container execution  
5762 interfaces validates the integrity and authenticity of the subsequent CRS stage before transferring control.
- 5763 3) If validation fails for any such startup stage, the CRS prevents startup of container execution services and  
5764 generates a security-relevant event.

5765 **3. Assessment preparation**

5766 The assessment shall have access to:

- 5767 • Documentation of the CRS startup chain for components delivered with the product (e.g. CRS launchers,  
5768 shims, daemons that initialize container execution services).

- 5769 • Documentation describing how each CRS startup stage validates the integrity and authenticity of the next CRS  
5770 stage (e.g. signature checks, hash verification) and how this can integrate with a platform or CRS-defined  
5771 chain of trust.
- 5772 • A test configuration where:
- 5773 - normal startup with valid CRS startup components is possible;
- 5774 - a controlled validation failure can be induced for at least one CRS startup stage (e.g. modified or  
5775 unsigned binary, untrusted key), consistent with vendor guidance.

#### 5776 4. Assessment activities

5777 The assessment shall at least:

- 5778 1) Review documentation to identify all CRS startup stages delivered with the product that execute before  
5779 container execution services are enabled, and the validation performed at each stage.
- 5780 2) Perform a normal CRS startup and verify, using logs or diagnostic outputs, that:
- 5781 - each CRS startup stage validates the integrity/authenticity of the next stage before passing control;
- 5782 - container execution services are enabled only after successful validation of all such stages.
- 5783 3) Introduce a controlled integrity/authenticity failure in one CRS startup stage and restart the CRS. Verify that:
- 5784 - the validation failure at that stage is detected;
- 5785 - container execution services are not started;
- 5786 - a security-relevant event is generated and recorded.

#### 5787 5. Assessment verdict

- 5788 • Pass: all CRS startup stages delivered with the product and executed before container execution services  
5789 validate the integrity and authenticity of subsequent CRS stages; validation failures are detected; and container  
5790 execution services are prevented from starting, with a security-relevant event generated.
- 5791 • Fail: some CRS startup stages do not validate subsequent stages; failures are not detected; or validation  
5792 failures do not block container execution services or produce a security-relevant event.

#### 5793 6. Assessment evidence

- 5794 • [DESIGN] Documentation of the CRS startup chain, per-stage validation mechanisms and chain-of-trust  
5795 integration.
- 5796 • [CONFIG] Startup configuration and key/reference-value configuration used for stage validation.
- 5797 • [LOG] / [OBSERVATION] Startup logs or traces from normal and failure scenarios showing successful  
5798 chained validation, detection of failed validation and prevention of container service startup with an associated  
5799 security event.

### 5800 6.4.1.4.3 Assessment Case AC-CRS-B-INT-003

#### 5801 1. Assessment reference

5802 Requirement: REQ-CRS-B-INT-003 (Advanced)

#### 5803 2. Assessment objective

5804 Verify that:

- 5805 1) The CRS supports consuming integrity and authenticity information exposed by platform verified boot or  
5806 integrity measurement mechanisms for operating system components and CRS components that have to be  
5807 loaded and initialized before container execution services are enabled.
- 5808 2) The CRS supports policies that prevent initialization of container execution services when the reported  
5809 verification status for any required component in this chain indicates failure or cannot be obtained.

### 5810 3. Assessment preparation

5811 The assessment shall have access to:

- 5812 • Documentation of the platform's verified boot or integrity measurement mechanism and the interface through  
5813 which verification status is exposed (e.g. APIs, status files, attestation logs).
- 5814 • Documentation describing which OS and CRS components are considered "required" before container  
5815 execution services are enabled and how their verification status is mapped into CRS policy decisions.
- 5816 • Documentation of CRS policies and configuration options that use this verification status to control container  
5817 startup.
- 5818 • A test configuration where:
  - 5819 - platform verified boot/integrity measurement is enabled and exposes status for required components;
  - 5820 - CRS is configured to consume this status and enforce startup policies;
  - 5821 - verification failures or missing status can be simulated in a controlled way, consistent with  
5822 platform/vendor guidance.

### 5823 4. Assessment activities

5824 The assessment shall at least:

- 5825 1) Review documentation to identify the required OS and CRS components whose integrity/measurement status  
5826 is consumed by the CRS and understand how this status is obtained.
- 5827 2) Configure the CRS policy so that container execution services are allowed only when verification status for all  
5828 required components is reported as successful.
- 5829 3) In a normal scenario, boot the platform and start the CRS, ensuring that platform verified boot/integrity  
5830 mechanisms report successful status for all required components. Verify that:
  - 5831 - the CRS successfully consumes the verification status;
  - 5832 - container execution services are initialized.
- 5833 4) Induce a failure or unavailable status for one required component (e.g. configuration that causes a component  
5834 to fail verification, or removal of its measurement from the platform status interface) and restart. Verify that:
  - 5835 - the CRS detects that verification status for that component indicates failure or cannot be obtained;
  - 5836 - container execution services are not initialized;
  - 5837 - a security-relevant event is generated and recorded if required by CRS logging behaviour.

### 5838 5. Assessment verdict

- 5839 • Pass: the CRS can consume platform-exposed verification status for required OS and CRS components; it  
5840 supports policies that prevent initialization of container execution services when status indicates failure or is  
5841 unavailable; and those policies behave as documented in normal and failure scenarios.
- 5842 • Fail: the CRS cannot effectively consume platform verification status; startup policies do not prevent container  
5843 execution services when status is failed or missing; or behaviour does not match documentation.

### 5844 6. Assessment evidence

- 5845 • [DESIGN] Documentation of CRS integration with platform verified boot/integrity measurement mechanisms  
5846 and the mapping of verification status to startup policy.
- 5847 • [PLT-CAP] Evidence of platform capabilities and interfaces used to expose verification status.
- 5848 • [CONFIG] CRS policy configuration controlling container startup based on verification status.
- 5849 • [LOG] / [OBSERVATION] Logs and status outputs from normal and failure tests showing consumed  
5850 verification status and corresponding startup decisions for container execution services.

## 5851 6.4.1.5 Assessment for Container Image Integrity Verification

### 5852 6.4.1.5.1 Assessment Case AC-CRS-IMG-INT-001

#### 5853 1. Assessment reference

5854 Requirement: REQ-CRS-IMG-INT-001 (Basic)

#### 5855 2. Assessment objective

5856 Verify that:

- 5857 1) The CRS validates that the image manifest and all referenced layers match their declared cryptographic digests  
5858 before instantiation.
- 5859 2) This integrity verification is invoked automatically before creating or starting any container from an image.
- 5860 3) If any mismatch, corruption, or removal of a referenced layer or manifest entry is detected, the CRS rejects the  
5861 image, prevents container instantiation from that image, and records a security-relevant event.

#### 5862 3. Assessment preparation

5863 The assessment shall have access to:

- 5864 • Documentation describing the container image integrity verification mechanism, including how manifest and  
5865 layer digests are obtained and validated.
- 5866 • Documentation of the container creation/startup sequence showing where integrity verification is invoked.
- 5867 • A test environment with:
  - 5868 - at least one image available through the normal image distribution mechanism (e.g. local store or  
5869 registry);
  - 5870 - the ability to create a valid image with consistent manifest and layer digests;
  - 5871 - the ability to create an inconsistent or corrupted image (e.g. modified layer content without updating the  
5872 digest, missing layer, or modified manifest entry) in a controlled manner.

#### 5873 4. Assessment activities

5874 The assessment shall at least:

- 5875 1) Review documentation and configuration to identify:
  - 5876 - which images are subject to integrity verification;
  - 5877 - how the CRS validates manifest and layer digests;
  - 5878 - that verification is invoked automatically prior to container creation/start.
- 5879 2) Using a valid image whose manifest and layers match their declared digests:
  - 5880 - request creation or start of a container from that image;
  - 5881 - verify, using logs or diagnostics if available, that integrity verification is performed;
  - 5882 - confirm that the container is successfully instantiated when verification succeeds.
- 5883 3) Using an image modified so that at least one layer or manifest entry does not match its declared digest, or a  
5884 referenced layer is unavailable:
  - 5885 - request creation or start of a container from that image;
  - 5886 - verify that the CRS detects the mismatch or missing component;
  - 5887 - confirm that the image is rejected, container instantiation is prevented, and a security-relevant event is  
5888 recorded.

#### 5889 5. Assessment verdict

- 5890 • Pass: the CRS automatically validates manifest and layer digests before container creation/start; images with  
5891 consistent digests are allowed; images with mismatches, corruption, or missing layers are rejected, containers  
5892 are not instantiated, and a security-relevant event is recorded.
- 5893 • Fail: integrity verification is not applied, not automatic, or can be bypassed; or images with digest mismatches  
5894 or missing layers can be used to instantiate containers without rejection and without a recorded security event.

## 5895 6. Assessment evidence

- 5896 • [DESIGN] Documentation of the container image integrity verification mechanism and its position in the  
5897 container lifecycle.
- 5898 • [CONFIG] CRS configuration enabling image integrity verification and describing image sources.
- 5899 • [LOG] Records of successful verification and container start for valid images, and records of detected  
5900 mismatches, rejected images, and corresponding security-relevant events.
- 5901 • [OBSERVATION] Test execution logs or command outputs demonstrating behaviour for valid and corrupted  
5902 images.

### 5903 6.4.1.5.2 Assessment Case AC-CRS-IMG-INT-002

#### 5904 1. Assessment reference

5905 Requirement: REQ-CRS-IMG-INT-002 (Elevated)

#### 5906 2. Assessment objective

5907 Verify that:

- 5908 1) The CRS verifies both the integrity and the authenticity of container images before execution.
- 5909 2) Authenticity verification establishes that each image originates from a trusted publisher or approval authority  
5910 and has not been altered after approval.
- 5911 3) The CRS prevents execution or instantiation of any container image whose integrity or authenticity cannot be  
5912 successfully verified according to the configured trust policy.

#### 5913 3. Assessment preparation

5914 The assessment shall have access to:

- 5915 • Documentation describing the image trust model and verification mechanisms, including:
- 5916 - how trusted publishers or approval authorities are represented and configured;
- 5917 - how integrity and authenticity checks are combined (e.g. signatures, certificates, provenance metadata,  
5918 equivalent cryptographic mechanisms).
- 5919 • A test environment with:
- 5920 - at least one container image produced and "approved" according to the documented process (e.g.  
5921 properly signed or otherwise bound to a trusted publisher);
- 5922 - one or more images for which integrity or authenticity will fail, such as:
- 5923 ▪ an image modified after approval;
- 5924 ▪ an image missing authenticity information;
- 5925 ▪ an image signed or bound to an untrusted publisher, prepared in a controlled manner;
- 5926 - a configured trust policy specifying which publishers/authorities are trusted.

#### 5927 4. Assessment activities

5928 The assessment shall at least:

- 5929 1) Review documentation and configuration to identify:

- 5930 - the trust anchors or metadata used to represent trusted publishers/authorities;
- 5931 - how the CRS enforces the trust policy when verifying images.
- 5932 2) For an image that is correctly produced and approved under the trust policy:
- 5933 - request creation or start of a container from that image;
- 5934 - verify that both integrity and authenticity verification are applied and succeed;
- 5935 - confirm that the container is instantiated.
- 5936 3) For each test image that should fail integrity or authenticity (modified content, missing/invalid signature,  
5937 untrusted publisher, or equivalent):
- 5938 - request creation or start of a container from that image;
- 5939 - verify that the CRS detects the failure to verify integrity and/or authenticity according to the configured  
5940 trust policy;
- 5941 - confirm that the CRS prevents execution or instantiation of the container from that image.

#### 5942 **5. Assessment verdict**

- 5943 • Pass: integrity and authenticity verification is applied to container images before execution; only images from  
5944 trusted publishers/authorities that have not been modified after approval can be instantiated; and images whose  
5945 integrity or authenticity cannot be successfully verified under the trust policy are blocked.
- 5946 • Fail: authenticity is not enforced or can be bypassed; untrusted or modified images can be instantiated; or  
5947 failure to verify integrity/authenticity does not prevent container instantiation.

#### 5948 **6. Assessment evidence**

- 5949 • [DESIGN] Documentation of the container image trust model, including integrity and authenticity verification  
5950 and the trust policy.
- 5951 • [CONFIG] Trust configuration (trusted publishers/authorities, key material, policy settings).
- 5952 • [LOG] Records showing successful verification and container instantiation for approved images, and failed  
5953 verification with blocked instantiation for images that do not satisfy the trust policy.
- 5954 • [OBSERVATION] Test execution outputs demonstrating enforcement of the trust policy.

#### 5955 **6.4.1.5.3 Assessment Case AC-CRS-IMG-INT-003**

##### 5956 **1. Assessment reference**

5957 Requirement: REQ-CRS-IMG-INT-003 (Advanced)

##### 5958 **2. Assessment objective**

5959 Verify that:

- 5960 1) The CRS performs integrity and authenticity verification of container images using trust anchors stored as  
5961 protected trust material.
- 5962 2) Protected trust material is not modifiable during normal operational state and cannot be modified by container  
5963 workloads, unprivileged users, or routine runtime operations.
- 5964 3) Protected trust material is updated or replaced only through a controlled, authenticated administrative process.
- 5965 4) If verification of the integrity or authenticity of a container image using this protected trust material fails, or  
5966 the verification status cannot be established, the CRS prevents execution or instantiation of that container  
5967 image and records a security-relevant event.

##### 5968 **3. Assessment preparation**

5969 The assessment shall have access to:

- 5970 • Documentation describing:
  - 5971 - what constitutes protected trust material (keys, certificates, trust anchors);
  - 5972 - how and where this material is stored;
  - 5973 - what protections prevent modification during normal operation;
  - 5974 - the process and required privileges for controlled update or replacement of trust material.
- 5975 • A test environment with:
  - 5976 - a CRS deployment configured to use protected trust material for image verification;
  - 5977 - at least one container image that verifies successfully against this material;
  - 5978 - the ability to simulate conditions where an image fails verification against the protected trust material or
  - 5979 where verification status is indeterminate, in a controlled manner.

#### 5980 4. Assessment activities

5981 The assessment shall at least:

- 5982 1) Review documentation and configuration to confirm:
  - 5983 - which trust anchors are used for image verification;
  - 5984 - where they are stored and how access is restricted;
  - 5985 - that updates require a controlled, authenticated administrative process distinct from routine runtime
  - 5986 operations.
- 5987 2) In normal operational state, attempt to modify or replace trust material using:
  - 5988 - container workloads;
  - 5989 - unprivileged user contexts;
  - 5990 - standard administrative or runtime interfaces that are not part of the documented controlled update
  - 5991 process.
- 5992 3) Verify that such attempts fail or are blocked.
- 5993 4) Using a container image correctly approved under the current protected trust material, request container
- 5994 creation/start and verify that:
  - 5995 - integrity and authenticity verification succeed;
  - 5996 - the container is instantiated.
- 5997 5) Create a test condition such that verification against the protected trust material fails or becomes indeterminate
- 5998 for a container image (e.g. image modified after approval, image associated with no matching trust anchor)
- 5999 and then request container creation/start. Verify that:
  - 6000 - the CRS detects the verification failure or inability to establish verification status;
  - 6001 - the container image is not allowed to execute or be instantiated;
  - 6002 - a security-relevant event is recorded.

#### 6003 5. Assessment verdict

- 6004 • Pass: the CRS uses protected trust material for image integrity/authenticity verification; this material cannot be
- 6005 modified by workloads, unprivileged users or routine operations; only controlled, authenticated processes can
- 6006 update it; and images that fail or cannot be verified against this material are blocked and generate a security-
- 6007 relevant event.

- 6008 • Fail: trust anchors are stored in locations modifiable during normal operation; workloads or unprivileged users
- 6009 can alter them; or images that fail verification against the protected trust material can still be instantiated
- 6010 without a recorded security event.

## 6011 6. Assessment evidence

- 6012 • [DESIGN] Documentation of protected trust material, its storage, protection model and update procedures.
- 6013 • [CONFIG] CRS configuration showing trust stores or equivalent structures used for image verification and
- 6014 their access-control settings.
- 6015 • [LOG] Records showing successful verification and container instantiation for valid images, and failed
- 6016 verification with corresponding security-relevant events for images that cannot be verified.
- 6017 • [OBSERVATION] Test outputs demonstrating blocked modification attempts on trust material and blocked
- 6018 instantiation of images that fail verification.

### 6019 6.4.1.6 Assessment for Runtime Integrity Protection

#### 6020 6.4.1.6.1 Assessment Case AC-CRS-RP-INT-002

##### 6021 1. Assessment reference

6022 Requirement: REQ-CRS-RP-INT-002 (Elevated)

##### 6023 2. Assessment objective

6024 Verify that:

- 6025 1) The CRS enforces integrity protection for its configuration and control plane so that only authenticated and
- 6026 authorized modifications can be applied.
- 6027 2) The CRS generates audit events for any security-relevant changes to configuration or internal control logic.

##### 6028 3. Assessment preparation

6029 The assessment shall have access to:

- 6030 • Documentation describing the CRS configuration and control plane, including:
  - 6031 - which configuration items and control-plane elements are considered security-relevant;
  - 6032 - which roles or identities are permitted to modify them;
  - 6033 - which local or remote interfaces are used for such modifications.
- 6034 • Documentation of the CRS logging/auditing mechanisms for configuration and control-plane changes.
- 6035 • A CRS deployment with:
  - 6036 - at least one authenticated administrative account/role with privileges to change security-relevant
  - 6037 configuration or control-plane logic;
  - 6038 - at least one authenticated non-privileged or Unauthorized account/role;
  - 6039 - access to the configuration and control-plane interfaces under test;
  - 6040 - logging/audit facilities enabled and accessible for inspection.

##### 6041 4. Assessment activities

6042 The assessment shall at least:

- 6043 1) Review documentation and CRS configuration to identify:
  - 6044 - the set of security-relevant configuration parameters and control-plane elements;
  - 6045 - the authentication and authorization mechanisms governing their modification;
  - 6046 - how audit events are produced and stored when such elements are changed.

- 6047 2) Using an authenticated and authorized administrative account:
- 6048 - perform representative security-relevant configuration or control-plane changes (for example  
6049 enabling/disabling security features, changing access-control policies, or modifying isolation-related  
6050 settings);
- 6051 - verify that the CRS accepts these changes;
- 6052 - verify that corresponding audit events are generated, including at least the acting identity, the type of  
6053 change, and the time of the event.
- 6054 3) Using an unauthenticated context, a non-privileged account, or an interface that should not allow such  
6055 changes:
- 6056 - attempt to perform the same or equivalent security-relevant modifications;
- 6057 - verify that these attempts are rejected and that configuration/control-plane state is not altered;
- 6058 - verify, where specified by product design, that failed or denied modification attempts are logged.

#### 6059 5. Assessment verdict

- 6060 • Pass: only authenticated and authorized entities can perform security-relevant configuration or control-plane  
6061 changes; attempts by unauthenticated or Unauthorized entities are rejected without altering state; and audit  
6062 events are generated for security-relevant changes to configuration or internal control logic.
- 6063 • Fail: security-relevant changes can be made without proper authentication or authorization; such changes do  
6064 not generate audit events; or Unauthorized attempts are not prevented.

#### 6065 6. Assessment evidence

- 6066 • [DESIGN] Documentation of the CRS configuration and control-plane model, including the definition of  
6067 security-relevant changes and the access-control model.
- 6068 • [CONFIG] CRS configuration showing role/permission assignments for configuration and control-plane  
6069 interfaces and logging/audit settings.
- 6070 • [LOG] Audit or event logs capturing successful authorized changes and rejected Unauthorized attempts, with  
6071 timestamps and identifiers.
- 6072 • [OBSERVATION] Test execution records (for example CLI/API outputs, management UI screenshots)  
6073 demonstrating acceptance of authorized changes and rejection of Unauthorized ones.

#### 6074 6.4.1.6.2 Assessment Case AC-CRS-RP-INT-003

##### 6075 1. Assessment reference

6076 Requirement: REQ-CRS-RP-INT-003 (Advanced)

##### 6077 2. Assessment objective

6078 Verify that:

- 6079 1) The CRS maintains the integrity of its security-critical runtime components and detects Unauthorized  
6080 modification of configuration state or control logic used to enforce security policies or isolation.
- 6081 2) The CRS maintains a trusted baseline of the integrity state of these security-critical runtime components and  
6082 performs periodic or continuous integrity validation against this baseline.
- 6083 3) Upon detecting a deviation from the trusted baseline, the CRS automatically triggers protective actions (such  
6084 as isolating the affected component, disabling affected functions, or restoring the component from a trusted  
6085 state) and generates a security alert that cannot be suppressed by the affected component.

##### 6086 3. Assessment preparation

6087 The assessment shall have access to:

- 6088 • Documentation that:

- 6089 - identifies which runtime components are considered security-critical (for example processes or modules
- 6090 enforcing container isolation, access control, security policies, integrity checks, and configuration data
- 6091 that directly influences these functions);
- 6092 - describes how the trusted baseline for these components is established, stored, and updated;
- 6093 - explains whether integrity validation is periodic, continuous, or event-driven, and how it is configured;
- 6094 - describes the protective actions that can be triggered when deviations are detected;
- 6095 - describes the alerting mechanism and how alerts are exposed to administrators or monitoring systems.
- 6096 • A CRS deployment where runtime integrity validation is enabled and observable (for example via logs, status
- 6097 commands, or telemetry).
- 6098 • A controlled means, consistent with vendor guidance, to simulate or induce an integrity deviation affecting at
- 6099 least one security-critical runtime component or its configuration state (for example a supported test hook,
- 6100 debug mode, test image, or vendor-provided integrity-test procedure).

#### 6101 4. Assessment activities

6102 The assessment shall at least:

- 6103 1) Review documentation and configuration to confirm:
  - 6104 - which runtime components are treated as security-critical and covered by the trusted baseline;
  - 6105 - how the baseline is created, protected, and updated;
  - 6106 - the schedule or trigger conditions for integrity checks;
  - 6107 - the configured protective actions and alerting behaviour when deviations are detected.
- 6108 2) In a normal, non-tampered configuration:
  - 6109 - operate the CRS under representative conditions for a suitable period;
  - 6110 - verify from logs, status outputs, or telemetry that integrity validation is active for the defined security-
  - 6111 critical components (for example periodic checks or continuous monitoring events);
  - 6112 - confirm that no integrity deviation alerts are reported while the baseline remains intact.
- 6113 3) Introduce a controlled integrity deviation affecting a security-critical runtime component or its configuration
- 6114 state using the sanctioned test method:
  - 6115 - allow the CRS to perform its runtime validation;
  - 6116 - verify that the deviation from the trusted baseline is detected;
  - 6117 - verify that the documented protective actions (such as isolating or disabling the affected component or
  - 6118 restoring it from a trusted state) are automatically triggered;
  - 6119 - verify that a security alert is generated and remains visible via CRS or management/monitoring
  - 6120 interfaces, and that this alert cannot be cleared or suppressed solely by the affected component.

#### 6121 5. Assessment verdict

- 6122 • Pass: security-critical runtime components and associated configuration are covered by a trusted baseline; the
- 6123 CRS performs periodic or continuous validation against this baseline; controlled deviations are detected;
- 6124 documented protective actions are automatically applied; and a security alert is generated that cannot be
- 6125 suppressed by the affected component.
- 6126 • Fail: runtime integrity validation does not cover security-critical components; is not active or observable;
- 6127 deviations are not detected; protective actions are not triggered; or alerts can be suppressed or removed by the
- 6128 affected component.

#### 6129 6. Assessment evidence

- 6130 • [DESIGN] Documentation describing security-critical runtime components, trusted baseline creation and  
6131 protection, runtime integrity mechanisms, and protective and alerting behaviour.
- 6132 • [CONFIG] Configuration enabling runtime integrity validation, including monitored components, check  
6133 parameters, and alerting/protection settings.
- 6134 • [LOG] Integrity-check logs and security alerts from normal operation and from the controlled deviation  
6135 scenario, including timestamps and identifiers of affected components.
- 6136 • [OBSERVATION] Test or monitoring outputs showing runtime integrity checks in operation, detection of the  
6137 induced deviation, execution of protective actions, and persistence of the generated alert.
- 6138 • [PLT-CAP] Where hardware- or platform-assisted mechanisms are used as part of runtime integrity validation,  
6139 evidence of the underlying platform capabilities and their configuration.

#### 6140 6.4.1.7 Assessment for Remote Attestation

##### 6141 6.4.1.7.1 Assessment Case AC-CRS-RA-INT-003

###### 6142 1. Assessment reference

6143 Requirement: REQ-CRS-RA-INT-003 (Advanced)

###### 6144 2. Assessment objective

6145 Verify that:

- 6146 1) The CRS supports measured launch and generates attestation evidence describing its core runtime engine,  
6147 security-critical configuration, and other runtime state relevant to the integrity of container execution.
- 6148 2) The attestation evidence is cryptographically verifiable by an authorized verifier and is bound to the specific  
6149 CRS instance for which the measurements were produced.
- 6150 3) Attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay.
- 6151 4) The attestation evidence is limited to the minimum information necessary to verify the integrity and  
6152 configuration state of the CRS.
- 6153 5) Where platform or external trust services are used, the CRS uses them as trust anchors for attestation keys and  
6154 measurements in accordance with the product documentation.

###### 6155 3. Assessment preparation

6156 The assessment shall have access to:

- 6157 • Documentation describing:
  - 6158 - the measured-launch and attestation design, including which CRS components and configuration items  
6159 are measured (core runtime engine, security-critical configuration, relevant runtime state);
  - 6160 - how measurements are collected, stored, and associated with a CRS instance;
  - 6161 - the format and export mechanism of the attestation evidence (for example data fields, encoding, transport  
6162 or API);
  - 6163 - the cryptographic mechanisms used to protect attestation evidence and bind it to a CRS instance (for  
6164 example signature algorithms, keys, certificate chains);
  - 6165 - the protection model for attestation keys and measurement data, including which entities are permitted to  
6166 read, manage, or update them;
  - 6167 - any use of platform trust services or externally protected trust material as trust anchors for attestation  
6168 keys or measurements.
- 6169 • A test configuration with:
  - 6170 - a CRS deployment with measured launch and attestation enabled according to product documentation;

- 6171 - an interface to request attestation evidence (for example CLI, API, or management interface);
- 6172 - a verification tool or reference verifier configured with the relevant trust anchors to validate CRS
- 6173 attestation evidence;
- 6174 - at least one container workload or non-privileged host context that can be used to attempt Unauthorized
- 6175 access to attestation keys or raw measurement data;
- 6176 - where applicable, a platform or environment that provides the claimed trust services used as anchors for
- 6177 attestation.

#### 6178 4. Assessment activities

6179 The assessment shall at least:

- 6180 1) Review documentation and CRS settings to identify:
  - 6181 - the set of CRS elements covered by measurements (core runtime engine, security-critical configuration,
  - 6182 relevant runtime state);
  - 6183 - how these measurements are represented in the attestation evidence;
  - 6184 - which cryptographic algorithms and keys are used to protect and sign the evidence;
  - 6185 - how the evidence is bound to a specific CRS instance (for example instance identifiers, platform identity,
  - 6186 or key binding);
  - 6187 - how attestation keys and measurement data are stored and access-controlled;
  - 6188 - where applicable, which platform or external trust services are used as trust anchors.
- 6189 2) After a normal startup of the CRS:
  - 6190 - request attestation evidence using the documented interface;
  - 6191 - use the verification tool to confirm that the evidence verifies successfully under the configured trust
  - 6192 anchors and corresponds to the expected CRS instance;
  - 6193 - compare fields in the evidence with the deployed CRS software and configuration (for example runtime
  - 6194 engine version, key security-critical settings) to confirm consistency;
  - 6195 - review the content of the evidence to confirm that it is limited to the information required to verify
  - 6196 integrity and configuration state and does not unnecessarily expose unrelated sensitive operational data.
- 6197 3) From a container workload, non-privileged host account, or other context that should not have direct access:
  - 6198 - attempt to read or export attestation private keys or raw measurement data using available interfaces;
  - 6199 - verify that such attempts are rejected or blocked and that keys and measurements cannot be obtained
  - 6200 through these paths.
- 6201 4) Where anti-replay behaviour is documented:
  - 6202 - capture a valid attestation response;
  - 6203 - attempt to reuse or replay this response in a context that expects fresh attestation;
  - 6204 - verify, using the verifier or tool, that replayed evidence is detected or rejected in accordance with the
  - 6205 documented behaviour.

#### 6206 5. Assessment verdict

- 6207 • Pass: the CRS generates attestation evidence covering the core runtime engine, security-critical configuration,
- 6208 and relevant runtime state; the evidence is cryptographically verifiable and bound to the specific CRS instance;
- 6209 attestation keys and measurement data are protected against Unauthorized access, disclosure, or replay; the
- 6210 evidence content is limited to the minimum necessary to verify the integrity and configuration state; and,
- 6211 where platform or external trust services are claimed as anchors, their use is consistent with the documentation
- 6212 and visible in the attestation design.

- 6213 • Fail: attestation evidence is not generated as specified; cannot be cryptographically verified or is not clearly
- 6214 bound to a specific instance; Unauthorized entities can access or export attestation keys or measurement data
- 6215 or successfully replay stale evidence where freshness is required; the evidence content includes unnecessary
- 6216 sensitive information beyond what is required to verify integrity and configuration state; or claimed use of
- 6217 platform or external trust services as trust anchors is not realized in the implementation.

## 6218 6. Assessment evidence

- 6219 • [DESIGN] Documentation of the CRS measured-launch and attestation design, including measured
- 6220 components, evidence format, cryptographic protection, instance binding, protection model for keys and
- 6221 measurements, and any use of platform or external trust services as trust anchors.
- 6222 • [CONFIG] CRS and, where applicable, platform configuration enabling measured launch and attestation and
- 6223 defining trust anchors used by verifiers.
- 6224 • [LOG] Records of attestation requests and responses and any logged events related to attestation key access
- 6225 failures or replay handling.
- 6226 • [OBSERVATION] Captured attestation evidence samples and verifier outputs showing successful validation
- 6227 for normal runs and rejection or detection of replayed or tampered evidence.
- 6228 • [PLT-CAP] Evidence of platform capabilities or external trust services used to support CRS attestation, where
- 6229 the implementation relies on such services.

### 6230 6.4.1.7.2 Assessment Case AC-CRS-RA-INT-004

#### 6231 1. Assessment reference

6232 Requirement: REQ-CRS-RA-INT-004 (Advanced)

#### 6233 2. Assessment objective

6234 Verify that, where CRS attestation evidence is intended to be used in remote verification scenarios and this capability is

6235 claimed:

- 6236 1) The CRS supports configuration of privacy-preserving attestation options that reduce linkability and
- 6237 correlation between attestation events.
- 6238 2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and
- 6239 configuration state of the CRS.
- 6240 3) The CRS allows configuration of attested information, triggering conditions, and attestation intervals in order
- 6241 to support data minimization and to prevent excessive or Unauthorized monitoring.

#### 6242 3. Assessment preparation

6243 The assessment shall have access to:

- 6244 • Documentation describing:
  - 6245 - the CRS remote attestation model, including how CRS identity and instance information are represented
  - 6246 in attestation evidence;
  - 6247 - the privacy-preserving options available (for example limiting attested information, configuration of
  - 6248 attestation triggers and intervals, use of pseudonymous identifiers or separation of identity and integrity
  - 6249 information);
  - 6250 - how these options are configured;
  - 6251 - the expected impact of each option on attestation evidence fields, attestation frequency, and verifier
  - 6252 behaviour.
- 6253 • A test configuration with:
  - 6254 - a CRS deployment where remote attestation to one or more authorized verifiers is enabled;
  - 6255 - the ability to configure and modify the documented privacy-preserving options;

- 6256 - tools or procedures to trigger multiple attestation events (for example on-demand, periodic, or event-  
6257 driven) and capture the resulting attestation evidence under different privacy and scheduling  
6258 configurations;
- 6259 - access to an authorized verifier or verification tool capable of validating integrity and configuration state  
6260 from the attestation evidence in both baseline and privacy-enhanced modes.

#### 6261 4. Assessment activities

6262 The assessment shall at least:

- 6263 1) Review documentation and CRS settings to identify:
- 6264 - the default behaviour of remote attestation when privacy-preserving options are not enabled;
- 6265 - the available privacy-preserving options and their documented effects on identifiers, metadata, attested  
6266 content, and attestation frequency or trigger conditions;
- 6267 - any prerequisites or constraints for enabling and configuring these options.
- 6268 2) With privacy-preserving options disabled or in their default state:
- 6269 - trigger multiple attestation events over time according to the default model;
- 6270 - collect the corresponding attestation evidence;
- 6271 - Analyze the evidence to determine the stability and linkability of identifiers and other potentially  
6272 identifying fields across events, and the amount of configuration and runtime information included;
- 6273 - use the authorized verifier or tool to confirm that the integrity and configuration state of the CRS can be  
6274 correctly assessed from this baseline evidence set.
- 6275 3) Enable and configure the documented privacy-preserving options (for example limiting data fields, adjusting  
6276 attestation triggers or intervals, enabling pseudonymous identifiers or separation of identity and integrity  
6277 information) as described by the manufacturer, then:
- 6278 - trigger multiple attestation events under the privacy-enhanced configuration;
- 6279 - collect the resulting attestation evidence;
- 6280 - compare this evidence set with the baseline to verify that linkability and correlation between attestation  
6281 events are reduced in the documented manner and that attested content is limited in line with data-  
6282 minimization objectives;
- 6283 - verify, using the authorized verifier or tool, that the integrity and configuration state of the CRS can still  
6284 be reliably assessed using the privacy-enhanced evidence and any associated metadata or policies.

#### 6285 5. Assessment verdict

- 6286 • Pass: when remote attestation is used and privacy support is claimed, the CRS provides configurable privacy-  
6287 preserving attestation options that limit attested information, allow control of attestation triggers and intervals,  
6288 and, where applicable, support techniques such as pseudonymous identifiers or separation of identity and  
6289 integrity information; when enabled, these options reduce linkability and correlation between attestation events  
6290 as documented while still allowing authorized verifiers to correctly assess the integrity and configuration state  
6291 of the CRS.
- 6292 • Fail: privacy-preserving options are not available despite being claimed; configuration changes do not  
6293 meaningfully reduce linkability or correlation between attestation events or do not support data minimization  
6294 and control of attestation frequency; or enabling these options prevents authorized verifiers from correctly  
6295 assessing the integrity and configuration state of the CRS.

#### 6296 6. Assessment evidence

- 6297 • [DESIGN] Documentation of the CRS remote attestation model, identity representation, and privacy-  
6298 preserving options, including their expected effects on attestation evidence, frequency, and verifier behaviour.

- 6299 • [CONFIG] CRS and verifier configuration showing remote attestation settings, selected privacy-preserving  
6300 options, and configured triggers or intervals.
- 6301 • [LOG] Records of attestation events and related configuration changes, including timestamps and any  
6302 metadata relevant to privacy options and scheduling.
- 6303 • [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options,  
6304 with verifier outputs showing successful integrity and configuration assessment in both modes and reduced  
6305 linkability or correlation when privacy options are enabled.

## 6306 6.4.1.8 Assessment for Administrative Authentication

### 6307 6.4.1.8.1 Assessment Case AC-CRS-ADMIN-AUTH-001

#### 6308 1. Assessment reference

6309 Requirement: REQ-CRS-ADMIN-AUTH-001 (Basic)

#### 6310 2. Assessment objective

6311 Verify that:

- 6312 1) CRS administrative interfaces require authentication before any administrative operation.
- 6313 2) Administrative accounts use unique, non-default credentials and default credentials are disabled or changed  
6314 before use.
- 6315 3) Password complexity and lockout mechanisms for administrative accounts are configurable and enforced to  
6316 mitigate brute-force attempts.
- 6317 4) Where password-based authentication is used, administrative passwords are stored only as salted hashes.

#### 6318 3. Assessment preparation

6319 The assessment shall have access to:

- 6320 • Declared CRS product composition and list of administrative interfaces (CLI, APIs, dashboards/consols).
- 6321 • Documentation on administrative account model, default credentials, password policy and lockout options, and  
6322 password storage mechanism.
- 6323 • At least one CRS configuration with administrative accounts and access to identified administrative interfaces.

#### 6324 4. Assessment activities

6325 The assessment shall at least:

- 6326 1) Identify administrative interfaces from documentation/configuration and verify that each requires  
6327 authentication in the assessed configuration.
- 6328 2) Review installation/hardening guidance and verify in the assessed configuration that any default administrative  
6329 credentials are disabled or enforced to be changed before administrative access is granted.
- 6330 3) Confirm that password complexity and lockout policies for administrative accounts can be configured and  
6331 verify that non-compliant passwords are rejected and that lockout is applied according to the configured  
6332 thresholds.
- 6333 4) Review documentation and, where feasible, configuration/implementation details to confirm that CRS  
6334 administrative passwords are stored only as salted hashes (no cleartext).

#### 6335 5. Assessment verdict

- 6336 • Pass: CRS administrative interfaces require authentication; administrative accounts do not rely on unchanged  
6337 default credentials; password complexity and lockout are configurable and enforced; and administrative  
6338 passwords are stored only as salted hashes.
- 6339 • Fail: any administrative interface allows unauthenticated administrative operations; or unchanged default  
6340 credentials can be used; or password complexity/lockout cannot be configured or are not enforced; or  
6341 administrative passwords are stored in cleartext.

6342 **6. Assessment evidence**

- 6343 • [SCOPE] CRS product composition and list of administrative interfaces.
- 6344 • [CONFIG] Administrative authentication, password policy and lockout configuration.
- 6345 • [DESIGN] Description of administrative access model, default account handling and password storage  
6346 mechanism.
- 6347 • [LOG] / [OBSERVATION] Evidence of authentication enforcement, rejection of weak passwords and lockout  
6348 behaviour.

6349 **6.4.1.8.2 Assessment Case AC-CRS-ADMIN-AUTH-002**6350 **1. Assessment reference**

6351 Requirement: REQ-CRS-ADMIN-AUTH-002 (Elevated)

6352 **2. Assessment objective**

6353 Verify that:

- 6354 1) CRS administrative interfaces require authenticated access and support cryptographic key-based authentication  
6355 for administrative accounts.
- 6356 2) Password-based login for CRS administrative interfaces can be disabled.
- 6357 3) Private keys used for CRS administrative authentication are protected from extraction or reuse by container  
6358 workloads or other unprivileged components.
- 6359 4) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile  
6360 declared by the manufacturer, aligned with [2].

6361 **3. Assessment preparation**

6362 The assessment shall have access to:

- 6363 • Documentation on key-based administrative authentication for each CRS administrative interface.
- 6364 • Documentation on applicable cryptographic profiles for administrative authentication (algorithms, key lengths,  
6365 disallowed options).
- 6366 • Documentation of key management and key storage for administrative private keys, including separation from  
6367 container workloads and unprivileged components.
- 6368 • At least one CRS configuration where administrative accounts use key-based authentication on relevant  
6369 interfaces and password-based login can be disabled.

6370 **4. Assessment activities**

6371 The assessment shall at least:

- 6372 1) Confirm from documentation/configuration that administrative accounts can use key-based authentication on  
6373 relevant CRS interfaces and verify that an administrative account can obtain administrative access using the  
6374 key-based mechanism.
- 6375 2) Confirm that the CRS can disable password-based login for administrative interfaces and verify that, where  
6376 disabled, password-only attempts are rejected while key-based authentication still permits authorized access.
- 6377 3) Review configuration and, where applicable, protocol characteristics to confirm that algorithms, key lengths  
6378 and protocol versions used for key-based administrative authentication comply with the declared cryptographic  
6379 profile and that weaker parameters marked as non-compliant are not used.
- 6380 4) Review key storage design and configuration to confirm that container workloads and other unprivileged  
6381 components cannot read, export or misuse administrative private keys.

6382 **5. Assessment verdict**

- 6383 • Pass: CRS administrative interfaces require authenticated access and support key-based administrative  
6384 authentication; password-based login can be disabled and is effectively blocked where configured;  
6385 cryptographic parameters comply with the declared profile; and administrative private keys are protected from  
6386 container workloads and unprivileged components.
- 6387 • Fail: key-based administrative authentication is not available or not functional; or password-based login cannot  
6388 be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared  
6389 profile; or administrative private keys are accessible to container workloads or other unprivileged components.

## 6390 6. Assessment evidence

- 6391 • [CONFIG] CRS configuration for key-based administrative authentication, password-login disablement and  
6392 crypto settings.
- 6393 • [DESIGN] Documentation of key management and key storage architecture and applied cryptographic profile.
- 6394 • [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of  
6395 password-only attempts when disabled.
- 6396 • [PLT-CAP] Where relevant, evidence of platform capabilities used for key protection.

### 6397 6.4.1.8.3 Assessment Case AC-CRS-ADMIN-AUTH-003

#### 6398 1. Assessment reference

6399 Requirement: REQ-CRS-ADMIN-AUTH-003 (Advanced)

#### 6400 2. Assessment objective

6401 Verify that:

- 6402 1) Strong, cryptographically verifiable authentication is supported and enforceable for CRS administrative  
6403 accounts on administrative interfaces in scope.
- 6404 2) CRS administrative identities can be bound to a trusted, managed credential source (e.g. PKI or external  
6405 identity provider).
- 6406 3) The CRS integrates with external identity or credential management systems to support controlled issuance,  
6407 rotation and revocation of administrative credentials.
- 6408 4) The CRS checks validity and revocation (or equivalent invalidation) status of externally managed  
6409 administrative credentials before granting administrative access.

#### 6410 3. Assessment preparation

6411 The assessment shall have access to:

- 6412 • Documentation of strong administrative authentication mechanisms (e.g. certificate-based) for CRS  
6413 administrative interface.
- 6414 • Documentation of supported integrations with external identity/credential management systems (PKI, IdP,  
6415 directory), including mapping of external identities to CRS administrative roles/accounts.
- 6416 • Documentation describing how externally managed administrative credentials are issued, rotated and revoked,  
6417 and how the CRS consumes this information (validity, revocation checks).
- 6418 • At least one CRS configuration where administrative accounts are bound to a trusted external credential  
6419 source, with usable valid and invalid/expired/revoked credentials (or equivalent conditions) for testing.

#### 6420 4. Assessment activities

6421 The assessment shall at least:

- 6422 1) Confirm from documentation/configuration that strong, cryptographically verifiable credentials can be  
6423 enforced for administrative accounts on CRS administrative interfaces in scope and verify that administrative  
6424 access is granted only when such a valid credential is presented.
- 6425 2) Confirm that CRS administrative identities can be bound to a trusted external credential source (e.g. PKI-  
6426 issued certificates, IdP-managed identities) and that this binding is reflected in the CRS role/access model.

6427 3) Confirm that the CRS integrates with external identity/credential management systems for administrative  
6428 credentials and that it can handle credential issuance and rotation as defined by those systems.

6429 4) Verify in the assessed configuration that the CRS checks credential validity period and, where supported,  
6430 revocation or equivalent invalidation status, and denies administrative access when credentials are invalid,  
6431 expired, revoked or otherwise not trusted.

## 6432 5. Assessment verdict

6433 • Pass: strong, cryptographically verifiable administrative authentication is enforceable on CRS administrative  
6434 interfaces in scope; administrative identities are bound to a trusted external credential source; integration with  
6435 external identity/credential management supports issuance, rotation and revocation; and the CRS verifies  
6436 validity and revocation (or equivalent) of such credentials before granting administrative access.

6437 • Fail: strong administrative authentication cannot be enforced on administrative interfaces in scope; or  
6438 administrative identities cannot be effectively bound to a trusted external credential source; or integration with  
6439 external identity/credential management is insufficient for controlled issuance, rotation and revocation; or the  
6440 CRS does not properly verify validity and revocation (or equivalent) before granting administrative access.

## 6441 6. Assessment evidence

6442 • [CONFIG] CRS configuration for strong administrative authentication, bindings to external credential sources,  
6443 and identity/credential integration settings.

6444 • [DESIGN] Documentation of CRS administrative authentication architecture, including external  
6445 identity/credential integration and credential lifecycle handling.

6446 • [LOG] / [OBSERVATION] Evidence of successful administrative access with valid external credentials and  
6447 denial of access with invalid, expired or revoked credentials.

6448 • [PLT-CAP] Where relevant, evidence of platform or infrastructure capabilities used to support strong  
6449 administrative authentication and external identity integration.

## 6450 6.4.1.9 Assessment for Service Authentication

### 6451 6.4.1.9.1 Assessment Case AC-CRS-SERV-AUTH-001

#### 6452 1. Assessment reference

6453 Requirement: REQ-CRS-SERV-AUTH-001 (Basic)

#### 6454 2. Assessment objective

6455 Verify that:

6456 1) External services accessing CRS APIs or control interfaces (e.g. orchestration, monitoring, backup) are  
6457 authenticated.

6458 2) Each service account uses distinct, non-default credentials and no built-in shared or hard-coded service  
6459 credentials are relied upon.

6460 3) Configurable lockout and/or rate-limit mechanisms exist for service authentication attempts and are enforced.

6461 4) Service credentials are stored in a way that prevents exposure via configuration files, container images or logs.

#### 6462 3. Assessment preparation

6463 The assessment shall have access to:

6464 • Declared CRS product composition and list of external services/interfaces accessing CRS APIs or control  
6465 interfaces.

6466 • Documentation on the service account model, credential provisioning, any default/built-in service credentials,  
6467 and authentication lockout/rate-limiting options.

6468 • Documentation on how service credentials are stored (e.g. secrets store, environment variables, KMS) and how  
6469 configuration, images and logs are handled.

- 6470 • At least one configuration with representative external services integrated using service accounts.

#### 6471 4. Assessment activities

6472 The assessment shall at least:

- 6473 1) Identify external services accessing CRS APIs or control interfaces and verify that each authenticates  
6474 successfully before invoking management or control operations.
- 6475 2) Review configuration and documentation to confirm that each service account has distinct credentials and that  
6476 normal operation does not require shared or hard-coded service credentials; verify for at least one service that  
6477 its credentials are unique and changeable.
- 6478 3) Confirm that lockout and/or rate-limit thresholds for service authentication attempts can be configured; verify  
6479 that repeated failed authentication attempts from a service trigger lockout or rate-limiting as configured.
- 6480 4) Review the documented storage locations and mechanisms for service credentials and inspect representative  
6481 configuration files, container images and logs to confirm that credentials are not stored or exposed in cleartext  
6482 or other directly reusable form in these artefacts.

#### 6483 5. Assessment verdict

- 6484 • Pass: external services accessing CRS APIs or control interfaces require authentication; each service account  
6485 uses distinct, non-default credentials with no reliance on shared/hard-coded credentials; lockout or rate-limit  
6486 mechanisms for service authentication attempts are configurable and enforced; and service credentials are not  
6487 exposed via configuration files, container images or logs.

- 6488 • Fail: unauthenticated service access to CRS APIs/control interfaces is possible; or shared/default/hard-coded  
6489 service credentials are required or used; or lockout/rate-limit mechanisms are not available, not configurable,  
6490 or not enforced; or service credentials are exposed in configuration files, images or logs.

#### 6491 6. Assessment evidence

- 6492 • [SCOPE] List of external services and CRS APIs/control interfaces in scope.
- 6493 • [CONFIG] Service authentication, credential, lockout and rate-limiting configuration.
- 6494 • [DESIGN] Documentation of the service account and credential storage model, including handling of  
6495 configuration, images and logs.
- 6496 • [LOG] / [OBSERVATION] Evidence of enforced authentication, triggered lockout or rate-limiting, and  
6497 absence of exposed credentials in sampled configuration files, images and logs.

#### 6498 6.4.1.9.2 Assessment Case AC-CRS-SERV-AUTH-002

##### 6499 1. Assessment reference

6500 Requirement: REQ-CRS-SERV-AUTH-002 (Elevated)

##### 6501 2. Assessment objective

6502 Verify that:

- 6503 1) The CRS supports mutual cryptographic authentication for service-to-CRS interactions that access  
6504 management functions or data, allowing both parties to verify each other's identity.
- 6505 2) Cryptographic materials used for service authentication (keys, certificates, tokens) are protected from  
6506 Unauthorized extraction or reuse by container workloads or other unprivileged components.
- 6507 3) Algorithms and key lengths used for service authentication conform to the cryptographic profile declared by  
6508 the manufacturer, aligned with [2].

##### 6509 3. Assessment preparation

6510 The assessment shall have access to:

- 6511 • Documentation on mutual cryptographic authentication mechanisms for service-to-CRS interactions (e.g.  
6512 mTLS, signed tokens).

- 6513 • Documentation on applicable cryptographic profiles for service authentication (approved algorithms,  
6514 minimum key lengths, disallowed options).
- 6515 • Documentation on key/token management and storage (e.g. keystores, HSM/KMS, secret stores) and how  
6516 separation from container workloads/unprivileged components is enforced.
- 6517 • At least one configuration where representative services access CRS management functions or data using  
6518 mutual cryptographic authentication.

#### 6519 4. Assessment activities

6520 The assessment shall at least:

- 6521 1) Confirm from documentation/configuration that mutual cryptographic authentication is supported for service-  
6522 to-CRS interactions that access management functions or data; verify in the assessed configuration that a  
6523 service using mutual authentication can establish an authenticated session and access management functions or  
6524 data only after successful mutual verification.
- 6525 2) Review configuration/protocol traces to confirm that algorithms, key lengths and protocol versions used for  
6526 mutual authentication comply with the declared cryptographic profile and that non-compliant algorithms or  
6527 key sizes are not negotiated.
- 6528 3) Review the design and configuration of storage for service authentication keys/certificates/tokens and confirm  
6529 that container workloads and other unprivileged components cannot read, export or misuse these materials  
6530 according to documented isolation and access-control mechanisms.

#### 6531 5. Assessment verdict

- 6532 • Pass: mutual cryptographic authentication is available and functional for service-to-CRS interactions that  
6533 access management functions or data; cryptographic parameters used conform to the declared profile; and  
6534 keys/certificates/tokens used for service authentication are protected from extraction or reuse by container  
6535 workloads and other unprivileged components.
- 6536 • Fail: mutual cryptographic authentication is not available or not functional where required; or cryptographic  
6537 parameters are weaker than the declared profile; or cryptographic materials for service authentication can be  
6538 accessed or reused by container workloads or other unprivileged components.

#### 6539 6. Assessment evidence

- 6540 • [CONFIG] Mutual authentication configuration for services (certificates/keys/tokens and crypto settings).
- 6541 • [DESIGN] Documentation of service authentication architecture, key/token management and storage, and  
6542 applied cryptographic profile.
- 6543 • [LOG] / [OBSERVATION] Evidence of successful mutual authentication and rejection of connections using  
6544 non-compliant parameters or missing authentication.
- 6545 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication materials.

#### 6546 6.4.1.9.3 Assessment Case AC-CRS-SERV-AUTH-003

##### 6547 1. Assessment reference

6548 Requirement: REQ-CRS-SERV-AUTH-003 (Advanced)

##### 6549 2. Assessment objective

6550 Verify that:

- 6551 1) The CRS enforces a verifiable binding between service identities and cryptographic credentials for service  
6552 interactions that access management functions or data.
- 6553 2) Service credentials are issued, rotated and revoked through trusted credential or identity management systems.
- 6554 3) The CRS automatically rejects revoked or expired service credentials during authentication.

- 6555 4) The CRS supports integration with external identity or credential management systems to provide traceability  
 6556 of service identities across deployments and to ensure that only services with valid, managed identities can  
 6557 invoke CRS management or control functions.

### 6558 3. Assessment preparation

6559 The assessment shall have access to:

- 6560 • Documentation describing how service identities are represented (e.g. certificate subject/SAN, token claims,  
 6561 identity IDs) and how they are bound to cryptographic credentials.
- 6562 • Documentation of trusted credential/identity management systems used (PKI, IdP, secret/credential manager)  
 6563 and the processes for issuing, rotating and revoking service credentials.
- 6564 • Documentation on how the CRS validates credential status (expiry, revocation, or equivalent) during  
 6565 authentication.
- 6566 • Documentation on integration with external identity/credential management for traceability (e.g. logs,  
 6567 mappings, cross-deployment identifiers).
- 6568 • At least one configuration where service credentials are managed by an external trusted system, and valid,  
 6569 expired and revoked credentials (or equivalent conditions) can be exercised.

### 6570 4. Assessment activities

6571 The assessment shall at least:

- 6572 1) Confirm that the CRS binds service identities to cryptographic credentials (e.g. through certificate identity or  
 6573 token claims) and verify that, in the assessed configuration, access to management or control functions  
 6574 depends on presenting a credential that matches the bound identity.
- 6575 2) Verify that service credentials used with the CRS are issued, rotated and revoked by a trusted  
 6576 credential/identity management system; confirm, for at least one service, that credential updates (rotation) and  
 6577 revocation performed in the external system are reflected in CRS behaviour.
- 6578 3) Verify that the CRS checks credential expiry and revocation (or equivalent status) during authentication and  
 6579 automatically rejects expired or revoked service credentials before granting access.
- 6580 4) Verify that integration with the external identity/credential management system allows tracing service  
 6581 identities across deployments (e.g. consistent identifiers or mappings) and that attempts to invoke CRS  
 6582 management or control functions from services without valid, managed identities are denied.

### 6583 5. Assessment verdict

- 6584 • Pass: the CRS enforces verifiable binding of service identities to cryptographic credentials for service  
 6585 interactions accessing management functions or data; service credentials are managed (issued, rotated,  
 6586 revoked) by trusted systems; revoked or expired credentials are automatically rejected; and integration with  
 6587 external identity/credential management provides traceability and ensures that only services with valid,  
 6588 managed identities can invoke CRS management or control functions.
- 6589 • Fail: service identity is not reliably bound to cryptographic credentials; or service credentials are not managed  
 6590 by trusted external systems; or revoked/expired credentials are not automatically rejected; or integration with  
 6591 external identity/credential management does not provide effective traceability or allow the CRS to block  
 6592 services without valid, managed identities.

### 6593 6. Assessment evidence

- 6594 • [CONFIG] CRS configuration for service identity binding, use of externally managed credentials and  
 6595 authentication status checks.
- 6596 • [DESIGN] Documentation of the service identity and credential-lifecycle model, including integration with  
 6597 external identity/credential management and cross-deployment identity handling.
- 6598 • [LOG] / [OBSERVATION] Evidence of successful authentication with valid managed credentials, rejection of  
 6599 expired/revoked credentials, and denial of access for services without valid managed identities.

- 6600 • [PLT-CAP] Where relevant, evidence of platform capabilities used to support identity-bound credentials and
- 6601 status checking (e.g. PKI libraries, identity client libraries).

#### 6602 6.4.1.10 Assessment for Administrative Authorization

##### 6603 6.4.1.10.1 Assessment Case AC-CRS-ADMIN-AUTHZ-001

###### 6604 1. Assessment reference

6605 Requirement: REQ-CRS-ADMIN-AUTHZ-001 (Basic)

###### 6606 2. Assessment objective

6607 Verify that:

- 6608 1) All administrative actions are restricted to authenticated administrative accounts.
- 6609 2) A default-deny policy is enforced for access to CRS management functions, permitting only explicitly
- 6610 authorized actions.

###### 6611 3. Assessment preparation

6612 The assessment shall have access to:

- 6613 • Documentation of CRS administrative accounts, roles and management interfaces.
- 6614 • Documentation of the administrative authorization model, including default behaviour.
- 6615 • At least one configuration with multiple administrative accounts and access to CRS management functions.

###### 6616 4. Assessment activities

6617 The assessment shall at least:

- 6618 1) Verify that unauthenticated users and non-administrative accounts cannot perform CRS administrative actions
- 6619 on any management interface.
- 6620 2) Confirm from configuration and tests that administrative actions not explicitly granted to an authenticated
- 6621 administrative account are denied by default.

###### 6622 5. Assessment verdict

- 6623 • Pass: only authenticated administrative accounts can perform CRS administrative actions and any non-
- 6624 authorized administrative action is denied by default.
- 6625 • Fail: unauthenticated/non-administrative users can perform administrative actions; or actions not explicitly
- 6626 authorized are allowed.

###### 6627 6. Assessment evidence

- 6628 • [CONFIG] Administrative authorization and default policy configuration.
- 6629 • [DESIGN] Description of CRS administrative authorization model and default-deny behaviour.
- 6630 • [LOG] / [OBSERVATION] Evidence of denied operations for unauthenticated/non-admin users and permitted
- 6631 operations only when explicitly authorized.

##### 6632 6.4.1.10.2 Assessment Case AC-CRS-ADMIN-AUTHZ-002

###### 6633 1. Assessment reference

6634 Requirement: REQ-CRS-ADMIN-AUTHZ-002 (Elevated)

###### 6635 2. Assessment objective

6636 Verify that:

- 6637 1) Granular RBAC or ABAC is enforced for CRS administrative accounts.

- 6638 2) Each administrative account is granted only the minimum privileges required for its operational role (least  
6639 privilege).

### 6640 3. Assessment preparation

6641 The assessment shall have access to:

- 6642 • Documentation describing RBAC/ABAC for CRS administrative accounts (roles, attributes, permissions).
- 6643 • At least one configuration with several administrative roles/attributes showing clearly differentiated privilege  
6644 levels.

### 6645 4. Assessment activities

6646 The assessment shall at least:

- 6647 1) Confirm that administrative permissions are defined and enforced via RBAC or ABAC (e.g. role-/attribute-  
6648 based rules), not via a single full-privilege administrator profile.
- 6649 2) For selected administrative accounts with different roles/attributes, verify that each account can perform only  
6650 those actions associated with its defined role/attributes and that higher-privilege actions are denied,  
6651 demonstrating least-privilege enforcement.

### 6652 5. Assessment verdict

- 6653 • Pass: RBAC/ABAC is in place and selected administrative accounts are restricted to the minimum privileges  
6654 defined for their roles/attributes.
- 6655 • Fail: RBAC/ABAC is not effectively enforced; or administrative accounts have broader privileges than  
6656 defined; or least-privilege cannot be demonstrated.

### 6657 6. Assessment evidence

- 6658 • [CONFIG] RBAC/ABAC configuration for administrative accounts (roles/attributes and permissions).
- 6659 • [DESIGN] Documentation of the CRS administrative authorization model and least-privilege approach.
- 6660 • [LOG] / [OBSERVATION] Evidence of allowed actions within role scope and denied actions outside that  
6661 scope.

## 6662 6.4.1.10.3 Assessment Case AC-CRS-ADMIN-AUTHZ-003

### 6663 1. Assessment reference

6664 Requirement: REQ-CRS-ADMIN-AUTHZ-003 (Advanced)

### 6665 2. Assessment objective

6666 Verify that:

- 6667 1) Fine-grained authorization policies are enforced for CRS administrative accounts.
- 6668 2) Separation of duties (SoD) for administrative functions is supported and can be enforced.
- 6669 3) Time-bound or just-in-time (JIT) privilege elevation is supported and correctly revoked when no longer  
6670 needed.
- 6671 4) Integration with external identity and policy decision systems is supported to enable centralized definition and  
6672 enforcement of administrative authorization policies.

### 6673 3. Assessment preparation

6674 The assessment shall have access to:

- 6675 • Documentation of policy-based administrative authorization (policy language/rules, SoD and JIT models).
- 6676 • Documentation of integration with external identity and policy decision systems (if supported).
- 6677 • At least one configuration with:
  - 6678 - distinct administrative roles/functions demonstrating SoD; and

6679 - an example of time-bound or JIT privilege elevation.

#### 6680 4. Assessment activities

6681 The assessment shall at least:

- 6682 1) Verify that administrative authorization rules are expressed as fine-grained policies and that different  
6683 administrative functions can be assigned to different accounts to support SoD (e.g. one account configuring  
6684 changes and another approving or deploying them).
- 6685 2) Verify that time-bound or JIT elevation can be configured for an administrative account, that elevation is  
6686 limited in scope and duration, and that privileges are automatically revoked after expiry or completion.
- 6687 3) Where integration with external identity and policy decision systems is supported, verify that the CRS can use  
6688 those systems to obtain administrative identity/attributes and authorization decisions, and that changes to  
6689 central policies are reflected in CRS administrative authorization behaviour.

#### 6690 5. Assessment verdict

- 6691 • Pass: administrative authorization is fine-grained and policy-driven; SoD and time-bound/JIT elevation are  
6692 supported and function as documented; and, where applicable, external identity and policy decision integration  
6693 is effective.
- 6694 • Fail: fine-grained policies cannot be used to control administrative authorization; SoD cannot be enforced;  
6695 JIT/time-bound elevation is not available or not effectively revoked; or supported external integration cannot  
6696 be used for centralized administrative authorization.

#### 6697 6. Assessment evidence

- 6698 • [CONFIG] Administrative policy configuration showing fine-grained rules, SoD and time-bound/JIT  
6699 elevation, including any external PDP integration settings.
- 6700 • [DESIGN] Documentation of policy-driven administrative authorization, SoD model and external  
6701 identity/policy decision integration.
- 6702 • [LOG] / [OBSERVATION] Evidence of SoD enforcement, temporary elevation and revocation, and, where  
6703 applicable, authorization decisions influenced by external systems.

### 6704 6.4.1.11 Assessment for Service Authorization

#### 6705 6.4.1.11.1 Assessment Case AC-CRS-SERV-AUTHZ-001

##### 6706 1. Assessment reference

6707 Requirement: REQ-CRS-SERV-AUTHZ-001 (Basic)

##### 6708 2. Assessment objective

6709 Verify that:

- 6710 1) A default-deny policy is enforced for all CRS service accounts.
- 6711 2) Service accounts are only permitted to perform actions explicitly authorized by CRS configuration or policy.

##### 6712 3. Assessment preparation

6713 The assessment shall have access to:

- 6714 • List of CRS service accounts and service-facing interfaces.
- 6715 • Documentation of the authorization model and default behaviour for service accounts.
- 6716 • At least one configuration with several service accounts, including one with no explicit permissions.

##### 6717 4. Assessment activities

6718 The assessment shall at least:

- 6719 1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny any  
6720 action not explicitly authorized.

- 6721 2) Verify that a service account with no explicit permissions cannot perform management or data-related  
6722 operations.
- 6723 3) For a service account with defined permissions, verify that actions within its configured permissions are  
6724 allowed and that non-authorized actions are denied.
- 6725 **5. Assessment verdict**
- 6726 • Pass: default behaviour for service accounts is deny; unprivileged service accounts cannot perform operations;  
6727 and only explicitly authorized actions are permitted.
- 6728 • Fail: unspecified actions are allowed for service accounts; or a service account without explicit permissions  
6729 can perform operations.
- 6730 **6. Assessment evidence**
- 6731 • [SCOPE] List of CRS service accounts and interfaces in scope.
- 6732 • [CONFIG] Service-account authorization and default policy configuration.
- 6733 • [DESIGN] Description of the CRS service-account authorization model and default-deny behaviour.
- 6734 • [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and successful  
6735 execution of explicitly authorized actions.
- 6736 **6.4.1.11.2 Assessment Case AC-CRS-SERV-AUTHZ-002**
- 6737 **1. Assessment reference**  
6738 Requirement: REQ-CRS-SERV-AUTHZ-002 (Elevated)
- 6739 **2. Assessment objective**  
6740 Verify that:
- 6741 1) Granular authorization is enforced for CRS service accounts, with permitted actions bound to the authenticated  
6742 service identity.
- 6743 2) Least-privilege permissions are enforced for each service account.
- 6744 **3. Assessment preparation**  
6745 The assessment shall have access to:
- 6746 • Documentation on how service identities are represented and bound to permissions.
- 6747 • Documentation on the permission or role model for CRS service accounts.
- 6748 • At least one configuration with several service accounts having different, clearly scoped permissions.
- 6749 **4. Assessment activities**  
6750 The assessment shall at least:
- 6751 1) Confirm that authorization decisions for service accounts are based on the authenticated service identity, not  
6752 solely on generic network or system properties.
- 6753 2) For at least two service accounts with different permissions, verify that each account can perform only the  
6754 actions explicitly bound to its identity and that actions belonging to another service identity are denied.
- 6755 3) Verify that a service account configured with a minimal permission set cannot perform operations outside that  
6756 defined set, demonstrating least privilege.
- 6757 **5. Assessment verdict**
- 6758 • Pass: authorization for service accounts is identity-bound and granular, and least-privilege permission sets can  
6759 be configured and are enforced for individual service accounts.
- 6760 • Fail: actions are not clearly bound to service identity; or service accounts can perform operations beyond their  
6761 assigned permissions; or least-privilege cannot be demonstrated.

6762 **6. Assessment evidence**

- 6763 • [CONFIG] Service-account authorization configuration showing bindings between specific identities and  
6764 allowed actions, including at least one minimal-permission account.
- 6765 • [DESIGN] Documentation of the CRS service-account authorization model, identity binding and least-  
6766 privilege approach.
- 6767 • [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and  
6768 denied operations outside that scope.

6769 **6.4.1.11.3 Assessment Case AC-CRS-SERV-AUTHZ-003**6770 **1. Assessment reference**

6771 Requirement: REQ-CRS-SERV-AUTHZ-003 (Advanced)

6772 **2. Assessment objective**

6773 Verify that:

- 6774 1) Fine-grained, policy-driven authorization is enforced for CRS service accounts.
- 6775 2) Permissions for service accounts are bound to cryptographic service identities.
- 6776 3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-  
6777 account authorization decisions.
- 6778 4) Integration with external authorization systems is supported, where available, to enable centralized  
6779 management and evaluation of service-account policies.

6780 **3. Assessment preparation**

6781 The assessment shall have access to:

- 6782 • Documentation of policy-driven authorization for CRS service accounts (policy language/rules, conditions,  
6783 effects).
- 6784 • Documentation on how cryptographic service identities (e.g. certificate identities, token claims) are bound to  
6785 permissions.
- 6786 • Documentation on context attributes available to policy evaluation (e.g. environment, namespace, workload  
6787 labels).
- 6788 • Documentation on integration with external authorization systems (PDPs/policy engines), if supported.
- 6789 • At least one configuration where:
- 6790 - service-account policies reference cryptographic identities and at least one context attribute; and
- 6791 - where supported, an external authorization system is integrated and used for policy evaluation.

6792 **4. Assessment activities**

6793 The assessment shall at least:

- 6794 1) Verify that CRS service-account authorization can be expressed and enforced via fine-grained policies and that  
6795 permissions for at least one service account are defined in such policies.
- 6796 2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed  
6797 configuration, authorization decisions for at least one service depend on its cryptographic identity.
- 6798 3) Verify that at least one context attribute (e.g. namespace, cluster, label) influences authorization decisions for  
6799 service accounts and that access is allowed or denied based on that context in the assessed configuration.
- 6800 4) Where integration with an external authorization system is supported, verify that the CRS consults or delegates  
6801 service-account authorization decisions to that system and that policy changes in the external system are  
6802 reflected in CRS authorization outcomes.

6803 **5. Assessment verdict**

- 6804 • Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic  
6805 service identities; context-aware enforcement is supported; and, where applicable, external authorization  
6806 integration allows centralized policy management and evaluation.
- 6807 • Fail: policies cannot express fine-grained service-account authorization; or permissions are not effectively  
6808 bound to cryptographic identities; or context attributes cannot influence authorization decisions; or supported  
6809 external authorization integration is not effective.

6810 **6. Assessment evidence**

- 6811 • [CONFIG] Policy configuration for service-account authorization referencing cryptographic identities and  
6812 context attributes, and, where applicable, external authorization integration settings.
- 6813 • [DESIGN] Documentation of the policy-driven CRS service authorization architecture, including  
6814 cryptographic identity binding, context handling and external authorization interfaces.
- 6815 • [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and  
6816 context, and of changes resulting from external authorization policy updates where supported.

6817 **6.4.1.12 Assessment for Confidentiality Protection**6818 **6.4.1.12.1 Assessment Case AC-CRS-CONF-001**6819 **1. Assessment reference**

6820 Requirement: REQ-CRS-CONF-001 (Basic)

6821 **2. Assessment objective**

6822 Verify that:

- 6823 1) The CRS provides access-control mechanisms to restrict access to sensitive data under its control, including  
6824 container images, configuration data, stored secrets, and security-relevant logs.
- 6825 2) Access to such sensitive data is permitted only to authenticated and authorized entities through CRS-controlled  
6826 interfaces and APIs, and not directly accessible to unauthenticated or Unauthorized contexts (including  
6827 container workloads).

6828 **3. Assessment preparation**

6829 The assessment shall have access to:

- 6830 • Documentation describing:
- 6831 - the types of sensitive data handled by the CRS (for example container images under CRS control,  
6832 configuration objects, secrets, security-relevant logs);
- 6833 - the storage locations and repositories where such data is kept;
- 6834 - the CRS interfaces and APIs through which this data can be accessed or managed;
- 6835 - the access-control model, including roles or identities authorized to access each type of sensitive data.
- 6836 • A test configuration with:
- 6837 - at least one administrative account/role authorized to access and manage sensitive data;
- 6838 - at least one non-privileged or operational account/role with limited permissions;
- 6839 - one or more container workloads under CRS control;
- 6840 - CRS management interfaces and APIs reachable for testing.

**6841 4. Assessment activities**

6842 The assessment shall at least:

- 6843 1) Review documentation and CRS configuration to identify:
- 6844 - which CRS repositories or objects correspond to container images, configuration data, stored secrets, and  
6845 security-relevant logs;
  - 6846 - which CRS-controlled interfaces and APIs are intended for accessing or managing each type of sensitive  
6847 data;
  - 6848 - which roles/accounts are authorized to use these interfaces.
- 6849 2) Using an authenticated administrative account:
- 6850 - access representative examples of each sensitive data type via the documented CRS interfaces/APIs (for  
6851 example listing and retrieving container images, viewing configuration objects, reading secrets via a CRS  
6852 secret-management API, accessing security logs);
  - 6853 - verify that such access is successful only via the CRS-controlled interfaces and behaves as documented.
- 6854 3) Using a non-privileged account, unauthenticated context, or from within a container workload:
- 6855 - attempt to access or modify the same sensitive data via CRS APIs without appropriate permissions;
  - 6856 - where underlying storage paths are visible, attempt direct access to data repositories outside of CRS-  
6857 controlled interfaces;
  - 6858 - verify that these attempts are denied or blocked and do not result in disclosure or modification of  
6859 sensitive data.

**6860 5. Assessment verdict**

- 6861 • Pass: access to container images, configuration data, stored secrets, and security-relevant logs is restricted to  
6862 authenticated and authorized entities via CRS-controlled interfaces; attempts by unauthenticated,  
6863 Unauthorized, or workload-level contexts to access or modify such data are prevented.
- 6864 • Fail: sensitive data can be read or modified without using CRS-controlled interfaces, or by unauthenticated or  
6865 Unauthorized contexts; or the implemented access-control behaviour differs from the documented model.

**6866 6. Assessment evidence**

- 6867 • [DESIGN] Documentation of sensitive data types, storage locations, CRS interfaces/APIs, and the access-  
6868 control model.
- 6869 • [CONFIG] CRS configuration showing role/permission assignments and access-control settings for sensitive  
6870 data.
- 6871 • [LOG] Audit or event logs capturing successful authorized access and rejected Unauthorized attempts, where  
6872 such logging is implemented.
- 6873 • [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating  
6874 authorized access succeeding and Unauthorized access failing.
- 6875 • [SCOPE] Evidence identifying which CRS components and data stores were considered in-scope for sensitive  
6876 data.

## 6877 6.4.1.12.2 Assessment Case AC-CRS-CONF-002

6878 **1. Assessment reference**

6879 Requirement: REQ-CRS-CONF-002 (Elevated)

6880 **2. Assessment objective**

6881 Verify that:

6882 1) The CRS supports encryption of sensitive data at rest, including container images, configuration data, stored  
6883 secrets, and security-relevant logs, either via native capabilities or by integrating with encrypted storage,  
6884 registries, or secret-management services.

6885 2) When configured, encryption is effectively applied to the relevant sensitive data repositories.

6886 3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by  
6887 container workloads and other unprivileged components.6888 **3. Assessment preparation**

6889 The assessment shall have access to:

6890 • Documentation describing:

6891 - the CRS model for encrypting sensitive data at rest (native encryption, use of encrypted  
6892 storage/registries, integration with secret-management services);

6893 - which data types and repositories are covered;

6894 - supported key-management approaches (for example internal key store, external KMS, platform  
6895 services);

6896 - how encryption and key management are configured and monitored;

6897 - restrictions that prevent container workloads and other unprivileged components from accessing keys.

6898 • A test configuration with:

6899 - one or more CRS data stores (for example image registry, configuration store, secret store, log store)  
6900 configured to use the documented data-at-rest protection mechanisms;

6901 - an administrative account authorized to configure and monitor encryption and key management;

6902 - one or more container workloads and a non-privileged host context for attempting access to encrypted  
6903 data and keys;6904 - where applicable, access to underlying storage or diagnostic tools to observe whether data is stored  
6905 encrypted.6906 **4. Assessment activities**

6907 The assessment shall at least:

6908 1) Review documentation and configuration to identify:

6909 - which sensitive data categories and storage locations are expected to be protected at rest;

6910 - how encryption is enabled and verified for each type (for example flags in CRS configuration, storage-  
6911 layer encryption indicators);

6912 - where encryption keys are stored and how access to them is restricted.

6913 2) With data-at-rest protection enabled according to product guidance:

6914 - create or update representative container images, configuration objects, stored secrets, and security logs;

6915 - use platform or storage diagnostics (for example tools showing block- or file-level contents, provider  
6916 encryption status APIs) to verify that these items are stored in encrypted form and not in recoverable  
6917 plaintext;

- 6918 - verify that disabling or revoking access to encryption keys prevents normal decryption and access to the  
6919 protected data as expected.
- 6920 3) From container workloads and other non-privileged contexts:
- 6921 - attempt to access encryption keys or key stores via CRS APIs, environment variables, mounted volumes,  
6922 or host-level resources;
- 6923 - verify that such attempts are denied and that key material is not readable or exportable by these contexts.
- 6924 4) Where external or platform key-management services are used:
- 6925 - confirm via their interfaces that keys used for CRS data-at-rest protection reside in those services;
- 6926 - verify that the CRS only obtains derived or wrapped keys as documented, and does not expose raw key  
6927 material to workloads.

## 6928 5. Assessment verdict

- 6929 • Pass: the CRS supports and correctly applies data-at-rest protection to the configured categories of sensitive  
6930 data; protected data is not stored in plaintext on underlying storage; and encryption keys are stored and  
6931 managed in a way that prevents access by container workloads and other unprivileged components.
- 6932 • Fail: sensitive data configured to be protected at rest is stored unencrypted or in recoverable plaintext; keys are  
6933 accessible to workloads or non-privileged contexts; or behaviour diverges from the documented data-at-rest  
6934 and key-management model.

## 6935 6. Assessment evidence

- 6936 • [DESIGN] Documentation of the CRS data-at-rest encryption model, including covered data types and key-  
6937 management design.
- 6938 • [CONFIG] CRS configuration showing enabled encryption settings for image, configuration, secret, and log  
6939 stores, and key-management configuration.
- 6940 • [LOG] Events or audit trails related to enabling/disabling encryption, key rotation, and failed access attempts  
6941 to protected data or keys.
- 6942 • [OBSERVATION] Test records and storage or provider-tool outputs demonstrating that data is stored  
6943 encrypted and is inaccessible when keys are revoked.
- 6944 • [PLT-CAP] Evidence of platform, storage, registry, or secret-management capabilities used to provide data-at-  
6945 rest encryption.

### 6946 6.4.1.12.3 Assessment Case AC-CRS-CONF-003

#### 6947 1. Assessment reference

6948 Requirement: REQ-CRS-CONF-003 (Elevated)

#### 6949 2. Assessment objective

6950 Verify that:

- 6951 1) The CRS supports the use of encrypted communication for CRS management and control interfaces, for  
6952 communication with image registries, and for other CRS APIs that handle sensitive data or control operations.
- 6953 2) When these communications are configured to use encrypted transport, network traffic is protected with  
6954 appropriate cryptographic mechanisms that ensure confidentiality, integrity, and peer authentication.
- 6955 3) The CRS prevents fallback or downgrade from configured encrypted transport to unencrypted or weakly  
6956 protected channels.

### 6957 3. Assessment preparation

6958 The assessment shall have access to:

- 6959 • Documentation describing:
  - 6960 - CRS management and control interfaces and APIs that handle sensitive data or control operations;
  - 6961 - communication paths between the CRS and image registries, secret stores, or other services handling  
6962 sensitive data;
  - 6963 - supported encrypted transport mechanisms (for example TLS configurations, integration with service  
6964 meshes or platform-provided secure channels);
  - 6965 - how anti-fallback or anti-downgrade behaviour is implemented and configured.
- 6966 • A test configuration with:
  - 6967 - CRS management and control interfaces reachable over the network;
  - 6968 - a configured image registry and any other CRS APIs that handle sensitive data;
  - 6969 - valid certificates or other credentials for establishing encrypted channels as documented;
  - 6970 - tools to capture and inspect network traffic on relevant links.

### 6971 4. Assessment activities

6972 The assessment shall at least:

- 6973 1) Review documentation and CRS configuration to identify:
    - 6974 - which CRS interfaces and API endpoints are in scope for encrypted communication;
    - 6975 - the recommended or required secure transport protocols and cipher configurations;
    - 6976 - any configuration settings that enforce or relax requirements for encryption.
  - 6977 2) Configure encrypted transport for:
    - 6978 - at least one CRS management or control interface;
    - 6979 - communication between the CRS and a configured image registry;
    - 6980 - - at least one additional CRS API that handles sensitive data or control operations.
- 6981 For each case, initiate typical management, registry, or API operations while capturing network traffic, and  
6982 verify that:
- 6983 - payload data is encrypted and not visible in clear text on the wire;
  - 6984 - negotiated cryptographic parameters conform to the configuration and are not weaker than documented.
- 6985 3) Attempt to force downgrade or fallback by:
    - 6986 - misconfiguring clients to request unencrypted protocols;
    - 6987 - proposing weak or deprecated cipher suites if the protocol allows negotiation;
    - 6988 - modifying CRS configuration, where possible, to request or allow unencrypted access despite a  
6989 requirement for encryption.

6990 Verify that, when encrypted transport is mandated, the CRS does not silently fall back to unencrypted or  
6991 weakly protected channels and instead either maintains encrypted communication or fails the connection in a  
6992 documented way.

6993 **5. Assessment verdict**

- 6994 • Pass: CRS management/control interfaces, registry communication, and relevant APIs can be configured to  
6995 use encrypted transport; captured traffic for these configured channels is encrypted and uses acceptable  
6996 cryptographic parameters; and attempts to downgrade to unencrypted or weakly protected channels are  
6997 prevented or cause connection failure as documented.
- 6998 • Fail: encrypted transport cannot be enabled as documented; traffic configured to be encrypted is observed in  
6999 plaintext or with weaker protection than configured; or the CRS silently falls back to unencrypted or weakly  
7000 protected channels under downgrade attempts.

7001 **6. Assessment evidence**

- 7002 • [DESIGN] Documentation of secure communication mechanisms for CRS management, registry access, and  
7003 sensitive APIs, including downgrade-prevention behaviour.
- 7004 • [CONFIG] CRS configuration showing enabled encrypted transport settings and certificate or credential  
7005 configuration.
- 7006 • [LOG] Events or audit records relating to secure channel establishment, certificate or key errors, and rejected  
7007 downgraded connections.
- 7008 • [OBSERVATION] Packet captures and analysis outputs demonstrating encrypted traffic for configured  
7009 channels and failed or rejected downgrade attempts.
- 7010 • [PLT-CAP] Evidence of platform or service-mesh capabilities used to provide secure communication, where  
7011 applicable.

7012 **6.4.1.12.4 Assessment Case AC-CRS-CONF-004**7013 **1. Assessment reference**

7014 Requirement: REQ-CRS-CONF-004 (Advanced)

7015 **2. Assessment objective**7016 Verify that, where the underlying platform provides mechanisms for isolated or protected execution environments and  
7017 this capability is used:

- 7018 1) The CRS supports provisioning and management of execution environments for containerized workloads in  
7019 which confidential data associated with those workloads is protected, during execution, from observation,  
7020 inspection, or access through the CRS, the host operating system, and other workloads.
- 7021 2) The CRS provides configuration and API interfaces to deploy workloads into such protected execution  
7022 environments.
- 7023 3) The CRS supports secure provisioning of secrets and key material to those workloads such that the secrets and  
7024 key material are not accessible to the CRS itself, the host operating system, or other workloads.

7025 **3. Assessment preparation**

7026 The assessment shall have access to:

- 7027 • Documentation describing:
- 7028 - which confidential-execution mechanisms are supported (for example trusted execution environments,  
7029 confidential containers, hardware-backed isolation mechanisms);
- 7030 - how the CRS exposes configuration and API interfaces to deploy workloads into these environments;
- 7031 - how secrets and key material are provisioned to protected workloads and which entities can access them;
- 7032 - any limitations on CRS or host visibility into confidential workload memory and execution state.
- 7033 • A test configuration with:
- 7034 - a platform providing the documented confidential-execution capabilities;

- 7035 - the CRS configured to enable and manage confidential or protected workloads as described;
- 7036 - at least one containerized workload deployable both as a normal container and as a confidential/protected  
7037 instance;
- 7038 - CRS, host, and workload-level tools that would normally be used to inspect container state, logs, or  
7039 memory.

#### 7040 4. Assessment activities

7041 The assessment shall at least:

- 7042 1) Review documentation and CRS configuration to identify:
  - 7043 - how an operator selects confidential or protected execution for a workload (for example annotations,  
7044 profiles, or API parameters);
  - 7045 - which platform features are used to enforce additional confidentiality;
  - 7046 - how secret and key provisioning is designed to avoid exposure to CRS or host components.
- 7047 2) Deploy the test workload as a normal container and, using standard CRS and host tools:
  - 7048 - confirm that the workload's memory, process state, and secrets (for example environment variables,  
7049 mounted secret volumes) are accessible as documented for non-confidential workloads.
- 7050 3) Deploy the equivalent workload into a confidential or protected execution environment via CRS configuration  
7051 or APIs and:
  - 7052 - verify via platform or CRS indicators that the workload is running in the protected mode;
  - 7053 - attempt to access workload memory, process state, or confidential data using CRS and host inspection  
7054 tools;
  - 7055 - verify that direct inspection of confidential memory or secrets is blocked or that retrieved data is  
7056 unintelligible and consistent with the confidential-execution model.
- 7057 4) For secrets and key material provisioned to the confidential workload:
  - 7058 - examine CRS and host configurations to confirm that secrets are delivered through mechanisms that do  
7059 not expose their plaintext to CRS control-plane components or to other workloads;
  - 7060 - attempt to read these secrets from CRS logs, configuration stores, or unrelated workloads;
  - 7061 - verify that such attempts fail or reveal only non-sensitive metadata.

#### 7062 5. Assessment verdict

- 7063 • Pass: the CRS can deploy workloads into platform-provided confidential or protected execution environments;  
7064 in such environments, confidential workload data in use cannot be observed or accessed via normal CRS or  
7065 host inspection mechanisms; and secrets and key material are provisioned in a way that avoids exposure to  
7066 CRS and other workloads, consistent with the documented model.
- 7067 • Fail: workloads deployed as confidential can still be inspected in clear text by CRS or host tools; secrets or  
7068 keys intended to be confined to protected workloads are visible to CRS or other workloads; or documented  
7069 confidential-execution behaviour is not achieved in practice.

#### 7070 6. Assessment evidence

- 7071 • [DESIGN] Documentation of the CRS integration with platform confidential-execution mechanisms and the  
7072 secret/key provisioning model for protected workloads.
- 7073 • [CONFIG] CRS and platform configuration showing how confidential workloads are enabled and identified,  
7074 and how secret delivery is configured.
- 7075 • [LOG] Events and audit records related to creation, configuration, and failure of confidential workloads,  
7076 including any attempted misuse of confidential-execution options.

- 7077 • [OBSERVATION] Test records and screenshots showing normal versus confidential workload behaviour,  
7078 including failed attempts to inspect confidential memory or secrets.
- 7079 • [PLT-CAP] Evidence of platform confidential-computing capabilities used by the CRS.

## 7080 6.4.1.13 Assessment for Availability and Resilience

### 7081 6.4.1.13.1 Assessment Case AC-CRS-AVAIL-001

#### 7082 1. Assessment reference

7083 Requirement: REQ-CRS-AVAIL-001 (Basic)

#### 7084 2. Assessment objective

7085 Verify that:

- 7086 1) The CRS maintains resource isolation between container workloads and protects CRS control-plane processes  
7087 from CPU and memory exhaustion caused by container workloads.
- 7088 2) Failure or abnormal termination of an individual container is contained within its execution context and, under  
7089 normal conditions and configured limits, does not compromise CRS control-plane operation or cause other  
7090 containers to terminate unexpectedly.

#### 7091 3. Assessment preparation

7092 The assessment shall have access to:

- 7093 • Documentation of CRS resource isolation mechanisms (namespaces, cgroups/quotas, limits) and how they  
7094 apply to CPU and memory.
- 7095 • Documentation of CRS control-plane components and their protection from workload resource use.
- 7096 • Documentation of how container failures are handled (restart policy, isolation, impact).
- 7097 • A test configuration with multiple containers, including at least one container capable of generating high load  
7098 and one or more normal workloads.

#### 7099 4. Assessment activities

7100 The assessment shall at least:

- 7101 1) Confirm that CPU and memory limits or equivalent isolation mechanisms can be configured per container or  
7102 workload, and that CRS control-plane processes are not scheduled within the same resource domain as  
7103 arbitrary workloads.
- 7104 2) Configure a container with aggressive CPU/memory usage within defined limits, generate high load, and  
7105 verify that CRS control-plane functions (e.g. scheduling, API server, management operations) remain  
7106 operational and responsive.
- 7107 3) Induce abnormal termination of a selected container (e.g. crash or kill) and verify that:
  - 7108 - the CRS control plane remains operational; and
  - 7109 - other containers continue to run unaffected under normal operating conditions and configured limits.

#### 7110 5. Assessment verdict

- 7111 • Pass: CPU and memory isolation mechanisms protect CRS control-plane processes from workload-induced  
7112 exhaustion; and failure of an individual container does not compromise CRS control-plane operation or trigger  
7113 unexpected termination of other containers under normal conditions.
- 7114 • Fail: control-plane processes are starved or disrupted by workload resource usage; or a single container failure  
7115 propagates to the control plane or causes other containers to terminate unexpectedly.

#### 7116 6. Assessment evidence

- 7117 • [CONFIG] Resource isolation (limits/quotas) configuration for containers and control-plane components.

- 7118      • [DESIGN] Documentation of CRS resource isolation and failure-handling architecture.
- 7119      • [LOG] / [OBSERVATION] Evidence showing control-plane availability under workload stress and  
7120      containment of individual container failures.
- 7121      **6.4.1.13.2 Assessment Case AC-CRS-AVAIL-002**
- 7122      **1. Assessment reference**
- 7123      Requirement: REQ-CRS-AVAIL-002 (Elevated)
- 7124      **2. Assessment objective**
- 7125      Verify that:
- 7126      1) The CRS provides automatic recovery mechanisms to restart failed containers based on defined policies,  
7127      without manual intervention.
- 7128      2) Where supported by the deployment environment, these mechanisms can be used with orchestration or  
7129      clustering systems to restore workloads on the same or an alternative host.
- 7130      **3. Assessment preparation**
- 7131      The assessment shall have access to:
- 7132      • Documentation of CRS container restart policies and automatic recovery behaviour (e.g. "always", "on-  
7133      failure", back-off behaviour).
- 7134      • Documentation of integration with orchestration or clustering systems, where supported (e.g. Kubernetes,  
7135      Swarm, other orchestrators).
- 7136      • A test environment with several containers, and where supported, at least two hosts under common  
7137      orchestration/cluster control.
- 7138      **4. Assessment activities**
- 7139      The assessment shall at least:
- 7140      1) Configure one or more containers with automatic restart policies as described in documentation.
- 7141      2) Induce container failure (e.g. crash, forced termination) and verify that the CRS automatically restarts the  
7142      container according to the defined policy, without manual actions.
- 7143      3) Where orchestration/clustering is supported, simulate host or node unavailability for a container workload and  
7144      verify that the orchestration/cluster in conjunction with CRS mechanisms restores the workload on the same or  
7145      an alternative host according to the configuration.
- 7146      **5. Assessment verdict**
- 7147      • Pass: failed containers configured with restart policies are automatically restarted without manual intervention;  
7148      and, where supported, orchestration/clustering combined with CRS behaviour restores workloads on the same  
7149      or another host as documented.
- 7150      • Fail: container restart requires manual intervention in contradiction with configured policies; or supported  
7151      orchestration/clustering cannot restore workloads as intended.
- 7152      **6. Assessment evidence**
- 7153      • [CONFIG] Container restart policy configuration and, where applicable, orchestration/cluster configuration.
- 7154      • [DESIGN] Documentation of CRS automatic recovery mechanisms and integrations with  
7155      orchestration/clustering systems.
- 7156      • [LOG] / [OBSERVATION] Evidence of automatic container restarts and, where applicable, restoration of  
7157      workloads on alternative hosts.

## 7158 6.4.1.13.3 Assessment Case AC-CRS-AVAIL-003

7159 **1. Assessment reference**

7160 Requirement: REQ-CRS-AVAIL-003 (Elevated)

7161 **2. Assessment objective**

7162 Verify that:

- 7163 1) The CRS implements resource scheduling or reservation mechanisms that allow configuration of minimum  
7164 resource allocations for CRS security and management functions (CPU, memory and I/O).
- 7165 2) These mechanisms protect the CRS control plane from resource starvation caused by container workloads or  
7166 external contention, ensuring that security and management functions remain able to operate under load.

7167 **3. Assessment preparation**

7168 The assessment shall have access to:

- 7169 • Documentation of CRS resource scheduling/reservation or priority mechanisms (e.g. control-plane  
7170 reservations, priorities, QoS classes).
- 7171 • Documentation describing how minimum CPU, memory and I/O resources can be reserved or prioritized for  
7172 CRS security and management components.
- 7173 • A test configuration with CRS security/management functions active and multiple container workloads  
7174 capable of generating high CPU, memory and I/O load.

7175 **4. Assessment activities**

7176 The assessment shall at least:

- 7177 1) Confirm that resource scheduling or reservation mechanisms exist and that minimum CPU, memory and I/O  
7178 allocations (or equivalent prioritization) can be configured for CRS security and management functions.
- 7179 2) Configure such minimum allocations/priorities in the assessed environment.
- 7180 3) Generate high resource load from containers and/or external sources (CPU- and I/O-intensive workloads) and  
7181 verify that:
- 7182 - CRS management and security interfaces remain responsive; and
  - 7183 - critical control-plane operations (e.g. container lifecycle operations, logging, policy enforcement)  
7184 continue to function correctly under load.

7185 **5. Assessment verdict**

- 7186 • Pass: resource scheduling/reservation mechanisms support configuration of minimum CPU, memory and I/O  
7187 allocations (or equivalent prioritization) for CRS security and management functions; and under high load  
7188 from container workloads or external contention, these functions remain able to operate as intended.
- 7189 • Fail: minimum allocations/priorities for security/management functions cannot be configured; or under load,  
7190 the CRS control plane or security/management functions become starved and unable to operate.

7191 **6. Assessment evidence**

- 7192 • [CONFIG] Resource scheduling/reservation or prioritization configuration for CRS security and management  
7193 components.
- 7194 • [DESIGN] Documentation of CRS resource scheduling architecture and its application to control-plane  
7195 protection.
- 7196 • [LOG] / [OBSERVATION] Evidence that CRS management and security functions continue to operate  
7197 correctly under stress conditions.

## 7198 6.4.1.13.4 Assessment Case AC-CRS-AVAIL-004

7199 **1. Assessment reference**

7200 Requirement: REQ-CRS-AVAIL-004 (Advanced)

7201 **2. Assessment objective**

7202 Verify that:

- 7203 1) Where the deployment environment supports workload relocation, the CRS supports checkpoint and restore of  
7204 the runtime state of selected container workloads, including process state and relevant in-memory execution  
7205 context.
- 7206 2) This checkpoint/restore capability can be used, together with orchestration or clustering mechanisms, to enable  
7207 failover with minimal disruption to workload execution when relocating to a standby instance or alternative  
7208 host.

7209 **3. Assessment preparation**

7210 The assessment shall have access to:

- 7211 • Documentation of CRS checkpoint/restore mechanisms for container workloads (scope of state, limitations,  
7212 required environment support).
- 7213 • Documentation of how orchestration or clustering systems can coordinate relocation and failover using  
7214 checkpoint/restore.
- 7215 • A test environment where workload relocation is supported (e.g. compatible hosts, shared or consistent  
7216 storage/network configuration) and at least one container workload configured for checkpoint/restore and  
7217 relocation.

7218 **4. Assessment activities**

7219 The assessment shall at least:

- 7220 1) Configure a selected container workload to use CRS checkpoint/restore as documented, including any required  
7221 integration with the orchestrator or cluster.
- 7222 2) Create a checkpoint of the running workload and verify that process state and relevant in-memory execution  
7223 context are captured according to the CRS documentation.
- 7224 3) Simulate a control-plane or host-level failure (or initiate a planned relocation) and verify that the workload is  
7225 restored from checkpoint on a standby instance or alternative host using orchestrator/cluster coordination.
- 7226 4) Observe workload behaviour and confirm that disruption is limited to the minimal interruption inherent to the  
7227 checkpoint/restore and relocation process, rather than a full re-deployment from scratch.

7228 **5. Assessment verdict**

- 7229 • Pass: checkpoint and restore of runtime state for selected container workloads is supported; when combined  
7230 with orchestration/clustering, the mechanism enables relocation/failover to a standby or alternative host with  
7231 minimal disruption to workload execution as documented.
- 7232 • Fail: checkpoint/restore of runtime state cannot be configured or does not function as specified; or  
7233 relocation/failover requires full restart without using checkpointed state; or workload disruption is inconsistent  
7234 with the intended "minimal disruption" behaviour.

7235 **6. Assessment evidence**

- 7236 • [CONFIG] Checkpoint/restore and relocation configuration for selected container workloads, including any  
7237 orchestrator/cluster settings.
- 7238 • [DESIGN] Documentation of CRS checkpoint/restore and relocation architecture, including scope of captured  
7239 state and expected failover behaviour.
- 7240 • [LOG] / [OBSERVATION] Evidence of checkpoint creation, restore operations and observed workload  
7241 behaviour during and after relocation/failover.

7242 **6.4.1.14 Assessment for Logging**7243 **6.4.1.14.1 Assessment Case AC-CRS-LOG-001**7244 **1. Assessment reference**

7245 Requirement: REQ-CRS-LOG-001 (Basic)

7246 **2. Assessment objective**

7247 Verify that:

- 7248 1) The CRS generates audit logs for security-relevant administrative actions and security events, including at  
7249 least:
- 7250 - authentication attempts to CRS administrative or control interfaces;
  - 7251 - configuration changes affecting CRS behaviour or security posture;
  - 7252 - container lifecycle operations (create, start, stop, delete, restart) initiated through the CRS.
- 7253 2) Audit logs are not writable or directly modifiable by container workloads or other unprivileged processes  
7254 running within container namespaces.

7255 **3. Assessment preparation**

7256 The assessment shall have access to:

- 7257 • Documentation of CRS logging/auditing capabilities and log storage locations.
- 7258 • Documentation of which actions/events are treated as security-relevant.
- 7259 • A configuration with:
  - 7260 - CRS administrative/control access enabled;
  - 7261 - at least one container workload that can be created, started, stopped, restarted and deleted.

7262 **4. Assessment activities**

7263 The assessment shall at least:

- 7264 1) Identify audit log sources and storage locations from documentation/configuration.
- 7265 2) In the assessed configuration, perform:
- 7266 - successful and failed authentication attempts to CRS administrative/control interfaces;
  - 7267 - CRS configuration changes affecting behaviour/security posture;
  - 7268 - container lifecycle operations (creation, start, stop, restart, deletion) via CRS mechanisms.
- 7269 Verify that corresponding audit records are generated, including at least action/event type, affected  
7270 object, timestamp and responsible account or process where applicable.
- 7271 3) From a container workload and from another unprivileged process, attempt to write to, modify or delete the  
7272 CRS audit log files or endpoints and verify that these operations are denied.

7273 **5. Assessment verdict**

- 7274 • Pass: the CRS generates audit logs for the specified security-relevant actions/events; and audit logs cannot be  
7275 written or directly modified by container workloads or other unprivileged processes in container namespaces.
- 7276 • Fail: one or more required events are not logged; or container workloads/unprivileged processes can write or  
7277 modify audit logs.

7278 **6. Assessment evidence**

- 7279 • [CONFIG] CRS logging configuration and log storage settings.

- 7280 • [DESIGN] Documentation of audit logging behaviour, logged event categories and log isolation from  
7281 workloads.
- 7282 • [LOG] / [OBSERVATION] Sample audit entries for authentication attempts, configuration changes and  
7283 container lifecycle operations, and evidence of denied write/modify attempts from containers/unprivileged  
7284 processes.

#### 7285 6.4.1.14.2 Assessment Case AC-CRS-LOG-002

##### 7286 1. Assessment reference

7287 Requirement: REQ-CRS-LOG-002 (Elevated)

##### 7288 2. Assessment objective

7289 Verify that:

- 7290 1) CRS audit logs are protected from Unauthorized access, modification or deletion.
- 7291 2) Access to CRS audit logs is restricted to authorized administrative roles or trusted system processes under  
7292 CRS or host OS control.
- 7293 3) The CRS supports secure export of audit logs to external log management or SIEM systems using  
7294 authenticated communication channels.

##### 7295 3. Assessment preparation

7296 The assessment shall have access to:

- 7297 • Documentation on audit log storage (files, services, APIs) and access-control mechanisms (roles, permissions,  
7298 OS controls).
- 7299 • Documentation identifying which administrative roles and trusted system processes are allowed to  
7300 access/manage CRS audit logs.
- 7301 • Documentation of CRS log export mechanisms and supported authenticated transport options.
- 7302 • A configuration with audit logging enabled and a reachable external log management/SIEM endpoint (test  
7303 instance) that can accept logs over an authenticated channel.

##### 7304 4. Assessment activities

7305 The assessment shall at least:

- 7306 1) Review configuration and underlying platform permissions to confirm that audit log storage can only be  
7307 accessed by authorized administrative roles or trusted system processes, and that modification/deletion by  
7308 other users/processes is prevented.
- 7309 2) Attempt to read, modify or delete audit logs using non-authorized accounts or processes and verify that these  
7310 attempts are denied.
- 7311 3) Configure export of CRS audit logs to the external log/SIEM endpoint using an authenticated communication  
7312 channel as documented (e.g. authenticated TLS, authenticated agent protocol).
- 7313 4) Verify, using configuration and where feasible protocol inspection, that:
- 7314 - the log export uses authenticated communication; and
- 7315 - the external system receives log records corresponding to those generated locally.

##### 7316 5. Assessment verdict

- 7317 • Pass: only authorized administrative roles/trusted processes can access CRS audit logs; Unauthorized  
7318 access/modification/deletion is prevented; and CRS audit logs can be exported to external systems over  
7319 authenticated channels.
- 7320 • Fail: Unauthorized users/processes can access or tamper with audit logs; or access controls on audit logs are  
7321 insufficient; or log export does not use authenticated channels or cannot be securely configured.

##### 7322 6. Assessment evidence

- 7323 • [CONFIG] Audit log storage permissions, role/process mappings and export configuration, including  
7324 authentication settings.
- 7325 • [DESIGN] Documentation of CRS audit log protection model and external export mechanisms.
- 7326 • [LOG] / [OBSERVATION] Evidence of denied access attempts by non-authorized entities and successful  
7327 authenticated export to an external log/SIEM system.

### 7328 6.4.1.14.3 Assessment Case AC-CRS-LOG-003

#### 7329 1. Assessment reference

7330 Requirement: REQ-CRS-LOG-003 (Advanced)

#### 7331 2. Assessment objective

7332 Verify that:

- 7333 1) The CRS supports cryptographic protection and trusted time-stamping of audit logs so that origin and integrity  
7334 of audit records can be verified during forensic analysis.
- 7335 2) Where external time or signing services are used, audit records include sufficient metadata to identify and  
7336 verify the trust source used for time-stamping and integrity protection.

#### 7337 3. Assessment preparation

7338 The assessment shall have access to:

- 7339 • Documentation describing cryptographic mechanisms used to protect CRS audit logs (e.g. signatures, MACs,  
7340 hash chains or log sealing), including how verification is performed.
- 7341 • Documentation of trusted time-stamping (e.g. internal secure time source, external time service, signing  
7342 service) and the metadata recorded in audit logs about the trust source.
- 7343 • A configuration with cryptographic log protection and trusted time-stamping enabled, with the ability to export  
7344 or access raw audit log data and any associated metadata for verification.

#### 7345 4. Assessment activities

7346 The assessment shall at least:

- 7347 1) Enable cryptographic protection and trusted time-stamping for CRS audit logs as documented.
- 7348 2) Generate a set of representative audit events (e.g. logins, configuration changes, container lifecycle operations)  
7349 and collect the corresponding audit records and associated metadata.
- 7350 3) Using the documented verification tools or procedures, verify that:
- 7351 - cryptographic checks over the audit records succeed, demonstrating integrity and origin as designed;
- 7352 - each audit record includes a trusted timestamp; and
- 7353 - where external time or signing services are used, the log metadata clearly identifies the trust source (e.g.  
7354 service identifier, certificate, or similar) and allows verification against that source.
- 7355 4) Modify a copy of the audit log (e.g. change or remove records) and confirm that cryptographic and/or time-  
7356 stamp verification fails or indicates tampering.

#### 7357 5. Assessment verdict

- 7358 • Pass: cryptographic mechanisms and trusted time-stamping for CRS audit logs are available; audit records  
7359 contain sufficient metadata to verify the origin, integrity and trust source; and tampering with logs is  
7360 detectable through the provided verification process.
- 7361 • Fail: cryptographic protection or trusted time-stamping cannot be enabled; audit records lack metadata needed  
7362 to verify the trust source; or modifications to audit logs cannot be reliably detected.

#### 7363 6. Assessment evidence

- 7364 • [CONFIG] Configuration enabling cryptographic protection and trusted time-stamping of CRS audit logs,  
7365 including any external signing/time service integration.
- 7366 • [DESIGN] Documentation of CRS log protection and time-stamping architecture, including metadata format  
7367 and verification procedures.
- 7368 • [LOG] / [OBSERVATION] Audit logs and verification outputs showing successful checks for unmodified  
7369 logs and detection of tampering and/or invalid trust source.

## 7370 6.4.1.15 Assessment for Secure Update

### 7371 6.4.1.15.1 Assessment Case AC-CRS-UPD-001

#### 7372 1. Assessment reference

7373 Requirement: REQ-CRS-UPD-001 (Basic)

#### 7374 2. Assessment objective

7375 Verify that:

- 7376 1) The CRS supports applying security updates to CRS components without a full reinstallation of the CRS.
- 7377 2) The CRS can be updated without requiring a full reinstallation of the host system.
- 7378 3) Applicable update mechanisms in the evaluated solution (e.g. host package manager, management appliance,  
7379 orchestration pipeline) can update CRS binaries and related components in-place.

#### 7380 3. Assessment preparation

7381 The assessment shall have access to:

- 7382 • Documentation of the CRS update mechanisms used in the evaluated solution (host package manager,  
7383 management appliance, orchestration/CI pipeline, image-based updates).
- 7384 • Documentation indicating which binaries and components are considered CRS components in scope.
- 7385 • A test environment where a newer CRS version or security patch is available for application using the  
7386 supported mechanism(s).

#### 7387 4. Assessment activities

7388 The assessment shall at least:

- 7389 1) Identify the mechanism(s) responsible for delivering and installing CRS component updates in the evaluated  
7390 solution.
- 7391 2) Apply a CRS update or security patch using the documented mechanism(s) and verify that the update  
7392 completes without performing a full CRS or host system reinstallation.
- 7393 3) After the update, verify that:
  - 7394 - the CRS version/build reflects the applied update; and
  - 7395 - existing CRS configuration and container workloads behave according to the product's documented  
7396 update behaviour.

#### 7397 5. Assessment verdict

- 7398 • Pass: CRS components are updated through the documented mechanisms without requiring full CRS or host  
7399 system reinstallation, and the updated CRS is active afterwards.
- 7400 • Fail: CRS updates require reinstalling the CRS or host system; or CRS components cannot be updated using  
7401 the documented product update mechanism.

#### 7402 6. Assessment evidence

- 7403 • [SCOPE] Description of CRS components in scope and the update mechanism(s) used.

- 7404 • [CONFIG] Update mechanism configuration (repositories, appliance/orchestrator settings, image update  
7405 policies).
- 7406 • [DESIGN] Documentation of the CRS update model and its relation to host OS/appliance/orchestration  
7407 services.
- 7408 • [LOG] / [OBSERVATION] Evidence of a successful in-place CRS update (before/after version, update logs,  
7409 continuity of workloads).

#### 7410 6.4.1.15.2 Assessment Case AC-CRS-UPD-002

##### 7411 1. Assessment reference

7412 Requirement: REQ-CRS-UPD-002 (Elevated)

##### 7413 2. Assessment objective

7414 Verify that:

- 7415 1) The CRS, or the update mechanisms in the evaluated solution, verify authenticity and integrity of all CRS  
7416 updates using digital signatures validated against trusted keys or certificates configured for the CRS.
- 7417 2) If authenticity or integrity verification fails, installation of the CRS patch or update is prevented and a security-  
7418 relevant event is recorded.
- 7419 3) Trusted keys or certificates used for CRS update verification cannot be modified by container workloads or  
7420 other unprivileged components during normal operation.

##### 7421 3. Assessment preparation

7422 The assessment shall have access to:

- 7423 • Documentation describing CRS update verification (signature formats, trust anchors, key/certificate  
7424 management).
- 7425 • Documentation showing where and how trusted keys/certificates for CRS update verification are stored and  
7426 protected.
- 7427 • Ability to obtain:
  - 7428 - a valid, correctly signed CRS update; and
  - 7429 - an update artefact with invalid/missing signatures or with intentional modification to simulate  
7430 verification failure.

##### 7431 4. Assessment activities

7432 The assessment shall at least:

- 7433 1) Apply a valid signed CRS update using the documented mechanism and verify that signature/integrity  
7434 verification is performed and the update is accepted and installed.
- 7435 2) Attempt to apply an invalid or tampered CRS update (e.g. corrupted or unsigned package) and verify that:
  - 7436 - authenticity/integrity verification fails;
  - 7437 - the update is not installed; and
  - 7438 - a security-relevant event is recorded in the CRS or host audit logs.
- 7439 3) Review configuration and underlying platform controls for the trusted keys/certificates used in CRS update  
7440 verification and verify that container workloads or other unprivileged components cannot modify or replace  
7441 these trust anchors under normal operation (e.g. by file permissions, store isolation, or hardware protection).

##### 7442 5. Assessment verdict

- 7443 • Pass: authenticity and integrity of CRS updates are verified against trusted keys/certificates; updates that fail  
7444 verification are not installed and generate a security-relevant log; and trust anchors for CRS update verification  
7445 cannot be modified by container workloads or other unprivileged components.

- 7446 • Fail: CRS updates can be installed without effective signature/integrity verification; or failed verification does  
7447 not block installation and/or is not logged as a security event; or trusted keys/certificates can be modified by  
7448 unprivileged components.

7449 **6. Assessment evidence**

- 7450 • [CONFIG] CRS update verification configuration, trust store settings and access controls on keys/certificates.  
7451 • [DESIGN] Documentation of the CRS update verification process and trust model, including how failures are  
7452 logged.  
7453 • [LOG] / [OBSERVATION] Evidence of successful installation of a valid update, blocked installation of a  
7454 tampered update, and corresponding security-relevant log entries.  
7455 • [PLT-CAP] Where relevant, evidence of platform capabilities used to protect trust anchors (e.g. OS keystore,  
7456 HSM, read-only image).

7457 **6.4.1.15.3 Assessment Case AC-CRS-UPD-003**

7458 **1. Assessment reference**

7459 Requirement: REQ-CRS-UPD-003 (Advanced)

7460 **2. Assessment objective**

7461 Verify that, where version rollback is supported for CRS components:

- 7462 1) The update mechanism implements rollback protection to prevent Unauthorized installation of outdated,  
7463 revoked or replayed CRS updates.  
7464 2) Authorized rollback to a previously trusted CRS version is only permitted when:  
7465 - the rollback image/package is verified for integrity and authenticity against a trusted key/certificate;  
7466 - the rollback action is explicitly authorized by an administrator with elevated privileges or equivalent  
7467 authorized control function; and  
7468 - the rollback operation is recorded as a security-relevant event in the audit logs, including the version  
7469 reverted from and the version reverted to.

7470 **3. Assessment preparation**

7471 The assessment shall have access to:

- 7472 • Documentation of rollback capabilities for CRS components and rollback protection mechanisms (version  
7473 tracking, anti-replay, revocation checks).  
7474 • Documentation of the roles or control functions authorized to perform rollback and of the audit logging of  
7475 update/rollback operations.  
7476 • A test configuration where:  
7477 - CRS rollback is supported by the evaluated update mechanism; and  
7478 - forward updates and rollback to a previous trusted version can be exercised.

7479 **4. Assessment activities**

7480 The assessment shall at least:

- 7481 1) Confirm from documentation/configuration that the update mechanism tracks CRS versions and prevents  
7482 installation of outdated, revoked or replayed updates except via an explicit, authorized rollback procedure.  
7483 2) Attempt to install an older or revoked CRS update package without following the documented rollback  
7484 procedure and verify that the mechanism blocks this as Unauthorized rollback/replay.  
7485 3) Perform an authorized rollback using the documented procedure and verify that:  
7486 - the rollback image/package is subject to integrity and authenticity verification before use;

- 7487 - the rollback requires elevated administrative privileges or equivalent control function; and
- 7488 - the rollback event is recorded as a security-relevant audit record, including at least the previous CRS
- 7489 version and the rollback target version.
- 7490 4) After rollback, verify that the CRS runs the expected previous trusted version and that normal operation is
- 7491 restored according to product documentation.

7492 **5. Assessment verdict**

- 7493 • Pass: Unauthorized installation of outdated/revoked/replayed CRS updates is prevented; and authorized
- 7494 rollback requires integrity/authenticity verification of the rollback image, elevated authorization, and produces
- 7495 an audit log recording from/to version.
- 7496 • Fail: outdated or revoked CRS updates can be installed without appropriate controls; or rollback does not
- 7497 require elevated authorization; or rollback events are not logged with sufficient detail.

7498 **6. Assessment evidence**

- 7499 • [CONFIG] CRS update/rollback configuration, including version control, rollback policy and administrative
- 7500 authorization settings.
- 7501 • [DESIGN] Documentation of rollback protection mechanisms and the authorized rollback process.
- 7502 • [LOG] / [OBSERVATION] Evidence of blocked Unauthorized rollback attempts and of logged authorized
- 7503 rollback operations, including version information before and after rollback.

7504 **6.4.1.16 Assessment for Secure Configuration and Default**

7505 **6.4.1.16.1 Assessment Case AC-CRS-CFG-001**

7506 **1. Assessment reference**

7507 Requirement: REQ-CRS-CFG-001 (Basic)

7508 **2. Assessment objective**

7509 Verify that:

- 7510 1) By default, the CRS disables interfaces and features not required for container execution and CRS
- 7511 administration, including remote debugging endpoints, unauthenticated control sockets, and developer
- 7512 interfaces exposing management or control capabilities.
- 7513 2) Only administrators can explicitly enable such interfaces and features through CRS configuration.

7514 **3. Assessment preparation**

7515 The assessment shall have access to:

- 7516 • Documentation listing CRS interfaces/features (remote debugging, control sockets, developer interfaces,
- 7517 admin APIs/UIs) and identifying those required for normal container execution and CRS administration.
- 7518 • Documentation of CRS configuration mechanisms and role/permission model for modifying configuration.
- 7519 • A fresh/default CRS deployment in its documented default configuration, with at least one non-administrative
- 7520 account or context (e.g. unprivileged user, container workload).

7521 **4. Assessment activities**

7522 The assessment shall at least:

- 7523 1) In the default configuration, attempt to access all documented non-essential interfaces/features (remote
- 7524 debugging endpoints, unauthenticated control sockets, developer interfaces) and verify they are disabled or not
- 7525 reachable.
- 7526 2) Confirm that interfaces/features required for container execution and CRS administration are available as
- 7527 documented.

- 7528 3) Using administrative credentials, enable one or more of the disabled interfaces/features via CRS configuration  
7529 and verify that:
- 7530 - the configuration change is accepted; and
- 7531 - the corresponding interface/feature becomes available.
- 7532 4) Attempt to perform the same enable operations using a non-administrative account/context and verify that the  
7533 configuration change is rejected or not permitted.

7534 **5. Assessment verdict**

- 7535 • Pass: non-essential/debug/developer interfaces and unauthenticated control sockets are disabled by default;  
7536 required interfaces for execution/administration are available; and only administrators can enable additional  
7537 interfaces/features.
- 7538 • Fail: non-essential or insecure interfaces are enabled by default; or non-administrative users can enable such  
7539 interfaces/features.

7540 **6. Assessment evidence**

- 7541 • [CONFIG] Default and modified CRS configuration showing interface/feature states and permissions required  
7542 for changes.
- 7543 • [DESIGN] Documentation of CRS interfaces, default exposure, and configuration/role model.
- 7544 • [OBSERVATION] / [LOG] Connection attempts and CLI/UI output showing:
- 7545 - non-availability of non-essential interfaces by default;
- 7546 - successful enable by admin;
- 7547 - failed enable attempts by non-admin.

7548 **6.4.1.16.2 Assessment Case AC-CRS-CFG-002**

7549 **1. Assessment reference**

7550 Requirement: REQ-CRS-CFG-002 (Elevated)

7551 **2. Assessment objective**

7552 Verify that:

- 7553 1) The CRS validates configuration parameters under its control before applying them.
- 7554 2) Configuration validation rejects settings that would:
- 7555 - disable or bypass mandatory security requirements defined in the present document; or
- 7556 - compromise container-to-container isolation; or
- 7557 - compromise controlled access to host resources.
- 7558 3) When a configuration change is rejected for security reasons, the CRS provides clear feedback to the  
7559 administrator indicating the reason.

7560 **3. Assessment preparation**

7561 The assessment shall have access to:

- 7562 • Documentation of CRS configuration parameters and validation behaviour, including examples of security-  
7563 relevant invalid settings (e.g. disabling required controls, removing isolation, granting unrestricted host  
7564 access).
- 7565 • Documentation of mandatory CRS security controls, isolation guarantees and host-access restrictions.
- 7566 • A test environment where CRS configuration can be modified via supported administrative interfaces.

7567 **4. Assessment activities**

7568 The assessment shall at least:

- 7569 1) Identify a set of configuration parameters where insecure values would:
- 7570 - disable/bypass mandatory security controls (e.g. turning off mandatory auth/logging/admission checks);
- 7571 - weaken or remove container-to-container isolation; or
- 7572 - grant uncontrolled access to host resources that should remain restricted.
- 7573 2) Attempt to apply such insecure configuration changes and verify that the CRS refuses to apply them as part of
- 7574 its validation logic.
- 7575 3) Capture the feedback presented to the administrator and verify that it clearly indicates the security-related
- 7576 reason for rejection (e.g. reference to disabled control, isolation violation, or host-access violation).
- 7577 4) Apply one or more valid configuration changes and verify they are accepted and enforced, demonstrating that
- 7578 validation focuses on insecure cases rather than blocking configuration in general.

7579 **5. Assessment verdict**

- 7580 • Pass: configuration parameters are validated before application; insecure settings that disable/bypass
- 7581 mandatory controls or compromise isolation/host access are rejected; and clear, security-related feedback is
- 7582 provided to the administrator.
- 7583 • Fail: unsafe configurations can be applied; or validation does not cover key security/isolation/host-access
- 7584 guarantees; or error messages do not clearly explain the security reason.

7585 **6. Assessment evidence**

- 7586 • [CONFIG] Examples of attempted insecure and valid configuration changes and their outcomes.
- 7587 • [DESIGN] Documentation of CRS configuration validation logic and its linkage to mandatory controls,
- 7588 isolation, and host-access rules.
- 7589 • [LOG] / [OBSERVATION] CLI/UI output showing rejection of insecure configurations with explicit reasons,
- 7590 and acceptance of secure configurations.

7591 **6.4.1.16.3 Assessment Case AC-CRS-CFG-003**

7592 **1. Assessment reference**

7593 Requirement: REQ-CRS-CFG-003 (Advanced)

7594 **2. Assessment objective**

7595 Verify that:

- 7596 1) The CRS supports defining and using security configuration baselines that specify secure settings for:
- 7597 - CRS daemon configuration; and
- 7598 - default container security profiles, including at least isolation-related options and default access to host
- 7599 resources.
- 7600 2) The CRS can automatically validate current CRS configuration and default container profile settings against a
- 7601 selected security baseline.
- 7602 3) The CRS generates alerts or audit events when non-compliant settings are detected.

7603 **3. Assessment preparation**

7604 The assessment shall have access to:

- 7605 • Documentation describing security configuration baselines (how they are defined, stored, selected and scoped
- 7606 to daemon config and default container profiles).

7607 • Documentation of baseline content, including isolation-related options (e.g. default capabilities, privilege  
7608 levels, namespaces) and default host resource access (e.g. host volumes, host network).

7609 • Documentation of automatic baseline validation mechanisms and alert/audit integration.

7610 • A test environment where at least one security baseline can be defined/selected and CRS/default-profile  
7611 settings can be altered to introduce non-compliance.

#### 7612 4. Assessment activities

7613 The assessment shall at least:

7614 1) Define or select a security baseline that includes secure settings for CRS daemon configuration and default  
7615 container security profiles, covering isolation and host-access defaults as supported by the product.

7616 2) Align CRS configuration and default container profiles with the baseline and invoke the baseline validation  
7617 function; verify that the CRS reports compliance and no non-compliance alerts/events are generated.

7618 3) Introduce deliberate deviations from the baseline (e.g. relaxing default container isolation, enabling privileged  
7619 default containers, broadening default host resource access, changing a daemon setting that weakens isolation).

7620 4) Run baseline validation again and verify that:

7621 - the CRS detects one or more non-compliant settings; and

7622 - alerts or audit log entries are generated identifying which settings deviate from the selected baseline.

#### 7623 5. Assessment verdict

7624 • Pass: security baselines can be defined and used for CRS daemon configuration and default container security  
7625 profiles; automatic validation against a selected baseline is available; and detected non-compliance results in  
7626 alerts or audit events that identify the deviations.

7627 • Fail: security baselines cannot be defined or do not cover both daemon config and default profiles; automatic  
7628 validation cannot be performed or is ineffective; or non-compliant settings do not trigger clear alerts/audit  
7629 events.

#### 7630 6. Assessment evidence

7631 • [CONFIG] Example security baseline definition and corresponding CRS configuration/default profile before  
7632 and after introducing deviations.

7633 • [DESIGN] Documentation of CRS baseline management, validation logic and alert/audit integration.

7634 • [LOG] / [OBSERVATION] Validation outputs and alert/audit records showing detection and reporting of non-  
7635 compliant settings.

### 7636 6.4.1.17 Assessment for Data Minimization

#### 7637 6.4.1.17.1 Assessment Case AC-CRS-DM-001

##### 7638 1. Assessment reference

7639 Requirement: REQ-CRS-DM-001 (Basic)

##### 7640 2. Assessment objective

7641 Verify that:

7642 1) CRS logging, telemetry and diagnostic outputs are limited to information necessary for runtime operation,  
7643 troubleshooting and security monitoring.

7644 2) Logs/telemetry/diagnostics do not unintentionally expose sensitive security material such as plaintext  
7645 credentials, full cryptographic keys or complete container memory contents.

7646 3) Payload content from container workloads is not included in CRS logs/telemetry/diagnostics by default.

- 7647 4) Where payload inclusion for debugging is supported, it is only enabled through explicit administrative  
7648 configuration and this state is clearly indicated to the administrator.

### 7649 3. Assessment preparation

7650 The assessment shall have access to:

- 7651 • Documentation of CRS logging, telemetry and diagnostic capabilities, including any "debug", "deep  
7652 inspection" or "payload capture" options.
- 7653 • Documentation specifying which types of data are explicitly excluded (credentials, keys, memory dumps,  
7654 application payloads).
- 7655 • A test configuration with:
  - 7656 - administrative access to configure logging/telemetry/diagnostics;
  - 7657 - at least one container workload processing identifiable test payload data (e.g. specific strings);
  - 7658 - the ability to generate authentication events and crypto operations with known test values.

### 7659 4. Assessment activities

7660 The assessment shall at least:

- 7661 1) Review documentation and default configuration to identify:
  - 7662 - which events and fields are logged/collected by default;
  - 7663 - any options that may enable payload or extended context capture.
- 7664 2) In default configuration, perform:
  - 7665 - successful and failed authentication attempts to CRS admin/control interfaces using known test  
7666 usernames/password patterns;
  - 7667 - cryptographic operations (e.g. key generation/use) with identifiable test keys;
  - 7668 - normal container workload activity with recognizable test payload data.

7669 Then inspect logs/telemetry/diagnostics to verify that they do not contain:

  - 7670 - plaintext credentials;
  - 7671 - full cryptographic keys or complete container memory contents;
  - 7672 - application/payload data from the containers.
- 7673 3) Where the CRS supports payload inclusion for debugging:
  - 7674 - enable the relevant debug/payload option using administrative configuration;
  - 7675 - verify that only an administrator can enable it and that the configuration/status clearly indicates that  
7676 payload capture is active;
  - 7677 - generate new test payload data and verify that such payload appears in logs/diagnostics as documented.

### 7678 5. Assessment verdict

- 7679 • Pass: by default, logs/telemetry/diagnostics are limited to necessary operational and security data, without  
7680 plaintext credentials, full keys, complete memory dumps or container payload; and any payload capture is only  
7681 enabled via explicit administrative configuration and is clearly indicated.
- 7682 • Fail: sensitive security material or container payload appears in default logs/telemetry/diagnostics; or payload  
7683 capture can be enabled without explicit admin action or clear indication.

### 7684 6. Assessment evidence

- 7685 • [CONFIG] Default and modified CRS logging/telemetry/diagnostic configuration, including any  
7686 debug/payload options.
- 7687 • [DESIGN] Documentation of data minimization behaviour and explicit exclusions (credentials, keys, memory,  
7688 payload).
- 7689 • [LOG] / [OBSERVATION] Samples of logs/telemetry/diagnostics under default and debug/payload-enabled  
7690 configurations, showing absence/presence of test secrets and payload as expected.

#### 7691 6.4.1.17.2 Assessment Case AC-CRS-DM-002

##### 7692 1. Assessment reference

7693 Requirement: REQ-CRS-DM-002 (Basic)

##### 7694 2. Assessment objective

7695 Verify that:

- 7696 1) The CRS provides administrative controls to disable or restrict collection of specific categories of log,  
7697 telemetry and runtime metric data.
- 7698 2) These controls allow administrators to disable or limit particular log types, telemetry streams, metric sets, or  
7699 debug payload capture to align data collection with operational and regulatory requirements.

##### 7700 3. Assessment preparation

7701 The assessment shall have access to:

- 7702 • Documentation describing configurable categories of logs (e.g. audit vs debug), telemetry streams, metric  
7703 groups and any debug/payload capture options.
- 7704 • Documentation of administrative roles/permissions required to modify data collection configuration.
- 7705 • A test configuration where different log/telemetry/metric categories can be toggled and where events can be  
7706 generated to exercise each category.

##### 7707 4. Assessment activities

7708 The assessment shall at least:

- 7709 1) Identify, from documentation and current configuration, the distinct categories controllable by administrators  
7710 (e.g. specific log types, telemetry streams, metric collections, payload capture).
- 7711 2) Using administrative controls, configure one or more categories (e.g. a debug log type, a telemetry stream, a  
7712 metrics group, or payload capture) to be disabled or restricted.
- 7713 3) Generate CRS and workload activity that would normally produce data in those categories and verify that:  
7714 - no new records are produced for disabled categories; and  
7715 - non-disabled categories continue to produce data as configured.
- 7716 4) Re-enable one or more previously disabled categories and verify that data for those categories is again  
7717 produced when new events occur.

##### 7718 5. Assessment verdict

- 7719 • Pass: administrators can selectively disable or restrict specific log types, telemetry streams, metrics and any  
7720 debug payload capture; disabled categories stop producing data while others continue; and re-enabling restores  
7721 data collection as configured.
- 7722 • Fail: data collection cannot be selectively controlled; or categories configured as disabled continue to produce  
7723 data.

##### 7724 6. Assessment evidence

- 7725 • [CONFIG] CRS data-collection configuration before and after disabling/re-enabling specific  
7726 log/telemetry/metric/payload categories.

- 7727 • [DESIGN] Documentation of administrative controls and category granularity for data collection, including  
7728 how these support operational/regulatory alignment.
- 7729 • [LOG] / [OBSERVATION] Evidence showing presence/absence of data from selected categories when  
7730 enabled vs disabled, demonstrating effective restriction of collection.

## 7731 6.4.2 CE

7732 [To be completed in a future version]

## 7733 6.4.3 CO

7734 [To be completed in a future version]

---

7735 **Annex A (informative):**  
7736 **Relationship between the present document and the**  
7737 **requirements of EU Regulation (EU) 2024/2847 - the**  
7738 **Cyber Resilience Act**

7739 The present document has been prepared under the Commission's standardisation request C(2025)618 [i.3] to provide  
7740 one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and  
7741 of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and  
7742 amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act)  
7743 [i.1]

7744 Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance  
7745 with the normative clauses of the present document given in Table A.1 confers, within the limits of the scope of the  
7746 present document, a presumption of conformity with the corresponding requirements of that Regulation and associated  
7747 EFTA regulations.

7748 **Table A.1: Relationship between the present document and**  
7749 **the requirements of Regulation (EU) 2024/2847 - the Cyber Resilience Act**

No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
1	The design, development, and production of products with digital elements ensures an appropriate level of cybersecurity based on the risks.	Annex I, Part I, (1)	Hypervisor: 5.1.1.1 5.1.1.2 5.1.1.3 5.1.1.4 5.1.1.5 5.1.1.6  CRS: 5.2.1.1 5.2.1.2 5.2.1.3 5.2.1.4 5.2.1.5 5.2.1.6	U	-
2	Products with digital elements are made available on the market without known exploitable vulnerabilities.	Annex I, Part I, (2)(a)	Hypervisor: 5.1.1.2 5.1.1.8 5.1.1.9 5.1.1.10  CRS: 5.2.1.2 5.2.1.8	U	

No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
			5.2.1.9 5.2.1.10		
3	Products with digital elements are made available on the market with a secure by default configuration, unless otherwise agreed between manufacturer and business user in relation to a tailor-made product with digital elements, including the possibility to reset the product to its original state.	Annex I, Part I, (2)(b)	Hypervisor: 5.1.1.9 5.1.1.10  CRS: 5.2.1.9 5.2.1.10	U	
4	Products with digital elements ensure that vulnerabilities can be addressed through security updates, including, where applicable, through automatic security updates that are installed within an appropriate timeframe enabled as a default setting, with a clear and easy-to-use opt-out mechanism, through the notification of available updates to users, and the option to temporarily postpone them.	Annex I, Part I, (2)(c)	Hypervisor: 5.1.1.8  CRS: 5.2.1.8	U	
5	Products with digital elements ensure protection from Unauthorized access by appropriate control mechanisms, including but not limited to authentication, identity or access management systems, and report on possible Unauthorized access.	Annex I, Part I, (2)(d)	Hypervisor: 5.1.1.3 5.1.1.4  CRS: 5.2.1.3 5.2.1.4	U	
6	Products with digital elements protect the confidentiality of stored, transmitted or otherwise processed data, personal or other, such as by encrypting relevant data at rest or in transit by state of the art mechanisms, and by using other technical means.	Annex I, Part I, (2)(e)	Hypervisor: 5.1.1.5  CRS: 5.2.1.5	U	
7	Products with digital elements protect the integrity of stored, transmitted or otherwise processed data, personal or other, commands, programs and configuration against any manipulation or modification not	Annex I, Part I, (2)(f)	Hypervisor: 5.1.1.2 5.1.1.7  CRS: 5.2.1.2 5.2.1.7	U	

No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
	authorized by the user, and report on corruptions.				
8	Products with digital elements process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital elements (data minimization).	Annex I, Part I, (2)(g)	Hypervisor: 5.1.1.11  CRS: 5.2.1.11	U	
9	Products with digital elements protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks.	Annex I, Part I, (2)(h)	Hypervisor: 5.1.1.6  CRS: 5.2.1.6	U	
10	Products with digital elements minimize the negative impact by the products themselves or connected products on the availability of services provided by other products or networks.	Annex I, Part I, (2)(i)	Hypervisor: 5.1.1.1 5.1.1.6  CRS: 5.2.1.1 5.2.1.6	U	
11	Products with digital elements are designed, developed and produced to limit attack surfaces, including external interfaces.	Annex I, Part I, (2)(j)	Hypervisor: 5.1.1.1 5.1.1.10  CRS: 5.2.1.1 5.2.1.10	U	
12	Products with digital elements are designed, developed and produced to reduce the impact of an incident using appropriate exploitation mitigation mechanisms and techniques.	Annex I, Part I, (2)(k)	Hypervisor: 5.1.1.1 5.1.1.2 5.1.1.6  CRS: 5.2.1.1 5.2.1.2 5.2.1.6	U	
13	Products with digital elements provide security related information by recording and monitoring relevant internal activity, including the access to or modification of data, services or functions, with an opt-out mechanism for the user.	Annex I, Part I, (2)(l)	Hypervisor: 5.1.1.7 5.1.1.11  CRS: 5.2.1.7 5.2.1.11	U	
14	Products with digital elements provide the possibility for users to securely and easily remove on a permanent basis all data and settings and, where such data can be transferred to other products or systems, ensure that this is done in a secure manner.	Annex I, Part I, (2)(m)	Hypervisor: 5.1.1.5 5.1.1.9  CRS: 5.2.1.5 5.2.1.9	U	

No	Description	Requirements of Regulation	Clause(s) of the present document	U/C	Condition
...	CRA Annex I - Part II: Vulnerability Handling Requirements		5.3		

7750

7751 **Key to columns:**7752 **Requirement:**7753 **No** A unique identifier for one row of the table which may be used to identify a requirement.7754 **Description** A textual reference to the requirement.7755 **Requirements of Regulation**

7756 Identification of article(s) defining the requirement in the Regulation.

7757 **Clause(s) of the present document**7758 Identification of clause(s) defining the requirement in the present document unless another  
7759 document is referenced explicitly.7760 **Requirement Conditionality:**7761 **U/C** Indicates whether the requirement is unconditionally applicable (U) or is conditional upon the  
7762 manufacturer's claimed functionality of the equipment (C).7763 **Condition** Explains the conditions when the requirement is or is not applicable for a requirement which is  
7764 classified "conditional".7765 Presumption of conformity stays valid only as long as a reference to the present document is maintained in the list  
7766 published in the Official Journal of the European Union. Users of the present document should consult frequently the  
7767 latest list published in the Official Journal of the European Union.

7768 Other Union legislation may be applicable to the product(s) falling within the scope of the present document.

7769

## 7770 Annex B (informative): Security Analysis

### 7771 B.1 Overview

7772 This clause is informative and non-binding. It provides a risk assessment methodology for assessing the cybersecurity  
7773 risk of a product based on its intended use case and deployment environment. This approach helps determine the risk  
7774 level for various use cases, guiding the application of appropriate security requirements.

7775 Manufacturers can use this methodology as guidance or apply their own established risk assessment methodologies.

7776 The risk assessment methodology used herein is a semi-quantitative approach, combining qualitative analysis with  
7777 numerical scoring. It provides a framework for:

- 7778 • Identifying product cybersecurity threats.
- 7779 • Determining the likelihood and magnitude of impact if those threats materialize.
- 7780 • Evaluating the overall cybersecurity risk for the product within its specific use case and deployment  
7781 environment.

7782 This methodology aligns with the general principles for assessing product cybersecurity risk, as defined in  
7783 prEN 40000-1-2 [i.11].

### 7784 B.2 Threats

#### 7785 B.2.1 Overview

7786 Table B.2.1-1 maps identified cybersecurity threats to the specific components they affect within the scope of the  
7787 present document: Hypervisor Type I, Hypervisor Type II, Orchestration & Management System, Container  
7788 Orchestrator, Container Engine, and Container Runtime System.

7789 The first column lists the threat categories considered in the present document, grouped by their primary security  
7790 impact, such as interference or escape, integrity compromise, Unauthorized access, data disclosure, denial of service,  
7791 logging issues, malicious updates, insecure configurations, and supply chain compromise.

7792 Each threat is referenced by its unique Threat ID, ensuring traceability to its detailed description in the threat catalogue  
7793 in the following clauses. This mapping provides a clear overview of which threats are applicable to each component  
7794 type, serving as the foundation for the risk assessment and for defining the corresponding security requirements and  
7795 controls.

7796 **Table B.2.1-1: Threats to components mapping**

Threat	Hypervisor Type I	Hypervisor Type II	Orchestration & Management	Container Orchestrator	Container Engine	Container Runtime System

<b>Interference / Escape</b>	T-HYP1-INTF-ESC	T-HYP2-HOST-ABUSE	T-ORCH-CTRL-PLANE	T-CO-INTF-ESC	T-CE-INTF-ESC	T-CRS-INTF-ESC
<b>Integrity Compromise</b>	T-HYP1-INTEG-COMP	T-HYP2-INTEG-COMP	T-ORCH-INTEG-TRUST	T-CO-INTEG-TRUST	T-CE-INTEG-COMP	T-CRS-INTEG-COMP
<b>Unauthorized Admin Access</b>	T-HYP1-UNAUTH-ADMIN	T-HYP2-UNAUTH-ADMIN	T-ORCH-UNAUTH-ADMIN	T-CO-UNAUTH-ADMIN	T-CE-UNAUTH-ADMIN	T-CRS-UNAUTH-ADMIN
<b>Unauthorized Privileged Actions</b>	T-HYP1-UNAUTH-FUNC	T-HYP2-UNAUTH-FUNC	T-ORCH-UNAUTH-FUNC	T-CO-UNAUTH-FUNC	T-CE-UNAUTH-FUNC	T-CRS-UNAUTH-FUNC
<b>Sensitive Data Disclosure</b>	T-HYP1-DATA-DISC	T-HYP2-DATA-DISC	T-ORCH-DATA-DISC	T-CO-DATA-DISC	T-CE-DATA-DISC	T-CRS-DATA-DISC
<b>DoS via Resource Exhaustion</b>	T-HYP1-AVAIL-DOS	T-HYP2-AVAIL-DOS	T-ORCH-AVAIL-DOS	T-CO-AVAIL-DOS	T-CE-AVAIL-DOS	T-CRS-AVAIL-DOS
<b>Insufficient / Tampered Logging</b>	T-HYP1-LOG-INT	T-HYP2-LOG-INT	T-ORCH-LOG-INT	T-CO-LOG-INT	T-CE-LOG-INT	T-CRS-LOG-INT
<b>Malicious / Compromised Update</b>	T-HYP1-UPD-COMP	T-HYP2-UPD-COMP	T-ORCH-UPD-COMP	T-CO-UPD-COMP	T-CE-UPD-COMP	T-CRS-UPD-COMP
<b>Insecure Default and Changed Configurations</b>	T-HYP1-CFG-COMP	T-HYP2-CFG-COMP	T-ORCH-CFG-COMP	T-CO-CFG-COMP	T-CE-CFG-COMP	T-CRS-CFG-COMP
<b>Supply Chain Compromise</b>	T-HYP1-SC-COMP	T-HYP2-SC-COMP	T-ORCH-SC-COMP	T-CO-SC-COMP	T-CE-SC-COMP	T-CRS-SC-COMP
<b>Compromise via hardware-level or platform trust components</b>	T-ALL-HW-ACCESS					

7797

7798 **B.2.2 VES**7799 **B.2.2.1 Hyper Type I**7800 **T-HYP1-INTF-ESC:** Inter-VM interference and escape via hypervisor/host interfaces7801 A malicious VM exploits flaws in the hypervisor, device emulation, vSwitch, or shared hardware to read/alter another  
7802 VM's memory/state or escape to the hypervisor/root.7803 **T-HYP1-INTEG-COMP:** Integrity compromise of workloads or platform components7804 A malicious actor tampers with hypervisor binaries, kernel modules, configuration files, or VM images on the host  
7805 storage, injecting malicious code or disabling isolation/security controls. The compromised hypervisor continues to  
7806 operate without detection by higher-level systems, allowing the attacker to manipulate or monitor workloads.7807 **T-HYP1-UNAUTH-ADMIN:** Unauthorized access to Type-1 hypervisor management interfaces7808 An attacker exploits missing or weak authentication on the hypervisor's direct management interfaces (e.g. vSphere  
7809 Client, ESXi Shell, SSH, vendor-specific APIs). This may be due to default credentials, weak password policies,  
7810 outdated authentication protocols, or disabled MFA.7811 **T-HYP1-UNAUTH-FUNC:** Unauthorized privileged actions or data access on a Type-1 hypervisor7812 An attacker with valid but limited credentials (e.g. VM operator role) or a compromised low-privileged API token  
7813 exploits insufficient authorization enforcement on the hypervisor management plane to perform high-impact actions or  
7814 access restricted data.7815 **T-HYP1-DATA-DISC:** Unauthorized disclosure of sensitive data in a Type-1 hypervisor environment7816 An attacker exploits vulnerabilities, weak encryption, or misconfigurations in the hypervisor or its management plane to  
7817 gain Unauthorized access to sensitive data from virtual machines or platform resources.

- 7818 This may include data in use (e.g. reading contents directly from RAM or CPU registers of running workloads), data at  
7819 rest (e.g. accessing virtual disk images or configuration files), and data in transit (e.g. intercepting network traffic  
7820 between VMs).
- 7821 **T-HYP1-AVAIL-DOS:** DoS via resource exhaustion in a Type-1 hypervisor environment
- 7822 An attacker either from a malicious or compromised guest VM, or via external network/API access, consumes  
7823 excessive CPU, memory, I/O bandwidth, or network capacity on the hypervisor host, leading to degraded performance  
7824 or full unavailability of hosted VMs.
- 7825 This also includes scenarios involving resource overcommitment, where CPU steal time, memory contention, or shared  
7826 I/O bottlenecks impact the performance of co-located VMs, potentially leading to degraded availability even when  
7827 individual VM resource limits are not exceeded.
- 7828 **T-HYP1-LOG-INT:** Insufficient logging or log tampering in a Type-1 hypervisor environment
- 7829 An attacker with access to a guest VM or hypervisor management interface exploits insufficient logging, disabled audit  
7830 trails, or insecure log storage to hide malicious actions (e.g. VM configuration changes, hypervisor setting  
7831 modifications). They may also tamper with existing logs to erase evidence of compromise.
- 7832 **T-HYP1-UPD-COMP:** Malicious or compromised update in a Type-1 hypervisor environment
- 7833 An attacker injects, replaces, or replays a hypervisor firmware or software update to gain control over the Virtualization  
7834 layer. Because Type-1 hypervisors run directly on hardware, compromise here provides full access to all guest VMs and  
7835 host resources.
- 7836 **T-HYP1-CFG-COMP:** Insecure or Unauthorized configuration in a Type-1 Hypervisor
- 7837 An attacker exploits insecure default settings or Unauthorized/insecure configuration changes to the hypervisor (via  
7838 CLI, web UI, API, or configuration files). This can weaken isolation, expose management services, or disable  
7839 logging/security controls, leading to Unauthorized access, privilege escalation, or compromise of workloads.
- 7840 **T-HYP1-SC-COMP:** Supply chain compromise in a Type-1 Hypervisor
- 7841 An attacker compromises a dependency used in the hypervisor's build process, such as a low-level driver, cryptographic  
7842 library, or firmware, to inject a backdoor or exploitable vulnerability.
- 7843 Because the malicious code is baked into the hypervisor's core, every VM running on the host inherits the exposure,  
7844 giving the attacker a privileged foothold.
- 7845 **B.2.2.2 Hyper Type II**
- 7846 **T-HYP2-HOST-ABUSE:** Guest-to-host escalation via host OS dependency and shared services
- 7847 A malicious VM abuses the host OS attack surface to access host resources or other VMs.
- 7848 **T-HYP2-INTEG-COMP:** Integrity compromise in hosted Virtualization
- 7849 An attacker compromises the host OS or hypervisor application binaries/configuration, or tampers with VM images in  
7850 the host filesystem. Because Type-2 relies on the host OS, compromise of the host environment compromises the  
7851 Virtualization layer and workloads.
- 7852 **T-HYP2-UNAUTH-ADMIN:** Unauthorized access to Type-2 hypervisor or host OS admin functions
- 7853 An attacker exploits weak authentication on the hypervisor application or on the host OS's administrative accounts.  
7854 Since Type-2 hypervisors run on a general-purpose OS, compromise of the host admin account grants full control over  
7855 the hypervisor and all guest VMs.
- 7856 **T-HYP2-UNAUTH-FUNC:** Unauthorized privileged actions or data access on a Type-2 hypervisor
- 7857 An attacker with authenticated but restricted access to the hypervisor application or underlying OS exploits inadequate  
7858 authorization to perform actions reserved for administrators. Since Type-2 hypervisors rely on the host OS for resource  
7859 access, weak separation can also allow escalation to host-level privileges.
- 7860 **T-HYP2-DATA-DISC:** Unauthorized disclosure of sensitive data in a Type-2 hypervisor environment

- 7861 An attacker leverages weaknesses in the hosted hypervisor or host OS to read sensitive data belonging to VMs or the  
7862 platform.
- 7863 This may include data in use (e.g. extracting secrets from process memory), data at rest (e.g. accessing VM image files  
7864 stored on the host), and data in transit (e.g. capturing inter-VM or host-VM communications).
- 7865 **T-HYP2-AVAIL-DOS:** DoS via resource exhaustion in a Type-2 hypervisor environment
- 7866 An attacker running on a guest VM or the host OS consumes excessive system resources (CPU, memory, storage,  
7867 network), causing other VMs or the hypervisor process to stall or crash.
- 7868 **T-HYP2-LOG-INT:** Insufficient logging or log tampering in a Type-2 hypervisor environment
- 7869 An attacker running on the host OS or inside a VM exploits weak logging practices to erase or manipulate event records  
7870 (e.g. VM lifecycle events, admin actions) to avoid detection. In a Type-2 setup, logs are often stored in user-accessible  
7871 locations on the host, making them easier to modify.
- 7872 **T-HYP2-UPD-COMP:** Malicious or compromised update in a Type-2 hypervisor environment
- 7873 An attacker injects or manipulates updates to the hypervisor application running on top of the host OS. Because these  
7874 updates often come via user-initiated downloads, attackers may deliver installers that have been modified with  
7875 malicious code or downgrade the software to a vulnerable version.
- 7876 **T-HYP2-CFG-COMP:** Unauthorized or insecure configuration in a Type-2 hypervisor
- 7877 A Type-2 hypervisor may be exposed to insecure configuration either through Unauthorized or insecure modifications  
7878 of its parameters via the host operating system environment, local malware, or its application interface, or through  
7879 insecure defaults inherited from the host operating system or shipped by the hypervisor itself. Examples include  
7880 permissive VM networking modes, host-guest shared folders enabled by default, or control channels without  
7881 authentication. Such insecure or Unauthorized configurations can weaken VM isolation, expose the host system, or  
7882 allow privilege escalation between guest VMs and the host.
- 7883 **T-HYP2-SC-COMP:** Supply chain compromise in a Type-2 Hypervisor
- 7884 An attacker compromises a dependency used by the Type-2 hypervisor, such as a driver, shared library, system  
7885 component, or packaged module, during development, upstream distribution, or packaging.
- 7886 When the hypervisor is installed on a user's OS, the malicious dependency executes with host-level privileges, allowing  
7887 the attacker to compromise both the hypervisor and the host OS.
- 7888 **B.2.2.3      Orchestration and Management system**
- 7889 **T-ORCH-CTRL-PLANE:** Control-plane abuse leading to cross-workload interference and privilege escalation
- 7890 An attacker with (or who obtains) control-plane/API access manipulates scheduling, networking, storage or policies to  
7891 read/impact other tenants' workloads or to break isolation at scale.
- 7892 **T-ORCH-INTEG-TRUST:** Misplaced trust in compromised platform
- 7893 An orchestrator and management system treats a hypervisor host as trustworthy without verifying its integrity (no  
7894 attestation, or attestation ignored). A compromised host can feed falsified state/configuration data back to the  
7895 orchestrator and run tampered workloads while appearing legitimate.
- 7896 **T-ORCH-UNAUTH-ADMIN:** Unauthorized access to orchestration/management plane
- 7897 An attacker gains Unauthorized access to the orchestration platform's administrative APIs, dashboards, or CLI tools  
7898 through weak authentication, missing identity federation controls, or compromised admin accounts. This provides  
7899 cluster-wide control of all connected hypervisors and workloads.
- 7900 **T-ORCH-UNAUTH-FUNC:** Unauthorized privileged actions or data access in orchestration/management systems
- 7901 An attacker with authenticated but restricted access to the orchestration platform exploits insufficient authorization on  
7902 APIs or dashboards to invoke privileged actions cluster-wide or access sensitive data.
- 7903 **T-ORCH-DATA-DISC:** Unauthorized disclosure of sensitive data via Virtualization orchestration/management system

- 7904 An attacker abuses vulnerabilities, excessive privileges, or misconfigured access controls in the orchestration platform  
7905 to obtain sensitive data from managed VMs or platform resources.
- 7906 This may include data in use (e.g. retrieving memory dumps from running workloads), data at rest (e.g. accessing stored  
7907 VM snapshots or configuration files), and data in transit (e.g. intercepting orchestrator-mediated traffic).
- 7908 **T-ORCH-AVAIL-DOS:** DoS via resource exhaustion in Virtualization orchestration/management systems
- 7909 An attacker overwhelms the orchestration control plane, APIs, or schedulers with excessive requests, malformed  
7910 workload definitions, or frequent provisioning/de-provisioning cycles. This can lead to unresponsiveness of  
7911 management tools and inability to operate workloads.
- 7912 **T-ORCH-LOG-INT:** Insufficient logging or log tampering in Virtualization orchestration/management systems
- 7913 An attacker targets the orchestration system's audit and logging functions to hide or falsify records of administrative  
7914 actions, API calls, or workload lifecycle events. These systems often serve as the central source of truth, so tampering  
7915 can mislead both human operators and automated security processes.
- 7916 **T-ORCH-UPD-COMP:** Malicious or compromised update in Virtualization orchestration/management systems
- 7917 An attacker tampers with updates for the orchestration platform or its plugins/extensions. A compromised orchestration  
7918 system can push malicious configurations or updates to all managed hypervisors and workloads, multiplying the impact.
- 7919 **T-ORCH-CFG-COMP:** Unauthorized or insecure configuration in Virtualization orchestration/management systems
- 7920 An attacker exploits insecure default settings (such as default administrator accounts, weak API tokens, unencrypted  
7921 management channels, or overly permissive role assignments) or performs Unauthorized modifications through the  
7922 orchestration system's management interfaces or APIs. Because the orchestration system controls multiple hypervisors  
7923 and cluster-wide policies, insecure defaults or misconfigurations can propagate across the entire environment,  
7924 weakening isolation, exposing management services, or granting attackers elevated privileges.
- 7925 **T-ORCH-SC-COMP:** Supply chain compromise in Virtualization orchestration/management systems
- 7926 An attacker injects malicious code into a dependency (for example, a module, API library, or plugin) used in the  
7927 orchestrator's control plane. Because orchestrators have administrative access to all managed hosts, the compromise  
7928 cascades to the entire infrastructure once deployed.
- 7929 **B.2.3 CES**
- 7930 **B.2.3.1 Container Orchestrator**
- 7931 **T-CO-INTF-ESC:** Orchestrator-level cross-workload interference
- 7932 An attacker with access to orchestration APIs or permissions in one namespace/project abuses orchestration  
7933 misconfigurations, policy gaps, or network segmentation flaws to interfere with workloads managed by the orchestrator.
- 7934 **T-CO-INTEG-TRUST:** Orchestrator trusts compromised nodes or components
- 7935 The orchestrator schedules workloads onto a container host (node) or engine whose binaries, configuration, or runtime  
7936 state have been tampered with, without verifying its integrity. This may be due to the absence of attestation, or  
7937 attestation being ignored or spoofed.
- 7938 **T-CO-UNAUTH-ADMIN:** Unauthorized access to orchestrator management plane
- 7939 An attacker gains access to the orchestrator's administrative interfaces (web dashboards, CLI tools, APIs) due to weak  
7940 or missing authentication controls, default credentials, or compromised admin accounts. Since the orchestrator manages  
7941 scheduling, scaling, and policy enforcement for all workloads, compromise can result in complete cluster control.
- 7942 **T-CO-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the orchestrator
- 7943 An attacker with valid but low-privileged credentials or a compromised service account exploits insufficient  
7944 authorization enforcement in the orchestrator's API server, CLI tools, or dashboards to perform privileged actions or  
7945 read restricted data.
- 7946 **T-CO-DATA-DISC:** Unauthorized disclosure of sensitive data via the container orchestrator

- 7947 An attacker exploits vulnerabilities, excessive privileges, or misconfigured access controls in the orchestrator's API  
7948 server, dashboard, or CLI to obtain sensitive data from managed workloads or control-plane resources.
- 7949 This may include data in use (e.g. reading container memory or runtime secrets), data at rest (e.g. accessing container  
7950 images, volumes, or configuration files), and data in transit (e.g. capturing pod-to-pod or service communications).
- 7951 **T-CO-AVAIL-DOS:** DoS via resource exhaustion in the container orchestrator
- 7952 An attacker overwhelms the orchestrator's control plane, APIs, or schedulers with excessive requests, malformed  
7953 deployment manifests, or frequent scaling operations. This can lead to delays or complete unavailability in workload  
7954 scheduling, management, and monitoring.
- 7955 **T-CO-LOG-INT:** Insufficient logging or log tampering in the Container Orchestrator
- 7956 An attacker targets the orchestrator's event and audit logging systems to suppress or alter records of critical cluster  
7957 events, administrative actions, or API calls. This can allow malicious changes—such as Unauthorized deployments,  
7958 RBAC policy modifications, or workload scheduling—to go undetected.
- 7959 **T-CO-UPD-COMP:** Malicious or compromised update in the Container Orchestrator
- 7960 An attacker injects, tampers with, or replays updates to the orchestrator's control plane components (API server,  
7961 scheduler, controller manager) or its plugins/extensions.
- 7962 A compromised orchestrator update could allow the attacker to modify workload scheduling, bypass RBAC rules, or  
7963 deploy malicious workloads across all managed nodes.
- 7964 **T-CO-CFG-COMP:** Unauthorized or insecure configuration in the Container Orchestrator
- 7965 An attacker exploits insecure default settings (such as anonymous API access, default administrator accounts,  
7966 unrestricted inter-pod communication, or permissive admission controller policies) or performs Unauthorized  
7967 modifications of orchestrator configurations through its management interfaces (API server, dashboard, CLI) or  
7968 configuration stores. Such weaknesses can compromise workload isolation, weaken network segmentation, bypass  
7969 RBAC enforcement, or allow unverified workloads to run. Because the orchestrator governs multiple containers and  
7970 services, insecure or Unauthorized configurations can rapidly escalate into cluster-wide compromise or lateral  
7971 movement between workloads. When orchestrator-level configuration is propagated to container engines or container  
7972 runtimes, the configuration source and propagation path shall be authenticated and authorized to prevent Unauthorized  
7973 configuration changes from cascading across the environment.
- 7974 **T-CO-SC-COMP:** Supply chain compromise in a Container Orchestrator
- 7975 An attacker injects malicious code into a dependency (for example, a module, plugin, or API library) used in the  
7976 orchestrator's control plane.
- 7977 Because the orchestrator controls all cluster workloads and nodes, the compromise propagates to the entire  
7978 environment.
- 7979 **B.2.3.2 Container Engine**
- 7980 **T-CE-INTF-ESC:** Engine-level cross-container interference
- 7981 A malicious container exploits weaknesses in engine-level isolation enforcement (configuration, API exposure, or  
7982 plugin vulnerabilities) to interfere with other containers running under the same engine.
- 7983 **T-CE-INTEG-COMP:** Integrity compromise of container engine and managed images
- 7984 An attacker tampers with the container engine's binaries, configuration files, or locally cached container images. The  
7985 compromised engine can then execute manipulated workloads or disable controls, while still appearing healthy to the  
7986 orchestrator.
- 7987 **T-CE-UNAUTH-ADMIN:** Unauthorized access to container engine control endpoints
- 7988 An attacker gains access to the container engine's local or remote control interfaces (e.g. Docker API socket, containerd  
7989 gRPC API) due to weak authentication, unprotected sockets, or lack of TLS. This gives direct control over container  
7990 lifecycle operations and engine configuration.

- 7991 **T-CE-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the container engine
- 7992 An attacker with authenticated but restricted engine API access (local socket or remote endpoint) exploits weak  
7993 authorization to invoke high-risk operations beyond their intended role.
- 7994 **T-CE-DATA-DISC:** Unauthorized disclosure of sensitive data via the container engine
- 7995 An attacker gains Unauthorized access to container image layers, configuration files, or secrets stored locally on the  
7996 engine host due to weak encryption, insufficient file permissions, or misconfigured access to control sockets.
- 7997 This may include data in use (e.g. reading secrets from running container processes), data at rest (e.g. accessing local  
7998 container image storage), and data in transit (e.g. intercepting traffic between containers using the engine's default  
7999 network).
- 8000 **T-CE-AVAIL-DOS:** DoS via resource exhaustion in the container engine
- 8001 An attacker exploits the container engine's local API or socket to trigger excessive image pulls, container starts/stops, or  
8002 logging operations, consuming CPU, memory, disk I/O, or network bandwidth on the host.
- 8003 **T-CE-LOG-INT:** Insufficient logging or log tampering in the Container Engine
- 8004 An attacker with access to the container engine API or control socket manipulates lifecycle logs (start/stop events,  
8005 image pulls, error messages) to remove or alter records of their actions. Since the engine directly manages container  
8006 execution, these logs are often key for local forensics.
- 8007 **T-CE-UPD-COMP:** Malicious or compromised update in the Container Engine
- 8008 An attacker injects or tampers with updates to the container engine binaries or API services. Since the engine directly  
8009 interfaces with the host OS and launches containers, a compromised update can be used to execute arbitrary code with  
8010 elevated privileges.
- 8011 **T-CE-CFG-COMP:** Unauthorized or insecure configuration in the Container Engine
- 8012 An attacker exploits insecure default settings (such as an exposed API socket without TLS/authentication, unrestricted  
8013 default bridge networking, or permission to run privileged containers) or performs Unauthorized modifications to  
8014 container engine daemon or runtime configurations (e.g. Docker daemon JSON, container runtime parameters). Such  
8015 weaknesses can bypass container isolation, allow unsafe container operations, or provide a direct path to escalate from  
8016 container workloads to the host operating system. Because the container engine mediates between containers, the  
8017 container runtime stack (CRS), and the host OS, insecure or Unauthorized configurations can undermine both workload  
8018 security and host-level protections. If configuration changes originate from the orchestrator or external automation, the  
8019 container engine shall only accept configuration updates from authenticated and authorized sources to prevent  
8020 unintended cluster-wide propagation of insecure settings.
- 8021 **T-CE-SC-COMP:** Supply chain compromise in a Container Engine
- 8022 A dependency used by the CE, such as a registry client library, image build tool, or network driver, is compromised  
8023 during development or distribution.
- 8024 This malicious code can be triggered when the CE pulls or builds images, or when it sets up container networking.
- 8025 **B.2.3.3 Container Runtime System**
- 8026 **T-CRS-INTF-ESC:** Runtime-level container breakout
- 8027 A malicious container exploits flaws in OS-level isolation mechanisms or runtime implementation to escape its sandbox  
8028 and interfere directly with the host or peer containers.
- 8029 **T-CRS-INTEG-COMP:** Runtime-level tampering and unverified trust
- 8030 An attacker modifies runtime binaries, configuration, or container state data (namespaces, cgroups, mounts) to disable  
8031 or bypass security controls. Because the orchestrator and engine may not verify the runtime's state, a compromised  
8032 runtime can operate undetected.
- 8033 **T-CRS-UNAUTH-ADMIN:** Unauthorized access to container runtime control mechanisms

- 8034 An attacker directly interacts with the container runtime binary or its control interfaces, bypassing higher-level engine  
8035 or orchestrator authentication. This may occur if the runtime is invoked manually or through a compromised local  
8036 account.
- 8037 **T-CRS-UNAUTH-FUNC:** Unauthorized privileged operations or data access in the container runtime system
- 8038 An attacker with authenticated local or delegated runtime access exploits insufficient authorization to manipulate  
8039 running containers or modify runtime configurations directly.
- 8040 **T-CRS-DATA-DISC:** Unauthorized disclosure of sensitive data via the container runtime system
- 8041 An attacker exploits runtime-level weaknesses to read sensitive data from running containers or host resources that  
8042 should be isolated.
- 8043 This may include data in use (e.g. reading contents from container or host memory), data at rest (e.g. accessing mounted  
8044 volumes or container filesystem layers), and data in transit (e.g. intercepting inter-container or container-host  
8045 communications).
- 8046 **T-CRS-AVAIL-DOS:** DoS via resource exhaustion in the container runtime system
- 8047 An attacker running inside a container exploits runtime isolation weaknesses to monopolize CPU, memory, or I/O  
8048 resources, or floods inter-container or container-to-host communication channels to degrade performance.
- 8049 **T-CRS-LOG-INT:** Insufficient logging or log tampering in the Container Runtime System
- 8050 An attacker operating inside a container exploits insufficient runtime logging or insecure log handling to conceal  
8051 actions such as privilege escalation attempts, restricted syscalls, or file system access violations. The runtime's own log,  
8052 if tampered with, may give a false picture of container activity.
- 8053 **T-CRS-UPD-COMP:** Malicious or compromised update in the Container Runtime System
- 8054 An attacker injects, tampers with, or replays updates to the low-level runtime responsible for creating and managing  
8055 containers. Since the runtime enforces isolation (namespaces, cgroups, seccomp), a compromised update could remove  
8056 or weaken these controls.
- 8057 **T-CRS-CFG-COMP:** Unauthorized or insecure configuration in the Container Runtime System
- 8058 An attacker exploits insecure defaults (such as lack of enforced seccomp/AppArmor profiles, missing cgroup limits for  
8059 CPU, memory, or I/O, or permissive filesystem mounts that expose sensitive host paths) or introduces Unauthorized  
8060 changes to runtime-level parameters and configuration files. Such weaknesses undermine container isolation, disable  
8061 syscall filtering, weaken resource constraints, or allow containers to access sensitive host resources. Because the  
8062 container runtime system directly enforces workload isolation and resource governance, insecure or Unauthorized  
8063 configurations at this layer can enable container escapes, denial-of-service attacks, or privilege escalation against the  
8064 host. Where runtime configuration is applied automatically from a container engine or orchestrator, the runtime shall  
8065 ensure that such configuration inputs are authenticated and authorized before application, to prevent malicious or  
8066 unintended configuration propagation.
- 8067 **T-CRS-SC-COMP:** Supply chain compromise in a Container Runtime System
- 8068 An attacker compromises a low-level runtime dependency, such as a system call handling library, sandboxing module,  
8069 or container image unpacker, used by the CRS.
- 8070 The malicious code executes whenever the CRS starts or runs containers, potentially enabling container escape.

## 8071 **B.2.4 Hardware-level common threat**

### 8072 **T-ALL-HW-ACCESS:** Compromise via hardware-level or platform trust components

- 8073 An attacker with access to hardware-level capabilities (for example compromised CPU, firmware, hardware security  
8074 modules, or physical access to deployed systems) may compromise the hypervisor, container runtime system, or  
8075 associated platform components below the software control plane. Such attacks may exploit manufacturer backdoors in  
8076 processors or firmware, supply chain implants in platform components, or physical tampering vectors.

8077 Because hardware-level compromise can operate at privilege levels that exceed the visibility and control of software-  
 8078 based protection mechanisms, these attacks may bypass or undermine boot integrity verification, runtime integrity  
 8079 checks, attestation, and isolation mechanisms defined in the present document.

8080 The present document does not require products to fully prevent all forms of hardware-level compromise. Instead, for  
 8081 requirements that rely on platform-anchored, hardware-assisted, or externally protected trust mechanisms,  
 8082 manufacturers are expected to:

- 8083 • use available platform trust anchors and protection mechanisms where appropriate;
- 8084 • provide software-based or hybrid protections where feasible to achieve equivalent security outcomes; and
- 8085 • document trust assumptions and residual risks associated with hardware components, including the potential  
 8086 impact of compromise consistent with T-ALL-HW-ACCESS.

## 8087 B.3 Risk Assessment Process

8088 The risk assessment process focuses on evaluating the product within its defined use case and deployment environment,  
 8089 following a logical sequence of steps:

- 8090 1) Define the product and its context: Identify the product being assessed, its intended use case, and the specific  
 8091 deployment environment it is intended for. This provides the foundational context for the assessment.
- 8092 2) Identify product threats: Identify potential cybersecurity threats that could exploit vulnerabilities.

8093 For each identified threat, the following steps should be performed:

- 8094 3) Analyze risk factors for likelihood and impact as defined in clause B.5: This step determines the product's  
 8095 cybersecurity risk by evaluating its characteristics related to potential threats. For each identified threat, and  
 8096 for every relevant risk factor, the following guiding question should be applied: *"How does this factor  
 8097 influence the likelihood or impact if the threat were to occur?"*. Each threat is evaluated against a defined set  
 8098 of risk factors covering both Likelihood and Impact. These factors are scored on a three-level scale (Low = 1,  
 8099 Medium = 2, High = 3) using the definitions provided in clause B.4.

8100 The analysis involves the following sub-steps:

- 8101 - Identifying key risk factors: Determine the main aspects of the product's operation within that  
 8102 environment that influence its cybersecurity risk (e.g. connectivity, administrative context, configuration  
 8103 volatility, impact of compromise, integration with critical systems). These include factors that affect  
 8104 either:

- 8105 1) the likelihood of threat exploitation; or
- 8106 2) the severity of consequences if the threat materializes.

8107 These factors serve as proxies for the likelihood of a threat materializing and the magnitude of its impact.

- 8108 - Defining scoring: Assign numerical scores (1 for low risk, 2 for medium, 3 for high) to each risk factor  
 8109 based on how it influences the likelihood of a threat event and the magnitude of its potential impact.

- 8110 4) Derive overall likelihood score: Sum the scores of all risk factors identified as influencing "Likelihood". Map  
 8111 this total score to the Likelihood level (1-3) using the thresholds defined in clause B.5.2.

- 8112 5) Derive overall impact score: Sum the scores of all risk factors identified as influencing "Impact". Map this  
 8113 total score to the Impact level (1-3) using the thresholds defined in clause B.5.3.

- 8114 6) Determine final risk level: Using the derived overall Likelihood score and overall Impact score, consult the  
 8115 risk matrix (clause B.6) to determine the final, overall risk level (Low, Medium, High) for the specific use  
 8116 case.

- 8117 7) Document and review: Record all assessment findings, including the rationale for scores and the determined  
 8118 risk level.

## 8119 B.4 Risk Factor Scoring

8120 Table B.4-1 provides the detailed scoring rubric to be used for assessing cybersecurity risks of products within their use  
8121 cases and deployment environments. Each risk factor is explicitly linked to whether it primarily influences likelihood or  
8122 impact. The 1-3 ratings of the risk factors are summed per type (Likelihood or Impact) to derive the overall likelihood  
8123 and impact scores.

Table B.4-1: Risk factors

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
EXT	External Network Exposure	Likelihood	Measures how reachable the VES/CES is from untrusted networks, including public internet and cross-tenant networks. Higher exposure increases the probability of remote exploitation.	Internal-only access; management on isolated VLAN/bastion; strict firewall segmentation.	Limited/controlled external access (VPN, allow-list, jump host).	Public internet-facing endpoints or broad cross-tenant network paths.
TEN	Tenant Multiplicity & Trust	Likelihood	Captures the number of tenants and their trust level. Multi-tenancy with untrusted tenants increases the chance of malicious insiders or cross-tenant attacks.	Single trusted tenant.	Few semi-trusted tenants.	Many untrusted tenants (public/multi-organization).
CTL	Control-Plane Exposure, Access Control & Detectability	Likelihood	Evaluates the exposure, strength of authentication/authorization on management interfaces, and the extent to which malicious actions on the control plane would be detectable.	Admin access isolated, MFA + RBAC enforced, centralized logging alerts on anomalous admin activity.	Internal network access, standard RBAC, partial monitoring or periodic audits.	Externally reachable control plane, weak RBAC, or limited/no monitoring of administrative actions.
SUP	Update & Supply-Chain Assurance	Likelihood	Assesses the security of update mechanisms, software/image sources, and supply-chain validation. Weak assurance can allow malicious updates or images.	Signed/verified only; private mirrors; staged/canary updates.	Vendor-signed updates/images; occasional direct pulls.	Direct pulls from public registries/vendor repositories; unsigned allowed.
PRV	Attack Complexity & Privileged Footprint	Likelihood	Represents the attacker skill/resources required to exploit the privileged software surface. Higher privileged footprint or weaker hardening → lower attacker skill required.	Minimal attack surface, compiler + runtime hardening (ASLR, SMEP/SMAP/CET), requires advanced attacker capability.	Mixed hardening, moderate privileged surface, requires skilled attacker with tools.	Large privileged surface, device pass-through, weak/no hardening — exploitable by unsophisticated attackers.
CHG	Change/Workload Volatility	Impact	Evaluates how frequently the configuration and workloads change. Higher volatility increases the likelihood of misconfigurations or unpatched instances.	Static, rarely changes.	Periodic changes (weekly/monthly).	Highly dynamic (daily/hourly), autoscaling, frequent reconfiguration.
ISO	Isolation Breakout Consequence	Impact	Assesses the damage if the VM/container isolation boundary is broken. Higher values mean more severe lateral movement or system compromise.	No sensitive assets exposed; minimal lateral movement possible.	Exposure of internal systems/data; contained blast radius.	Cross-tenant or critical system compromise; admin plane reachable.

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
<b>DAT</b>	Data/Safety/Criticality	Impact	Measures the sensitivity of the data and any safety or regulatory implications of compromise. Higher scores indicate severe operational, safety, or compliance impacts.	Public or low-sensitivity data; no safety impact.	Confidential/business data; moderate operational disruption.	Regulated/PII/financial/eHealth data; safety hazard or critical infrastructure outage.
<b>ORB</b>	Orchestration Dependency & Blast Radius	Impact	Evaluates how much damage an orchestrator compromise could cause. Higher values reflect large-scale or critical workload control.	Stand-alone host; no orchestrator or read-only integration.	Basic orchestrator; tenant- or namespace-scoped control.	Full control of critical workloads across multi-node clusters/sites.
<b>DRC</b>	Detection & Recovery Capability	Impact	Measures the ability to detect, contain, and recover from a compromise. Poor detection/recovery increases the residual impact of an incident.	Centralized logging/alerting; immutable backups; tested recovery plans.	Adequate logging/backups; periodic recovery tests.	Minimal logging/alerting; weak or untested recovery procedures.

## B.5 Calculation and Mapping to Likelihood/Impact Levels

### B.5.1 General

This clause defines how the total scores from Table B.4-1 in clause B.4 are translated into the 1-3 Likelihood and Impact levels used in the risk matrix (clause B.6).

### B.5.2 Likelihood Score

Table B.4-1 in clause B.4 defines five Likelihood factors:

- EXT (External Network Exposure)
- TEN (Tenant Multiplicity & Trust)
- CTL (Control-Plane Exposure, Access Control & Detectability)
- SUP (Update & Supply-Chain Assurance)
- PRV (Privileged Footprint & Hardening)

Each factor is scored from 1 (Low) to 3 (High).

- Maximum possible Likelihood score: 5 factors  $\times$  3 = 15
- Minimum possible Likelihood score: 5 factors  $\times$  1 = 5

The resulting total score is mapped to three levels as follows:

**Table B.5.2-1: Likelihood score**

Total Likelihood Score Range	Likelihood Level	Description
5 - 8	Low (1)	The event is highly unlikely to occur under normal circumstances.
9 - 12	Medium (2)	The event may occur occasionally in enterprise or cloud deployments.
13 - 15	High (3)	The event is expected to occur or has a high probability of occurring.

### B.5.3 Impact Score

Table B.4-1 in clause B.4 defines five Impact factors:

- CHG (Change/Workload Volatility)
- ISO (Isolation Breakout Consequence)
- DAT (Data/Safety/Criticality)
- ORB (Orchestration Dependency & Blast Radius)
- DRC (Detection & Recovery Capability)

Each factor is scored from 1 (Low) to 3 (High).

- Maximum possible Impact score: 5 factors  $\times$  3 = 15
- Minimum possible Impact score: 5 factors  $\times$  1 = 5

The resulting total score is mapped to three levels as follows:

**Table B.5.3-1: Impact score**

Total Impact Score Range	Impact Level	Description
5 - 8	Low (1)	Minimal disruption. No significant loss of confidentiality, integrity, or availability. No impact on safety or critical infrastructure.
9 - 12	Medium (2)	Noticeable disruption requiring recovery. Moderate impact on confidentiality, integrity, or availability. Potential for indirect impact on safety or critical infrastructure.
13 - 15	High (3)	Severe disruption, service interruption, or data loss. Major impact on confidentiality, integrity, or availability. May directly affect safety or critical infrastructure.

## B.6 Risk Matrix

The final overall risk level is determined by combining the calculated likelihood level (clause B.5.2) and impact level (clause B.5.3) using the matrix below:

**Table B.6-1: Risk matrix**

Impact \ Likelihood	1 = Low	2 = Medium	3 = High
3 = High	Medium	High	High
2 = Medium	Low	Medium	High
1 = Low	Low	Low	Medium

## B.7 Risk Methodology Applied to Use Cases

### B.7.1 General

This clause illustrates the application of the methodology to products deployed within the VES use cases (UC-V1, UC-V2, UC-V3) and CES use cases (UC-C1, UC-C2, UC-C3). The use cases are defined in clause 4.7, while the corresponding threats applicable to these products are described in clause B.2.

### B.7.2 Risk Evaluation of VES use cases

#### B.7.2.1 Hypervisor Type I

Table B.7.2.1-1: Hypervisor Type I - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-HYP1-INTF-ESC: Inter-VM interference / VM escape</b>														
<b>V1</b>	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1 the Type-1 hypervisor runs on a dedicated, isolated host with a very small set of non-critical VMs, limited external connectivity (EXT=1, TEN=1) and strictly local administration. Even if a VM escape occurs, lateral movement remains confined to that single host and low-value workloads, so both likelihood and impact are assessed as Low.
<b>V2</b>	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, the Type-1 hypervisor is part of an enterprise cluster with more VMs per tenant, shared management networks and standard hardening (EXT=1, TEN=2, CTL=3). A successful VM escape would expose all VMs owned by that enterprise tenant on that cluster, making both the feasibility and the consequences noticeably higher than in UC-V1.
<b>V3</b>	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin multi-tenant critical clouds where untrusted tenants co-exist and regulated or safety-relevant workloads are hosted (TEN=3, DAT=3, ORB=3). A VM escape here enables cross-tenant compromise and direct access to high-criticality data, so both likelihood and impact are rated High.
<b>T-HYP1-INTEG-COMP: Integrity compromise of workloads / hypervisor components</b>														
<b>V1</b>	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	For UC-V1, tampering with hypervisor binaries or VM images typically requires local access to a single host; workloads are static and non-critical, and orchestration blast radius is limited (ORB=1). As a result, integrity compromise remains a localized issue with Low impact on operations.
<b>V2</b>	1	2	3	2	2	2	2	2	2	3	10 → Med	11 → Med	Medium	In UC-V2, hypervisor binaries, kernel modules and VM images are centrally managed for business workloads. If an attacker tampers with these components on an enterprise VES, they can gain persistent control over all VMs of that tenant (ISO=2, ORB=2) and manipulate business services, so both likelihood and impact are Medium.
<b>V3</b>	3	3	3	3	3	3	2	3	3	3	15 → High	14 → High	High	In UC-V3, a compromise of Type-1 hypervisor components (e.g. signed modules, microcode integrations) affects multi-tenant, high-assurance environments. Any persistent backdoor or binary tampering undermines the entire trust anchor for critical workloads

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														across tenants (ISO=2 but DAT=3, ORB=3), justifying High likelihood and High impact.
<b>T-HYP1-UNAUTH-ADMIN: Unauthorized access to management interfaces</b>														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1, direct admin interfaces of the Type-1 hypervisor are usually reachable only from an isolated management network or via local console, and the host runs non-critical VMs. Even if authentication is weak, the lack of broad network exposure and the low value of the workloads keep both likelihood and impact Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 management plane (e.g. vSphere, vendor APIs) is reachable from the enterprise network and is a high-value target (EXT=1, CTL=3). A compromise of admin credentials or interfaces gives full control over all enterprise VMs, but the effects are still contained within that tenant/domain, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 management APIs, even if segregated, are critical control points for multi-tenant and regulated workloads. Unauthorized access to these interfaces would allow massive cross-tenant operations, uncontrolled VM management, and potential breach of critical systems, warranting High likelihood and High impact.
<b>T-HYP1-UNAUTH-FUNC: Unauthorized privileged actions</b>														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	Within UC-V1, a misconfigured RBAC or over-permissive role on the Type-1 hypervisor can allow privilege abuse (e.g. VM snapshots, power-off) but only on a few local, non-critical VMs. Given strong isolation from external networks and the limited blast radius, both likelihood and impact remain Low.
V2	1	2	3	1	3	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, enterprise deployments may assign operator roles to multiple administrators or automation tools. If authorization checks are weak (CTL=3, PRV=3), a compromised low-privilege account can perform administrative actions (e.g. attach disks, modify vSwitches) across the tenant's VMs, making both likelihood and impact Medium.
V3	2	3	2	3	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, weak or inconsistent authorization on the Type-1 management plane allows a compromised operator account to perform privileged actions affecting many untrusted tenants (TEN=3, ORB=3) and high-criticality workloads. This drives both likelihood and impact to High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-HYP1-DATA-DISC: Data disclosure</b>														
<b>V1</b>	1	1	2	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, Type-1 hypervisors host non-critical workloads with low-sensitivity data (DAT=1) and minimal connectivity. Even if hypervisor misconfigurations or weak encryption expose VM disks or inter-VM traffic, the resulting confidentiality impact is limited, so both likelihood and impact are Low.
<b>V2</b>	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 hypervisor stores and processes business data and may support backups or snapshots. Misconfigurations or vulnerabilities leading to access to VM images or memory dumps would significantly affect the enterprise tenant, but not multiple independent tenants, leading to Medium risk.
<b>V3</b>	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin clouds handling regulated and critical workloads (e.g. eHealth, finance, telecom core). A breakout that accesses VM RAM, disks, or east-west traffic can disclose highly sensitive multi-tenant data (DAT=3, ISO=3, ORB=3), resulting in High likelihood and High impact.
<b>T-HYP1-AVAIL-DOS: DoS via resource exhaustion</b>														
<b>V1</b>	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, resource exhaustion attacks generally require local or tightly scoped access, target a small number of VMs, and affect functions that are not business- or safety-critical. Even if CPU, memory or I/O are saturated, only a single host and low-value workloads are impacted, so likelihood and impact are Low.
<b>V2</b>	1	2	2	2	2	2	2	2	2	2	9 → Med	10 → Med	Medium	For UC-V2, noisy-neighbour effects or malicious workloads can cause CPU steal, memory pressure, or shared-I/O bottlenecks on Type-1 hosts that carry many enterprise workloads. While blast radius is still tenant-scoped, such events can cause notable service disruption (CHG=2, ORB=2), driving likelihood and impact to Medium.
<b>V3</b>	2	3	3	2	3	3	3	2	3	2	13 → High	13 → High	High	In UC-V3, shared Type-1 hypervisors run heterogeneous, often untrusted tenants at high density. Misconfigured overcommitment or deliberate resource exhaustion can create systemic availability issues across many critical services (TEN=3, CHG=3, ORB=3). Therefore, both likelihood and impact are assessed as High.
<b>T-HYP1-LOG-INT: Insufficient logging / log tampering</b>														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	Low	In UC-V1, logging on the Type-1 hypervisor may be minimal, and local administrators or malware could clear or modify logs. However, the environment has low data sensitivity and limited orchestration (DAT=1, ORB=1), so the inability to trace events has a Low overall impact.
V2	1	2	2	2	2	2	1	2	2	2	9 → Med	9 → Med	Medium	In UC-V2, insufficient audit trails on hypervisor admin actions, VM lifecycle events, and configuration changes delay detection of targeted attacks on business workloads. Enterprise security monitoring exists but may not fully cover hypervisor logs (DRC=2), so both likelihood and impact are Medium.
V3	2	3	3	2	3	2	2	3	3	3	13 → Med	13 → High	High	In UC-V3, gaps or tampering in Type-1 logging and audit mechanisms seriously undermine forensic readiness and incident containment. In multi-tenant critical clouds, missing or untrustworthy logs mean that cross-tenant attacks may remain undetected and unbounded in time, resulting in High likelihood but High impact.
<b>T-HYP1-UPD-COMP: Malicious or compromised update</b>														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the Type-1 hypervisor may be done manually or infrequently, sometimes from offline media or vendor repositories. A compromised image would affect a single host with non-critical workloads, and some manual verification is often possible, so both likelihood and impact stay Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, centralized yet sometimes ad-hoc update procedures (SUP=2) mean that delayed or inconsistent patching can allow malicious or tampered hypervisor updates to be applied across an enterprise cluster. This can compromise all VMs owned by that tenant, giving Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-1 hypervisors rely on highly automated, large-scale update pipelines. A compromise of the update supply chain or signing infrastructure would propagate to many critical hosts simultaneously (SUP=3, ORB=3), making both likelihood and impact High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-HYP1-CFG-COMP: Insecure or Unauthorized configuration</b>														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigurations (e.g. leaving a debug interface enabled or using permissive vSwitch settings) are usually applied locally and affect a small, static VM set. The environment is not exposed to untrusted tenants, so while isolation can be weakened, the overall risk remains Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	Medium	For UC-V2, centralized management of many VMs introduces more frequent changes and complex configuration (CHG=2). Errors such as exposing management services, disabling security features or misconfiguring virtual networks can expose sensitive enterprise workloads, leading to Medium likelihood and Medium impact.
V3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, configuration baselines have to enforce strong isolation, least privilege, and hardened defaults. Insecure or Unauthorized changes (e.g. tenant-visible management networks, overly permissive resource mappings) can immediately affect multiple critical tenants and services, so both likelihood and impact are High.
<b>T-HYP1-SC-COMP: Supply chain compromise</b>														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a compromised driver, library, or firmware component used by the Type-1 hypervisor yields a backdoor on one isolated host. Since workloads are low-value and connectivity is limited, the damage is confined, so likelihood and impact are Low.
V2	2	2	2	3	2	2	2	2	2	2	11 → Med	10 → Med	Medium	In UC-V2, the Type-1 stack depends on a broader supply chain (Virtualization platform, OS components, drivers). A poisoned component can introduce persistent backdoors affecting all enterprise VMs on the cluster, but still within one tenant domain, resulting in Medium likelihood and Medium impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, supply-chain compromise of hypervisor components (e.g. low-level drivers or cryptographic libraries) scales across many tenants and critical services. Every VM on the affected hosts inherits the compromised code path, giving the attacker a privileged foothold with High likelihood and High impact.

## B.7.2.2 Hypervisor Type II

Table B.7.2.2-1: Hypervisor Type II - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-HYP2-HOST-ABUSE: Guest-to-host escalation</b>														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a Type-2 hypervisor runs on a single host OS with few non-critical VMs and limited services exposed. An attacker in a guest can theoretically abuse host APIs or shared services, but low exposure and low data value keep both likelihood and impact Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, the host OS provides shared services (e.g. file shares, clipboard, device redirection) to multiple enterprise VMs. Weak isolation or host hardening allows a malicious VM to escalate into the host OS, from which it can access all internal workloads on that machine, giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, hosted Virtualization may underpin critical or multi-tenant workloads; a successful guest-to-host escalation immediately compromises the host OS and all VMs it controls (ISO=3, DAT=3, ORB=3). Given the large attack surface of general-purpose OSes, both likelihood and impact are High.
<b>T-HYP2-INTEG-COMP: Integrity compromise of hosted hypervisor</b>														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-V1, tampering with the Type-2 hypervisor binary or VM images on the host filesystem typically requires local access and affects only non-critical workloads. The blast radius is limited to one host, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, compromise of host OS or hypervisor application binaries directly affects the integrity of all VMs on that host and may propagate through shared images or templates within the enterprise. However, the effect is still scoped to one internal tenant environment, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, hosted Virtualization is used for high-assurance workloads; any manipulation of binaries, modules or VM images via the host OS or packaging system undermines trust in multiple critical services. This justifies High likelihood and High impact.
<b>T-HYP2-UNAUTH-ADMIN: Unauthorized admin access</b>														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-V1, the host OS admin account and the Type-2 hypervisor console are typically accessible only locally or on a restricted network, and control non-critical VMs. Even if passwords are weak, compromise remains local to a single device, making both likelihood and impact Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	For UC-V2, host OS and hypervisor admin interfaces are reachable from the enterprise network and control multiple business VMs (CTL=3). Unauthorized access to these accounts enables full control over internal workloads, but still within one enterprise boundary, so the risk is Medium.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, internet-reachable or widely exposed host OS admin accounts and Type-2 management consoles represent a systemic control point. Their compromise yields administrative control over critical or regulated workloads and potentially multiple tenants, so both likelihood and impact are High.
<b>T-HYP2-UNAUTH-FUNC: Unauthorized privileged actions</b>														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigured permissions in the Type-2 application or host OS may allow local users to perform limited privileged actions, but the small number of non-critical VMs and limited connectivity keep the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprise deployments often have multiple operators or automation agents interacting with the hypervisor application. Weak separation of roles or coarse-grained permissions allow Unauthorized privileged actions that affect many internal workloads (PRV=2, ISO=2), giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, insufficient authorization controls in the Type-2 hypervisor or host OS allow a compromised user or process to perform high-impact operations (e.g. mount host paths, change networking, access snapshots) across critical tenants. This drives both likelihood and impact to High.
<b>T-HYP2-DATA-DISC: Sensitive data disclosure</b>														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, VMs on a hosted hypervisor process low-sensitivity data and are often isolated physically or logically. Even if an attacker

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														reads VM image files or host-mediated traffic, the confidentiality impact is minimal, resulting in Low likelihood and Low impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, a Type-2 hypervisor can host enterprise workloads whose VM image files, snapshots, or memory may contain confidential business data. Exploiting host OS weaknesses or hypervisor flaws to access these artefacts affects the enterprise tenant significantly, but not other independent tenants, leading to Medium risk.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-2 environments may handle highly sensitive or regulated data on general-purpose OSes. If an attacker gains access to VM images, RAM, or inter-VM traffic via the host, multiple critical services and tenants can be affected, so both likelihood and impact are High.
<b>T-HYP2-AVAIL-DOS: Resource exhaustion</b>														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, a DoS caused by a single VM or host-level process consuming CPU, memory, or I/O affects only a standalone host with non-critical workloads, yielding Low likelihood and impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	For UC-V2, resource exhaustion on a host OS running a Type-2 hypervisor can disrupt many internal business VMs (e.g. via runaway processes or misconfigured resource limits). However, the effect is still constrained to one enterprise environment, so likelihood and impact are Medium.
V3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-V3, hosted Virtualization may carry critical services or multi-tenant workloads; noisy-neighbour effects or deliberate resource abuse at host level can degrade or interrupt many high-criticality services simultaneously (CHG=3, ORB=3). This justifies High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-HYP2-LOG-INT: Logging weakness/tampering</b>														
V1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, logs for the Type-2 hypervisor and host OS are often stored locally and may not be centrally monitored. While this allows some tampering, the low-sensitivity nature of the workloads limits the overall impact, so risk is Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, weak or inconsistent logging of VM lifecycle events, admin actions, or host OS changes makes enterprise incident detection slower. Attackers can persist longer within internal workloads (DRC=2), which supports Medium likelihood and impact.
V3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, insufficient or tamperable logs on the host OS and Type-2 hypervisor significantly weaken forensic and recovery capabilities across critical workloads. Multi-tenant or regulated services then lack reliable evidence of compromise, resulting in High likelihood and High impact.
<b>T-HYP2-UPD-COMP: Malicious/compromised update</b>														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the hosted hypervisor application are often done manually or automatically by local admins. A malicious installer can compromise that single host and its few non-critical VMs, but the limited scale and occasional update frequency make the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, admins may download and install Type-2 hypervisor updates directly from vendors or repositories on multiple enterprise hosts. A poisoned installer or downgraded package can compromise many internal workloads, so both likelihood and impact are Medium.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, automated or large-scale update processes for hosted hypervisors and host OS components mean a single compromised package or repository can propagate quickly to many critical nodes (SUP=3, ORB=3). This supports High likelihood and High impact.
<b>T-HYP2-CFG-COMP: Insecure or Unauthorized configuration</b>														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, insecure defaults (e.g. permissive host-guest shared folders, bridged networking) or local misconfigurations affect only a small number of non-critical VMs. The lack of multi-tenancy and external exposure keeps likelihood and impact Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-V2, Type-2 hypervisors on enterprise hosts may be configured with convenient but insecure settings (e.g. broad host path mappings, weak isolation between lab and production VMs). These configurations enable lateral movement and Unauthorized access between internal workloads, giving Medium risk.
V3	2	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-V3, insecure Type-2 configurations (such as open shared folders, permissive host networking, or control channels without authentication) directly break isolation between high-assurance tenants and critical services. Given the strong reliance on configuration correctness, both likelihood and impact are High.
<b>T-HYP2-SC-COMP: Supply chain compromise</b>														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a malicious driver or library used by the Type-2 hypervisor executes only on a single, isolated host. While it provides a backdoor to that system, the limited connectivity and low-value workloads constrain the impact, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprises may deploy Type-2 hypervisors at scale using OS packages or vendor installers. A compromised dependency (e.g. kernel module, library) can introduce persistent, hard-to-detect backdoors on multiple hosts and affect many internal workloads, leading to Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-2 hypervisors and host OS components depend on broader upstream ecosystems. A supply-chain attack on drivers, libraries, or packaged modules can propagate widely, compromising host OS and hypervisor instances across critical tenants, thus supporting High likelihood and High impact.

### B.7.2.3 M&O System

[To be completed in a future version]

## B.7.3 Risk Evaluation of CES use cases

### B.7.3.1 CRS

**Table B.7.3.1-1: CRS - Risk evaluation of CES use cases**

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-CRS-INTF-ESC: Runtime-level container breakout</b>														
<b>C1</b>	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, the CRS runs on isolated devices or personal environments where containers host non-critical workloads and have limited external exposure. A runtime-level escape (e.g. via runc or namespace flaws) is technically possible, but any breakout only affects a single device and low-sensitivity applications, so both likelihood and impact are assessed as Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS underpins enterprise or edge container platforms that host multiple business microservices. A successful runtime-level escape allows a compromised container to interfere with peer containers on that node (and possibly across nodes, if combined with orchestration weaknesses), exposing all business workloads for that tenant. This justifies Medium likelihood and Medium impact.
<b>C3</b>	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, the CRS is part of high-assurance, multi-tenant container platforms processing critical or regulated workloads. Exploits that break container isolation (e.g. sandbox escapes, kernel namespace bugs) can lead to cross-tenant compromise, access to critical data, and potential safety or regulatory failures, so both likelihood and impact are High.
<b>T-CRS-INTEG-COMP: Runtime-level tampering</b>														
<b>C1</b>	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, local tampering with runtime binaries or configuration (e.g. disabling seccomp or cgroup enforcement) requires direct access to a single device and affects only non-critical applications. The orchestrator, if present, is minimal or absent, and the blast radius is small, so both likelihood and impact remain Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the same CRS binaries and configs may be deployed across many enterprise nodes. If an attacker modifies runtime settings (e.g. disables AppArmor profiles, relaxes namespaces or cgroup limits), multiple business services lose isolation and resource

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														controls. Although the effect is tenant-scoped, it impacts many workloads, leading to Medium likelihood and Medium impact.
<b>C3</b>	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, runtime integrity is a core assurance anchor for critical and regulated container workloads. Tampered runtime binaries or unverified state (e.g. modified seccomp profiles, spoofed namespaces) can silently bypass orchestrator policies across many tenants and nodes. Therefore, both likelihood and impact are rated High.
<b>T-CRS-UNAUTH-ADMIN: Unauthorized runtime control access</b>														
<b>C1</b>	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-C1, the container runtime is typically driven by local commands (e.g. docker, ctr, podman) on a single, non-critical device. Misuse of these binaries via a local user or shared account is possible but affects only that device and low-value workloads, so likelihood and impact are Low.
<b>C2</b>	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, internal users or automation (CI/CD agents, deployment scripts) may have direct access to runtime control (e.g. privileged socket access or debug binaries). If authentication around such access is weak, insiders or compromised accounts can bypass orchestrator RBAC and start, stop, or modify containers directly on enterprise nodes. This yields Medium likelihood and Medium impact across the tenant's workloads.
<b>C3</b>	3	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-C3, any exposed or weakly protected runtime control interface (e.g. an unprotected runtime socket, debug endpoint, or local root account on critical nodes) allows attackers to circumvent orchestrator policy and manipulate containers running critical/regulated workloads. In multi-tenant or federated scenarios, this can impact many tenants and critical services, hence High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-CRS-UNAUTH-FUNC: Unauthorized privileged ops</b>														
<b>C1</b>	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, users with legitimate but broad permissions (e.g. local "developer" accounts) can perform privileged runtime actions such as starting privileged containers or mounting host paths. However, these actions are confined to a single device and non-critical apps, so overall risk remains Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise or edge environments often delegate limited runtime access to developers, operators, or automation tools. If authorization is insufficiently granular, these principals can escalate to privileged operations (e.g. modifying seccomp profiles, running containers as root, altering cgroup limits) and affect many containers belonging to that tenant, justifying Medium likelihood and Medium impact.
<b>C3</b>	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, inadequate authorization in the CRS enables authenticated but mis-scoped identities to perform powerful runtime operations across nodes hosting critical or regulated workloads. This can override intended least-privilege policies, create new attack surfaces, or directly expose sensitive data, so both likelihood and impact are High.
<b>T-CRS-DATA-DISC: Data disclosure via runtime</b>														
<b>C1</b>	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, containers typically host low-sensitivity data (e.g. local telemetry, hobbyist applications). Even if CRS flaws allow access to container filesystems, mounts, or memory, the confidentiality impact is minimal, so both likelihood and impact are Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS may host business microservices processing confidential data (e.g. HR, analytics, internal APIs). Runtime-level weaknesses that expose mounted volumes, layered filesystems, or shared memory segments can leak sensitive internal data of the enterprise tenant, leading to Medium likelihood and Medium impact.
<b>C3</b>	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, containers may handle financial, e-identity, or medical data in multi-tenant and federated setups. If runtime isolation fails, attackers can extract highly sensitive data from multiple tenants. This supports High likelihood and High impact.
<b>T-CRS-AVAIL-DOS: Resource exhaustion</b>														

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale	
<b>C1</b>	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, a container that saturates CPU, memory, or I/O via weak cgroup settings will only affect a single device and non-critical services. The resulting DoS is limited to local functionality, so both likelihood and impact are Low.	
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES nodes often run many business microservices. If the CRS does not enforce appropriate cgroup limits or priority settings, a single misbehaving or malicious container can starve other services on a node, causing noticeable but tenant-scoped outages, which is reflected as Medium likelihood and Medium impact.	
<b>C3</b>	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-C3, container platforms run critical or regulated services at scale. Insufficient isolation or resource governance in the CRS (e.g. no hard limits, shared resources across tenants) enables noisy-neighbour or deliberate resource exhaustion that can degrade or interrupt critical services across multiple tenants and edge sites, so both likelihood and impact are High.	
<b>T-CRS-LOG-INT: Logging weakness/tampering</b>															
<b>C1</b>	1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, CRS logs (e.g. container start/stop events, seccomp denials) may be minimal or stored locally without protection. Attackers running inside containers could modify or delete such logs, but given the low sensitivity and small scale, the inability to reconstruct activity has limited impact, so risk remains Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES platforms rely on CRS logs for troubleshooting and incident response (e.g. detecting abnormal restarts, denied syscalls, mount failures). Weak logging or easy log tampering delays detection of attacks against business microservices and complicates forensics, warranting Medium likelihood and Medium impact.
<b>C3</b>	2	3	3	2	3	2	3	3	3	3	3	13 → High	14 → High	High	In UC-C3, logging from the CRS is essential to maintain evidential integrity across critical workloads and tenants. If runtime logs are incomplete, not centrally collected, or modifiable by compromised containers, attackers can conduct stealthy operations (e.g. privilege escalation, forbidden syscalls) without trace across multi-tenant environments. This justifies High likelihood and High impact.
<b>T-CRS-UPD-COMP: Malicious/compromised update</b>															

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, CRS updates (e.g. runc, containerd) are often applied manually on single devices, sometimes via direct downloads. A compromised package or replayed older runtime can weaken isolation on that device, but the effect is limited to one low-critical node, so likelihood is Low and impact Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may use repositories or CI/CD pipelines to distribute runtime updates. If these are compromised or unsigned, a malicious runtime can be rolled out across many nodes, weakening container isolation and seccomp/cgroup enforcement for all workloads of that tenant. This supports Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, large-scale or automated update pipelines for CRS components mean that a poisoned runtime package can rapidly propagate across nodes hosting critical, multi-tenant workloads, effectively undermining isolation guarantees everywhere it is deployed. Hence both likelihood and impact are High.
<b>T-CRS-CFG-COMP: Insecure or Unauthorized configuration</b>														
C1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-C1, insecure runtime settings (e.g. no default seccomp/AppArmor profile, permissive host mounts) stem mainly from local experimentation or convenience. Misconfigurations are limited to one device and non-critical containers, so although isolation is reduced, overall risk remains Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, runtime configuration (default seccomp profiles, namespace settings, resource limits) is often propagated via container engines or orchestrators. Weak defaults (e.g. unbounded CPU/memory, privileged containers, hostPath mounts) or Unauthorized changes can significantly increase the attack surface across enterprise workloads, giving Medium likelihood and Medium impact.
C3	2	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, CRS configuration is a primary control for maintaining strict isolation and resource separation in critical or regulated environments. Insecure or Unauthorized runtime settings can directly enable container escape, lateral movement and high-impact DoS across tenants, so both likelihood and impact are High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
<b>T-CRS-SC-COMP: Supply chain compromise</b>														
<b>C1</b>	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, a compromised runtime component (e.g. unpacker library, sandboxing module) affects a single isolated host. While it may permit container escape or tampering, the limited connectivity and low-value workloads keep the overall impact low; the Likelihood is Low.
<b>C2</b>	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may deploy CRS packages from shared repositories or base images. A poisoned dependency embedded in the runtime can introduce persistent backdoors or isolation weaknesses across many enterprise nodes, impacting all internal workloads but still within one organizational tenant, hence Medium likelihood and Medium impact.
<b>C3</b>	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, CRS supply-chain integrity is critical. Compromise of a low-level runtime dependency (e.g. syscall handling library, sandbox module) used in container platforms for multiple tenants means every container instantiation can execute malicious code with high privileges. This can enable systemic escape and cross-tenant compromise in public or federated CES, leading to High likelihood and High impact.

### B.7.3.2 CE

[To be completed in a future version]

### B.7.3.3 CO

[To be completed in a future version]

## B.7.4 Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS

The threat T-ALL-HW-ACCESS (Compromise via hardware-level or platform trust components) is common to both Virtualization Execution Stacks (VES) and Container Execution Stacks (CES).

It models compromise of the underlying hardware or platform trust components in a way that can bypass or undermine software-based security mechanisms, including boot integrity verification, runtime integrity checks, attestation, and isolation controls.

Because hardware-level compromise typically requires a high level of attacker sophistication and is heavily dependent on platform supply-chain assurance, the likelihood remains relatively low to medium, while the impact can be catastrophic for high-assurance, multi-tenant or critical deployments.

Table B.7.4-1 applies the risk assessment methodology of clause B to this threat for the VES and CES use cases.

**Table B.7.4-1: Risk evaluation of the common hardware-level threat T-ALL-HW-ACCESS across VES and CES use cases**

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, hypervisors run on isolated hosts with a small number of non-critical VMs and limited connectivity. Hardware-level compromise (e.g. tampered firmware or physical access) would give full control of that single host, but the data is low-sensitivity and there is no large orchestration blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and attacker incentive (EXT=1, TEN=1, CTL=1, PRV=1) are low. Overall likelihood and impact are therefore assessed as Low.
V2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-V2, Type-1/Type-2 hypervisors support general business workloads in enterprise/private clouds and connected edge clusters. Commodity hardware and shared infrastructure increase the relevance of hardware supply-chain issues (EXT=2, TEN=2, SUP=2), but the attack still requires high sophistication (PRV=1). A successful hardware-level compromise affects multiple business VMs and nodes within one organization (ISO=2, DAT=2, ORB=2), and standard enterprise monitoring/recovery exists (DRC=2), so both likelihood and impact are Medium.
V3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-V3, hypervisors are the foundation of high-assurance, multi-tenant infrastructures processing critical or regulated workloads. These platforms are high-value targets for sophisticated hardware-level attacks (EXT=3, TEN=3, SUP=3) even though such attacks remain complex (PRV=1). A compromised CPU, firmware, or hardware security module can bypass secure boot, attestation, and isolation controls, impacting all tenants and critical services across clusters (ISO=3, DAT=3, ORB=3). Detection and recovery from deep hardware compromise are limited (DRC=3). As a result, likelihood is Medium but impact is High, and the overall risk level is High.
C1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, the CES run on devices with limited connectivity (e.g. hobbyist setups, isolated test/dev hosts) that execute non-critical containers. Hardware-level compromise would allow full control of the host and all containers, but the workloads and data are low-sensitivity and there is no meaningful multi-tenant blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and tenant multiplicity are low (EXT=1, TEN=1, CTL=1, PRV=1), so both likelihood and impact are Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-C2, the CES supports general business microservices and edge workloads. A hardware-level compromise of the underlying platform (e.g. compromised firmware or SoC) affects all CRS instances and containers on affected nodes, exposing sensitive internal data and services within one organization (ISO=2, DAT=2, ORB=2). Commodity hardware and common supply chains (SUP=2) increase the exposure compared to UC-C1, but attacks still require high sophistication (PRV=1). This results in Medium likelihood and Medium impact.
C3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-C3, the CES underpins critical or regulated containerized workloads in public, federated, or zero-trust environments, including edge sites performing safety-relevant processing. Hardware-level compromise of CPUs, firmware or platform trust components can silently bypass all CRS and orchestrator controls, enabling arbitrary access to containers and data across many tenants and sites (ISO=3, DAT=3, ORB=3). These platforms are attractive targets for well-resourced attackers (EXT=3, TEN=3, SUP=3), and deep hardware compromise is hard to detect and remediate (DRC=3). Accordingly, likelihood is Medium but impact is High, giving an overall High risk.

## B.8 Threat-to-Requirement Traceability Matrix

### B.8.1 VES

#### B.8.1.1 Hypervisor

**Table B.8.1.1-1: Hypervisor - threats to requirements mapping**

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-HYP1-INTF-ESC/ T-HYP2-HOST-ABUSE	REQ-H-VM-ISO-001 REQ-H-CP-ISO-001 REQ-H-NP-ISO-001	REQ-H-VM-ISO-001 REQ-H-CP-ISO-002 REQ-H-NP-ISO-002	REQ-H-VM-ISO-003 REQ-H-CP-ISO-003 REQ-H-NP-ISO-003
T-HYP1-INTEG-COMP/T-HYP2-INTEG-CO	REQ-H-B-INT-001 REQ-H-IMG-INT-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002	REQ-H-B-INT-003 REQ-H-IMG-INT-003 REQ-H-RP-INT-003 REQ-H-RA-INT-003 REQ-H-RA-INT-004
T-HYP1-UNAUTH-ADMIN/ T-HYP2-UNAUTH-ADMIN	REQ-H-ADMIN-AUTH-001 REQ-H-SERV-AUTH-001	REQ-H-ADMIN-AUTH-002 REQ-H-SERV-AUTH-002	REQ-H-ADMIN-AUTH-003 REQ-H-SERV-AUTH-003
T-HYP1-UNAUTH-FUNC/ T-HYP2-UNAUTH-FUNC	REQ-H-ADMIN-AUTHZ-001 REQ-H-SERV-AUTHZ-001	REQ-H-ADMIN-AUTHZ-002 REQ-H-SERV-AUTHZ-002	REQ-H-ADMIN-AUTHZ-003 REQ-H-SERV-AUTHZ-003
T-HYP1-DATA-DISC/ T-HYP2-DATA-DISC	REQ-H-CONF-001 REQ-H-DM-001 REQ-H-DM-002	REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-CONF-004
T-HYP1-AVAIL-DOS/ T-HYP2-AVAIL-DOS	REQ-H-AVAIL-001	REQ-H-AVAIL-002 REQ-H-AVAIL-003	REQ-H-AVAIL-004
T-HYP1-LOG-INT/ T-HYP2-LOG-INT	REQ-H-LOG-001	REQ-H-LOG-002	REQ-H-LOG-003
T-HYP1-UPD-COMP/ T-HYP2-UPD-COMP	REQ-H-UPD-001	REQ-H-UPD-002	REQ-H-UPD-003
T-HYP1-CFG-COMP/ T-HYP2-CFG-COMP	REQ-H-CFG-001	REQ-H-CFG-002	REQ-H-CFG-003
T-HYP1-SC-COMP/ T-HYP2-SC-COMP	REQ-H-B-INT-001 REQ-H-IMG-INT-001 REQ-H-UPD-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-UPD-002	REQ-H-B-INT-003 REQ-H-IMG-INT-003 REQ-H-RP-INT-003 REQ-H-RA-INT-003 REQ-H-RA-INT-004 REQ-H-UPD-003

### B.8.1.2 M&O System

[To be completed in a future version]

## B.8.2 CES

## B.8.2.1 CRS

Table B.8.2.1-1: CRS - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-CRS-INTF-ESC	REQ-CRS-CN-ISO-001 REQ-CRS-CP-ISO-001 REQ-CRS-NP-ISO-001	REQ-CRS-CN-ISO-002 REQ-CRS-CP-ISO-002 REQ-CRS-NP-ISO-002	REQ-CRS-CN-ISO-003 REQ-CRS-CP-ISO-003 REQ-CRS-NP-ISO-003
T-CRS-INTEG-COMP	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002	REQ-CRS-B-INT-003 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-003 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004
T-CRS-UNAUTH-ADMIN	REQ-CRS-ADMIN-AUTH-001 REQ-CRS-SERV-AUTH-001	REQ-CRS-ADMIN-AUTH-002 REQ-CRS-SERV-AUTH-002	REQ-CRS-ADMIN-AUTH-003 REQ-CRS-SERV-AUTH-003
T-CRS-UNAUTH-FUNC	REQ-CRS-ADMIN-AUTHZ-001 REQ-CRS-SERV-AUTHZ-001	REQ-CRS-ADMIN-AUTHZ-002 REQ-CRS-SERV-AUTHZ-002	REQ-CRS-ADMIN-AUTHZ-003 REQ-CRS-SERV-AUTHZ-003
T-CRS-DATA-DISC	REQ-CRS-CONF-001 REQ-CRS-DM-001 REQ-CRS-DM-002	REQ-CRS-CONF-002 EQ-CRS-CONF-003	REQ-CRS-CONF-004
T-CRS-AVAIL-DOS	REQ-CRS-AVAIL-001	REQ-CRS-AVAIL-002 REQ-CRS-AVAIL-003	REQ-CRS-AVAIL-004
T-CRS-LOG-INT	REQ-CRS-LOG-001	REQ-CRS-LOG-002	REQ-CRS-LOG-003
T-CRS-UPD-COMP	REQ-CRS-UPD-001	REQ-CRS-UPD-002	REQ-CRS-UPD-003
T-CRS-CFG-COMP	REQ-CRS-CFG-001	REQ-CRS-CFG-002	REQ-CRS-CFG-003
T-CRS-SC-COMP	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001 REQ-CRS-UPD-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-UPD-002	REQ-CRS-B-INT-003 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-003 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004 REQ-CRS-UPD-003

## B.8.2.2 CE

[To be completed in a future version]

## B.8.2.3 CO

[To be completed in a future version]

## B.8.3 Hardware-level common threat

Table B.8.3-1: Hypervisor - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-H-B-INT-001 REQ-H-IMG-INT-001 REQ-H-CONF-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-B-INT-003 REQ-H-IMG-INT-003 REQ-H-RP-INT-003 REQ-H-RA-INT-003, REQ-H-RA-INT-004 REQ-H-CONF-004

Table B.8.3-2: CRS - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-CRS-B-INT-001 REQ-CRS-IMG-INT-001 REQ-CRS-CONF-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-CONF-002 REQ-CRS-CONF-003	REQ-H-CRS-INT-003 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-003 REQ-CRS-RA-INT-003, REQ-CRS-RA-INT-004 REQ-CRS-CONF-004

## Annex K: Cryptographic mechanisms

### K.1 General

This annex identifies, for the requirements that directly require cryptographic functionality, the relevant families of agreed cryptographic mechanisms from the ECCG Agreed Cryptographic Mechanisms document [2].

For cryptographic constructions and protocols, the use of agreed mechanisms is subject to the ECCG principle that agreed constructions rely on agreed underlying primitives. The concrete choice of algorithm, parameter size, protocol profile, and associated conditions remains subject to the applicable ECCG clauses, including their notes, restrictions, and validity status.

This annex does not imply that each requirement listed below mandates a single implementation technology. Where the requirement is technology-neutral, the table identifies the ECCG mechanism family that is most directly relevant to the required cryptographic objective.

The mapping is intended to support consistent identification of the ECCG cryptographic mechanism families relevant to the implementation of the corresponding requirements. It does not replace the requirement text itself and does not reduce or alter the applicability, scope, or assurance expectations of the mapped requirement.

### K.2 Mapping of Hypervisor requirements to agreed cryptographic mechanisms (ACM)

**Table K.2: Mapping of Hypervisor cryptographic requirements to ACM**

Hypervisor requirement	Cryptographic operation	ACM direct clause(s) [2]	Supporting ACM clause(s) [2]
REQ-H-CP-ISO-002	Protected communication channel for network-accessible administrative interfaces	6.1 TLS	8 Key Management
REQ-H-CP-ISO-003	Mutually authenticated protected communication channel for network-accessible administrative interfaces	6.1 TLS	5.3 Asymmetric Entity Authentication Schemes; 8 Key Management
REQ-H-NP-ISO-003	Cryptographically protected management-network channel where shared networks are used	6.1 TLS	8 Key Management
REQ-H-B-INT-001	Integrity verification against a trusted reference value	2.3 Hash Functions	NA
REQ-H-B-INT-002	Authenticity and integrity verification of boot stages	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-H-B-INT-003	Reliance on authenticated integrity information for the boot chain	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-H-IMG-INT-001	Integrity verification of guest VM images	2.3 Hash Functions	NA
REQ-H-IMG-INT-002	Authenticity and integrity verification of guest VM images	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-H-IMG-INT-003	Verification of guest VM images using protected trust material	5.2 Digital Signature	8 Key Management; 2.3 Hash Functions
REQ-H-RA-INT-003	Cryptographically verifiable attestation evidence and protection of attestation keys	5.2 Digital Signature	8 Key Management; 2.3 Hash Functions; 7 Random Generator
REQ-H-RA-INT-004	Protected attestation credentials for privacy-aware attestation options	5.2 Digital Signature	8 Key Management
REQ-H-ADMIN-AUTH-002	Cryptographic key-based administrative authentication	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 7 Random Generator
REQ-H-ADMIN-AUTH-003	Strong administrative authentication, including certificate-based authentication where used	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 6.1 TLS
REQ-H-SERV-AUTH-002	Certificate-based mutual authentication for external services	5.3 Asymmetric Entity Authentication Schemes	6.1 TLS; 8 Key Management

REQ-H-SERV-AUTH-003	Certificate-based mutual authentication with PKI path validation for external services	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 6.1 TLS
REQ-H-CONF-002	Encryption of sensitive data at rest	3.2 Specific Confidentiality Modes: Disk Encryption	3.1 Confidentiality Modes of Operation; 8 Key Management
REQ-H-CONF-003	Confidentiality and integrity protection for data in transit	3.5 Authenticated Encryption	6.1 TLS; 8 Key Management
REQ-H-LOG-002	Authenticated and encrypted export of audit logs	6.1 TLS	8 Key Management
REQ-H-LOG-003	Cryptographic integrity/origin protection and tamper-evident protection of audit logs	5.2 Digital Signature; 3.3 Integrity Modes: Message Authentication Codes	2.3 Hash Functions; 8 Key Management
REQ-H-UPD-002	Authenticity and integrity verification of updates before installation	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-H-UPD-003	Cryptographic support for trusted update provenance and authorized rollback decisions	8 Key Management	5.2 Digital Signature; 2.3 Hash Functions

### K.3 Mapping of CRS requirements to agreed cryptographic mechanisms (ACM)

**Table K.3: Mapping of CRS cryptographic requirements to ACM**

CRS requirement	Cryptographic operation	ACM direct clause(s) [2]	Supporting ACM clause(s) [2]
REQ-CRS-CP-ISO-003	Protected and mutually authenticated communication channels for remote or network-exposed CRS management and control interfaces	6.1 TLS	5.3 Asymmetric Entity Authentication Schemes; 8 Key Management
REQ-CRS-NP-ISO-003	Protected and mutually authenticated communication channels for CRS management and control traffic traversing shared or untrusted networks	6.1 TLS	5.3 Asymmetric Entity Authentication Schemes; 8 Key Management
REQ-CRS-B-INT-001	Integrity verification of CRS startup executables against trusted reference values	2.3 Hash Functions	NA
REQ-CRS-B-INT-002	Authenticity and integrity verification of CRS startup stages	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-CRS-B-INT-003	Reliance on authenticated integrity and authenticity status for CRS startup components	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-CRS-IMG-INT-001	Integrity verification of container images using cryptographic digests of manifests and referenced layers	2.3 Hash Functions	NA
REQ-CRS-IMG-INT-002	Authenticity and integrity verification of container images and their trusted provenance	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-CRS-IMG-INT-003	Verification of container images using protected trust material	5.2 Digital Signature	8 Key Management; 2.3 Hash Functions
REQ-CRS-RA-INT-003	Cryptographically verifiable attestation evidence and protection of attestation keys and measurements	5.2 Digital Signature	8 Key Management; 2.3 Hash Functions; 7 Random Generator
REQ-CRS-RA-INT-004	Protected attestation credentials and privacy-aware attestation options for remote verification scenarios	5.2 Digital Signature	8 Key Management
REQ-CRS-ADMIN-AUTH-001	Non-reversible, salted password hashing for administrative passwords where password-based authentication is used	3.8 Password Protection / Password Hashing Mechanisms	NA
REQ-CRS-ADMIN-AUTH-002	Cryptographic key-based administrative authentication	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 7 Random Generator

REQ-CRS-ADMIN-AUTH-003	Strong administrative authentication, including certificate-based authentication where used	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 6.1 TLS
REQ-CRS-SERV-AUTH-002	Mutual cryptographic authentication for service-to-CRS interactions	5.3 Asymmetric Entity Authentication Schemes	6.1 TLS; 8 Key Management
REQ-CRS-SERV-AUTH-003	Verifiable binding of service identities to cryptographic credentials, with managed rotation and revocation	5.3 Asymmetric Entity Authentication Schemes	8 Key Management; 6.1 TLS
REQ-CRS-CONF-002	Encryption of sensitive data at rest	3.2 Specific Confidentiality Modes: Disk Encryption	3.1 Confidentiality Modes of Operation; 8 Key Management
REQ-CRS-CONF-003	Confidentiality and integrity protection for CRS data in transit	3.5 Authenticated Encryption	6.1 TLS; 8 Key Management
REQ-CRS-CONF-004	Protection and controlled provisioning of secrets and key material for confidential execution environments	8 Key Management	3.6 Key Protection
REQ-CRS-LOG-002	Authenticated and encrypted export of CRS audit logs	6.1 TLS	8 Key Management
REQ-CRS-LOG-003	Cryptographic integrity/origin protection of audit logs	5.2 Digital Signature; 3.3 Integrity Modes: Message Authentication Codes	2.3 Hash Functions; 8 Key Management
REQ-CRS-UPD-002	Authenticity and integrity verification of CRS updates before installation	5.2 Digital Signature	2.3 Hash Functions; 8 Key Management
REQ-CRS-UPD-003	Cryptographic support for trusted CRS update provenance and authorized rollback decisions	8 Key Management	5.2 Digital Signature; 2.3 Hash Functions

---

## Annex R: Additional provisions for products relying on remote data processing solutions (RDPS)

---

### Annex (informative): Bibliography

## Annex (informative): Change history

Date	Version	Information about changes
June 2025	0.0.1	<ul style="list-style-type: none"> <li>• Added representative use cases for hypervisors and container runtime systems, classified by risk level.</li> <li>• Defined cumulative security levels (Basic, Elevated, Advanced) to guide requirement selection.</li> <li>• Defined threats</li> <li>• Defined generic security requirements</li> <li>• Defined instantiated security requirements for isolation, integrity, authentication, authorization and confidentiality with assigned security levels across product types (Type 1/2 Hypervisors, CRS).</li> <li>• Added Annex A: Risk assessment methodology</li> <li>• Added Annex B: Mapping between GR requirements and CRA.</li> <li>• Added Annex C: Threat-to-Requirement-to-Control mapping table for traceability.</li> <li>• Added Annex D: Illustrative security controls</li> </ul>
August 2025	0.0.2	<ul style="list-style-type: none"> <li>• Extend the scope to cover the entire container execution stack, including the Container Runtime System (CRS), the Container Engine (CE), and the Container Orchestrator (CO).</li> <li>• Extend the scope to cover the entire Virtualization execution stack, including the hypervisor and the orchestration and management system.</li> <li>• Update scope figures.</li> <li>• Replace "security level" with "security category level".</li> <li>• Add new requirements specific to the container engine (CE) and container orchestrator (CO).</li> <li>• Revise existing CRS requirements to align with the extended scope and updated terminology.</li> <li>• Update the risk assessment methodology in Annex A to: <ul style="list-style-type: none"> <li>◦ Clarify how threats are factored into risk determination.</li> <li>◦ Describe how to assess the overall product risk level.</li> </ul> </li> <li>• Introduce an "Assessment" clause, including an example test case for one of the defined security requirements.</li> <li>• Replace the "Assumptions" clause with a new "Operational Environment Objectives" clause.</li> </ul>
August 2025	0.0.3	<ul style="list-style-type: none"> <li>• Updated scope figures.</li> <li>• Revised list of threats, now with specific threats defined per component.</li> <li>• New clause on Objectives for the Operational Environment, replacing the previous <i>Assumptions</i> clause.</li> <li>• New and updated requirements, including: <ul style="list-style-type: none"> <li>◦ New requirements for the management and orchestration system.</li> <li>◦ Updates to isolation and integrity requirements for hypervisors.</li> </ul> </li> <li>• Addition of three options for structuring the assessment of requirements: <ul style="list-style-type: none"> <li>◦ Test case-based approach</li> <li>◦ Simplified PT2</li> <li>◦ ETSI TS 103 645 / ETSI EN 303 645 approach</li> </ul> </li> </ul>
September 2025	0.0.4	<ul style="list-style-type: none"> <li>• Addressed new comments received (see Excel sheet v5).</li> <li>• Updated the titles of representative use cases in Clause 5.</li> <li>• Updated requirement categorization and selection approaches in Clause 6: <ul style="list-style-type: none"> <li>◦ The label Enhanced has been replaced by Advanced.</li> <li>◦ A new option for creating a custom security profile has been introduced for cases where the product use case does not fit one of the representative use cases in the present document, or where the manufacturer wishes to conduct a dedicated risk assessment.</li> <li>◦ A categorization approach: Cumulative, Exclusive, and Single - has been introduced to clarify the risk-based selection of requirements.</li> </ul> </li> <li>• Transformed generic requirements into security objectives (Clause 8).</li> <li>• Updated VES and CES requirements following received comments; additional elements have been added, and the new categorization approach has been applied to a subset of them (see Clauses 10 and 11).</li> <li>• Added new requirements on vulnerability handling and SBOM in Clause 12.</li> <li>• Updated the risk assessment methodology: simplification of risk factors and application of the updated risk factors to the assessment of use cases.</li> </ul>

Date	Version	Information about changes
		<ul style="list-style-type: none"> <li>• Moved implementation notes intended as guidance for manufacturers from the requirements clauses to Annex D.</li> </ul>
September 2025	0.0.5	<ul style="list-style-type: none"> <li>• Addressed new comments received from ETSI.</li> <li>• Added a time synchronization objective and related requirements for the hypervisor, with implementation notes included in Annex D.</li> <li>• Completed and updated authentication requirements for the hypervisor, covering both administrative and service authentication.</li> <li>• Annex A: CRA essential requirements have been mapped to the existing hypervisor requirements.</li> <li>• Annex C: Now designated as a normative annex. Threats to hypervisor requirements are partially mapped as a sample for presentation; this mapping will be completed once a stable set of requirements with fixed IDs is available.</li> <li>• Reordered clauses: The clause defining labels, categories, and SCLs, along with the approaches for applying requirements, has been moved to appear just before the requirements clause, making the present document easier to follow.</li> <li>• Updated use case descriptions, including illustrative business applications and revised example environments.</li> <li>• Requirement categorization, labelling, and classification approaches have been clarified for manufacturers. The guidance is now more direct and less perspective-based.</li> <li>• The label Standard has been replaced by Elevated.</li> </ul>
October 2025	0.0.6	<ul style="list-style-type: none"> <li>• Conformance assessment approach described, including evaluation of hypervisor isolation and boot chain integrity requirements.</li> <li>• Incorporated new comments received.</li> <li>• Added definitions of Tenant and Multi-Tenancy.</li> <li>• Slightly updated use cases UC-V3 and UC-C3 to also cover multi-tenancy scenarios within the same organization.</li> <li>• Updated risk assessment methodology applied to UC-V1/2/3, with risk evaluated per threat for hypervisors.</li> <li>• Mapped hypervisor threats to corresponding technical requirements to CRA requirements.</li> <li>• Moved implementation notes to Annex D.</li> <li>• Replaced the 'Label' term with 'Class'.</li> </ul>
October 2025	0.0.7	<ul style="list-style-type: none"> <li>• Update following the ETSI/CEN-CENELEC rapporteurs meeting on the alignment of CRA vertical standards: <ul style="list-style-type: none"> <li>◦ Document structure made more compact and logically sequenced.</li> <li>◦ Assessment approach optimized for consistency.</li> </ul> </li> <li>• Addition of a new clause: "Product variants" clarifying the scope of products under the present document, identifying core mandatory components, and distinguishing optional elements.</li> <li>• Update of use cases to remove inconsistencies and confusion related to the terms single-tenant and multi-tenant; consolidation achieved through additional illustrative examples.</li> <li>• All requirements in this version now follow a harmonised structure, with the class-based approach applied consistently across all defined requirements.</li> <li>• Update of requirements leveraging hardware-based technologies to ensure they are described in a more neutral, technology-agnostic, and risk-based manner.</li> <li>• CRS requirements are completed.</li> <li>• Addition of a new table clarifying the applicability of hypervisor requirements to Type I, Type II, and Hybrid architectures.</li> <li>• Application of the new assessment approach to both Hypervisor and CRS requirements related to Isolation and Integrity Protection.</li> <li>• Completion of the mapping between CRA essential requirements and the corresponding Hypervisor and CRS requirements.</li> <li>• Completion of the mapping between identified Threats and corresponding Requirements for both Hypervisor and CRS components.</li> </ul>

Date	Version	Information about changes
November 2025	0.0.8	<ul style="list-style-type: none"> <li>• Addressed most of the comments received during the pre-open consultation.</li> <li>• Addressed most of the comments received from the ETSI team working on the present document.</li> <li>• Updated the use cases section, including the addition of edge deployment scenarios.</li> <li>• Clarified the assessment approach and added requirements for the assessment report and the evidence package.</li> <li>• Added applicability notes to several security requirements where relevant.</li> <li>• Added a new clause, "Roles and Responsibilities," clarifying the responsibilities of the manufacturer and the product integrator.</li> <li>• Added additional assessment cases for confidentiality and availability requirements for Hyper and CRS components.</li> <li>• Updated the applicability table for hypervisor requirements, replacing "partially applicable" with "conditionally applicable."</li> <li>• Added references to prEN 40000-1-1, prEN 40000-1-2, and prEN 40000-1-3.</li> <li>• Referenced prEN 40000-1-3 for SBOM and vulnerability handling requirements.</li> <li>• Updated several requirements following received comments, including changes in classification for some of them.</li> </ul>
November 2025	0.0.9	<ul style="list-style-type: none"> <li>• Added a new hardware-related threat and performed the corresponding risk assessment.</li> <li>• Updated the applicability table for requirements on Type I, Type II and Hybrid hypervisors to clarify that a requirement may be implemented directly by the hypervisor or delegated to other OE components.</li> <li>• Updated the isolation, integrity and confidentiality requirements: Advanced class may leverage software- or hardware-based mechanisms; hardware-assisted requirements are now optional ("may").</li> <li>• Completed the assessment descriptions for all Hypervisor and CRS requirements.</li> <li>• Updated the risk assessment for the use cases and refined the rationales to better reflect the context of the risk factors and use cases.</li> <li>• Performed an end-to-end alignment in preparation for the HAS review.</li> </ul>
December 2025	0.0.10	<ul style="list-style-type: none"> <li>• Revised based on ETSI editing support comments to improve compliance with ETSI drafting requirements.</li> </ul>
March 2026	0.0.11	<ul style="list-style-type: none"> <li>• Update based on OC1/2/3 comments</li> <li>• Update based on the HAS comments received on Clause 4 (except comments #7 and #16)</li> </ul>
March 2026	0.0.12	<ul style="list-style-type: none"> <li>• Update based on the HAS comments received on Clause 4/5</li> </ul>
April 2026	0.0.13	<ul style="list-style-type: none"> <li>• Update based on the HAS comments received on Clause 4/5</li> <li>• Update based on stakeholders' comments</li> </ul>
April 2026	0.0.14	<ul style="list-style-type: none"> <li>• Update based on the HAS comments received on Clause 5</li> <li>• Update based on stakeholders' comments (Google, XCP-ng OSI)</li> <li>• Addition of mapping between requirements and crypto operations from ACM</li> <li>• Alignment with the cross vertical skeleton</li> </ul>
April 2026	0.0.15	<ul style="list-style-type: none"> <li>• Clauses 5 and 6 updated based on the received HAS comments</li> <li>• Document updated based on stakeholders' comments, including XCP-ng / OSI feedback</li> <li>• Terminology updated: SCL replaced by Security Profile (SP)</li> <li>• Technology-specific terms (TEE / TPM / HSM) replaced by generic wording</li> <li>• Threat-to-requirement mapping tables moved to Annex B</li> <li>• Applicability tables for UCs / SPs added under each requirements subclause</li> <li>• Document structure aligned with the cross-vertical skeleton</li> <li>• Annex R (RDPS) added as a placeholder, pending integration of a stable and endorsed version</li> </ul>

---

## History

Document history		
V0.0.6	October 2025	Clean-up done by <i>editHelp!</i> E-mail: <a href="mailto:edithelp@etsi.org">mailto:edithelp@etsi.org</a>
V0.0.10	December 2025	Clean-up done by <i>editHelp!</i> E-mail: <a href="mailto:edithelp@etsi.org">mailto:edithelp@etsi.org</a>