

Draft ETSI EN 304 623 V0.1.1 (2026-05)



Cyber Security (CYBER); Cyber Resilience Act (CRA); Cybersecurity requirements for boot managers

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 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: [STAN4CRA / EN 304 623 Boot Managers · GitLab](#) and submit your comments as “issues” following the guidelines provided on this site. Alternatively, you may also send your comments to: 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-008

Keywords

CRA, Cybersecurity, Boot

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.

48
49
50
51

© ETSI 2026.
All rights reserved.

52	Contents		
53	Intellectual Property Rights		11
54	Foreword.....		11
55	Modal verbs terminology		12
56	Introduction		12
57	1 Scope.....		13
58	1.1 General.....		13
59	1.2 In-scope products.....		13
60	2 References		14
61	2.1 Normative references.....		14
62	2.2 Informative references		14
63	3 Definition of terms, symbols and abbreviations.....		15
64	3.1 Terms		15
65	3.2 Symbols		18
66	3.3 Abbreviations.....		18
67	4 Product description (informative)		18
68	4.1 Intended purpose and reasonably foreseeable use		18
69	4.1.1 Intended purpose		18
70	4.1.2 Reasonably foreseeable use.....		19
71	4.1.3 Reasonably foreseeable misuse		19
72	4.2 Product functions		20
73	4.3 Architecture		20
74	4.3.1 Boot architecture		20
75	4.3.1.1 Unverified boot.....		20
76	4.3.1.2 Verified boot.....		21
77	4.3.1.3 Measured boot		22
78	4.3.2 Functional extensions.....		22
79	4.3.3 Product capabilities		23
80	4.3.3.1 Introduction		23
81	4.3.3.2 Update capability.....		23
82	4.3.3.3 Boot source		24
83	4.3.3.4 Configuration management		24
84	4.3.3.5 Hardware security.....		24
85	4.3.3.6 Recovery capability		25
86	4.3.3.7 Logging capability.....		25
87	4.3.3.8 Authentication capability.....		25
88	4.3.3.9 Debug interface		26
89	4.3.3.10 Key provisioning		26
90	4.3.4 Resource constraints.....		26
91	4.4 Operational environment		27
92	4.4.1 Introduction.....		27
93	4.4.2 Physical security context.....		27
94	4.4.3 Network exposure		27
95	4.4.4 Data sensitivity and value		27
96	4.4.5 System criticality.....		28
97	4.4.6 Lifecycle expectations.....		28
98	4.4.7 Administrative control.....		28
99	4.4.8 Availability requirements.....		28
100	4.4.9 Additional environmental characteristics		29
101	4.4.9.1 Environmental stress.....		29
102	4.4.9.2 Update frequency expectations.....		29
103	4.4.9.3 Interfaces		29
104	4.5 Distribution of security functions		30
105	4.5.1 Introduction		30
106	4.5.2 Security functions provided by the boot manager		30

107	4.5.3	Security functions required from the platform	30
108	4.5.4	Security functions delegated to boot target	31
109	4.5.5	Trust boundaries.....	31
110	4.6	Users	32
111	4.6.1	Introduction	32
112	4.6.2	User categories	32
113	4.6.3	User interaction patterns.....	32
114	4.7	Risks, threats and security context.....	33
115	4.8	Use cases.....	33
116	4.8.1	Purpose	33
117	4.8.2	Capabilities as risk exposures.....	33
118	4.8.3	Use case definitions.....	33
119	4.8.3.1	UC-IMM: Immutable	34
120	4.8.3.2	UC-VER: Verified and updateable.....	34
121	4.8.3.3	UC-HW: Hardware-assisted security.....	34
122	4.8.4	Functional capabilities and additional requirements	35
123	4.8.5	Use case selection	35
124	5	Requirements (normative).....	36
125	5.1	Introduction: Applicability of the requirements.....	36
126	5.1.1	General.....	36
127	5.1.2	Use case declaration.....	36
128	5.1.2.1	[RQ-UC-IMM]	36
129	5.1.2.2	[RQ-UC-VER].....	36
130	5.1.2.3	[RQ-UC-HW]	36
131	5.1.3	Not applicable verdict	37
132	5.2	No known exploitable vulnerabilities	37
133	5.2.1	Introduction	37
134	5.2.2	[REQ-BM-KEV-001].....	37
135	5.2.3	[REQ-BM-KEV-002].....	37
136	5.2.4	[REQ-BM-KEV-003].....	37
137	5.3	Secure by default configuration	37
138	5.3.1	Introduction to secure by default configuration requirements.....	37
139	5.3.2	[REQ-BM-SBD-001]	38
140	5.3.3	[REQ-BM-SBD-002]	38
141	5.3.4	[REQ-BM-SBD-003]	38
142	5.3.5	[REQ-BM-SBD-004]	38
143	5.3.6	[REQ-BM-SBD-005]	38
144	5.3.7	[REQ-BM-SBD-006]	38
145	5.4	Secure updates	39
146	5.4.1	Introduction to secure update requirements	39
147	5.4.2	[REQ-BM-SU-001].....	39
148	5.4.3	[REQ-BM-SU-002].....	39
149	5.4.4	[REQ-BM-SU-003].....	39
150	5.4.5	[REQ-BM-SU-004].....	39
151	5.4.6	[REQ-BM-SU-005].....	39
152	5.4.7	[REQ-BM-SU-006].....	40
153	5.5	Authentication and access control	40
154	5.5.1	Introduction to authentication and access control requirements.....	40
155	5.5.2	[REQ-BM-AAC-001].....	40
156	5.5.3	[REQ-BM-AAC-002].....	40
157	5.5.4	[REQ-BM-AAC-003].....	40
158	5.5.5	[REQ-BM-AAC-004].....	41
159	5.5.6	[REQ-BM-AAC-005].....	41
160	5.5.7	[REQ-BM-AAC-006].....	41
161	5.5.8	[REQ-BM-AAC-007].....	41
162	5.6	Confidentiality	41
163	5.6.1	Introduction to confidentiality requirements.....	41
164	5.6.2	[REQ-BM-CON-001].....	41
165	5.6.3	[REQ-BM-CON-002].....	41
166	5.6.4	[REQ-BM-CON-003].....	42
167	5.6.5	[REQ-BM-CON-004].....	42

168	5.6.6	[REQ-BM-CON-005].....	42
169	5.6.7	[REQ-BM-CON-006].....	42
170	5.6.8	[REQ-BM-CON-007].....	42
171	5.6.9	[REQ-BM-CON-008].....	43
172	5.6.10	[REQ-BM-CON-009].....	43
173	5.6.11	[REQ-BM-CON-010].....	43
174	5.7	Integrity	43
175	5.7.1	Introduction to integrity requirements.....	43
176	5.7.2	[REQ-BM-INT-001].....	43
177	5.7.3	[REQ-BM-INT-002].....	43
178	5.7.4	[REQ-BM-INT-003].....	44
179	5.7.5	[REQ-BM-INT-004].....	44
180	5.7.6	[REQ-BM-INT-005].....	44
181	5.7.7	[REQ-BM-INT-006].....	44
182	5.7.8	[REQ-BM-INT-007].....	44
183	5.7.9	[REQ-BM-INT-008].....	45
184	5.7.10	[REQ-BM-INT-009].....	45
185	5.7.11	[REQ-BM-INT-010].....	45
186	5.7.12	[REQ-BM-INT-011].....	45
187	5.7.13	[REQ-BM-INT-012].....	45
188	5.7.14	[REQ-BM-INT-013].....	45
189	5.7.15	[REQ-BM-INT-014].....	46
190	5.7.16	[REQ-BM-INT-015].....	46
191	5.7.17	[REQ-BM-INT-016].....	46
192	5.7.18	[REQ-BM-INT-017].....	46
193	5.7.19	[REQ-BM-INT-018].....	46
194	5.7.20	[REQ-BM-INT-019].....	46
195	5.7.21	[REQ-BM-INT-020].....	47
196	5.7.22	[REQ-BM-INT-021].....	47
197	5.7.23	[REQ-BM-INT-022].....	47
198	5.7.24	[REQ-BM-INT-023].....	47
199	5.7.25	[REQ-BM-INT-024].....	47
200	5.8	Data minimisation.....	48
201	5.8.1	Introduction to data minimisation requirements.....	48
202	5.8.2	[REQ-BM-DM-001].....	48
203	5.8.3	[REQ-BM-DM-002].....	48
204	5.9	Availability protection	48
205	5.9.1	Introduction to availability protection requirements	48
206	5.9.2	[REQ-BM-AP-001].....	48
207	5.9.3	[REQ-BM-AP-002].....	49
208	5.9.4	[REQ-BM-AP-003].....	49
209	5.9.5	[REQ-BM-AP-004].....	49
210	5.9.6	[REQ-BM-AP-005].....	49
211	5.9.7	[REQ-BM-AP-006].....	49
212	5.9.8	[REQ-BM-AP-007].....	49
213	5.9.9	[REQ-BM-AP-008].....	50
214	5.9.10	[REQ-BM-AP-009].....	50
215	5.9.11	[REQ-BM-AP-010].....	50
216	5.9.12	[REQ-BM-AP-011].....	50
217	5.9.13	[REQ-BM-AP-012].....	50
218	5.9.14	[REQ-BM-AP-013].....	51
219	5.9.15	[REQ-BM-AP-014].....	51
220	5.9.16	[REQ-BM-AP-015].....	51
221	5.9.17	[REQ-BM-AP-016].....	51
222	5.9.18	[REQ-BM-AP-017].....	51
223	5.9.19	[REQ-BM-AP-018].....	51
224	5.10	Impact minimisation	52
225	5.10.1	Introduction to impact minimisation requirements	52
226	5.10.2	[REQ-BM-IM-001].....	52
227	5.11	Minimisation of attack surfaces	52
228	5.11.1	Introduction to attack surfaces minimisation requirements.....	52
229	5.11.2	[REQ-BM-MAS-001].....	52

230	5.11.3	[REQ-BM-MAS-002]	52
231	5.11.4	[REQ-BM-MAS-003]	53
232	5.11.5	[REQ-BM-MAS-004]	53
233	5.11.6	[REQ-BM-MAS-005]	53
234	5.11.7	[REQ-BM-MAS-006]	53
235	5.11.8	[REQ-BM-MAS-007]	53
236	5.12	Exploitation mitigation mechanisms.....	54
237	5.13	Logging and monitoring	54
238	5.13.1	Introduction to logging and monitoring requirements.....	54
239	5.13.2	[REQ-BM-LOG-001]	54
240	5.13.3	[REQ-BM-LOG-002]	54
241	5.13.4	[REQ-BM-LOG-003]	55
242	5.13.5	[REQ-BM-LOG-004]	55
243	5.14	Data removal and transparency.....	55
244	5.15	Vulnerability handling	55
245	6	Conformity assessment and testing (informative).....	56
246	6.1	Introduction to the assessment and compliance criteria.....	56
247	6.1.1	Boot manager assessment considerations.....	57
248	6.1.2	Scope of assessment	57
249	6.1.3	Assessment report requirements.....	58
250	6.2	No known exploitable vulnerabilities	58
251	6.2.1	[ACC-BM-KEV-001]	58
252	6.2.2	[ACC-BM-KEV-002]	59
253	6.2.3	[ACC-BM-KEV-003]	60
254	6.3	Secure by default configuration	60
255	6.3.1	[ACC-BM-SBD-001]	61
256	6.3.2	[ACC-BM-SBD-002]	62
257	6.3.3	[ACC-BM-SBD-003]	62
258	6.3.4	[ACC-BM-SBD-004]	63
259	6.3.5	[ACC-BM-SBD-005]	64
260	6.3.6	[ACC-BM-SBD-006]	64
261	6.4	Secure updates	65
262	6.4.1	[ACC-BM-SU-001]	66
263	6.4.2	[ACC-BM-SU-002]	66
264	6.4.3	[ACC-BM-SU-003]	67
265	6.4.4	[ACC-BM-SU-004]	68
266	6.4.5	[ACC-BM-SU-005]	68
267	6.4.6	[ACC-BM-SU-006]	69
268	6.5	Authentication and access control	70
269	6.5.1	[ACC-BM-AAC-001]	70
270	6.5.2	[ACC-BM-AAC-002]	71
271	6.5.3	[ACC-BM-AAC-003]	72
272	6.5.4	[ACC-BM-AAC-004]	73
273	6.5.5	[ACC-BM-AAC-005]	73
274	6.5.6	[ACC-BM-AAC-006]	74
275	6.5.7	[ACC-BM-AAC-007]	75
276	6.6	Confidentiality	76
277	6.6.1	[ACC-BM-CON-001]	76
278	6.6.2	[ACC-BM-CON-002]	77
279	6.6.3	[ACC-BM-CON-003]	78
280	6.6.4	[ACC-BM-CON-004]	78
281	6.6.5	[ACC-BM-CON-005]	79
282	6.6.6	[ACC-BM-CON-006]	80
283	6.6.7	[ACC-BM-CON-007]	80
284	6.6.8	[ACC-BM-CON-008]	81
285	6.6.9	[ACC-BM-CON-009]	82
286	6.6.10	[ACC-BM-CON-010]	82
287	6.7	Integrity	83
288	6.7.1	[ACC-BM-INT-001]	86
289	6.7.2	[ACC-BM-INT-002]	86
290	6.7.3	[ACC-BM-INT-003]	87

291	6.7.4	[ACC-BM-INT-004].....	88
292	6.7.5	[ACC-BM-INT-005].....	88
293	6.7.6	[ACC-BM-INT-006].....	89
294	6.7.7	[ACC-BM-INT-007].....	90
295	6.7.8	[ACC-BM-INT-008].....	90
296	6.7.9	[ACC-BM-INT-009].....	91
297	6.7.10	[ACC-BM-INT-010].....	91
298	6.7.11	[ACC-BM-INT-011].....	92
299	6.7.12	[ACC-BM-INT-012].....	93
300	6.7.13	[ACC-BM-INT-013].....	93
301	6.7.14	[ACC-BM-INT-014].....	94
302	6.7.15	[ACC-BM-INT-015].....	95
303	6.7.16	[ACC-BM-INT-016].....	95
304	6.7.17	[ACC-BM-INT-017].....	96
305	6.7.18	[ACC-BM-INT-018].....	97
306	6.7.19	[ACC-BM-INT-019].....	97
307	6.7.20	[ACC-BM-INT-020].....	98
308	6.7.21	[ACC-BM-INT-021].....	99
309	6.7.22	[ACC-BM-INT-022].....	99
310	6.7.23	[ACC-BM-INT-023].....	100
311	6.7.24	[ACC-BM-INT-024].....	101
312	6.8	Data minimisation.....	101
313	6.8.1	[ACC-BM-DM-001].....	102
314	6.8.2	[ACC-BM-DM-002].....	102
315	6.9	Availability protection.....	103
316	6.9.1	[ACC-BM-AP-001].....	104
317	6.9.2	[ACC-BM-AP-002].....	105
318	6.9.3	[ACC-BM-AP-003].....	106
319	6.9.4	[ACC-BM-AP-004].....	106
320	6.9.5	[ACC-BM-AP-005].....	107
321	6.9.6	[ACC-BM-AP-006].....	108
322	6.9.7	[ACC-BM-AP-007].....	108
323	6.9.8	[ACC-BM-AP-008].....	109
324	6.9.9	[ACC-BM-AP-009].....	110
325	6.9.10	[ACC-BM-AP-010].....	110
326	6.9.11	[ACC-BM-AP-011].....	111
327	6.9.12	[ACC-BM-AP-012].....	112
328	6.9.13	[ACC-BM-AP-013].....	112
329	6.9.14	[ACC-BM-AP-014].....	113
330	6.9.15	[ACC-BM-AP-015].....	113
331	6.9.16	[ACC-BM-AP-016].....	114
332	6.9.17	[ACC-BM-AP-017].....	115
333	6.9.18	[ACC-BM-AP-018].....	115
334	6.10	Impact minimisation.....	116
335	6.10.1	[ACC-BM-IM-001].....	116
336	6.11	Minimisation of attack surfaces.....	117
337	6.11.1	[ACC-BM-MAS-001].....	117
338	6.11.2	[ACC-BM-MAS-002].....	118
339	6.11.3	[ACC-BM-MAS-003].....	119
340	6.11.4	[ACC-BM-MAS-004].....	119
341	6.11.5	[ACC-BM-MAS-005].....	120
342	6.11.6	[ACC-BM-MAS-006].....	121
343	6.11.7	[ACC-BM-MAS-007].....	121
344	6.12	Exploitation mitigation mechanisms.....	122
345	6.13	Logging and monitoring.....	122
346	6.13.1	[ACC-BM-LOG-001].....	123
347	6.13.2	[ACC-BM-LOG-002].....	124
348	6.13.3	[ACC-BM-LOG-003].....	124
349	6.13.4	[ACC-BM-LOG-004].....	125
350	6.14	Data removal and transparency.....	126
351	6.15	Vulnerability handling.....	126

352	Annex A (normative): Relationship between the present document and the essential requirements of EU	
353	Regulation 2024/2847	127
354	Annex B (informative): Security analysis	130
355	B.1 Assets	130
356	B.1.1 A-CRYPT: Cryptographic keys and certificates	130
357	B.1.2 A-ROLLBACK: Rollback protection data	130
358	B.1.3 A-CONFIG: Configuration data	130
359	B.1.4 A-AUTH: Authentication credentials	131
360	B.1.5 A-MEASURE: Measurement and attestation data	131
361	B.1.6 A-AUDIT: Audit logs	131
362	B.1.7 A-RUNTIME: Runtime state data	131
363	B.1.8 A-CODE: Boot manager executable code	132
364	B.1.9 A-UPDATE: Update packages and delivery mechanisms	132
365	B.2 Risk Factors	132
366	B.2.1 General	132
367	B.2.2 RF-SURFACE: Capability-driven attack surface	132
368	B.2.3 RF-NET: Network reachability	133
369	B.2.4 RF-AVAIL: Operational continuity	133
370	B.2.5 RF-IMPACT: Compromise severity	133
371	B.3 Assumptions	134
372	B.3.1 General	134
373	B.3.2 Hardware platform assumptions	134
374	B.3.3 Process assumptions	134
375	B.3.4 Boot target assumptions	134
376	B.4 Threats	135
377	B.4.1 Security property impacts	135
378	B.4.1.1 General	135
379	B.4.1.2 Confidentiality	135
380	B.4.1.3 Integrity	136
381	B.4.1.4 Availability	136
382	B.4.1.5 Impact of other devices	136
383	B.4.2 Threat identification methodology	137
384	B.4.2.1 Development approach	137
385	B.4.2.2 Threat sources	137
386	B.4.2.3 Categorization rationale	137
387	B.4.2.4 Limitations	137
388	B.4.3 Threat catalogue	138
389	B.4.3.1 Introduction	138
390	B.4.3.2 T-INTEGRITY: Boot integrity attacks	138
391	B.4.3.2.1 Code execution attacks	138
392	B.4.3.2.2 Configuration tampering	138
393	B.4.3.2.3 Cryptographic compromise	139
394	B.4.3.2.4 Parser and input validation attacks	139
395	B.4.3.2.5 Rollback attacks	140
396	B.4.3.3 T-PERSIST: Persistent firmware threats	140
397	B.4.3.4 T-PHYS: Physical attacks	140
398	B.4.3.5 T-SUPPLY: Supply chain attacks	141
399	B.4.3.6 T-NET: Network-based attacks	142
400	B.4.3.7 T-AVAIL: Availability and resilience threats	142
401	B.5 Mapping of risk factors to use cases	143
402	B.6 Mapping of risk factors to security profiles	144
403	B.6.1 General	144
404	B.6.2 UC-IMM: LOW	144
405	B.6.3 UC-VER: MEDIUM	144
406	B.6.4 UC-HW: HIGH	144
407	Annex C (informative): Product Documentation	145
408	C.1 Introduction	145
409	C.2 Security design documentation	145
410	C.3 Physical security documentation	146
411	C.4 Integrity mechanism documentation	146

412	C.5	Configuration security documentation.....	146
413	C.6	Vulnerability handling documentation.....	146
414	C.7	Security testing evidence	146
415	Annex E (informative): Relation to NIST SP 800-193		148
416	E.1	Purpose	148
417	E.2	Protection (NIST SP 800-193, Section 4.2).....	148
418	E.2.1	Authenticated update mechanism (NIST SP 800-193, Section 4.2.1.1).....	148
419	E.2.2	Integrity protection (NIST SP 800-193, Section 4.2.1.2).....	148
420	E.2.3	Non-bypassability (NIST SP 800-193, Section 4.2.1.3).....	149
421	E.3	Detection (NIST SP 800-193, Section 4.3).....	149
422	E.3.1	Detection of corrupted code (NIST SP 800-193, Section 4.3.1)	149
423	E.3.2	Detection of corrupted critical data (NIST SP 800-193, Section 4.3.2)	149
424	E.3.3	Notification	150
425	E.4	Recovery (NIST SP 800-193, Section 4.4).....	150
426	E.4.1	Recovery of mutable code (NIST SP 800-193, Section 4.4.1).....	150
427	E.4.2	Recovery of critical data (NIST SP 800-193, Section 4.4.2).....	150
428	E.5	Firmware categories (NIST SP 800-193).....	150
429	Annex K (normative): Generic requirements and assessment criteria for the use of state of the art		
430	cryptography V 0.0.7 (2026-04-10)		152
431	K.1	State of the Art Cryptography (CRY-SOTA)	152
432	K.1.1	Requirement.....	152
433	K.1.2	Assessment criteria	153
434	K.1.2.1	Assessment objective	153
435	K.1.2.1.1	Assessment preparation.....	153
436	K.1.2.1.2	Assessment activities	153
437	K.1.2.2	Assessment objective	153
438	K.1.2.2.1	Assessment preparation.....	154
439	K.1.2.2.2	Assessment activities	154
440	K.2	Crypto agility	156
441	K.2.1	Requirement.....	156
442	K.2.2	Assessment criteria	156
443	K.2.2.1	Assessment objective	156
444	K.2.2.1.1	Assessment preparation.....	157
445	K.2.2.1.2	Assessment activities	157
446	K.2.2.1.3	Supporting Evidence.....	157
447	K.2.2.1.4	Assignment of verdict	157
448	History		157
449			
450			

451 Intellectual Property Rights

452 Essential patents

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

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

462 Trademarks

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

468 **DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its
 469 Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and
 470 of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its
 471 Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by
 472 the GSM Association.

473 Foreword

474 This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber
 475 Security (CYBER), and is now submitted for the combined Public Enquiry and Vote phase of the ETSI
 476 Standardisation Request deliverable Approval Procedure (SRdAP).

477 The present document has been prepared under the Commission's standardisation request C(2025) 618
 478 [i.3] final to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847
 479 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity
 480 requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No
 481 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) [i.1].

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

486

Proposed national transposition dates	
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

487

488

Modal verbs terminology

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

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

493

Introduction

494 The present document is a European harmonised standard that defines technical cybersecurity
 495 requirements for boot managers as products with digital elements according to Regulation (EU) 2024/2847
 496 [i.1], the Cyber Resilience Act (CRA).

497 How to use this document

498 Clause 4 describes the product context: architecture, capabilities, operational environment, and use cases.
 499 This clause is informative and contains no requirements.

500 Clause 4.8 defines three use cases for boot manager products (UC-IMM, UC-VER and UC-HW), each
 501 associated with a security profile (LOW, MEDIUM or HIGH).

502 The manufacturer declares the use case matching the product's capabilities per Table 4.8.4-1.

503 Clause 5 contains the normative technical requirements. Requirements apply based on the declared
 504 security profile and on the functional capabilities the product implements. Clause 5.1 specifies the
 505 applicability rules.

506 Clause 6 defines assessment criteria for each requirement.

507 Annex A maps requirements in this standard to CRA essential requirements.

508 Annex B documents the security analysis underlying the requirements.

509 Annex C contains information on the product documentation.

510 Annex E provides an informative mapping to NIST SP 800-193

511 Annex K defines generic requirements and assessment criteria for the use of state of the art cryptography.
 512 This Annex is normative.

513

514

515 1 Scope

516 1.1 General

517 The present document specifies technical cybersecurity product requirements for boot managers based on
518 the Essential Cybersecurity Requirements in the Regulation (EU) 2024/2847 (Cyber Resilience Act) [i.1].

519 The scope covers software and firmware components that manage the boot process from power-on
520 through establishment of the chain of trust to handoff to the boot target.

521 Requirements apply using a risk-based approach determined by risk factors in Annex B.2 and
522 capability-based conditions defined in clause 4.3.3.

523 1.2 In-scope products

524 Products in scope include boot management software and firmware regardless of distribution model or
525 integration level. These are:

- 526 • **System firmware** that performs hardware initialization and boot management
- 527 • **Bootloaders** that manage boot target selection, verification, and loading
- 528 • **Embedded boot firmware** in IoT and embedded devices
- 529 • **Network boot implementations** enabling remote boot capabilities
- 530 • Boot managers that **integrate with hardware security components** for chain of trust establishment

531 NOTE 1: Boot managers may be single-stage (direct loading) or multi-stage (staged verification).

532 NOTE 2: For microcontrollers (MCUs) and microprocessors (MPUs):

- 533 • **Silicon-integrated immutable firmware:** Mask ROM, fused code, or boot firmware integrated
534 during chip manufacturing is assessed as part of MCU/MPU hardware under semiconductor
535 standards.
- 536 • **Updateable boot managers:** Boot software in flash storage (including OTP programmed
537 post-manufacture) is assessed using the present document when distinctly identifiable or
538 independently updatable.

539 NOTE 3: Runtime services executing after boot target handoff (such as secure monitor mode handlers
540 or attestation services) are in scope only if they provide verification or attestation services to
541 the boot process itself, not to the boot target.

542 2 References

543 2.1 Normative references

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

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

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

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

552 [1] prEN 40000-1-3: "Cybersecurity requirements for products with digital elements - Part 1-
553 3: Vulnerability Handling", (produced by CEN).

554 [2] ENISA Report 1747792503: "[European Cybersecurity Certification Group Sub-group on
555 Cryptography Agreed Cryptographic Mechanisms](#)" - version 2 - April 2025.

556 2.2 Informative references

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

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

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

564 [i.1] [Regulation \(EU\) 2024/2847](#) of the European Parliament and of the Council of 23 October
565 2024 on horizontal cybersecurity requirements for products with digital elements and
566 amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU)
567 2020/1828 (Cyber Resilience Act).

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

572 [i.3] [C\(2025\)618 - Standardisation request M/606](#): Commission Implementing decision of
573 3.2.2025 on a standardisation request to the European Committee for Standardisation
574 (CEN), the European Committee for Electrotechnical Standardisation (Cenelec) and the
575 European Telecommunications Standards Institute (ETSI) as regards products with digital
576 elements in support of Regulation (EU) 2024/2847 of the European Parliament and of
577 the Council of 23 October 2024 on horizontal cybersecurity requirements for products
578 with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020
579 and Directive (EU) 2020/1828 (Cyber Resilience Act).

580	[i.4]	EN 17927:2023: "Security Evaluation Standard for IoT Platforms (SESIP)".
581	[i.5]	ISO/IEC 15408: "Common Criteria for Information Technology Security Evaluation".
582	[i.6]	NIST SP 800-193: "Platform Firmware Resiliency Guidelines", National Institute of
583		Standards and Technology.
584	[i.7]	CAPEC™: "Common Attack Pattern Enumeration and Classification", MITRE Corporation.
585	[i.8]	MITRE EMB3D™: "Embedded Device Security Threat Model", MITRE Corporation.
586	[i.9]	CWE: "Common Weakness Enumeration", MITRE Corporation.
587	[i.10]	UEFI Specification Version 2.11 (December 2024) UEFI Specification Version 2.11, UEFI
588		Forum.
589	[i.11]	TPM 2.0 Keys for Device Identity and Attestation v1.10, Trusted Computing Group.
590	[i.12]	DICE Layering Architecture Version 1.0, Trusted Computing Group.
591	[i.13]	DICE Attestation Architecture v1.2, Trusted Computing Group.
592	[i.14]	ETSI TS 104 875 V0.0.7 (2026-04): Cyber Security (CYBER); Hardware-Based Root of Trust
593		Specification.
594		

595 3 Definition of terms, symbols and abbreviations

596 3.1 Terms

597 For the purposes of the present document, the terms given in Regulation (EU) 2024/2847 [i.1] and the
598 following apply:

599 **attestation:** Process of providing cryptographic evidence about boot state, configuration, and
600 measurements to a local or remote verifier.

601 NOTE: Attestation enables trust decisions based on demonstrated boot integrity without necessarily
602 preventing execution of unverified code.

603 EXAMPLE: Attestation may include measurement logs, signed attestation statements, and cryptographic
604 proofs of boot component identity

605 **authorised entity:** person, system, or process permitted to perform security-relevant operations on the
606 boot manager, including configuration changes, trust anchor modification, update installation, and recovery
607 mode activation.

608 NOTE: An authorised entity may be a manufacturer, integrator, administrator, or end user depending on the
609 lifecycle phase and the access control policy configured for the boot manager.

610 **boot component:** any executable code, firmware module, configuration data, or cryptographic material
611 loaded and used during the boot process

612 **boot device:** any storage medium or interface from which boot code can be loaded and executed, including
613 internal storage, removable media, or network sources.

614 **boot managers:** Software or firmware component that controls the boot process and allows management
615 of multiple boot targets or boot configurations after power-on or reset.

616 NOTE: This term includes both single-purpose bootloaders and multi-function firmware with boot
617 management capabilities.

618 **boot sequence:** ordered series of operations from power-on through hardware initialization, boot manager
619 execution, and handoff to the boot target

620 **boot source:** Logical origin from which boot code can be loaded and executed.

621 EXAMPLE: boot devices, network interfaces, and debug interfaces.

622 NOTE: Each source presents different security boundaries and threat exposures.

623 **boot target:** software component to which the boot manager transfers control upon successful completion
624 of its designated function, such as an operating system kernel, hypervisor, another bootloader, or an
625 embedded application

626 **chain of trust:** sequential verification where each boot component validates the next, also known as
627 transitive trust or trust chain, establishing security from hardware root to the boot target

628 **certificate:** A data structure that associates a public key with an identity, optionally including usage
629 constraints, validity period, and other attributes.

630 NOTE: A certificate may be signed by a certification authority or may be self-signed. In the context of
631 the present document, "certificate" refers to both ITU-T X.509-compliant certificates and
632 minimal public key serializations, as specified in the relevant requirements.

633 **critical data:** data whose corruption would prevent successful boot completion or compromise security
634 verification, including boot code, verification keys, anti-rollback counters, and security policy settings

635 **defence-in-depth:** security strategy employing multiple, overlapping, and diverse protection mechanisms so
636 that compromise of any single mechanism does not result in complete security failure.

637 NOTE: In the present document, "threat" is used synonymously with "cyber threat" as defined in
638 Regulation (EU) 2024/2847 [i.1].

639 **enforcement:** active prevention of unauthorised actions through cryptographic verification and blocking of
640 execution when verification fails, as distinct from detection-only approaches that record but do not prevent
641 execution

642 **essential operation:** operation required for the boot manager to complete its primary function of loading
643 and transferring control to a boot target

644 **evaluation activity:** assessment procedure addressing a group of related requirements

645 **firmware:** low-level software stored in non-volatile memory on the device/product that provides hardware
646 initialization and runtime services, including boot managers

647 **firmware runtime services:** functions provided by boot manager firmware that remain accessible to the
648 operating system after boot handoff, typically including variable storage, time services, and system
649 management functions

650 **immutable component:** component whose code cannot be modified after manufacture or initial
651 programming, typically implemented in ROM (Read-Only Memory), OTP (One-Time Programmable) storage,
652 e-fuses, or write-protected flash memory

653 **handoff:** transfer of execution control from the boot manager to the boot target, marking the completion of
654 the boot manager's function and the beginning of boot target execution

655 **hardware security component:** generic term for hardware-based security modules that provide
656 cryptographic services, secure storage, or isolated execution environments

657 NOTE: TPM and HSM are examples of hardware security components. The generic term 'hardware
658 security component' is used throughout the present document unless referring to specific
659 implementation examples.

660 **measured boot:** recording cryptographic measurements of boot components before execution for
661 attestation without blocking execution

662 **monotonic counter:** counter value that can only increase, never decrease, used for anti-rollback protection
663 to make version downgrade detectable

664 **network boot:** loading boot components from network sources using protocols such as PXE or HTTPS Boot

665 **physical presence:** Requirement for a user to be physically at the device location with direct physical access
666 to the device to perform certain security-sensitive operations, typically verified through local keyboard
667 input or button presses

668 NOTE: Physical presence can also encompass operations conducted in a controlled environment such
669 as a manufacturing plant.

670 **recovery boot:** specialized boot process designed to restore system functionality after failures

671 **rollback protection:** security mechanism preventing unauthenticated installation of older firmware versions
672 that may contain known vulnerabilities

673 **Root of Trust:** Component providing foundational trust for the system through hardware-protected
674 execution and storage

675 NOTE: Root of Trust is typically implemented in immutable ROM, hardware security modules, or
676 physically isolated processors. Serves as the anchor for chain of trust verification.

677 **secure boot:** specific UEFI implementation of verified boot as defined in the UEFI Specification

678 **security-critical configuration:** configuration settings whose modification would reduce security posture,
679 including verification enable/disable settings, trust anchor databases, certificates, user-enrolled keys, boot
680 order, and recovery mode settings

681 **sensitive data:** Data requiring protection from disclosure

682 EXAMPLE: Sensitive data include cryptographic private keys and symmetric keys.

683 NOTE: Unauthenticated access to sensitive data enables impersonation, forgery, or decryption of
684 protected content.

685 **trust anchor:** public key, certificate, or cryptographic hash that serves as the starting point for verification
686 and is trusted without requiring further validation

687 **update capability:** ability to modify boot manager code, configuration, or security policies after initial
688 deployment, ranging from configuration-only updates to full firmware replacement

689 **verified boot:** cryptographic verification of boot components before execution to ensure only authorised
690 code runs during system initialization

691 NOTE: Implementations verify authenticity using cryptographic signatures or authenticated hashes bound to
692 cryptographic keys or hardware roots of trust

693 3.2 Symbols

694 Void.

695 3.3 Abbreviations

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

697	BIOS	Basic Input/Output System
698	CDI	Compound Device Identity
699	CRC	Cyclic Redundancy Check
700	CRL	Certificate Revocation List
701	DICE	Device Identifier Composition Engine
702	DMA	Direct Memory Access
703	HSE	Hardware Security Enforcement
704	HSM	Hardware Security Module
705	HSS	Hardware Security Storage
706	MAC	Message Authentication Code
707	NVRAM	Non-Volatile Random Access Memory
708	OCSP	Online Certificate Status Protocol
709	OS	Operating System
710	OTP	One-Time Programmable
711	PCR	Platform Configuration Register
712	PXE	Preboot Execution Environment
713	ROM	Read-Only Memory
714	SBOM	Software Bill of Materials
715	TEE	Trusted Execution Environment
716	TOCTOU	Time-Of-Check-Time-Of-Use
717	TPM	Trusted Platform Module
718	UEFI	Unified Extensible Firmware Interface

719 4 Product description (informative)

720 4.1 Intended purpose and reasonably foreseeable use

721 4.1.1 Intended purpose

722 Boot managers establish initial system trust and manage the boot process from power-on through handoff
723 to the boot target. They serve as the foundational trust component that executes before operating systems
724 or security services are available.

725 The intended purpose includes:

- 726 • Establishing the initial chain of trust for the system.

- 727 • Loading and transferring control to boot targets (such as another bootloader, operating systems,
728 hypervisors, or embedded applications).
- 729 • Managing boot configuration and security policy.

730 Where capable it also includes:

- 731 • Verifying integrity of boot components.
- 732 • Authentication of boot components.
- 733 • Recording measurements for attestation.
- 734 • Providing recovery mechanisms.

735 4.1.2 Reasonably foreseeable use

736 Boot managers are components that may be:

- 737 • Integrated into diverse hardware platforms across all industry verticals.
- 738 • Deployed in security contexts ranging from consumer devices to critical infrastructure.
- 739 • Configured by users with varying levels of technical expertise.
- 740 • Operated in environments ranging from physically secure to hostile and untrusted.
- 741 • Transferred to secondary markets without factory reset.
- 742 • Operated beyond the manufacturer's active support lifecycle.
- 743 • Combined with other components in composite products.
- 744 • Subjected to configuration changes by downstream integrators.
- 745 • Used in safety-critical systems where availability requirements may conflict with security update
746 procedures.

747 These reasonably foreseeable uses inform the requirements in clause 5 and the risk analysis methodology in
748 Annex B.

749 Reasonably foreseeable deployment contexts determine risk level determination per Annex B.2. Boot
750 managers are commonly composed of components from multiple sources including silicon vendors,
751 independent BIOS vendors, OEMs, and open-source projects. Each component may have distinct update
752 mechanisms, vulnerability response capabilities, and maintenance lifecycles.

753 4.1.3 Reasonably foreseeable misuse

754 Reasonably foreseeable misuse includes use of the boot manager in ways not intended but predictable from
755 human behaviour or system interactions. Examples include:

- 756 • Disabling security features for convenience
- 757 • Using development/debug modes in production
- 758 • Operating without applying available security updates
- 759 • Connecting to untrusted networks or storage

760 Requirements in Clause 5 address these scenarios through secure defaults and fail-safe behaviour.

761 4.2 Product functions

762 Boot managers establish system trust and load the boot target. This process includes verifying or measuring
763 boot components, managing boot configuration, and transferring execution control to the selected target.

764 Security functions vary by design. Verified boot implementations block execution of unverified code through
765 cryptographic signature verification. Measured boot implementations record cryptographic measurements
766 of boot components for attestation purposes. Both approaches protect boot configuration from
767 unauthorised modification and may include rollback protection against version downgrade attacks.

768 Configuration capabilities determine how boot behaviour can be modified. Some implementations have
769 fixed behaviour determined at manufacturing. Others allow modification of security policies during
770 subsequent boots, cryptographic credentials, and boot target selection. Boot managers with configuration
771 capability involves access controls to prevent unauthorised changes.

772 Lifecycle management includes update, recovery, and reset functions. Updates modify boot manager
773 firmware or configuration post-deployment. Recovery functions provide alternative boot paths when
774 primary operations fail. Reset capabilities support secure decommissioning by erasing cryptographic
775 material and restoring default configurations, where technically feasible. However, when cryptographic
776 material or configuration data are stored in immutable components such as e-fuses or ROM, complete
777 erasure or restoration may not be possible.

778 4.3 Architecture

779 4.3.1 Boot architecture

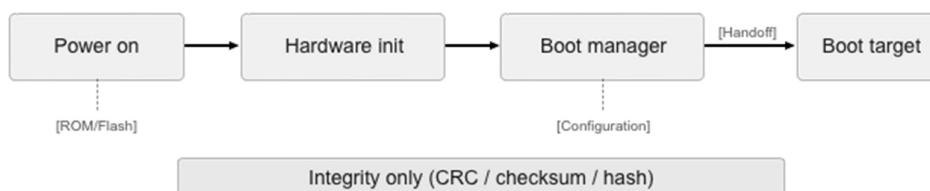
780 4.3.1.1 Unverified boot

781 Unverified Boot implements sequential stage execution without cryptographic verification mechanisms.

782 The system begins with power-on, proceeds through hardware initialization, and continues with sequential
783 stage execution where each stage directly loads and executes the next until the boot target is reached.

784 Unverified boot implements no cryptographic authentication mechanisms, but includes simple integrity
785 checks such as CRC, checksums or hashes. These mechanisms may detect accidental corruption but do not
786 provide protection against deliberate modification or substitution attacks.

787 Unverified boot is appropriate for environments where physical security controls provide the primary
788 protection, or for device categories (e.g. sensors) where risk assessment determines that the assets stored
789 or transmitted do not warrant verified boot implementation.



790

791 4.3.1.2 Verified boot

792 Verified boot implements cryptographic verification with enforcement throughout the boot process.
793 Components are cryptographically verified before execution, creating a verification chain from power-on
794 through boot target execution.

795 NOTE: Verification may be performed either immediately before execution (on-boot verification) or
796 prior to installation with integrity-protected storage retained between updates (on-update
797 verification, also referred to as capsule update). Both patterns establish that the code executed
798 on the device has been cryptographically verified at some point in time before its execution. A
799 boot manager may implement either pattern or a combination of both.

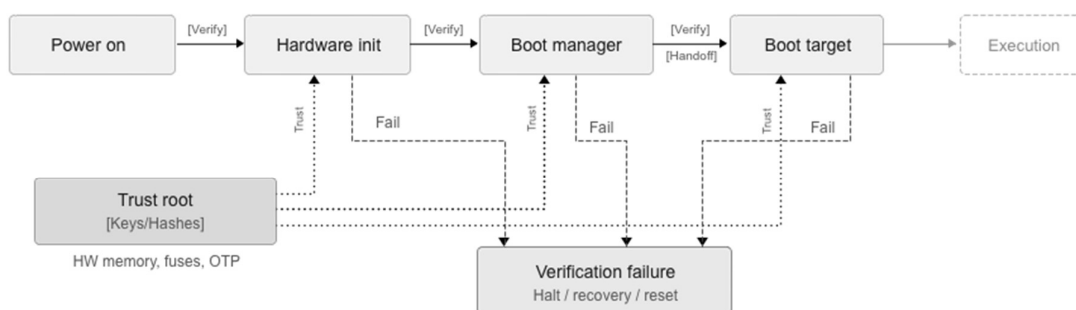
800 Components are verified against stored cryptographic signatures or authenticated hash values and if
801 verification fails the system halts the boot process and potentially triggers recovery mode. Initial verification
802 material (keys or hashes) may be stored in hardware-protected memory, fuses, or one-time programmable
803 storage. Systems implementing verified boot need cryptographic verification capabilities and secure storage
804 for verification material. Verified boot employs cryptographic authentication mechanisms using either:

- 805 • Asymmetric cryptography: Digital signatures verified using public keys stored in the trust root (e.g.
806 RSA, ECDSA, EdDSA signatures)
- 807 • Symmetric cryptography: Message Authentication Codes (MACs) or authenticated encryption
808 verified against pre-shared secrets (e.g. HMAC, CMAC, AES-GCM)
- 809 • One-way functions, such as cryptographic hash functions, which are a necessary component in
810 signature systems, and map data and public keys to protected trust anchors

811 NOTE: See Annex K for approved cryptographic mechanisms and deprecation schedules.

812 Both approaches provide cryptographic assurance that boot components have not been modified and
813 originate from a trusted source. When verification fails, the system halts execution and may transition to a
814 degraded secure state, trigger recovery mode, or perform a power reset to prevent execution of
815 unauthorised code.

816 The choice between asymmetric and symmetric authentication (e.g. authenticated hashes), and hash-based
817 verification depends on the threat model, key management capabilities, available cryptographic hardware
818 support, and performance requirements particularly in safety-critical contexts where boot time is
819 constrained.



820
821

822 4.3.1.3 Measured boot

823 Measured boot records cryptographic measurements of each boot stage without preventing execution.
824 During initialization, the system activates measurement capabilities and records a cryptographic hash of
825 each subsequent stage into secure storage.

826 Each stage computes a cryptographic hash of the next component and stores or extends it into designated
827 registers, and the boot process continues regardless of measurement values. The resulting measurement
828 log provides an attestable record of the boot sequence that can be verified against known-good values or
829 reference measurements.

830 This architecture relies on hardware or firmware capable of secure measurement process, storage and
831 reporting. The fundamental difference from verified boot is in the enforcement policy. The system records
832 measurements in the log but continues the boot process regardless of values. This allows:

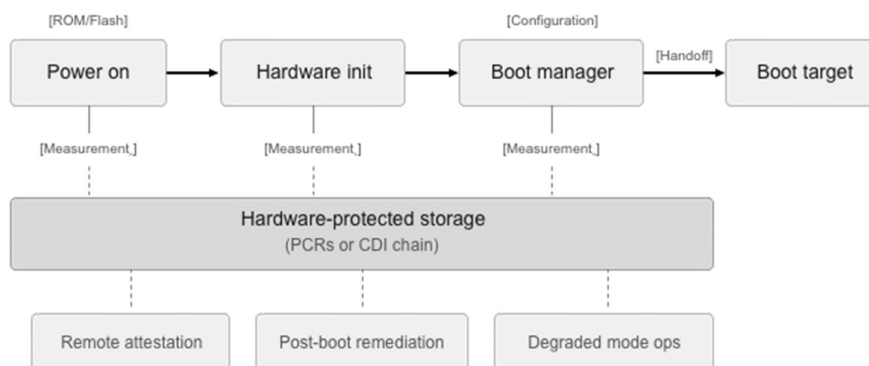
- 833 • Attestation of the actual boot state to remote verifiers.
- 834 • Post-boot analysis and remediation decisions.
- 835 • Operation in degraded mode with logged anomalies.
- 836 • Flexibility for development and recovery scenarios.
- 837 • Execute parts of the system software only if the measured stages are valid

838 The measurement log provides a cryptographically protected, auditable record that can be verified locally or
839 remotely against known-good reference values or security policies.

840 Measured boot employs the same cryptographic authentication mechanisms as verified boot.

841 **EXAMPLE:** The diagram below illustrates the general measured boot architecture.

842 Each boot stage computes a cryptographic hash and records it into hardware-protected storage
843 (hardware-based Root of Trust), which may be implemented using a variety of technologies including PCRs
844 in a TPM, CDI derivation in DICE, or other secure measurement storage mechanisms.



845

846 4.3.2 Functional extensions

847 Boot managers may provide additional functions beyond core boot architecture. These functions are
848 classified as either security-enhancing or functional (see clause 4.3.3).

849 Security-enhancing capabilities improve the security posture of the boot manager and characterise the use
850 cases defined in clause 4.8.

851 Functional capabilities provide features that may be required by the product's intended purpose but
 852 introduce additional attack surface. Where a functional capability is implemented, the requirements in
 853 clause 5 that reference that capability apply. Both types of capability can be combined with any boot
 854 architecture (unverified, verified, or measured).

855 4.3.3 Product capabilities

856 4.3.3.1 Introduction

857 The following subclauses describe each capability. Table 4.3.3-1 summarises the classification defined in
 858 clause 4.3.2. Where applicability statements in clause 5 refer to boot managers with a particular capability,
 859 they apply to boot managers implementing any option listed in the corresponding sub-clause other than the
 860 "no [capability]" option, unless the sub-clause defines a different threshold.

861 **Table 4.3.3-1: Capability classification**

Security-enhancing	Functional
Verified boot (4.3.1.2)	Network boot (4.3.3.3)
Update capability (4.3.3.2)	Configuration management (4.3.3.3)
Logging (4.3.3.7)	Measured boot (4.3.1.3)
Hardware security (4.3.3.5)	Recovery capability (4.3.3.6)
Key provisioning (4.3.3.10)	Authentication capability (4.3.3.8)
	Debug interface (4.3.3.9)

862

863 4.3.3.2 Update capability

864 Update capability describes whether and how the boot manager can be modified after initial deployment.

- 865 • **No updates:** Boot manager code and configuration cannot be modified after manufacture.
- 866 • **Configuration updates:** Only boot configuration, security policies, and trusted keys can be updated.
867 Boot manager code remains immutable.
- 868 • **Partial updates:** Specific components can be updated while core components remain immutable.
869 Root of trust and initial boot stages cannot be modified.
- 870 • **Full updates:** Complete boot manager firmware can be updated including all boot stages.
- 871 • **Runtime updates:** Boot manager can be updated while system is running without requiring reboot
872 until next boot cycle.
- 873 • **Recovery-mode updates:** Updates can only be applied through a separate recovery mode or
874 alternate boot path.
- 875 • **Network updates:** Boot manager can fetch and apply updates directly from network sources.
- 876 • **Enclave-managed updates:** Updates are managed by a separate security processor or trusted
877 execution environment.

878 Products may combine multiple update capabilities (e.g. configuration updates via network, full updates via
 879 recovery mode), except products with no update capability. For the purposes of this document, references
 880 to "update capability" in clause 5 apply to boot managers implementing partial updates, full updates,
 881 runtime updates, recovery-mode updates, network updates, or enclave-managed updates. Boot managers
 882 implementing only "no updates" or "configuration updates" are not considered to have update capability.

883 4.3.3.3 Boot source

884 Boot source identifies media types from which the boot manager can load and execute code.

- 885 • **Internal storage:** Boot exclusively from non-volatile storage integrated within the same package or
886 die as the processor.
- 887 • **Fixed storage:** Boot exclusively from non-removable storage physically attached to the system but
888 in a separate package.
- 889 • **Removable storage:** Can boot from removable storage devices such as USB, SD cards, or optical
890 media.
- 891 • **Network boot:** Can boot from network sources including PXE, HTTP, iSCSI, or cloud services.
- 892 • **External expansion:** Can boot from external expansion buses such as Thunderbolt or ExpressCard.
- 893 • **Debug interfaces:** Can boot or load code via debug or programming interfaces such as JTAG, SWD,
894 or UART.

895 Multiple boot sources are common. Each additional source increases attack surface.

896 NOTE: Requirements referencing "network boot" apply when the boot manager includes Network
897 boot capability. Requirements referencing "network update" apply when the boot manager
898 includes Network updates under update capability. These may overlap but are distinct:
899 network boot loads the entire boot image from network while network update modifies
900 persistent boot manager components.

901 4.3.3.4 Configuration management

902 Configuration capability describes whether boot behaviour can be modified through settings.

- 903 • **No configuration:** Fixed behaviour with no configurable settings, or configuration set once via
904 one-time programmable mechanisms.
- 905 • **Static configuration:** Configuration can be set but involves physical access or special tools to modify.
- 906 • **Local configuration:** Configuration modifiable through local interfaces requiring physical presence.
- 907 • **Protected configuration:** Configuration modifiable with authentication but without physical
908 presence requirement.
- 909 • **Remote configuration:** Configuration modifiable through network interfaces.
- 910 • **Runtime configuration:** Configuration modifiable by the operating system through runtime
911 services.

912 For the purposes of this document, references to "configuration capability" in clause 5 apply to boot
913 managers implementing local, protected, remote, or runtime configuration. Boot managers implementing
914 only "no configuration" or "static configuration" are not considered to have configuration capability.

915 4.3.3.5 Hardware security

916 Hardware security describes whether the boot manager utilizes dedicated security hardware.

917 **Software only:** Boot manager operates entirely in software without dedicated security hardware.

918 **Hardware storage:** Utilizes hardware components for protected key storage and cryptographic operations.

919 **Hardware enforcement:** Hardware actively enforces security policies during boot.

920 Hardware enforcement typically includes hardware storage.

921 4.3.3.6 Recovery capability

922 Recovery capability describes how the system can recover from boot failures, corruption, or attacks.

- 923 • **No recovery:** No separate recovery path or fallback boot mechanism.
- 924 • **Basic recovery:** Simple recovery mode or fallback boot option such as safe mode or previous
925 known-good configuration.
- 926 • **Full recovery:** Comprehensive recovery environment with diagnostics, repair tools, and restore
927 capabilities.

928 Recovery mechanisms provide resilience but introduce additional attack surface.

929 4.3.3.7 Logging capability

930 Logging capability describes whether and how the boot manager can record, store, and communicate
931 security-relevant events and operational states during the boot process. The presence, type, and durability
932 of logging functions depend on the boot manager's architecture, resource constraints, and deployment
933 context.

- 934 • **No logging:** The boot manager does not implement any logging functionality. No events or states
935 are recorded during boot in persistent storage.

936 NOTE 1: This variant is typical for resource-constrained devices or early boot stages where logging
937 infrastructure is not available.

- 938 • **Volatile logging:** The boot manager records events in volatile memory during boot. Logs are lost
939 upon power cycle or reset and are accessible only during the current boot session.

940 NOTE 2: Requirements referring to persistent log retrieval cannot be fulfilled when only volatile logging
941 is implemented.

- 942 • **Persistent logging:** The boot manager records events in persistent storage (e.g. flash, NVRAM,
943 dedicated log partition). Logs survive power cycles and can be accessed after reboot.
- 944 • **Hardware-protected logging:** The boot manager utilizes hardware security components (e.g. TPM,
945 secure element, write-once memory) to protect log integrity and prevent tampering.
- 946 • **External logging:** The boot manager exports log events to external interfaces (e.g. serial console,
947 debug ports) for real-time monitoring or remote analysis.

948 Logging capability enables security monitoring and forensic analysis but uses storage resources.

949 4.3.3.8 Authentication capability

950 Authentication capability describes whether the boot manager involves identity verification for privileged
951 operations.

- 952 • **No authentication:** No identity verification for any operations.

- 953 • **Physical presence:** Operations require physical presence detection (button press, jumper) but no
954 credential verification.
- 955 • **Password authentication:** Operations require password or PIN verification.
- 956 • **Cryptographic authentication:** Operations require cryptographic credential verification (smart card,
957 certificate, hardware token).
- 958 • **Multi-factor authentication:** Operations require multiple authentication mechanisms.

959 Authentication capability protects configuration and security policy changes from unauthorised
960 modification.

961 4.3.3.9 Debug interface

962 Debug interface describes whether the boot manager exposes diagnostic or programming interfaces.

- 963 • **No debug interface:** No debug, diagnostic, or programming interfaces present in production.
- 964 • **Disabled debug:** Debug interfaces physically present but permanently disabled or fused off.
- 965 • **Authenticated debug:** Debug interfaces require authentication before activation.
- 966 • **Conditional debug:** Debug interfaces available only in specific modes (e.g. manufacturing,
967 recovery).
- 968 • **Open debug:** Debug interfaces accessible without restriction.

969 Debug interfaces enable development and diagnostics but significantly increase attack surface when
970 accessible. For the purposes of this document, references to "debug capability" in clause 5 apply to boot
971 managers implementing authenticated, conditional, or open debug interfaces. Boot managers with no
972 debug interface, or with permanently disabled debug interfaces, are not considered to have debug
973 capability.

974 4.3.3.10 Key provisioning

975 Key provisioning describes whether and how trust anchors used for boot verification can be provisioned by
976 the integrator of the boot manager.

- 977 • **No key provisioning:** Trust anchors are fixed at manufacture and cannot be modified.
- 978 • **Manufacturing provisioning:** Trust anchors are provisioned during manufacturing or integration
979 using dedicated tools or interfaces.
- 980 • **Updateable provisioning:** Trust anchors can be modified post-deployment through authenticated
981 mechanisms.

982 Key provisioning does not imply that end users can modify trust anchors. It describes a mechanism available
983 to the integrator. Whether the integrator exposes trust anchor management to downstream entities is
984 outside the scope of this standard.

985 4.3.4 Resource constraints

986 Boot managers operate under inherent resource limitations determined by their implementation and
987 hardware platform:

- 988 • **Storage:** Restricted firmware storage capacity shared with other components.
 - 989 • **Computation:** No OS services, drivers, or shared libraries during early stages.
 - 990 • **Time:** Boot performance requirements limiting security operations.
 - 991 • **Memory:** Limited RAM available during early boot stages.
- 992 Resource constraints influence implementation choices and inform requirement applicability.

993 4.4 Operational environment

994 4.4.1 Introduction

995 The operational environment describes deployment context affecting risk factor assessment per Annex B.2.
996 Environmental characteristics are independent of product characteristics and describe where and how the
997 boot manager is deployed.

998 4.4.2 Physical security context

- 999 • **Secure:** Physically secure locations (data centres, secure facilities).
- 1000 • **Controlled:** Controlled access environments (offices, industrial sites).
- 1001 • **Restricted:** Limited public access with some monitoring (hospitals, retail).
- 1002 • **Public:** Publicly accessible with minimal security (kiosks, ATMs, digital signage).
- 1003 • **Hostile:** Uncontrolled, adversarial environments (consumer devices, field deployment).
- 1004 • **Mobile:** Travels through varying physical environments (laptops, smartphones, tablets - should
1005 assume hostile).

1006 4.4.3 Network exposure

1007 Boot managers may be deployed with varying levels of network exposure:

- 1008 • **None (air-gapped):** No network connectivity.
- 1009 • **Isolated:** Private/isolated networks (LAN, industrial networks).
- 1010 • **Managed:** Internet-connected within managed environment (VPN, enterprise).
- 1011 • **Internet:** Direct internet exposure.
- 1012 • **Cellular:** Mobile/cellular networks.

1013 4.4.4 Data sensitivity and value

1014 The sensitivity and value of data accessible through or protected by the boot manager affects attacker
1015 motivation and consequence severity:

- 1016 • **Minimal:** No significant sensitive data (basic sensors, simple IoT).
- 1017 • **Personal:** Personal user data (consumer devices, personal files).

- 1018 • **Financial:** Financial/payment data (payment terminals).
- 1019 • **Business:** Business confidential data (enterprise systems, intellectual property).
- 1020 • **Health:** Protected health information (medical devices, health records).
- 1021 • **Critical:** Critical infrastructure control (power grid, water systems).
- 1022 • **Classified:** Government classified information.

1023 4.4.5 System criticality

1024 Impact severity if the boot manager is compromised, considering availability, integrity, and safety:

- 1025 • **Non-critical:** Low business impact, non-critical functions, general purpose devices.
- 1026 • **Business impact:** Moderate business impact, internal systems, productivity loss, reputation
1027 damage.
- 1028 • **Safety-critical:** Systems where compromise causes physical harm, critical infrastructure disruption,
1029 major financial loss or widespread service disruption.

1030 4.4.6 Lifecycle expectations

- 1031 • **Short:** < 5 years.
- 1032 • **Medium:** 5 to 15 years.
- 1033 • **Long:** > 15 years.

1034 4.4.7 Administrative control

1035 The model for managing configuration and updates:

- 1036 • **Centrally managed:** IT department or security operations centre controls all configuration and
1037 updates, security policies centrally enforced, compliance monitoring in place.
- 1038 • **User managed:** Individual end-users control configuration and updates. Varying security expertise,
1039 inconsistent update adoption.
- 1040 • **Vendor managed:** Manufacturer retains control over updates and configuration. User cannot
1041 modify boot security settings.
- 1042 • **Hybrid managed:** Shared responsibility between multiple parties, needs clear responsibility
1043 documentation.
- 1044 • **Autonomous:** No administrative interface, device operates independently. Updates require physical
1045 access or hardware replacement.

1046 4.4.8 Availability requirements

- 1047 • **Flexible:** Downtime acceptable for updates/maintenance.
- 1048 • **Standard:** Normal business availability expectations.
- 1049 • **High:** High uptime required.

- 1050 • **Critical:** Continuous operation required for safety or critical infrastructure.

1051 4.4.9 Additional environmental characteristics

1052 4.4.9.1 Environmental stress

1053 Harsh environmental conditions affect boot manager reliability and security:

- 1054 • **Temperature extremes:** Harsh thermal environments affecting non-volatile memory reliability and
1055 boot component integrity.

- 1056 • **Vibration and shock:** Mechanical stress causing storage connection failures and data corruption
1057 risks.

- 1058 • **Radiation:** Environments with ionizing radiation causing storage bit flips requiring error correction
1059 and redundancy.

- 1060 • **Humidity and corrosion:** Outdoor deployment, marine environments affecting hardware integrity.

- 1061 • **Aging effects:** Flash wear (limited write cycles), bit rot, electromigration degradation over extended
1062 lifecycles.

1063 These stresses increase probability of boot failures, requiring enhanced error detection, redundant storage,
1064 and robust recovery mechanisms. See REQ-BM-AP-006 and REQ-BM-AP-007.

1065 4.4.9.2 Update frequency expectations

1066 Boot manager update patterns vary by deployment:

- 1067 • **Never:** ROM-based, OTP firmware, no post-manufacture updates possible.
- 1068 • **Rare:** Major version updates only (e.g. every 2 to 5 years) for long-lifecycle industrial equipment.
- 1069 • **Periodic:** Scheduled maintenance windows (quarterly, annually) for enterprise/infrastructure.
- 1070 • **Responsive:** Updates deployed when vulnerabilities discovered, testing-dependent timeline.
- 1071 • **Continuous:** Rapid update capability for internet-connected consumer devices.

1072 Update frequency affects vulnerability exposure window and influences defence-in-depth requirements and
1073 may affect also the lifetime of memory used. See Annex B.2.2 (RF-SURFACE).

1074 4.4.9.3 Interfaces

1075 Boot managers interact through various technical interfaces:

- 1076 • **Storage:** SPI flash, USB, eMMC, NVMe, SATA with different protection mechanisms.
- 1077 • **Network:** Ethernet, WiFi supporting PXE, HTTP Boot, iSCSI protocols.
- 1078 • **Debug:** JTAG, SWD, UART, USB, SPI programmers requiring access control.
- 1079 • **User:** Serial console, display output, keyboard, remote management (IPMI/BMC).
- 1080 • **Hardware:** Hardware security component connections (SPI/I2C or other protocols).
- 1081 • **Runtime:** Firmware runtime services, ACPI tables, SMM/SMI interfaces.

1082 Boot managers receive inputs such as hardware reset signals, configuration data, boot media, network
1083 packets, or user input, and produce outputs including system state, measurement logs, or error codes.
1084 Control interfaces include configuration utilities, update mechanisms, recovery modes, and remote
1085 management protocols.

1086 The interface types listed above are comprehensive. The specific protocols and technologies listed under
1087 each type are representative examples; additional interfaces assignable to these types are subject to the
1088 same security considerations described in Clause 5.

1089 4.5 Distribution of security functions

1090 4.5.1 Introduction

1091 As a component within a larger system, boot managers both provide and depend upon security functions.
1092 This clause clarifies the security boundaries and responsibilities.

1093 4.5.2 Security functions provided by the boot manager

1094 Depending on capabilities, boot managers provide:

- 1095 • Initial trust establishment before any other software executes.
- 1096 • Boot chain integrity verification.
- 1097 • Cryptographic measurement recording for attestation.
- 1098 • Secure handoff to boot target.
- 1099 • Protection of boot configuration and policy.
- 1100 • Rollback protection for boot components.
- 1101 • Root of trust for measurement.
- 1102 • Root of trust for verification.
- 1103 • Root of trust for storage.
- 1104 • Root of trust for reporting.

1105 NOTE: Verification, measurement, and rollback capabilities are implementation dependent.

1106 4.5.3 Security functions required from the platform

1107 The boot manager may depend on the following security functions from the hardware platform. These
1108 dependencies are documented as assumptions in Annex B.3.2. Where a platform does not provide a listed
1109 function, the corresponding threats identified in Annex B.3.2 are not mitigated by this standard.

- 1110 • Hardware root of trust (immutable boot ROM or equivalent)
- 1111 • Secure storage for cryptographic keys and configuration
- 1112 • Protected execution environment (where available)
- 1113 • Hardware security components for key storage (where available)

- 1114 • Physical security mechanisms (where available)

1115 NOTE: Hardware security mechanisms may include various implementations such as TPM, HSM,
1116 secure elements, or write protection mechanisms that prevent tampering of boot manager
1117 code by the operating system or other software executing after boot handoff.

1118 4.5.4 Security functions delegated to boot target

1119 The following security functions are outside the scope of the boot manager and are the responsibility of the
1120 loaded boot target:

- 1121 • Runtime security monitoring and threat detection
- 1122 • User authentication and authorisation (post-boot)
- 1123 • Network security and firewall functions
- 1124 • Application security and sandboxing
- 1125 • Ongoing vulnerability management and patching
- 1126 • Security event logging (post-handoff)

1127 NOTE: Some boot managers provide runtime services that remain accessible after boot target handoff
1128 (e.g. UEFI runtime services for variable storage, time, capsule updates or attestation). These
1129 runtime services are within scope when they affect boot security, integrity, or configuration.

1130 4.5.5 Trust boundaries

1131 Boot managers operate across multiple trust boundaries where different security domains interface.

1132 Boot managers interface with multiple trust domains, each requiring different verification approaches, for
1133 example root of trust by hardware vendor, security hashes for different stages by firmware or board OEM,
1134 or certificates by operating system vendors.

- 1135 • **Hardware:** Components the boot manager trusts without verification capability, including CPU
1136 microcode, platform security processors providing opaque services, immutable boot ROM and/or
1137 e-fuses.
- 1138 • **Storage:** Components requiring verification before use, including all data read from storage media,
1139 configuration data, and boot components that may have been tampered with or corrupted.
- 1140 • **Network:** Untrusted input requiring complete verification, including all network-sourced boot
1141 components, protocol data, and remote configuration.
- 1142 • **Operating system:** Components to which the boot manager transfers control, including operating
1143 systems, hypervisors, and runtime services. This boundary is bidirectional when the operating
1144 system performs updates to the boot manager.

1145 These boundaries define where boot managers can enforce security properties versus where they rely
1146 instead on trust assumptions or external security mechanisms.

1147 4.6 Users

1148 4.6.1 Introduction

1149 Boot managers operate mostly without any user interaction during normal operation, with security
1150 decisions predetermined by configuration rather than runtime user input. In the requirements of clause 5,
1151 "authorised entity" is used in place of "user" to avoid ambiguity between the CRA definition of user and
1152 technical access roles. See clause 3.1.

1153 4.6.2 User categories

1154 Boot managers interact with different user types across their lifecycle:

- 1155 • Manufacturers: during production and initial provisioning.
- 1156 • System integrators: during deployment.
- 1157 • System administrators: for enterprise configuration.
- 1158 • End users: for boot selection.
- 1159 • Service technicians: during maintenance.

1160 4.6.3 User interaction patterns

- 1161 • Manufacturing phase: Fuse programming, key provisioning, initial firmware installation, testing.
- 1162 • Configuration phase: Use of setup and configuration tools
- 1163 • NOTE: In some business models, operations such as fuse programming or key provisioning may be
1164 deferred from the manufacturing phase to the configuration phase, allowing OEMs to customise or
1165 finalise device configuration post-manufacture.
- 1166 • Boot phase: Physical presence detection, boot target selection, recovery mode activation.
- 1167 • Maintenance phase: Updates, rollbacks, resets.
- 1168 • Decommissioning phase: Secure erasure, memory clean-up, key destruction, decommissioning
1169 verification.

1170 These interactions represent points where boot manager security can be modified, compromised or
1171 bypassed, requiring appropriate access controls and authentication mechanisms.

1172 Interactions requiring special attention:

- 1173 • Device resale or transfer.
- 1174 • Return for service/warranty.
- 1175 • Employee termination (enterprise devices).
- 1176 • End-of-life disposal.

1177 These scenarios present opportunities for security policy changes, credential transfer, or secure erasure
1178 failures that could compromise subsequent users or violate data protection obligations.

1179 User interactions requiring authentication are addressed in clause 5.4. Physical presence verification
1180 provides protection for security-critical configuration changes.

1181 4.7 Risks, threats and security context

1182 The risk analysis underlying the requirements in clause 5 is documented in Annex B. Annex B identifies
1183 assets requiring protection (B.1), risk factors (B.2), assumptions (B.3), threats (B.4), and maps risk factors to
1184 use cases (B.5) and security profiles (B.6).

1185 4.8 Use cases

1186 4.8.1 Purpose

1187 This clause describes use cases that characterise boot manager products by the security-enhancing
1188 capabilities they implement.

1189 A boot manager manufacturer declares the use case that corresponds to the capabilities of their product as
1190 placed on the market. Each use case is associated with a security profile (LOW, MEDIUM, or HIGH) that
1191 determines which requirements from clause 5 apply.

1192 Separately, boot manager products may implement functional capabilities (see 4.8.4) that are not
1193 associated with any specific use case but that, when present, trigger additional requirements in clause 5 to
1194 mitigate the risks those capabilities introduce. The normative requirements for use case declaration and
1195 requirement application are specified in clause 5.1.

1196 4.8.2 Capabilities as risk exposures

1197 Most capabilities described in clause 4.3.3 do not inherently improve the security of the boot manager.
1198 They provide functionality that may be required by the product's intended purpose, but each capability
1199 introduces additional attack surface and associated risks. A boot manager that does not implement a given
1200 capability is not subject to the risks that capability introduces.

1201 Where a boot manager implements a functional capability, the requirements in clause 5 that reference that
1202 capability apply. These requirements mitigate the risks introduced by the capability. Their applicability is
1203 conditional on the capability being present, not on the use case or security profile.

1204 A limited set of capabilities genuinely enhance the security posture of the system. These capabilities are
1205 associated with the use cases defined in clause 4.8.3 because they establish the security foundation of the
1206 product. See Table 4.3.3-1 for the classification of capabilities.

1207 4.8.3 Use case definitions

1208 Three use cases are defined. Each use case is characterised by the security-enhancing capabilities the boot
1209 manager implements and is associated with a security profile. Use cases are cumulative: each higher use
1210 case includes all requirements of the use cases below it.

1211 The rationale for the association between use cases and security profiles is described in Annex B.

1212

Table 4.8.3-1: Use cases and security profiles

Use case	Security profile	Characterised by
UC-IMM	LOW	Immutable code + trust anchors
UC-VER	MEDIUM	Verified boot + update + logging + key provisioning
UC-HW	HIGH	UC-VER + hardware-backed security

1213

1214 **4.8.3.1 UC-IMM: Immutable**

1215 UC-IMM describes a boot manager where the executable code and trust anchors are stored in hardware-
1216 enforced immutable storage and cannot be modified after deployment.

1217 Associated security profile: LOW. All requirements for security profile LOW apply.

1218 NOTE 1: The security posture of UC-IMM relies on immutability as the primary security property. The
1219 absence of update, configuration, and network boot capabilities eliminates entire classes of
1220 attack vectors.

1221 NOTE 2: A UC-IMM boot manager cannot be remediated if a vulnerability is discovered post-
1222 deployment.

1223 NOTE 3: UC-IMM addresses immutability of the boot manager only. Properties of the post-handoff
1224 stage are an environmental assumption (see Annex B.3.4, AS-TARGET-2). Where this
1225 assumption does not hold, UC-IMM's security posture does not extend to the running system.

1226 **4.8.3.2 UC-VER: Verified and updateable**

1227 UC-VER describes a boot manager that implements verified boot, update capability, logging, and key
1228 provisioning as its security foundation.

1229 Associated security profile: MEDIUM. All requirements for security profiles LOW and MEDIUM apply.

1230 NOTE 1: The association of verified boot and update capability is deliberate: verified boot without
1231 update capability prevents remediation of compromised keys or signing infrastructure; update
1232 capability without verified boot provides no integrity assurance for updates.

1233 NOTE 2: Key provisioning is a capability of the boot manager product. Whether the device manufacturer
1234 in which the boot manager is integrated exposes key management to the device owner is a
1235 decision for the device manufacturer. The boot manager provides the mechanism; it does not
1236 determine downstream key management policy.

1237 **4.8.3.3 UC-HW: Hardware-assisted security**

1238 UC-HW describes a boot manager that implements all UC-VER capabilities plus hardware-backed security
1239 mechanisms.

1240 Associated security profile: HIGH. All requirements for security profiles LOW, MEDIUM and HIGH apply.

1241 NOTE: UC-HW does not imply resistance to advanced physical attacks (e.g., side-channel analysis, fault
1242 injection, chip decapsulation).

1243 4.8.4 Functional capabilities and additional requirements

1244 Functional capabilities (see Table 4.3.3-1) are not associated with any specific use case. They may be
 1245 implemented based on the product's intended purpose. Where a functional capability is implemented, all
 1246 requirements in clause 5 that reference that capability become applicable, regardless of the declared use
 1247 case. These requirements mitigate the risks introduced by the capability.

1248 **Table 4.8.4-1: Capability requirements by use case**

Capability	UC-IMM	UC-VER	UC-HW
Verified boot (see note 1)	Not expected	Required	Required
Update capability	Not permitted	Required	Required
Logging	Optional	Required	Required
Key provisioning	Not expected (see note 2)	Required	Required
Hardware security	Optional	Optional	Required
Network boot	Not permitted (see note 3)	Optional	Optional
Configuration management	Limited (see note 4)	Optional	Optional
Measured boot	Not expected	Optional	Optional
Recovery	Not expected	Optional	Optional
Authentication	Optional	Optional	Optional
Debug interface	Optional	Optional	Optional

1249

1250 NOTE 1: "Verified boot" covers both on-boot verification and on-update verification (capsule update) as
 1251 defined in 4.3.1.2. Individual requirements in clause 5 may apply specifically to on-boot or on-
 1252 update verification; such scope is stated in the applicability statement of the requirement.

1253 NOTE 2: "Not expected" means the capability serves no purpose given the use case characterisation but
 1254 is not prohibited. If nevertheless implemented, the corresponding requirements apply.

1255 NOTE 3: "Not permitted" means the capability is incompatible with the use case characterisation. A
 1256 boot manager implementing that capability cannot declare that use case.

1257 NOTE 4: Under UC-IMM, configuration capability is limited to operational parameters that do not affect
 1258 the security properties of the boot manager.

1259 4.8.5 Use case selection

1260 Use case selection is based on the security-enhancing capabilities the boot manager implements as placed
 1261 on the market:

- 1262 a) If the boot manager code, configuration, and trust anchors are immutable: UC-IMM.
- 1263 b) If the boot manager implements verified boot, update capability, logging, and key provisioning: UC-
 1264 VER.
- 1265 c) If the boot manager additionally uses hardware-backed security mechanisms: UC-HW.

1266 Where a boot manager implements some but not all capabilities of a higher use case, the manufacturer
 1267 declares the highest use case for which all required capabilities are met.

1268 NOTE: Use case selection is based on the capabilities of the boot manager product. The deployment
 1269 context of the device in which the boot manager is integrated is the responsibility of the device
 1270 manufacturer's product conformity assessment, not of the boot manager manufacturer.

1271 5 Requirements (normative)

1272 5.1 Introduction: Applicability of the requirements

1273 5.1.1 General

1274 This clause establishes cybersecurity requirements implementing Annex I Part I and Part II of Regulation
1275 (EU) 2024/2847 [i.1].

1276 Requirements apply based on:

1277 a) the security profile associated with the declared use case (clause 4.8); and

1278 b) implemented functional capabilities: requirements that reference a functional capability apply when
1279 that capability is present, regardless of the declared use case.

1280 Both conditions are evaluated independently. A requirement applies when the security profile condition
1281 and the capability condition (if any) are both met.

1282 5.1.2 Use case declaration

1283 The manufacturer shall declare the use case (UC-IMM, UC-VER, or UC-HW) applicable to the boot manager
1284 product. The declared use case shall be consistent with the capability requirements defined in Table 4.8.4-1.

1285 5.1.2.1 [RQ-UC-IMM]

1286 Where the boot manager is declared as UC-IMM, the boot manager shall meet all of the following
1287 conditions:

1288 a) the boot manager executable code and trust anchors reside in hardware-enforced immutable storage;

1289 b) the boot manager does not implement update capability;

1290 c) the boot manager does not implement network boot.

1291 5.1.2.2 [RQ-UC-VER]

1292 Where the boot manager is declared as UC-VER, the boot manager shall implement:

1293 a) verified boot as defined in clause 4.3.1.2;

1294 b) update capability as defined in clause 4.3.3.2;

1295 c) logging capability as defined in clause 4.3.3.7;

1296 d) key provisioning as defined in clause 4.3.3.10.

1297 5.1.2.3 [RQ-UC-HW]

1298 Where the boot manager is declared as UC-HW, the boot manager shall, in addition to meeting [RQ-UC-
1299 VER], implement hardware-backed security mechanisms for key storage and
1300 cryptographic operations that protect the root of trust against remote software attacks.

1301 5.1.3 Not applicable verdict

1302 A requirement verdict of NOT APPLICABLE shall be justified only when all of the following are met:

1303 a) the capability referenced in the requirement's applicability statement is not implemented by the
1304 product;

1305 b) the declared use case per Table 4.8.4-1 does not assume that capability;

1306 c) residual risks arising from the absence of that capability are documented in the product technical
1307 documentation.

1308 5.2 No known exploitable vulnerabilities

1309 5.2.1 Introduction

1310 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(a). Boot managers shall be
1311 placed on the market without known exploitable vulnerabilities. Due to the pre-OS execution environment,
1312 boot manager vulnerability management relies on verified update mechanisms rather than runtime
1313 scanning or auto-update at first use.

1314 5.2.2 [REQ-BM-KEV-001]

1315 **Requirement:** The boot manager shall not be placed on the market with known exploitable vulnerabilities in
1316 its code, configuration, or cryptographic mechanisms.

1317 **Applicability:** Applies to all boot managers (all security profiles).

1318 5.2.3 [REQ-BM-KEV-002]

1319 **Requirement:** Where the boot manager implements update capability, security updates addressing known
1320 exploitable vulnerabilities shall be available for the duration of the support period.

1321 **Applicability:** Applies to boot managers with update capability (MEDIUM or HIGH security profile).

1322 5.2.4 [REQ-BM-KEV-003]

1323 **Requirement:** Where the boot manager does not implement update capability, the product technical
1324 documentation shall state the absence of update capability as a residual risk, the date of the last
1325 vulnerability assessment performed prior to placing the product on the market, and the defence-in-depth
1326 measures implemented to mitigate the inability to remediate post-deployment vulnerabilities.

1327 **Applicability:** Applies to boot managers without update capability (LOW security profile).

1328 5.3 Secure by default configuration

1329 5.3.1 Introduction to secure by default configuration requirements

1330 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(b). Boot managers shall be
1331 put on the market with security features enabled, no default credentials, and protection matching available
1332 hardware and software resources.

1333 NOTE: The technical implementation of security features (i.e. digital signature validation, or
1334 authenticated hash verification, logging capabilities, rollback protection) and the responsibility
1335 for enabling them by default may vary depending on the deployment model, device
1336 capabilities, and security policy. In many cases, the OEM or system integrator is responsible for
1337 activating these features during device configuration, rather than the silicon supplier, that is in
1338 charge to provide those features.

1339 5.3.2 [REQ-BM-SBD-001]

1340 **Requirement:** The boot manager shall enable cryptographic validation by default.

1341 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1342 NOTE: See clause 5.3.1 for details on implementation and activation details responsibilities.

1343 5.3.3 [REQ-BM-SBD-002]

1344 **Requirement:** The boot manager shall not implement unauthorised or undocumented mechanisms that
1345 allow bypassing authentication or security controls.

1346 **Applicability:** Applies to all boot managers (all security profiles).

1347 NOTE: Examples of such mechanisms include: a) default or hardcoded passwords; b) maintenance or
1348 service backdoors; c) undocumented access interfaces or commands.

1349 5.3.4 [REQ-BM-SBD-003]

1350 **Requirement:** The boot manager shall require passwords that are used to protect interfaces which can
1351 compromise core security guarantees of the system to be established during initial deployment.

1352 **Applicability:** Applies to boot managers with configuration capability using password authentication (all
1353 security profiles).

1354 5.3.5 [REQ-BM-SBD-004]

1355 **Requirement:** The boot manager shall prevent automatic fallback to a boot mode that disables or weakens
1356 verification or rollback protection without explicit authorisation.

1357 **Applicability:** Applies to boot managers with verified or measured boot (MEDIUM or HIGH security profile).

1358 5.3.6 [REQ-BM-SBD-005]

1359 **Requirement:** The boot manager shall provide a prominent indication appropriate for the device's interface
1360 capabilities (e.g. user-visible message, audio notification, LED signal, or logged event) when a security-
1361 relevant protection is reduced or disabled.

1362 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1363 5.3.7 [REQ-BM-SBD-006]

1364 **Requirement:** The boot manager shall support restoring the system to secure default configuration settings.

1365 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1366 NOTE 1: Where configuration or keys are immutable (e.g., ROM/fuses), restoration may be limited; such
1367 limitations needs to be documented.

1368 NOTE 2: For boot managers, "reset to the original state" per CRA Annex I, Part I, point (2)(b) refers to
1369 configuration reset. Reverting the boot manager firmware to an earlier version is intentionally
1370 constrained by the anti-rollback requirements in 5.7 (REQ-BM-INT-017, REQ-BM-INT-018) to
1371 prevent reintroduction of known vulnerabilities. Device-level reset to a pristine state is the
1372 responsibility of the product integrator rather than the boot manager.

1373 5.4 Secure updates

1374 5.4.1 Introduction to secure update requirements

1375 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(c). Where update capability
1376 is implemented, the boot manager shall authenticate update sources, protect update-related keys, and
1377 isolate update logic from the normal boot path.

1378 5.4.2 [REQ-BM-SU-001]

1379 **Requirement:** Where a user interface is available, the boot manager shall provide status indicators for
1380 available updates or rely on authorised components to present such indicators.

1381 **Applicability:** Applies to boot managers with update and configuration capabilities (MEDIUM or HIGH
1382 security profile).

1383 NOTE: Authorised components may include the operating system, firmware management services, or
1384 other trusted update-management mechanisms.

1385 5.4.3 [REQ-BM-SU-002]

1386 **Requirement:** When network capability is available, the boot manager shall support mechanisms to check
1387 for the availability of updates or rely on authorised components to do so.

1388 **Applicability:** Applies to boot managers with update capability (MEDIUM or HIGH security profile).

1389 NOTE: Authorised components may include the operating system, firmware management services, or
1390 other trusted update-management mechanisms.

1391 5.4.4 [REQ-BM-SU-003]

1392 **Requirement:** The boot manager shall enable security policy updates through configuration, allow disabling
1393 features to address vulnerabilities, and maintain configuration update capability throughout product
1394 lifetime.

1395 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1396 5.4.5 [REQ-BM-SU-004]

1397 **Requirement:** The boot manager shall accept updates only from authenticated and authorised sources.

1398 **Applicability:** Applies to boot managers with update capability where the boot manager orchestrates the
1399 update (MEDIUM or HIGH security profile).

1400 5.4.6 [REQ-BM-SU-005]

1401 **Requirement:** When hardware security components are available, the boot manager shall store update-
1402 verification keys using those components.

1403 **Applicability:** Applies to boot managers with update capability where the boot manager performs on-
1404 update verification (MEDIUM or HIGH security profile).

1405 5.4.7 [REQ-BM-SU-006]

1406 **Requirement:** The boot manager shall isolate update logic from normal boot path.

1407 **Applicability:** Applies to boot managers with update capability where the boot manager orchestrates the
1408 update (MEDIUM or HIGH security profile).

1409 5.5 Authentication and access control

1410 5.5.1 Introduction to authentication and access control requirements

1411 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(d). Boot managers control
1412 code execution authorisation and configuration protection through cryptographic verification, physical
1413 presence detection, and hardware-based protections.

1414 5.5.2 [REQ-BM-AAC-001]

1415 **Requirement:** The boot manager shall require that any configuration changes to trusted keys, certificates,
1416 or trust anchor databases are authorised only through one of the following mechanisms: (a) direct physical
1417 presence at the device; or (b) submission of a signed artefact whose authenticity and integrity are
1418 cryptographically verified by the boot manager prior to application.

1419 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1420 NOTE: For boot managers deployed in critical systems or environments with elevated security
1421 requirements, manufacturers may consider requiring a combination of independent
1422 authentication mechanisms (e.g. physical presence combined with cryptographic credential
1423 verification) for trust anchor modifications. This defence-in-depth approach mitigates risks
1424 from compromise of a single authentication factor.

1425 5.5.3 [REQ-BM-AAC-002]

1426 **Requirement:** The boot manager shall protect each configuration setting as needed according to the
1427 system's threat model.

1428 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1429 NOTE 1: In embedded system scenarios, replacing or revoking trusted keys or certificates is often
1430 performed via authenticated software updates, without requiring physical presence.

1431 NOTE 2: Examples of configuration settings related to the boot process include: a) boot order; b) boot
1432 parameters; c) selection of boot targets or modes.

1433 5.5.4 [REQ-BM-AAC-003]

1434 **Requirement:** The boot manager shall require explicit user action and authentication before any
1435 modification of security-critical configuration settings.

1436 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1437 NOTE: Examples of such actions include: a) weakening or disabling secure default settings; b)
1438 downgrading security-related configuration parameters; c) modifying security-critical options
1439 such as verification policies or trust anchors.

1440 5.5.5 [REQ-BM-AAC-004]

1441 **Requirement:** When passwords are used for authentication, the boot manager shall limit authentication
1442 attempts with hardware bound throttling (limit tries or increase delays, or via cryptographically designed to
1443 be slow key derivation functions, where passwords are used).

1444 **Applicability:** Applies to boot managers with configuration capability using password authentication (all
1445 security profiles).

1446 5.5.6 [REQ-BM-AAC-005]

1447 **Requirement:** Where security configuration policies are provided as separate signed artefacts, the boot
1448 manager shall verify their authenticity and integrity before importing or enforcing them.

1449 **Applicability:** Applies to boot managers with verified or measured boot and configuration capability
1450 (MEDIUM or HIGH security profile).

1451 5.5.7 [REQ-BM-AAC-006]

1452 **Requirement:** The boot manager shall indicate when running with non-default security-critical configuration
1453 through persistent visual indication or, where logging capability is supported, through logged events.

1454 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1455 NOTE 1: The indication may be visual (when a user interface is available) or may consist of recorded
1456 events in secure logs.

1457 NOTE 2: Examples of indications include: a) a persistent on-screen message or icon; b) a status LED or
1458 display element; c) a logged event recorded in a secure audit log.

1459 5.5.8 [REQ-BM-AAC-007]

1460 **Requirement:** The boot manager shall protect trusted certificate stores from unauthorised modification.

1461 **Applicability:** Applies to boot managers with verified or measured boot (MEDIUM or HIGH security profile).

1462 5.6 Confidentiality

1463 5.6.1 Introduction to confidentiality requirements

1464 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(e). Boot managers shall
1465 protect cryptographic key material, credentials, and security policies in resource-constrained environments
1466 against persistent physical access threats.

1467 5.6.2 [REQ-BM-CON-001]

1468 **Requirement:** The boot manager shall restrict access to cryptographic key material to authorised boot
1469 components as defined by the boot manager's security policy.

1470 **Applicability:** Applies to boot managers using cryptographic keys (all security profiles).

1471 5.6.3 [REQ-BM-CON-002]

1472 **Requirement:** Device-specific keys shall be used for symmetric verified boot operations. The use of global
1473 secret keys for verified boot purposes is explicitly forbidden.

1474 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1475 NOTE: Use of symmetric device-unique keys mitigates the risk of large-scale attacks resulting from
1476 compromise of a single global key.

1477 5.6.4 [REQ-BM-CON-003]

1478 **Requirement:** The boot manager shall prevent key material from being used for cryptographic purposes
1479 other than those for which it was designated.

1480 **Applicability:** Applies to boot managers with verified or measured boot (MEDIUM or HIGH security profile).

1481 5.6.5 [REQ-BM-CON-004]

1482 **Requirement:** The boot manager shall prevent sensitive data from being included in crash dumps or logs.

1483 **Applicability:** Applies to boot managers with logging capability (MEDIUM or HIGH security profile).

1484 NOTE: Examples of sensitive data include: a) cryptographic key material; b) authentication credentials;
1485 c) security-critical configuration parameters; d) personal or otherwise confidential information
1486 managed during the boot process.

1487 5.6.6 [REQ-BM-CON-005]

1488 **Requirement:** The boot manager shall protect stored credentials using approved mechanisms that prevent
1489 recovery of the original credential values.

1490 **Applicability:** Applies to boot managers with configuration and recovery capability (all security profiles).

1491 NOTE 1: Examples of protected credential types include user passwords, recovery keys, and other
1492 authentication secrets.

1493 NOTE 2: Examples of protection mechanisms include: a) key-derivation functions (e.g. salted password
1494 hashing); b) hardware-bound storage or sealed storage; c) encryption using keys not accessible
1495 to unauthorised entities; d) key-wrapping or secret-derivation schemes.

1496 5.6.7 [REQ-BM-CON-006]

1497 **Requirement:** The boot manager shall protect network boot parameters containing authentication
1498 information using approved mechanisms that prevent unauthorised disclosure.

1499 **Applicability:** Applies to boot managers with configuration capability and network boot (MEDIUM or HIGH
1500 security profile).

1501 5.6.8 [REQ-BM-CON-007]

1502 **Requirement:** The boot manager shall cryptographically erase all sensitive data and clear all security-critical
1503 configuration during secure disposal, where technically feasible (see NOTE).

1504 **Applicability:** Applies to boot managers with verified or measured boot and configuration capability
1505 (MEDIUM or HIGH security profile).

1506 NOTE: When cryptographic material or sensitive data are stored in immutable components such as e-
1507 fuses or ROM, complete erasure may not be possible.

1508 5.6.9 [REQ-BM-CON-008]

1509 **Requirement:** When secure disposal is supported, the boot manager shall provide an indication upon
1510 successful completion of the sanitization process.

1511 **Applicability:** Applies to boot managers that implement secure disposal (all security profiles).

1512 5.6.10 [REQ-BM-CON-009]

1513 **Requirement:** The boot manager shall protect the confidentiality and integrity of boot configuration data
1514 transmitted over network.

1515 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1516 5.6.11 [REQ-BM-CON-010]

1517 **Requirement:** The boot manager shall overwrite confidential or secret data after use, clear from memory
1518 credentials and temporary data structures containing such before boot target handoff and not persist
1519 authentication credentials or confidential cryptographic material beyond the operating system handoff. The
1520 measurements and credentials used for attestation shall persist.

1521 **Applicability:** Applies to all boot managers (all security profiles).

1522 NOTE 1: Examples of sensitive data include authentication credentials, cryptographic key material, and
1523 temporary data structures created during the boot process.

1524 NOTE 2: Examples of mechanisms for removing sensitive data include: a) overwriting sensitive data
1525 after use; b) clearing temporary data structures and credentials before handing off control to
1526 the boot target; c) avoiding persistence of authentication credentials or cryptographic material
1527 beyond the operating system handoff.

1528 5.7 Integrity

1529 5.7.1 Introduction to integrity requirements

1530 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(f). Boot managers establish
1531 trust chains through cryptographic verification, measurement, and protection mechanisms that resist
1532 component substitution, rollback, and TOCTOU attacks.

1533 5.7.2 [REQ-BM-INT-001]

1534 **Requirement:** The boot manager shall enforce privilege boundaries so that code executed in one boot stage
1535 cannot obtain privileges assigned to another boot stage or cross established trust boundaries.

1536 **Applicability:** Applies to all boot managers (all security profiles).

1537 NOTE: Examples of privilege boundaries include: a) restrictions on access to memory regions; b)
1538 restrictions on execution of privileged instructions; c) limitations on access to security-sensitive
1539 hardware resources.

1540 5.7.3 [REQ-BM-INT-002]

1541 **Requirement:** The boot manager shall verify integrity and authenticity of each boot stage using approved
1542 cryptographic mechanisms and shall establish a chain of trust to a root trust anchor before transferring

1543 control.

1544 **Applicability:** Applies to boot managers with on-boot verification (MEDIUM or HIGH security profile).

1545 5.7.4 [REQ-BM-INT-003]

1546 **Requirement:** The boot manager shall prevent boot continuation with components that fail integrity or
1547 authenticity verification.

1548 **Applicability:** Applies to boot managers with on-boot verification (MEDIUM or HIGH security profile).

1549 NOTE 1: Examples of verification failures include: a) invalid, expired, or revoked cryptographic
1550 credentials; b) substituted or tampered components; c) missing or untrusted verification data.

1551 NOTE 2: Cryptographic verification mechanisms may include digital signatures, message authentication
1552 codes (MACs), or other approved methods providing equivalent assurance.

1553 5.7.5 [REQ-BM-INT-004]

1554 **Requirement:** The boot manager shall compute and record a cryptographic measurement of each boot
1555 stage into hardware-protected storage before transferring control, establishing a measurement chain rooted
1556 in a hardware-based trust anchor.

1557 **Applicability:** Applies to boot managers with measured boot (all security profiles).

1558 NOTE 1: Unlike verified boot, measured boot does not prevent execution of unverified components.
1559 Instead, it provides an attestable record that enables a remote or local verifier to determine
1560 whether the boot sequence matches expected values.

1561 NOTE 2: Measurement mechanisms may include: a) extending hash values into hardware-protected
1562 registers (e.g. TPM PCRs); b) deriving compound device identifiers (e.g. DICE CDI); c) recording
1563 measurements in other tamper-evident storage.

1564 NOTE 3: The integrity of the measurement chain depends on the trustworthiness of the initial
1565 measurement by the hardware root of trust and the correct extension of measurements at
1566 each subsequent stage

1567 5.7.6 [REQ-BM-INT-005]

1568 **Requirement:** The boot manager shall verify authenticity and integrity of update packages before
1569 installation, and additionally after network transfer where updates are written to protected storage.

1570 **Applicability:** Applies to boot managers with update capability where the boot manager performs on-
1571 update verification (MEDIUM or HIGH security profile).

1572 5.7.7 [REQ-BM-INT-006]

1573 **Requirement:** When hardware security components are unavailable, the boot manager shall implement
1574 software-based cryptographic verification with keys stored in protected storage.

1575 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1576 5.7.8 [REQ-BM-INT-007]

1577 **Requirement:** The boot manager shall enable configuration among a set of approved signing algorithms.

1578 **Applicability:** Applies to boot managers with verified boot and with configuration capability, where the trust
1579 anchor is not immutable (MEDIUM or HIGH security profile).

1580 NOTE: A set of length one is acceptable. The requirement establishes the capability to substitute one
1581 approved algorithm for another, e.g. when an algorithm becomes deprecated.

1582 5.7.9 [REQ-BM-INT-008]

1583 **Requirement:** The boot manager shall support concurrent verification using more than one approved
1584 signing algorithm.

1585 **Applicability:** Applies to boot managers with verified boot and with configuration capability, where the trust
1586 anchor is not immutable (MEDIUM or HIGH security profile).

1587 5.7.10 [REQ-BM-INT-009]

1588 **Requirement:** The boot manager shall require authorisation before allowing verification bypass.

1589 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1590 NOTE: Authorisation may be evidenced by means appropriate to the deployment context, including
1591 physical presence, cryptographic credential, or explicit user action. The required strength of
1592 authorisation is determined by the security profile and the integrator.

1593 5.7.11 [REQ-BM-INT-010]

1594 **Requirement:** The boot manager shall prevent unauthorised modification between verification and
1595 execution.

1596 **Applicability:** Applies to boot managers with verified boot capability (MEDIUM or HIGH security profile).

1597 5.7.12 [REQ-BM-INT-011]

1598 **Requirement :** The boot manager shall load components into protected memory before verification, when
1599 supported by the underlying hardware platform.

1600 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1601 5.7.13 [REQ-BM-INT-012]

1602 **Requirement:** The boot manager shall protect the integrity and authenticity of sensitive configuration
1603 settings.

1604 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1605 NOTE: Mechanisms to satisfy this requirement may include hardware-protected storage, or
1606 authenticated encryption with freshness protection (e.g. nonces or monotonic counters) and
1607 non-deterministic encryption.

1608 5.7.14 [REQ-BM-INT-013]

1609 **Requirement:** The boot manager shall restore secure default configuration settings when configuration
1610 corruption is detected, where technically feasible.

1611 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1612 NOTE: When cryptographic material or configuration data are stored in immutable components such
1613 as e-fuses or ROM, complete erasure or restoration may not be possible.

1614 5.7.15 [REQ-BM-INT-014]

1615 **Requirement:** The boot manager shall authenticate network boot servers using cryptographic certificates
1616 before accepting actions or configuration data from them.

1617 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1618 5.7.16 [REQ-BM-INT-015]

1619 **Requirement:** The boot manager shall reject network boot connections with invalid, expired, or revoked
1620 server certificates.

1621 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1622 5.7.17 [REQ-BM-INT-016]

1623 **Requirement:** The boot manager shall verify that network configuration and boot-related responses
1624 originate from authorised infrastructure.

1625 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1626 NOTE: Examples of network configuration and boot-related responses include: a) DHCP responses; b)
1627 PXE boot offers; c) network-provided boot configuration data.

1628 5.7.18 [REQ-BM-INT-017]

1629 **Requirement:** The boot manager shall enforce rollback protection when verifying boot stages and
1630 configuration data.

1631 **Applicability:** Applies to boot managers with update or configuration capability (MEDIUM or HIGH security
1632 profile).

1633 NOTE: In some business models, anti-rollback protection can be configured by the OEM. Refer to note
1634 in clause 5.3.

1635 5.7.19 [REQ-BM-INT-018]

1636 **Requirement:** The boot manager shall store anti-rollback counters in hardware-backed or tamper-evident
1637 storage and verify signed version metadata before accepting updates.

1638 **Applicability:** Applies to boot managers with HIGH security profile and update capability.

1639 5.7.20 [REQ-BM-INT-019]

1640 **Requirement:** The boot manager shall verify certificate chains against a trusted root before using
1641 certificates for security-relevant operations.

1642 **Applicability:** Applies to boot managers with verified boot and measured boot, where measured boot
1643 includes remote attestation (MEDIUM or HIGH security profile).

1644 NOTE: Certificate-chain verification may include checking: a) the full chain of trust from the end-entity
1645 certificate to the trusted root; b) certificate validity periods; c) revocation status through
1646 mechanisms such as CRLs or OCSP; d) compliance with certificate policies or usage constraint.

1647 5.7.21 [REQ-BM-INT-020]

1648 **Requirement:** The boot manager shall support revocation of mutable compromised keys and certificates.

1649 **Applicability:** Applies to boot managers with verified boot and measured boot, where measured boot
1650 includes remote attestation (MEDIUM or HIGH security profile).

1651 NOTE 1: In boot managers without connectivity capability, revocation can be managed manually (e.g.
1652 via dedicated tools or physical programming) or through authenticated software updates.

1653 NOTE 2: In many embedded systems, the number of revocation slots (especially for verified boot) is
1654 limited by available storage or hardware design, so the number of possible key revocations may
1655 be constrained.

1656 NOTE 3: Revocation mechanisms may include: a) a revocation database; b) configuration or policy
1657 updates; c) certificate revocation lists (CRLs); d) online or offline status information (e.g. OCSP
1658 responses); e) manufacturer- or owner-provided revocation metadata.

1659 5.7.22 [REQ-BM-INT-021]

1660 **Requirement:** The boot manager shall use cryptographic algorithms, key sizes, and parameters in
1661 accordance with Annex K.

1662 **Applicability:** Applies to all boot managers (all security profiles).

1663 5.7.23 [REQ-BM-INT-022]

1664 **Requirement:** The boot manager shall support migration between approved cryptographic algorithms.

1665 **Applicability:** Applies to boot managers with update capability, where the trust anchor is not immutable
1666 (MEDIUM or HIGH security profile).

1667 NOTE: Migration may be achieved through configuration metadata, controlled algorithm selection
1668 mechanisms, or replacement of the boot manager itself, in alignment with the update
1669 capability declared in 4.3.3.2.

1670 5.7.24 [REQ-BM-INT-023]

1671 **Requirement:** The boot manager shall use collision- and preimage-resistant one-way functions for origin
1672 authentication and verification of boot code provenance.

1673 **Applicability:** Applies to boot managers with verified or measured boot capability (MEDIUM or HIGH
1674 security profile).

1675 NOTE 1: Examples of collision- and preimage-resistant one-way functions include SHA-2 and SHA-3

1676 NOTE 2: Non-collision-resistant hashes (e.g. GHASH in AES-GCM) may be used for integrity protection
1677 only when combined with device-specific keys

1678 5.7.25 [REQ-BM-INT-024]

1679 **Requirement:** The boot manager shall support replacement of trust anchors used for boot verification.

1680 **Applicability:** Applies to boot managers with update capability, where the trust anchor is not immutable
1681 (MEDIUM or HIGH security profile).

1682 NOTE: Trust anchor replacement supports recovery from key compromise and aligns with the key
1683 provisioning capability declared in 4.3.3.10.

1684 5.8 Data minimisation

1685 5.8.1 Introduction to data minimisation requirements

1686 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(g). Boot managers shall
1687 process only data necessary for boot operations, security verification, and error indication, given memory
1688 and storage constraints.

1689 5.8.2 [REQ-BM-DM-001]

1690 **Requirement:** The boot manager shall minimize disclosure of information to network infrastructure by
1691 limiting transmitted data to what is necessary for boot operations.

1692 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1693 NOTE: Examples of measures to minimize network disclosure include: a) requesting only the files
1694 required for the network boot process; b) negotiating only protocol parameters essential for
1695 establishing the boot session; c) restricting identifiers exposed on the network (e.g. to MAC
1696 address and device class).

1697 5.8.3 [REQ-BM-DM-002]

1698 **Requirement:** The boot manager shall prevent disclosure of sensitive device information and shall ensure
1699 that temporary network credentials are not retained beyond their required use.

1700 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1701 NOTE: Examples of sensitive device information that should not be disclosed include:

- 1702 a) firmware version information;
- 1703 b) hardware serial numbers;
- 1704 c) internal configuration or diagnostic data.

1705 5.9 Availability protection

1706 5.9.1 Introduction to availability protection requirements

1707 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(h). Boot managers shall
1708 maintain availability during failures or attacks through recovery mechanisms, failsafe operations, and
1709 resistance to denial-of-service conditions.

1710 Where this clause specifies limits without numeric values, the manufacturer shall document specific
1711 thresholds in security documentation per Annex C.

1712 5.9.2 [REQ-BM-AP-001]

1713 **Requirement:** The boot manager shall support fallback to previous known-good configuration.

1714 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1715 5.9.3 [REQ-BM-AP-002]

1716 **Requirement:** The boot manager shall be able to automatically recover from any interruptions during the
1717 update process.

1718 **Applicability:** Applies to boot managers with HIGH security profile.

1719 5.9.4 [REQ-BM-AP-003]

1720 **Requirement:** The boot manager shall enforce timeouts to prevent indefinite blocking during boot-related
1721 operations.

1722 **Applicability:** Applies to all boot managers (all security profiles).

1723 NOTE: Examples of operations that may require timeouts include: a) parsing or validation of
1724 configuration data; b) network discovery, negotiation, or file retrieval; c) loading or verifying
1725 boot components.

1726 5.9.5 [REQ-BM-AP-004]

1727 **Requirement:** The boot manager shall limit retry attempts and resource consumption for signature
1728 verification and network operations.

1729 **Applicability:** Applies to boot managers with verified boot or network boot (MEDIUM or HIGH security
1730 profile).

1731 5.9.6 [REQ-BM-AP-005]

1732 **Requirement:** The boot manager shall prevent unauthorised bypass of security verification steps during
1733 normal operation.

1734 **Applicability:** Applies to all boot managers (all security profiles).

1735 NOTE: In specific, controlled scenarios such as in-field debug mode, disabling security verification
1736 steps may be permitted for issue analysis, provided that such actions are strictly authenticated.

1737 5.9.7 [REQ-BM-AP-006]

1738 **Requirement:** The boot manager shall detect errors in critical data (e.g. cryptographic keys, configuration
1739 data) and mitigate the impact of detected errors. Where the underlying platform supports error correction
1740 for critical data, the boot manager shall additionally use those mechanisms to correct errors.

1741 **Applicability:** Applies to boot managers with MEDIUM or HIGH security profile.

1742 NOTE 1: Mitigation may include refusing to boot, falling back to a known-good configuration, or
1743 initiating a recovery path. The objective is to prevent execution in a corrupted state.

1744 NOTE 2: Where critical data is stored in immutable elements such as e-fuses, error correction can be
1745 implemented using Forward Error Correction (FEC) codes or similar techniques.

1746 5.9.8 [REQ-BM-AP-007]

1747 **Requirement:** The boot manager shall implement a mechanism to maintain availability of essential boot
1748 code in the presence of partial storage corruption or failure of stages following the initial boot code.

1749 **Applicability:** Applies to boot managers with MEDIUM or HIGH security profile, where the boot manager is
1750 not stored in immutable memory.

1751 NOTE 1: Mechanisms may include redundant storage with selection of uncorrupted copies, fallback
1752 boot partitions, or integrity-driven repair where sufficient information is available. The
1753 implementation shall be documented in the product description.

1754 NOTE 2: Essential boot code includes components required to initialize the platform and load
1755 subsequent boot stages. The initial boot code (e.g. reset vector and immediate successor
1756 instructions) is excluded from this requirement, as recovery from corruption at that stage
1757 requires hardware mechanisms outside the boot manager's control.

1758 5.9.9 [REQ-BM-AP-008]

1759 **Requirement:** The boot manager shall require authentication or physical presence before performing
1760 sensitive actions in recovery mode.

1761 **Applicability:** Applies to boot managers with recovery capability (all security profiles).

1762 NOTE: Recovery mode may be triggered automatically upon boot failures; however, any critical or
1763 security-relevant operations performed during recovery mode require authentication or
1764 physical presence.

1765 5.9.10 [REQ-BM-AP-009]

1766 **Requirement:** When network boot is supported, the boot manager shall handle network boot failures in a
1767 manner that maintains boot availability.

1768 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1769 NOTE: Mechanisms for handling network boot failures may include: a) enforcing timeouts; b) falling
1770 back to a local boot mechanism; c) performing exponential backoff retries; d) attempting
1771 multiple authorised network boot servers.

1772 5.9.11 [REQ-BM-AP-010]

1773 **Requirement:** When network boot is supported, the boot manager shall maintain local boot capability
1774 when network services are unavailable or under denial-of-service conditions.

1775 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1776 5.9.12 [REQ-BM-AP-011]

1777 **Requirement:** The boot manager shall generate logs for security-relevant failure conditions.

1778 **Applicability:** Applies to boot managers with verified boot or measured boot and logging capability
1779 (MEDIUM or HIGH security profile).

1780 NOTE: Security-relevant failure conditions may include verification failures, integrity violations,
1781 authentication failures, or unexpected modification of boot components

1782 5.9.13 [REQ-BM-AP-012]

1783 **Requirement:** The boot manager shall protect security-relevant logs against tampering.

1784 **Applicability:** Applies to boot managers with verified boot or measured boot and logging capability
1785 (MEDIUM or HIGH security profile).

1786 5.9.14 [REQ-BM-AP-013]

1787 **Requirement:** The boot manager shall use time representations that remain valid beyond 2038.

1788 **Applicability:** Applies to all boot managers (all security profiles).

1789 NOTE 1: This requirement applies to all security-relevant uses of time, including:

1790 a) certificate validity periods;

1791 b) internal system time used for boot decisions;

1792 c) timestamps recorded in logs or audit records.

1793 NOTE 2: This requirement helps prevent failures associated with 32-bit time representations (commonly
1794 known as the Year-2038 problem).

1795 5.9.15 [REQ-BM-AP-014]

1796 **Requirement:** The boot manager shall protect recovery mechanisms from unauthorised modification or
1797 disablement.

1798 **Applicability:** Applies to boot managers with recovery capability (all security profiles).

1799 5.9.16 [REQ-BM-AP-015]

1800 **Requirement:** The boot manager shall prevent partial execution of boot components when verification has
1801 not completed successfully.

1802 **Applicability:** Applies to boot managers with verified boot (MEDIUM or HIGH security profile).

1803 5.9.17 [REQ-BM-AP-016]

1804 **Requirement:** The boot manager shall not perform security-sensitive operations when required
1805 cryptographic components or services are unavailable.

1806 **Applicability:** Applies to all boot managers (all security profiles).

1807 5.9.18 [REQ-BM-AP-017]

1808 **Requirement:** The boot manager shall not bypass security controls when handling security violations or
1809 verification failures.

1810 **Applicability:** Applies to boot managers with recovery capability (all security profiles).

1811 5.9.19 [REQ-BM-AP-018]

1812 **Requirement:** The boot manager shall preserve recovery capability across firmware or configuration
1813 updates.

1814 **Applicability:** Applies to boot managers with update and recovery capability (MEDIUM or HIGH security
1815 profile).

1816 5.10 Impact minimisation

1817 5.10.1 Introduction to impact minimisation requirements

1818 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(i). Boot managers shall limit
1819 resource consumption, release hardware properly, and prevent cascading failures to connected devices or
1820 networks.

1821 5.10.2 [REQ-BM-IM-001]

1822 **Requirement:** The boot manager shall avoid generating harmful or disruptive network behaviour.

1823 **Applicability:** Applies to boot managers with network boot (MEDIUM or HIGH security profile).

1824 NOTE: Examples of harmful or disruptive network behaviour include: a) broadcast storms; b) network
1825 loops.

1826 5.11 Minimisation of attack surfaces

1827 5.11.1 Introduction to attack surfaces minimisation requirements

1828 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(j). Boot managers shall
1829 minimise the attack surface by limiting functionality, eliminating debug interfaces, and restricting all
1830 non-essential capabilities in production builds.

1831 5.11.2 [REQ-BM-MAS-001]

1832 **Requirement:** The boot manager shall, before handing off control to the boot target, ensure that only the
1833 resources necessary for the boot target remain enabled.

1834 **Applicability:** Applies to all boot managers (all security profiles).

1835 NOTE: Examples of resources that may need to be disabled or released include, but are not limited to:

1836 a) debug or diagnostic interfaces;

1837 b) DMA-capable devices or peripherals;

1838 c) hardware resources not required by the boot target;

1839 d) allocated volatile memory;

1840 e) network or communication buffers; and

1841 f) memory regions that are not required for the operation of the boot target.

1842 5.11.3 [REQ-BM-MAS-002]

1843 **Requirement:** The boot manager shall exclude non-essential code from production builds.

1844 **Applicability:** Applies to all boot managers (all security profiles).

1845 NOTE: Examples for exclude non-essential code may include test code, debug instrumentation,
1846 coverage tools, experimental features, and unsupported platform code.

1847 5.11.4 [REQ-BM-MAS-003]

1848 **Requirement:** The boot manager shall disable or protect interfaces and functions that are not intended for
1849 operational use.

1850 **Applicability:** Applies to all boot managers (all security profiles).

1851 NOTE: Examples of interfaces and functions that should be disabled in production configuration
1852 include: a) debug interfaces; b) diagnostic consoles; c) test or manufacturing modes; d) verbose
1853 or developer-oriented logging.

1854 5.11.5 [REQ-BM-MAS-004]

1855 **Requirement:** The boot manager shall remove unused or non-essential interfaces and protocols to reduce
1856 the attack surface.

1857 **Applicability:** Applies to all boot managers (all security profiles).

1858 NOTE: Examples of interfaces and protocols that should be disabled include: a) unused hardware
1859 interfaces; b) legacy or deprecated protocols; c) non-essential network services or discovery
1860 mechanisms.

1861 5.11.6 [REQ-BM-MAS-005]

1862 **Requirement:** The boot manager shall validate all inputs to ensure they conform to expected formats.

1863 **Applicability:** Applies to all boot managers (all security profiles).

1864 NOTE 1: Input validation mechanisms may include:

- 1865 a) bounds checking;
- 1866 b) enforcement of size limits;
- 1867 c) rejection of malformed, truncated, or oversized data.

1868 NOTE 2: Inputs in this context may include:

- 1869 a) boot images;
- 1870 b) certificates;
- 1871 c) configuration;
- 1872 d) cryptographic data before processing.

1873 5.11.7 [REQ-BM-MAS-006]

1874 **Requirement:** The boot manager shall detect errors in critical operations and reject non-compliant inputs;
1875 transition to error handling on faults.

1876 **Applicability:** Applies to all boot managers (all security profiles).

1877 5.11.8 [REQ-BM-MAS-007]

1878 **Requirement:** The boot manager shall provide a mechanism to disable configuration options that are not
1879 required by the deployment integrator.

1880 **Applicability:** Applies to boot managers with configuration capability (all profiles).

1881 5.12 Exploitation mitigation mechanisms

1882 This clause addresses the essential requirement in CRA Annex I, Part I, point (2)(k) on mitigating the impact
1883 of an incident. The mechanisms available to boot managers are constrained by the pre-OS execution
1884 environment. Requirements relevant to exploitation mitigation are specified in other clauses of the present
1885 document:

- 1886 • privilege separation and trust-boundary enforcement: REQ-BM-INT-001 (see 5.7);
- 1887 • resource minimisation and DMA restriction at handoff: REQ-BM-MAS-001 (see 5.11);
- 1888 • access control on cryptographic key material: REQ-BM-CON-001 (see 5.6);
- 1889 • attack-surface reduction: REQ-BM-MAS-002, REQ-BM-MAS-003, REQ-BM-MAS-004, REQ-BM-MAS-
1890 007 (see 5.11);
- 1891 • input validation and error handling: REQ-BM-MAS-005, REQ-BM-MAS-006 (see 5.11).

1892 NOTE: Where the target toolchain and architecture support them, compiler-level and platform-level
1893 exploit mitigation features (e.g. position-independent code, stack protection, control-flow
1894 integrity, execute-only memory) reduce the exploitability of residual defects in the boot
1895 manager.

1896 5.13 Logging and monitoring

1897 5.13.1 Introduction to logging and monitoring requirements

1898 This clause addresses the requirements in the CRA [i.1] Annex I, Part I, point (2)(l). Boot managers provide
1899 visibility into boot processes through logs and attestation within pre-OS constraints of limited storage and
1900 no file systems.

1901 5.13.2 [REQ-BM-LOG-001]

1902 **Requirement:** The boot manager shall record measurements of boot components and security-critical
1903 configuration in a tamper-evident way (e.g. tamper-proof storage, or extended into a platform configuration
1904 register) before handoff.

1905 **Applicability:** Applies to boot managers with measured boot and logging capability (MEDIUM or HIGH
1906 security profile).

1907 NOTE: Tamper-evident retention does not require persistent storage; a platform configuration register
1908 or equivalent volatile mechanism satisfies this requirement. Write-endurance constraints apply
1909 where persistent storage is used.

1910 5.13.3 [REQ-BM-LOG-002]

1911 **Requirement:** The boot manager shall provide measurement records in a format that supports remote
1912 attestation and includes mechanisms to ensure freshness against replay.

1913 **Applicability:** Applies to boot managers with measured boot and logging capability (MEDIUM or HIGH
1914 security profile).

1915 NOTE 1: Freshness mechanisms may include nonces, monotonic counters, timestamps, or challenge-
1916 response protocols.

1917 NOTE 2: Measurement records may include information such as:

- 1918 a) integrity digests of boot components;
- 1919 b) indices or identifiers of measured registers or structures;
- 1920 c) event types or measurement categories;
- 1921 d) version or revision information of components.

1922 5.13.4 [REQ-BM-LOG-003]

1923 **Requirement:** The boot manager shall indicate security-relevant failures and state changes.

1924 **Applicability:** Applies to boot managers with logging capability (MEDIUM or HIGH security profile).

1925 NOTE: Examples security-relevant failures and state changes includes:

- 1926 a) verification failures;
- 1927 b) authentication failures;
- 1928 c) security policy violations;
- 1929 d) recovery mode activation;
- 1930 e) execution of unsigned code.

1931 5.13.5 [REQ-BM-LOG-004]

1932 **Requirement:** The boot manager shall provide version information accessible to operating system,
1933 management systems or authorised entity.

1934 **Applicability:** Applies to all boot managers (all security profiles).

1935 5.14 Data removal and transparency

1936 This clause addresses the essential requirement in CRA Annex I, Part I, point (2)(m) on data removal and
1937 transparency. Requirements relevant to data removal are specified in other clauses of the present
1938 document:

- 1939 • overwrite of confidential or secret data after use: REQ-BM-CON-010 (see 5.6);
- 1940 • secure erasure of sensitive data at decommissioning: REQ-BM-CON-007 (see 5.6).

1941 Transparency toward the user on data processed by the boot manager is not a boot manager product
1942 characteristic; boot managers do not process user personal data in the course of normal boot operations.

1943 NOTE: Where end-of-life data removal procedures require user action, product documentation
1944 describes the data categories processed by the boot manager and the steps to remove them.

1945 5.15 Vulnerability handling

1946 This clause addresses the requirements in the CRA [i.1] Annex I Part II.

1947 The requirements specified in CEN/CLC JT013090:2026 (CEN/CLC prEN 40000-1-3) [1] shall be fulfilled for
1948 the product.

1949 6 Conformity assessment and testing (informative)

1950 6.1 Introduction to the assessment and compliance criteria

1951 This clause provides objective and reproducible assessment criteria to determine whether a product
1952 complies with the technical security requirements of clause 5, based on the use case and security profile it
1953 declares for placement on the Union market.

1954 For each requirement defined in clause 5, the following clauses specify assessment criteria to determine
1955 whether the requirement is met. The mapping from clause 5 requirements to clause 6 assessment criteria is
1956 one-to-one via the shared identifier suffix.

1957 The assessment criteria for each requirement are structured as follows:

- 1958 • **Assessment objective:** defines the security property or capability that shall be verified, ensuring
1959 that the assessment remains focused on the intent of the requirement. It includes the reference of
1960 the requirement(s) it aims to assess.
- 1961 • **Assessment preparation:** describes the environment, setup and preconditions required before
1962 executing the assessment, including as applicable:
 - 1963 ○ Test environment: hardware, software and network setup, including versions, topology and
1964 relevant dependencies.
 - 1965 ○ Preconditions: configurations, credentials or operational states that shall be established
1966 before the assessment (e.g. product initialised, certificates loaded, user roles created).
 - 1967 ○ Required tools: tools or software necessary to perform the assessment (e.g. vulnerability
1968 scanners, protocol fuzzers, traffic analysers, static code analysers, cryptographic test suites).
 - 1969 ○ Required information and documentation for the assessment.
 - 1970 ○ Reference to any vendor-provided setup guides, configuration instructions or operational
1971 manuals, and to any relevant standards or technical notes, that define how the product
1972 shall be configured or operated for the assessment.
- 1973 • **Assessment activities:** execution steps to be performed. Assessment activities may include, as
1974 applicable:
 - 1975 ○ Review of the information and documentation to confirm that the described
1976 implementation matches the requirement.
 - 1977 ○ Security functional tests to verify the completeness and correctness of the information and
1978 documentation.
 - 1979 ○ Security functional or penetration tests to verify that implemented controls are correctly
1980 implemented (e.g. attempting to log in with invalid credentials to test lockout enforcement,
1981 or attempting to modify protected configuration without authorisation).
 - 1982 ○ Code or binary analysis to identify potential security weaknesses or misconfigurations (e.g.
1983 static analysis for hardcoded credentials, dynamic analysis for buffer overflow or injection
1984 vulnerabilities).
 - 1985 ○ Configuration inspection to ensure that required security parameters are correctly applied.

- 1986 ○ Observation of runtime behaviour to confirm that protections such as cryptographic
1987 verification, authentication and integrity verification operate as intended.
- 1988 • **Assessment verdict:** Defines the pass/fail criteria.
- 1989 ○ **Pass:** the product demonstrably fulfils the requirement and meets the defined thresholds:
- 1990 ○ **Fail:** the requirement is not fulfilled, or the defined thresholds are not achieved.
- 1991 • **Assessment evidence** : defines the artefacts and documentation collected to demonstrate that the
1992 requirement has been assessed and fulfilled. The evidence shall be sufficient to enable independent
1993 verification of the assessment results and to demonstrate compliance with the relevant CRA
1994 essential requirements. Supporting evidence includes, where applicable:
- 1995 ○ Test or assessment reports showing the steps performed and results obtained;
- 1996 ○ Logs, configuration files, or audit traces demonstrating the implementation of the
1997 requirement;
- 1998 ○ Screenshots, captures, or console outputs confirming the correct execution or protection
1999 behaviour;
- 2000 ○ Relevant vendor or design documentation describing the applied security measures;
- 2001 The indexing structure used in the present document is: **ACC-BM-ESR-NNN** where ACC identifies an
2002 assessment and compliance criterion, BM is the product short name (boot manager), ESR is the
2003 abbreviation of the essential requirement per clause 5, and NNN is the sequence number, matching the
2004 corresponding requirement identifier in clause 5.

2005 6.1.1 Boot manager assessment considerations

2006 Boot manager assessment may require:

- 2007 • Platform-specific test environments due to hardware dependency.
- 2008 • Specialized access to pre-OS execution environment.
- 2009 • System integration context for component-level products.
- 2010 • Physical access to hardware interfaces.

2011 The assessment relies on manufacturer documentation of the test environment and any platform-specific
2012 limitations affecting assessment.

2013 6.1.2 Scope of assessment

2014 The present document defines requirements applicable to boot managers as products. Conformity
2015 assessment against the present document addresses the boot manager as the unit under assessment,
2016 independently of any finished product into which the boot manager may be integrated.

2017 Where a requirement in Clause 5 is conditional upon a capability of the platform or the boot target (see
2018 Clause 4.5), the assessment uses the platform assumptions recorded in the product technical
2019 documentation. Verification that those assumptions hold in a deployed system is outside the scope of the
2020 present document.

2021 6.1.3 Assessment report requirements

2022 The assessment report includes:

- 2023 • the product identifier and version;
- 2024 • the declared use case and security profile per Table 4.8.4-1;
- 2025 • for each requirement in Clause 5:
 - 2026 ▪ the requirement identifier;
 - 2027 ▪ a verdict of PASS, FAIL, or NOT APPLICABLE;
 - 2028 ▪ a reference to the evidence supporting the verdict;
 - 2029 ▪ for a NOT APPLICABLE verdict, a justification meeting the criteria defined in clause 5.1.3;
- 2030 • an overall conformity statement.

2031 A verdict of NOT APPLICABLE for a requirement shall be justified only when all of the following conditions
2032 are met:

- 2033 • the capability referenced in the requirement's Applicability statement is not implemented by the
2034 product;
- 2035 • the product's declared use case per Table 4.8.4-1 does not assume that capability; and
- 2036 • residual risks arising from the absence of that capability are documented in the product technical
2037 documentation.

2038 6.2 No known exploitable vulnerabilities

2039 The present clause specifies assessment criteria for the requirements in Clause 5.2.

2040 **Table 6.2.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-KEV-001	No known exploitable vulns at market entry	Pre-market vulnerability assessment; SBOM and EUVD cross-reference	All
REQ-BM-KEV-002	Security updates during support period	Support period + update channel operational + release history	MEDIUM, HIGH
REQ-BM-KEV-003	Residual-risk documentation for non-updateable	Product technical documentation inspection	LOW only

2041

2042 6.2.1 [ACC-BM-KEV-001]

2043 **Assessment objective:** verify that the boot manager is placed on the market without known exploitable
2044 vulnerabilities in its code, configuration, or cryptographic mechanisms. References REQ-BM-KEV-001.

2045 **Assessment preparation:**

- 2046 • Test environment: a representative build of the boot manager as placed on the market.

2047 • Required information: the manufacturer's pre-market vulnerability assessment record, the SBOM,
2048 the list of cryptographic mechanisms used, and the declared component and dependency versions.

2049 • Required tools: vulnerability scanners; SBOM analysis tools; cross-reference against recognised
2050 vulnerability databases including EUVD.

2051 **Assessment activities:**

2052 • Review the pre-market vulnerability assessment record and verify that its scope covers code,
2053 configuration and cryptographic mechanisms.

2054 • Cross-reference the SBOM and component inventory against EUVD and other recognised
2055 vulnerability databases as of a defined cut-off date.

2056 • Inspect the shipped configuration for known insecure settings (e.g. deprecated cryptographic
2057 algorithms, disabled verification).

2058 • Verify that any known vulnerabilities identified pre-market are either resolved or documented as
2059 accepted residual risks with justification.

2060 **Assessment verdict:**

2061 • Pass: a pre-market vulnerability assessment is documented and no unresolved known exploitable
2062 vulnerability is present at market placement.

2063 • Fail: an unresolved known exploitable vulnerability is present, or the pre-market assessment is
2064 missing or demonstrably incomplete.

2065 **Assessment evidence:** pre-market vulnerability assessment report; SBOM; record of the EUVD cross-
2066 reference with cut-off date; record of the configuration inspection; records of accepted residual risks, if any.

2067 **6.2.2 [ACC-BM-KEV-002]**

2068 **Assessment objective:** verify that security updates addressing known exploitable vulnerabilities are
2069 available for the duration of the declared support period. References REQ-BM-KEV-002.

2070 **Assessment preparation:**

2071 • Test environment: the boot manager in its update-receiving configuration.

2072 • Required information: the manufacturer's declared support period, the update distribution
2073 channel, the update release history, and the mapping between addressed vulnerabilities and
2074 released updates.

2075 • Required tools: access to the update channel; signature verification tools.

2076 **Assessment activities:**

2077 • Review the declared support period and confirm it is stated in the product technical documentation.

2078 • Verify that the manufacturer's vulnerability-handling process produces updates addressing known
2079 exploitable vulnerabilities within the support period, consistent with CEN/CLC prEN 40000-1-3 [1].

2080 • Confirm that the update distribution channel is operational and that delivered updates are signed
2081 and verifiable (see the assessment of REQ-BM-INT-005 in 6.7).

2082 • Sample the update release history to confirm that past known vulnerabilities have been addressed.

2083 **Assessment verdict:**

2084 • Pass: a support period is declared; the update channel is operational; evidence exists of
2085 vulnerability-driven updates during the declared period.

2086 • Fail: no support period is declared; the update channel is unavailable or non-functional; known
2087 vulnerabilities during the declared period are not addressed.

2088 **Assessment evidence:** support period statement; update release history; sample signed update artefacts;
2089 records correlating vulnerabilities with the updates that addressed them.

2090 **6.2.3 [ACC-BM-KEV-003]**

2091 **Assessment objective:** verify that the product technical documentation of a boot manager without update
2092 capability records the absence of update capability as a residual risk, the date of the last pre-market
2093 vulnerability assessment, and the defence-in-depth measures mitigating the inability to remediate post-
2094 deployment vulnerabilities. References REQ-BM-KEV-003.

2095 **Assessment preparation:**

2096 • Required information: the product technical documentation.

2097 **Assessment activities:**

2098 • Inspect the product technical documentation for each of the following:

2099 ○ an explicit statement of the absence of update capability as a residual risk; the date of the
2100 last pre-market vulnerability assessment;

2101 ○ a description of the defence-in-depth measures implemented to mitigate the inability to
2102 remediate post-deployment vulnerabilities.

2103 **Assessment verdict:**

2104 • Pass: all three elements are present and legible in the product technical documentation.

2105 • Fail: any of the three elements is missing or incomplete.

2106 **Assessment evidence:** relevant extracts from the product technical documentation.

2107 **6.3 Secure by default configuration**

2108 The present clause specifies assessment criteria for the requirements in Clause 5.3.

2109

Table 6.3.1-1: Requirements addressed in the present clause

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-SBD-001	Cryptographic validation enabled by default	Default-state boot test with trusted and untrusted images	MEDIUM, HIGH (LOW if implemented)
REQ-BM-SBD-002	No undocumented bypass mechanisms	Interface enumeration, credential testing, binary review	All
REQ-BM-SBD-003	Passwords established at initial deployment	Initial-deployment workflow test per identified interface	All (if implemented)
REQ-BM-SBD-004	No auto-fallback weakening verification	Fault injection per documented fallback mode	MEDIUM, HIGH (LOW if implemented)
REQ-BM-SBD-005	User-visible indication for reduced protection	UI state observation per documented protection	All (if implemented)
REQ-BM-SBD-006	Restoration to secure default configuration	Restoration mechanism against specified secure default	All (if implemented)

2110

2111 **6.3.1 [ACC-BM-SBD-001]**

2112 **Assessment objective:** verify that the boot manager has cryptographic validation enabled by default when
 2113 placed on the market. References REQ-BM-SBD-001.

2114 **Assessment preparation:**

- 2115 • Test environment: a representative build of the boot manager in its shipped state, without any post-
 2116 manufacturing configuration change.
- 2117 • Required information: the manufacturer's documentation of the default state of cryptographic
 2118 validation; the interfaces and mechanisms controlling the validation state.
- 2119 • Required tools: a test boot image signed with a trusted key and a test boot image that is unsigned
 2120 or signed with an untrusted key.

2121 **Assessment activities:**

- 2122 • Boot the shipped image with the trusted-signature test boot image and confirm successful
 2123 validation and boot completion.
- 2124 • Boot the shipped image with the unsigned or untrusted-signature test boot image and confirm that
 2125 validation rejects the image and the boot is halted.
- 2126 • Inspect the boot manager's default configuration and confirm that no step to enable cryptographic
 2127 validation is required before the tests above can succeed.

2128 **Assessment verdict:**

- 2129 • Pass: cryptographic validation is active in the shipped state without any enabling step; untrusted
 2130 images are rejected; trusted images are accepted.
- 2131 • Fail: cryptographic validation is disabled in the shipped state or requires an enabling step; untrusted
 2132 images are accepted.

2133 **Assessment evidence:** record of the default-state configuration; record of boot with a signed image; record
2134 of rejection of an untrusted image.

2135 6.3.2 [ACC-BM-SBD-002]

2136 **Assessment objective:** verify that the boot manager does not implement unauthorised or undocumented
2137 mechanisms that would allow bypassing authentication or security controls. References REQ-BM-SBD-002.

2138 **Assessment preparation:**

- 2139 • Test environment: the boot manager in its shipped state with access to all exposed interfaces.
- 2140 • Required information: the manufacturer's documented list of interfaces, commands, and
2141 authentication mechanisms; the product technical documentation.
- 2142 • Required tools: interface enumeration tools appropriate to the platform; credential testing tools;
2143 binary analysis tools where source is not available.

2144 **Assessment activities:**

- 2145 • Enumerate the interfaces exposed by the boot manager (e.g. shell, management commands, debug
2146 endpoints) and compare against the documented interface list; investigate any undocumented
2147 interface.
- 2148 • Test each authenticated interface against common default and hardcoded credentials and confirm
2149 that no default credential grants access.
- 2150 • Inspect documentation and binary for maintenance or service modes; where present, confirm that
2151 entry is authenticated, documented, and does not bypass authentication or security controls.
- 2152 • Review the binary for strings or code paths suggestive of backdoor or bypass functionality.

2153 **Assessment verdict:**

- 2154 • Pass: no undocumented interface provides access; no default or hardcoded credential grants
2155 access; any maintenance or service mode is documented and does not bypass authentication or
2156 security controls.
- 2157 • Fail: an undocumented interface grants access; a default or hardcoded credential is accepted; a
2158 maintenance or service mechanism bypasses authentication or security controls.

2159 **Assessment evidence:** record of the interface enumeration; record of credential tests; analysis of
2160 maintenance mode; findings from binary inspection.

2161 6.3.3 [ACC-BM-SBD-003]

2162 **Assessment objective:** verify that passwords protecting interfaces which can compromise core security
2163 guarantees are required to be established during initial deployment. References REQ-BM-SBD-003.

2164 **Assessment preparation:**

- 2165 • Test environment: the boot manager in a state representing initial deployment (first boot,
2166 manufacturer defaults).
- 2167 • Required information: the manufacturer's identification of interfaces that can compromise core
2168 security guarantees and are protected by passwords; the initial-deployment workflow.

- 2169 • Required tools: interaction tool for the identified interfaces.

2170 **Assessment activities:**

- 2171 • Identify, from manufacturer documentation, each interface that can compromise core security
2172 guarantees and is protected by password authentication.

- 2173 • Execute the initial-deployment workflow and confirm that access to each such interface either
2174 requires a password to be established during that workflow or is denied until a password is
2175 established.

- 2176 • Confirm that no empty, default, or manufacturer-shared password grants access to the identified
2177 interfaces in the deployed state.

2178 **Assessment verdict:**

- 2179 • Pass: each identified interface requires password establishment during initial deployment; empty,
2180 default, or manufacturer-shared passwords do not grant access post-deployment.

- 2181 • Fail: any identified interface can be accessed without a password being established during initial
2182 deployment, or accepts an empty, default, or manufacturer-shared password.

2183 **Assessment evidence:** list of identified interfaces; record of the initial deployment workflow; records of
2184 access attempts after deployment.

2185 **6.3.4 [ACC-BM-SBD-004]**

2186 **Assessment objective:** verify that the boot manager does not automatically fall back to a boot mode that
2187 disables or weakens verification or rollback protection without explicit authorisation. References REQ-BM-
2188 SBD-004.

2189 **Assessment preparation:**

- 2190 • Test environment: the boot manager configured with verification and rollback protection active,
2191 with the ability to induce failure conditions in normal boot paths.

- 2192 • Required information: the manufacturer's enumeration of fallback boot modes, the conditions
2193 under which they may be entered, and the authorisation step required for each.

- 2194 • Required tools: fault-injection tools capable of triggering documented failure conditions (e.g.
2195 corrupted primary image, failed rollback counter read).

2196 **Assessment activities:**

- 2197 • For each documented fallback mode, induce the failure condition that would normally trigger
2198 consideration of fallback.

- 2199 • Confirm that entry into a fallback mode weakening verification or rollback protection does not
2200 occur automatically and requires the documented explicit authorisation step.

- 2201 • Confirm that in the absence of the authorisation step, the boot manager halts or enters a mode
2202 that preserves verification and rollback protection.

2203 **Assessment verdict:**

- 2204 • Pass: no documented fallback mode weakening verification or rollback protection is entered
2205 without explicit authorisation under any induced failure condition.

- 2206 • Fail: a weakened fallback mode is entered automatically under an induced failure condition.

2207 **Assessment evidence:** enumeration of fallback modes; records of fault-injection tests; records of
2208 verification of authorisation requirements.

2209 6.3.5 [ACC-BM-SBD-005]

2210 **Assessment objective:** verify that the boot manager provides a user-visible indication when a security-
2211 relevant protection is reduced or disabled, where a user interface is available. References REQ-BM-SBD-005.

2212 **Assessment preparation:**

- 2213 • Test environment: the boot manager operated through the user interface available on the target
2214 platform.

- 2215 • Required information: the manufacturer's enumeration of security-relevant protections whose
2216 reduction or disablement triggers an indication; the form and location of the indication.

- 2217 • Required tools: configuration tool to reduce or disable each enumerated protection; display or
2218 output capture for the user interface.

2219 **Assessment activities:**

- 2220 • For each enumerated security-relevant protection, reduce or disable the protection via its
2221 configuration interface.

- 2222 • Observe the user interface during and after the change and confirm that the documented indication
2223 is presented.

- 2224 • Restore the protection and confirm that the indication is removed or updated consistently with the
2225 documented behaviour.

2226 **Assessment verdict:**

- 2227 • Pass: reducing or disabling each enumerated protection produces the documented user-visible
2228 indication; restoring the protection updates or removes the indication consistently.

- 2229 • Fail: reducing or disabling a protection produces no user-visible indication, or the indication is
2230 inconsistent with the documented behaviour.

2231 **Assessment evidence:** enumeration of security-relevant protections; records of UI capture for each state
2232 transition.

2233 6.3.6 [ACC-BM-SBD-006]

2234 **Assessment objective:** verify that the boot manager supports restoring the system to secure default
2235 configuration settings. References REQ-BM-SBD-006.

2236 **Assessment preparation:**

- 2237 • Test environment: the boot manager configured to a non-default state via documented
2238 configuration interfaces.

- 2239 • Required information: the manufacturer's specification of the secure default configuration; the
2240 restoration mechanism; any documented limitations to restoration arising from immutable

2241 configuration or keys; the manufacturer's statement on the scope of restoration for the purposes of
2242 CRA Annex I, Part I, point (2)(b) per the NOTE in 5.3.7.

- 2243 • Required tools: configuration tool capable of applying and reading configuration state; inspection
2244 tool for immutable-configuration elements where relevant.

2245 **Assessment activities:**

- 2246 • Apply a documented non-default configuration and confirm the change is reflected in the boot
2247 manager's configuration state.

- 2248 • Invoke the restoration mechanism per manufacturer documentation.

- 2249 • Compare the resulting configuration against the specified secure default; confirm that all mutable
2250 configuration items are restored and that any excluded items correspond to documented
2251 immutable-configuration limitations.

- 2252 • Confirm that the scope of restoration matches the manufacturer's statement for CRA Annex I, Part I,
2253 point (2)(b) and that limitations are explicit in the product technical documentation.

2254 **Assessment verdict:**

- 2255 • Pass: the restoration mechanism returns all mutable configuration items to the specified secure
2256 default; any excluded items are documented as limited by immutable configuration or keys.

- 2257 • Fail: mutable configuration items are not restored to the specified secure default, or limitations are
2258 undocumented.

2259 **Assessment evidence:** specification of secure defaults; record of non-default configuration; record of
2260 configuration after restoration; reference to documented limitations.

2261 6.4 Secure updates

2262 The present clause specifies assessment criteria for the requirements in Clause 5.4.

2263 **Table 6.4.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-SU-001	Update status indicators (UI or authorised components)	Per-status indicator observation or authorised-component interface check	MEDIUM, HIGH (if UI implemented)
REQ-BM-SU-002	Check for update availability	Availability-present / availability-absent test	MEDIUM, HIGH
REQ-BM-SU-003	Policy updates and feature disablement via config	Policy-update and feature-disablement test	All (if implemented)
REQ-BM-SU-004	Accept updates only from authenticated sources	Tests of update sources: authorised; unauthenticated; authenticated but not authorised	MEDIUM, HIGH
REQ-BM-SU-005	Update-verification keys in hardware security	Key-storage location inspection; hardware-interface usage	MEDIUM, HIGH (if HW present)
REQ-BM-SU-006	Update logic isolated from boot path	Code-path and state-region observation for both modes	MEDIUM, HIGH

2264

2265 6.4.1 [ACC-BM-SU-001]

2266 **Assessment objective:** verify that the boot manager, where a user interface is available, presents update
2267 status indicators or relies on authorised components to present them. References REQ-BM-SU-001.

2268 **Assessment preparation:**

2269 • Test environment: the boot manager operated through its user interface, with the ability to induce
2270 each documented update status (e.g. no update pending, update pending, update in progress,
2271 update failed).

2272 • Required information: the manufacturer's specification of the update statuses indicated through
2273 the user interface and, where reliance on authorised components is declared, the list of those
2274 components and the mechanism used to communicate status to them.

2275 • Required tools: display or output capture for the user interface; inspection tool for any interface
2276 exposed to authorised components.

2277 **Assessment activities:**

2278 • For each documented update status, induce the status condition and observe the user interface;
2279 confirm that the corresponding indication is presented as documented.

2280 • Where reliance on authorised components is declared in place of a user-visible indication, confirm
2281 that a documented interface exposes the update status to the authorised component and that the
2282 status value is consistent with the induced condition.

2283 **Assessment verdict:**

2284 • Pass: each documented update status produces the documented indication through the user
2285 interface, or is exposed to the documented authorised components through the declared interface.

2286 • Fail: an update status does not produce a documented indication and is not exposed through the
2287 declared interface.

2288 **Assessment evidence:** enumeration of update statuses; records of UI capture per status; records of
2289 interface inspection for code paths that rely on authorised components.

2290 6.4.2 [ACC-BM-SU-002]

2291 **Assessment objective:** verify that the boot manager supports mechanisms to check for update availability
2292 when network capability is available, or relies on authorised components to do so. References REQ-BM-SU-
2293 002.

2294 **Assessment preparation:**

2295 • Test environment: the boot manager configured with network capability, in a controlled
2296 environment where the update source can be populated with or deprived of available updates.

2297 • Required information: the manufacturer's specification of the update-availability check mechanism
2298 (including protocol, endpoint, and schedule, where applicable) or the list of authorised components
2299 relied upon.

2300 • Required tools: controllable update source or emulator; network traffic analyser; inspection tool for
2301 any interface to authorised components.

2302 **Assessment activities:**

- 2303 • Populate the update source with an available update and invoke the check mechanism; confirm
2304 that the boot manager, or the authorised component it relies on, detects and reports the
2305 availability.
- 2306 • Deprive the update source of available updates and invoke the check mechanism; confirm the
2307 reported availability is consistent.
- 2308 • Where reliance on authorised components is declared, confirm the interface to those components
2309 is documented and functional.

2310 **Assessment verdict:**

- 2311 • Pass: the update-availability check correctly reports availability and non-availability through either
2312 the boot manager's own mechanism or the authorised-component interface.
- 2313 • Fail: the check mechanism does not correctly report availability, or neither the boot manager nor
2314 the authorised component is able to perform the check.

2315 **Assessment evidence:** specification of the check mechanism; records of tests with availability present and
2316 with availability absent; records of inspection of the authorised-component interface.

2317 **6.4.3 [ACC-BM-SU-003]**

2318 **Assessment objective:** verify that the boot manager enables security policy updates through configuration,
2319 permits disabling features to address vulnerabilities, and maintains configuration update capability
2320 throughout product lifetime. References REQ-BM-SU-003.

2321 **Assessment preparation:**

- 2322 • Test environment: the boot manager accessible through its configuration interface.
- 2323 • Required information: the manufacturer's specification of the security policy items that can be
2324 updated through configuration; the features that can be disabled through configuration; the
2325 commitment on configuration update availability throughout product lifetime.
- 2326 • Required tools: configuration tool; inspection tool for the boot manager's active policy and feature
2327 state.

2328 **Assessment activities:**

- 2329 • Apply a documented security policy update through the configuration interface; confirm the new
2330 policy is reflected in the boot manager's state.
- 2331 • Disable a documented feature through the configuration interface; confirm the feature is no longer
2332 active in the boot manager's state.
- 2333 • Review the manufacturer's statement on configuration update availability throughout product
2334 lifetime and verify consistency with the declared support period of the product.

2335 **Assessment verdict:**

- 2336 • Pass: security policy updates apply; feature-disabling configuration applies; configuration update
2337 availability is committed for the declared product lifetime.

- 2338 • Fail: security policy updates fail to apply; features cannot be disabled through configuration;
2339 configuration update availability is not committed for the declared product lifetime.

2340 **Assessment evidence:** record of policy-update tests; record of feature-disablement tests; manufacturer
2341 statement on configuration update availability throughout product lifetime.

2342 6.4.4 [ACC-BM-SU-004]

2343 **Assessment objective:** verify that the boot manager, where it orchestrates updates, accepts updates only
2344 from authenticated and authorised sources. References REQ-BM-SU-004.

2345 **Assessment preparation:**

- 2346 • Test environment: the boot manager in its update-orchestrating configuration.
- 2347 • Required information: the manufacturer's specification of the source authentication mechanism
2348 (e.g. signed update manifests, authenticated transport, trusted source identifiers) and the
2349 authorisation policy applied to authenticated sources.
- 2350 • Required tools: test update source representing an authorised source; test update source
2351 representing an unauthorised source; test update source representing an authenticated but
2352 unauthorised source.

2353 **Assessment activities:**

- 2354 • Submit an update from the authorised source and confirm it is accepted for the subsequent update
2355 steps.
- 2356 • Submit an update from the unauthorised (unauthenticated) source and confirm rejection before
2357 any update action is taken.
- 2358 • Submit an update from an authenticated but unauthorised source (e.g. valid signature, wrong
2359 signer identity) and confirm rejection.
- 2360 • Confirm that the authentication and authorisation decisions are performed prior to invoking the
2361 package verification defined for REQ-BM-INT-005 in 6.7.

2362 **Assessment verdict:**

- 2363 • Pass: authorised-source updates are accepted; unauthenticated and sources that are authenticated
2364 but not authorised are rejected before any update action.
- 2365 • Fail: an unauthenticated or unauthorised source is accepted, or the source check is performed after
2366 an update action has begun.

2367 **Assessment evidence:** source authentication specification; test records for each of the three source
2368 categories.

2369 6.4.5 [ACC-BM-SU-005]

2370 **Assessment objective:** verify that the boot manager, when hardware security components are available,
2371 stores update-verification keys using those components where the boot manager verifies update packages.
2372 References REQ-BM-SU-005.

2373 **Assessment preparation:**

- 2374 • Test environment: the boot manager on a platform whose hardware security components are
2375 documented and inspectable.
- 2376 • Required information: the manufacturer's declaration of whether hardware security components
2377 are available on the target platform; where available, the identification of those components and
2378 the key-storage interface used.
- 2379 • Required tools: inspection tool for the hardware security component appropriate to the platform
2380 (e.g. TPM command interface, secure-element debug interface within manufacturer-authorised
2381 scope).

2382 **Assessment activities:**

- 2383 • Confirm the availability of hardware security components on the platform under assessment;
2384 where unavailable, record the REQ as not applicable and terminate this assessment per 5.1.3.
- 2385 • Where hardware security components are available, inspect the storage location of the update-
2386 verification keys; confirm the keys are held by the hardware security component and not by
2387 software-accessible storage.
- 2388 • Confirm that update verification by the boot manager uses the keys through the hardware security
2389 component's protected interface, without exporting the key material.

2390 **Assessment verdict:**

- 2391 • Pass: update-verification keys are stored in the available hardware security components;
2392 verification uses the keys through the component's protected interface without exposing the key
2393 material.
- 2394 • Fail: update-verification keys are stored in software-accessible storage while hardware security
2395 components are available; keys are exported from the hardware security component for
2396 verification.

2397 **Assessment evidence:** record of platform hardware security component availability; record of inspection of
2398 the key storage location; record of the key-use interface.

2399 **6.4.6 [ACC-BM-SU-006]**

2400 **Assessment objective:** verify that the boot manager, where it orchestrates updates, isolates update logic
2401 from the normal boot path. References REQ-BM-SU-006.

2402 **Assessment preparation:**

- 2403 • Test environment: the boot manager in both normal boot and update-orchestration modes.
- 2404 • Required information: the manufacturer's description of the separation mechanism between the
2405 update logic and the normal boot path (e.g. separate execution context, memory isolation, distinct
2406 code paths, state machine separation).
- 2407 • Required tools: inspection tool for the boot manager's runtime state and code-path activation as
2408 supported by the platform.

2409 **Assessment activities:**

- 2410 • Execute a normal boot and inspect the code paths and state regions activated; confirm that the
2411 update logic is not activated.

- 2412 • Execute an update orchestration and inspect the code paths and state regions activated; confirm
 2413 that the normal-boot code paths are not entered concurrently with update logic in a way that
 2414 would allow cross-influence.
- 2415 • Where the separation mechanism involves shared resources, confirm that resource handoff
 2416 between the two paths is documented and does not allow update logic to manipulate normal-boot
 2417 state, or vice versa.
- 2418 **Assessment verdict:**
- 2419 • Pass: update logic is not activated during normal boot; update orchestration does not enter normal-
 2420 boot code paths in a way that allows cross-influence; any shared-resource handoff is documented
 2421 and isolated.
- 2422 • Fail: update logic is activated during normal boot; update orchestration and normal-boot code
 2423 paths share state or resources in a way that allows cross-influence.
- 2424 **Assessment evidence:** execution trace of normal boot; execution trace of update orchestration; record of
 2425 inspection of the separation mechanism.

2426 6.5 Authentication and access control

2427 The present clause specifies assessment criteria for the requirements in Clause 5.5.

2428 **Table 6.5.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-AAC-001	Key/cert changes require physical presence or signed artefact	Four-way test: no-auth, presence, valid signed, invalid signed	All (if implemented)
REQ-BM-AAC-002	Config settings protected per threat model	Per-setting test mapping threats to protection mechanisms	All (if implemented)
REQ-BM-AAC-003	Explicit user action + auth for critical config	No-action, no-auth, both-present, and implicit-path tests	All (if implemented)
REQ-BM-AAC-004	Password auth throttling (HW-bound or slow KDF)	Per-attempt cost measurement; power-cycle persistence check	All (if implemented)
REQ-BM-AAC-005	Verify signed configuration policy artefacts	Trusted, altered, untrusted-signer artefact tests	MEDIUM, HIGH (LOW if verified/measured boot implemented)
REQ-BM-AAC-006	Indication for non-default security config	Per-item UI or log capture	All (if implemented)
REQ-BM-AAC-007	Protect trusted certificate stores	Per-interface modification + direct-storage alteration tests	MEDIUM, HIGH (LOW if verified/measured boot implemented)

2429

2430 6.5.1 [ACC-BM-AAC-001]

2431 **Assessment objective:** verify that configuration changes to trusted keys, certificates, or trust anchor
 2432 databases are authorised only through direct physical presence at the device, or through submission of a
 2433 signed artefact whose authenticity and integrity are cryptographically verified by the boot manager prior to
 2434 application. References REQ-BM-AAC-001.

2435 **Assessment preparation:**

- 2436 • Test environment: the boot manager configured with an established trust anchor database, a
2437 physical-presence detection mechanism where supported, and a signed-artefact submission
2438 interface.
- 2439 • Required information: the manufacturer's specification of the physical-presence detection
2440 mechanism (where implemented) and the signed-artefact format, signer identities, and verification
2441 algorithm.
- 2442 • Required tools: hardware means of asserting and de-asserting physical presence (e.g. jumper,
2443 button, hardware key); signed-artefact generation tool using trusted and untrusted signers.

2444 **Assessment activities:**

- 2445 • Attempt a configuration change to a trusted key, certificate, or trust anchor without asserting
2446 physical presence and without submitting a signed artefact; confirm rejection.
- 2447 • Attempt the same change with physical presence asserted; confirm acceptance.
- 2448 • Attempt the same change via submission of an artefact signed by a trusted signer with valid
2449 authenticity and integrity; confirm acceptance.
- 2450 • Attempt the same change via submission of an artefact signed by an untrusted signer, or with
2451 invalid signature or altered content; confirm rejection prior to application.

2452 **Assessment verdict:**

- 2453 • Pass: changes are accepted only with asserted physical presence or with a cryptographically verified
2454 signed artefact from a trusted signer; all other attempts are rejected.
- 2455 • Fail: any change is accepted without one of the two documented mechanisms, or with an unverified
2456 artefact.

2457 **Assessment evidence:** specification of the physical-presence mechanism; specification of artefact
2458 submission; test records for each of the four attempt categories.

2459 **6.5.2 [ACC-BM-AAC-002]**

2460 **Assessment objective:** verify that each configuration setting is protected as needed according to the
2461 system's threat model. References REQ-BM-AAC-002.

2462 **Assessment preparation:**

- 2463 • Test environment: the boot manager accessible through its configuration interface.
- 2464 • Required information: the manufacturer's threat model applicable to the boot manager; the
2465 inventory of configuration settings with, for each, the threats applicable and the protection applied
2466 (e.g. authentication, authorisation, integrity protection, confidentiality protection, rate limiting,
2467 physical-presence requirement).
- 2468 • Required tools: configuration tool; inspection tool for the boot manager's configuration protection
2469 state.

2470 **Assessment activities:**

- 2471 • For each configuration setting, enumerate the threats applicable per the documented threat model.

2472 • For each applicable threat, verify that the documented protection is implemented by testing an
 2473 attempt that the protection is intended to prevent (e.g. unauthorised modification attempt,
 2474 integrity violation attempt); confirm rejection.

2475 • Confirm that the documented protection level for each setting is consistent with the setting's role in
 2476 the security guarantees of the boot manager.

2477 **Assessment verdict:**

2478 • Pass: every configuration setting has protections consistent with the documented threat model, and
 2479 test attempts targeting applicable threats are rejected.

2480 • Fail: a configuration setting lacks a protection required by the documented threat model, or a test
 2481 attempt targeting an applicable threat succeeds.

2482 **Assessment evidence:** inventory of configuration settings with mapping of threats to protection
 2483 mechanisms; records of attempts per setting.

2484 **6.5.3 [ACC-BM-AAC-003]**

2485 **Assessment objective:** verify that modification of security-critical configuration settings requires explicit
 2486 user action and authentication. References REQ-BM-AAC-003.

2487 **Assessment preparation:**

2488 • Test environment: the boot manager in an authenticated configuration session.

2489 • Required information: the manufacturer's enumeration of security-critical configuration settings;
 2490 the documented user action required for modification; the documented authentication mechanism.

2491 • Required tools: configuration interaction tool; authentication bypass testing tool (for negative
 2492 cases).

2493 **Assessment activities:**

2494 • For each security-critical setting, attempt a modification without performing the documented user
 2495 action; confirm rejection.

2496 • Attempt a modification with the user action but without the documented authentication; confirm
 2497 rejection.

2498 • Attempt a modification with both user action and authentication; confirm acceptance.

2499 • Attempt a modification via an automated or implicit path that does not involve explicit user action
 2500 (e.g. by replaying a prior session's credentials, scripted invocation bypassing the UI); confirm
 2501 rejection.

2502 **Assessment verdict:**

2503 • Pass: modifications to security-critical settings succeed only when both the documented user action
 2504 and authentication are performed; automated or implicit modification paths are rejected.

2505 • Fail: any security-critical setting can be modified without the documented user action, without
 2506 authentication, or through an implicit path.

2507 **Assessment evidence:** enumeration of security-critical settings; test records for each of the four attempt
 2508 categories.

2509 6.5.4 [ACC-BM-AAC-004]

2510 **Assessment objective:** verify that, where passwords are used for authentication, authentication attempts
2511 are limited by hardware-bound throttling, try-count limits, increasing delays, or cryptographically slow key
2512 derivation functions. References REQ-BM-AAC-004.

2513 **Assessment preparation:**

- 2514 • Test environment: the boot manager configured with password authentication on a documented
2515 interface.
- 2516 • Required information: the manufacturer's specification of the throttling mechanism (type,
2517 parameters, whether hardware-bound or KDF-based); the minimum effective cost per
2518 authentication attempt.
- 2519 • Required tools: automated authentication attempt tool; timing measurement tool.

2520 **Assessment activities:**

- 2521 • Measure the effective cost per authentication attempt at nominal rates and under sustained
2522 unsuccessful attempts (e.g. time per attempt, attempt count before lockout, delay pattern across
2523 attempts).
- 2524 • Confirm that the measured behaviour is consistent with the documented throttling mechanism.
- 2525 • Where the mechanism is hardware-bound, confirm that the throttling state is maintained across
2526 power cycles or equivalent, such that repeated attempts across power cycles do not reset the
2527 throttling.
- 2528 • Where a slow KDF is the mechanism, confirm that the documented parameters (e.g. iteration
2529 count, memory cost) yield a per-attempt cost consistent with the documented minimum.

2530 **Assessment verdict:**

- 2531 • Pass: measured per-attempt cost and lockout or delay behaviour match the documented throttling
2532 mechanism; hardware-bound throttling state persists across power cycles where claimed.
- 2533 • Fail: throttling is absent, inconsistent with documentation, or bypassable (e.g. reset across power
2534 cycles when claimed hardware-bound).

2535 **Assessment evidence:** throttling specification; cost measurements per attempt; records of persistence
2536 across power cycles where applicable.

2537 6.5.5 [ACC-BM-AAC-005]

2538 **Assessment objective:** verify that, where security configuration policies are provided as separate signed
2539 artefacts, the boot manager verifies their authenticity and integrity before importing or enforcing them.
2540 References REQ-BM-AAC-005.

2541 **Assessment preparation:**

- 2542 • Test environment: the boot manager with the import or enforcement interface for security
2543 configuration policies accessible.

2544 • Required information: the manufacturer's declaration of whether security configuration policies are
 2545 provided as separate signed artefacts; where they are, the artefact format, signer identities, and
 2546 verification algorithm.

2547 • Required tools: artefact generation tool capable of producing artefacts signed by trusted and
 2548 untrusted signers, with valid and altered contents.

2549 **Assessment activities:**

2550 • Where the manufacturer declares that policies are not provided as separate signed artefacts (e.g.
 2551 policies baked into the boot manager), record the REQ as not applicable per 5.1.3 and terminate
 2552 this assessment.

2553 • Where policies are provided as separate signed artefacts, submit an artefact with valid trusted-
 2554 signer signature and unaltered content; confirm the policy is imported and enforced.

2555 • Submit an artefact with valid signature but altered content after signing; confirm rejection before
 2556 import or enforcement.

2557 • Submit an artefact signed by an untrusted signer; confirm rejection before import or enforcement.

2558 **Assessment verdict:**

2559 • Pass: valid trusted-signer artefacts with unaltered content are imported and enforced; altered or
 2560 untrusted-signer artefacts are rejected before import or enforcement.

2561 • Fail: an altered or untrusted-signer artefact is imported or enforced; verification occurs after import
 2562 or enforcement.

2563 **Assessment evidence:** artefact format specification; test records for each of the three artefact categories;
 2564 record of import timing confirming verification precedes enforcement.

2565 **6.5.6 [ACC-BM-AAC-006]**

2566 **Assessment objective:** verify that the boot manager indicates when running with non-default security-
 2567 critical configuration, either through persistent visual indication or, where logging capability is supported,
 2568 through logged events. References REQ-BM-AAC-006.

2569 **Assessment preparation:**

2570 • Test environment: the boot manager with a documented user interface or logging capability
 2571 available.

2572 • Required information: the manufacturer's enumeration of security-critical configuration items
 2573 whose non-default state triggers an indication; the indication mechanism declared (visual, logged,
 2574 or both) and its form.

2575 • Required tools: configuration tool; display or output capture for visual indications; log capture tool
 2576 for logged-event indications.

2577 **Assessment activities:**

2578 • Place each enumerated security-critical configuration item into its non-default state.

2579 • Where a visual indication is declared, observe the user interface across the boot manager's
 2580 execution and confirm the indication is persistent and consistent with documentation.

2581 • Where a logged event is declared, capture the log output and confirm that the documented event
2582 entry is produced.

2583 • Restore the item to its default state and confirm the indication is removed or superseded.

2584 **Assessment verdict:**

2585 • Pass: each non-default security-critical configuration item triggers the declared indication (visual,
2586 logged, or both); restoration to default removes or supersedes the indication.

2587 • Fail: a non-default security-critical configuration item produces no declared indication, or the
2588 indication is transient where persistent behaviour is declared.

2589 **Assessment evidence:** enumeration of security-critical items; records of visual indication capture; records of
2590 logged event capture.

2591 **6.5.7 [ACC-BM-AAC-007]**

2592 **Assessment objective:** verify that trusted certificate stores are protected from unauthorised modification.
2593 References REQ-BM-AAC-007.

2594 **Assessment preparation:**

2595 • Test environment: the boot manager with the trusted certificate store populated and accessible
2596 through its documented interfaces.

2597 • Required information: the manufacturer's description of the trusted certificate store, its storage
2598 location, the protection mechanism (e.g. access control, integrity protection, hardware-backed
2599 storage), and the authorised modification path.

2600 • Required tools: certificate-store inspection tool; modification attempt tools for each interface
2601 exposing the store.

2602 **Assessment activities:**

2603 • Attempt modification of the trusted certificate store through each documented interface without
2604 the authorised modification path (e.g. without physical presence, without a signed artefact,
2605 without authentication); confirm rejection.

2606 • Attempt modification through undocumented interfaces where they exist (e.g. direct memory
2607 access paths, debug interfaces); confirm rejection.

2608 • Modify the store content directly in its storage location where inspection access is available;
2609 confirm that on subsequent use the integrity check detects the alteration.

2610 **Assessment verdict:**

2611 • Pass: modification attempts through any interface without the authorised path are rejected; direct-
2612 storage alteration is detected by integrity check prior to use.

2613 • Fail: modification succeeds through any interface without the authorised path, or direct-storage
2614 alteration is not detected before use.

2615 **Assessment evidence:** specification of the certificate store; records of modification attempts per interface;
2616 record of detection of direct storage alteration.

2617 6.6 Confidentiality

2618 The present clause specifies assessment criteria for the requirements in Clause 5.6.

2619 **Table 6.6.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-CON-001	Access control on cryptographic key material	Authorised / non-authorised component access tests; reachability inspection	All (if implemented)
REQ-BM-CON-002	Device-specific keys for symmetric verified boot	Provisioning-path documentation + cross-device comparison	MEDIUM, HIGH (LOW if implemented)
REQ-BM-CON-003	Keys used only for designated purposes	Per-key observed-operation vs designation; misuse-attempt tests	MEDIUM, HIGH (LOW if implemented)
REQ-BM-CON-004	No sensitive data in crash dumps or logs	Crash and log captures searched for sensitive-data categories	All (if implemented)
REQ-BM-CON-005	Stored credentials non-recoverable	Storage-format inspection; Annex K mechanism check; recovery-attempt tests	All (if implemented)
REQ-BM-CON-006	Network-boot auth parameters protected	Storage-form + transmission-form inspection; plaintext-read attempts	MEDIUM, HIGH (if implemented)
REQ-BM-CON-007	Secure disposal cryptographic erase	Post-disposal inspection per sensitive-data item; feasibility-limit documentation	MEDIUM, HIGH (LOW if implemented)
REQ-BM-CON-008	Sanitization completion indication	Successful and interrupted disposal outcome observation	All (if implemented)
REQ-BM-CON-009	Network boot config protected in transit	Transport capture; MITM integrity test; passive decryption attempt	MEDIUM, HIGH (if implemented)
REQ-BM-CON-010	Clear sensitive data before handoff	Handoff memory inspection plus persistence check after power cycling	All

2620

2621 6.6.1 [ACC-BM-CON-001]

2622 **Assessment objective:** verify that access to cryptographic key material is restricted to authorised boot
2623 components as defined by the boot manager's security policy. References REQ-BM-CON-001.

2624 **Assessment preparation:**

- 2625 • Test environment: the boot manager with cryptographic keys provisioned and the security policy
2626 active.
- 2627 • Required information: the manufacturer's security policy identifying key material and the
2628 authorised boot components for each; the access-control mechanism used (e.g. isolation
2629 boundaries, access tokens, protected storage with enforced access).
- 2630 • Required tools: inspection tool for memory regions and access-control state on the target platform;
2631 test harness capable of invoking non-authorised components that attempt key access.

2632 **Assessment activities:**

- 2633 • For each key item, identify the authorised component set per the security policy.
- 2634 • Attempt access to the key from each authorised component and confirm acceptance; attempt
2635 access from a non-authorised component and confirm rejection.

- 2636 • Inspect memory regions and access-control state to confirm that key material is not reachable from
2637 components outside the documented authorised set, including via shared resources (e.g. caches,
2638 DMA paths).

2639 **Assessment verdict:**

- 2640 • Pass: authorised components access key material; non-authorised components are denied; no
2641 unintended reachability path exists.

- 2642 • Fail: a non-authorised component reaches key material, or an unintended reachability path is
2643 identified.

2644 **Assessment evidence:** mapping of keys to components per security policy; records of access tests per
2645 component; record of memory and access-control inspection.

2646 **6.6.2 [ACC-BM-CON-002]**

2647 **Assessment objective:** verify that symmetric verified-boot operations use device-specific keys and that
2648 global secret keys are not used for verified-boot purposes. References REQ-BM-CON-002.

2649 **Assessment preparation:**

- 2650 • Test environment: the boot manager configured for verified boot using symmetric cryptography.

- 2651 • Required information: the manufacturer's key-provisioning documentation identifying the
2652 derivation or generation path for each symmetric verified-boot key and demonstrating its device-
2653 specific nature (e.g. derivation from a per-device hardware secret, per-device factory provisioning).

- 2654 • Required tools: tooling to extract or compare key values across two distinct devices of the same
2655 model; key-provenance inspection where manufacturer documentation does not suffice.

2656 **Assessment activities:**

- 2657 • For each symmetric key used in verified boot, inspect the documented derivation or provisioning
2658 path and confirm it yields per-device unique values.

- 2659 • Compare the symmetric verified-boot keys across two distinct devices of the same model; confirm
2660 non-equality.

- 2661 • Where asymmetric keys are used for verified boot, confirm this REQ applies only to symmetric
2662 paths and mark asymmetric paths as outside the scope of this ACC.

2663 **Assessment verdict:**

- 2664 • Pass: every symmetric verified-boot key is device-specific by documentation and by cross-device
2665 comparison.

- 2666 • Fail: a symmetric verified-boot key is shared across devices, or the provisioning path does not yield
2667 per-device unique values.

2668 **Assessment evidence:** documentation of key provisioning; record of key comparison across devices; record
2669 of provenance inspection where used.

2670 6.6.3 [ACC-BM-CON-003]

2671 **Assessment objective:** verify that key material is not used for cryptographic purposes other than those for
2672 which it was designated. References REQ-BM-CON-003.

2673 **Assessment preparation:**

- 2674 • Test environment: the boot manager with cryptographic operations observable.
- 2675 • Required information: the manufacturer's key-designation table listing each key and the set of
2676 cryptographic purposes for which it is authorised (e.g. this key for signature verification only; that
2677 key for measurement attestation only).
- 2678 • Required tools: inspection tool for the cryptographic operations performed by the boot manager,
2679 including the key used in each operation.

2680 **Assessment activities:**

- 2681 • For each key, observe the cryptographic operations performed during a representative boot and
2682 record which key is used for which purpose.
- 2683 • Compare observed use against the key-designation table; flag any use of a key outside its
2684 designated purpose set.
- 2685 • Attempt to invoke the boot manager in ways that would require using a key outside its designation
2686 (e.g. crafted input that could solicit misuse); confirm the boot manager does not proceed.

2687 **Assessment verdict:**

- 2688 • Pass: every observed cryptographic operation uses a key within its designated purpose; misuse
2689 attempts are not processed.
- 2690 • Fail: a key is used outside its designated purpose, or a misuse attempt is processed.

2691 **Assessment evidence:** table of key designations; record of observed operations; records of misuse-attempt
2692 tests.

2693 6.6.4 [ACC-BM-CON-004]

2694 **Assessment objective:** verify that sensitive data is not included in crash dumps or log output. References
2695 REQ-BM-CON-004.

2696 **Assessment preparation:**

- 2697 • Test environment: the boot manager in a state where crash handling and log emission can be
2698 exercised; the sensitive-data categories documented per the manufacturer.
- 2699 • Required information: the manufacturer's enumeration of sensitive data categories; the crash-
2700 dump and log-emission paths.
- 2701 • Required tools: fault-injection tool capable of inducing each documented crash condition; log and
2702 crash-dump capture tool.

2703 **Assessment activities:**

- 2704 • Induce each documented crash condition and capture the resulting crash dump.

- 2705 • Generate log output under nominal and error conditions and capture the log content.
- 2706 • Search both capture sets for each sensitive-data category; confirm no sensitive data appears.
- 2707 **Assessment verdict:**
- 2708 • Pass: no sensitive data from any documented category appears in crash dumps or log output under
2709 any induced condition.
- 2710 • Fail: a sensitive-data category appears in a crash dump or log entry.

2711 **Assessment evidence:** list of sensitive data categories; crash-dump captures; log captures; search results
2712 per category.

2713 6.6.5 [ACC-BM-CON-005]

2714 **Assessment objective:** verify that stored credentials are protected using approved mechanisms that
2715 prevent recovery of the original credential values. References REQ-BM-CON-005.

2716 **Assessment preparation:**

- 2717 • Test environment: the boot manager with credentials provisioned and the credential storage
2718 inspectable.
- 2719 • Required information: the manufacturer's documentation of the credential-storage mechanism
2720 (e.g. salted hash function, hardware-protected key-wrap) and the claim that original credentials
2721 cannot be recovered; reference to Annex K for approved mechanisms.
- 2722 • Required tools: credential-storage inspection tool; cryptanalysis or recovery-attempt tooling
2723 appropriate to the declared mechanism.

2724 **Assessment activities:**

- 2725 • Inspect the credential-storage format and confirm it matches the declared mechanism and that the
2726 mechanism is listed in Annex K.
- 2727 • Confirm the stored form does not contain the original credential in recoverable form (e.g. plaintext,
2728 reversible encoding).
- 2729 • Where applicable, attempt offline credential recovery against the stored form using tools
2730 proportionate to the declared mechanism's attack surface; confirm recovery is not feasible within
2731 the declared strength.

2732 **Assessment verdict:**

- 2733 • Pass: credentials are stored using an Annex K approved mechanism; the stored form is not directly
2734 reversible; recovery attempts are infeasible within the declared strength.
- 2735 • Fail: credentials are stored in plaintext or a reversible form; the mechanism is not listed in Annex K;
2736 recovery is feasible within the declared strength.

2737 **Assessment evidence:** specification of credential storage; record of storage-form inspection; record of
2738 recovery-attempt tests where applicable.

2739 6.6.6 [ACC-BM-CON-006]

2740 **Assessment objective:** verify that network boot parameters containing authentication information are
2741 protected by approved mechanisms preventing unauthorised disclosure. References REQ-BM-CON-006.

2742 **Assessment preparation:**

- 2743 • Test environment: the boot manager with network boot configured and the network-boot
2744 parameter storage and transmission paths accessible.
- 2745 • Required information: the manufacturer's identification of network-boot parameters that contain
2746 authentication information; the applied protection mechanism for each (at rest, in transit); Annex K
2747 reference for approved mechanisms.
- 2748 • Required tools: parameter-storage inspection tool; network traffic analyser capable of inspecting
2749 transmitted parameters.

2750 **Assessment activities:**

- 2751 • For each identified parameter, inspect its storage form and confirm it is protected using the
2752 declared mechanism listed in Annex K.
- 2753 • Capture transmissions containing the parameters and confirm the transmitted form is protected
2754 (e.g. encrypted transport, encrypted payload).
- 2755 • Attempt to read the parameter in plaintext from storage and from the transmission; confirm neither
2756 yields the authentication information.

2757 **Assessment verdict:**

- 2758 • Pass: network-boot authentication parameters are protected at rest and in transit using Annex K
2759 approved mechanisms; neither storage inspection nor traffic capture yields plaintext authentication
2760 information.
- 2761 • Fail: an authentication parameter is accessible in plaintext from storage or transmission; the
2762 protection mechanism is not listed in Annex K.

2763 **Assessment evidence:** record of parameter identification; record of storage inspection; record of
2764 transmission capture.

2765 6.6.7 [ACC-BM-CON-007]

2766 **Assessment objective:** verify that secure disposal cryptographically erases sensitive data and clears
2767 security-critical configuration, where technically feasible. References REQ-BM-CON-007.

2768 **Assessment preparation:**

- 2769 • Test environment: the boot manager provisioned with sensitive data and security-critical
2770 configuration, with the secure-disposal operation accessible.
- 2771 • Required information: the manufacturer's specification of the sensitive-data and security-critical
2772 configuration items subject to secure disposal; the cryptographic-erase mechanism; the
2773 documented technical-feasibility limitations and the rationale for each.
- 2774 • Required tools: storage inspection tool for sensitive-data and configuration regions after disposal.

2775 **Assessment activities:**

- 2776 • Invoke the secure-disposal operation with representative sensitive data and security-critical
2777 configuration populated.
- 2778 • Inspect the post-disposal storage for each enumerated item; confirm cryptographic erasure or
2779 clearance per the declared mechanism.
- 2780 • For items documented as outside technical feasibility, confirm the rationale is recorded and the
2781 corresponding residual-risk entry exists in the product technical documentation.

2782 **Assessment verdict:**

- 2783 • Pass: every item subject to secure disposal is cryptographically erased or cleared; any items
2784 excluded by technical feasibility are documented with rationale and residual-risk entry.
- 2785 • Fail: an item subject to secure disposal retains recoverable content after the operation, or an
2786 excluded item lacks documented rationale.

2787 **Assessment evidence:** inventory of sensitive data and configuration; records of inspection after disposal;
2788 documentation of technical feasibility and residual risk.

2789 **6.6.8 [ACC-BM-CON-008]**

2790 **Assessment objective:** verify that the boot manager, where secure disposal is supported, indicates
2791 successful completion of sanitization. References REQ-BM-CON-008.

2792 **Assessment preparation:**

- 2793 • Test environment: the boot manager with secure disposal supported and the completion-indication
2794 channel accessible.
- 2795 • Required information: the manufacturer's declaration of whether secure disposal is supported;
2796 where supported, the form and channel of the completion indication.
- 2797 • Required tools: observation tool for the completion-indication channel (visual, log output,
2798 programmatic response).

2799 **Assessment activities:**

- 2800 • Where the manufacturer declares secure disposal is not supported, record the REQ as not
2801 applicable per 5.1.3 and terminate this assessment.
- 2802 • Where supported, invoke secure disposal and observe the documented completion-indication
2803 channel; confirm the indication is presented upon successful completion.
- 2804 • Abort or interrupt a secure-disposal operation and confirm that the success indication is not
2805 presented.

2806 **Assessment verdict:**

- 2807 • Pass: successful secure disposal produces the documented completion indication; unsuccessful
2808 operations do not.
- 2809 • Fail: successful secure disposal does not produce the indication, or the indication is presented for
2810 unsuccessful operations.

2811 **Assessment evidence:** declaration of support for secure disposal; record of indication on successful
2812 completion; record of non-indication when the operation is interrupted.

2813 6.6.9 [ACC-BM-CON-009]

2814 **Assessment objective:** verify that boot configuration data transmitted over the network is protected for
2815 confidentiality and integrity. References REQ-BM-CON-009.

2816 **Assessment preparation:**

- 2817 • Test environment: the boot manager configured for network boot in a controlled environment
2818 supporting active man-in-the-middle modifications.
- 2819 • Required information: the manufacturer's documentation of the transport protection mechanism
2820 applied to boot configuration data (e.g. authenticated encryption, TLS, payload that is both signed
2821 and encrypted); Annex K reference.
- 2822 • Required tools: passive network capture tool; active man-in-the-middle tool capable of modifying
2823 transmitted data.

2824 **Assessment activities:**

- 2825 • Capture network transmissions of boot configuration data and confirm the payload is not in
2826 plaintext form and is protected by the declared mechanism listed in Annex K.
- 2827 • Execute a man-in-the-middle modification of a boot configuration data payload; confirm the boot
2828 manager detects the integrity violation and rejects the modified data.
- 2829 • Attempt a passive decryption of the captured payload using tools proportionate to the declared
2830 mechanism's strength; confirm infeasibility.

2831 **Assessment verdict:**

- 2832 • Pass: transmitted boot configuration data is protected by an Annex K mechanism for confidentiality
2833 and integrity; integrity violations are detected; passive decryption is infeasible.
- 2834 • Fail: a payload is transmitted in plaintext; integrity violations are not detected; passive decryption
2835 succeeds within declared strength.

2836 **Assessment evidence:** specification of transport protection; record of captured payload; record of integrity-
2837 violation detection.

2838 6.6.10 [ACC-BM-CON-010]

2839 **Assessment objective:** verify that the boot manager overwrites confidential or secret data after use, clears
2840 credentials and temporary data structures before boot target handoff, and does not persist authentication
2841 credentials or confidential cryptographic material beyond the operating system handoff, with the exception
2842 of measurements and credentials used for attestation. References REQ-BM-CON-010.

2843 **Assessment preparation:**

- 2844 • Test environment: the boot manager with memory inspection capability at handoff and across
2845 power-cycle boundaries.
- 2846 • Required information: the manufacturer's enumeration of confidential or secret data, credentials,
2847 and temporary data structures processed during boot; the documented attestation material exempt
2848 from clearance; the clearing mechanism for each item.

- 2849 • Required tools: memory inspection tool at handoff; persistent-storage inspection tool across power
2850 cycles.

2851 **Assessment activities:**

- 2852 • Drive the boot manager through a complete boot with enumerated sensitive items exercised.
- 2853 • Immediately before handoff, inspect memory and relevant data structures for residual sensitive
2854 content; confirm overwriting or clearance for all enumerated items except the documented
2855 attestation exceptions.
- 2856 • Power-cycle the device and inspect persistent storage for authentication credentials or confidential
2857 cryptographic material; confirm no persistence beyond the current boot cycle except for
2858 documented attestation material.

2859 **Assessment verdict:**

- 2860 • Pass: at handoff, enumerated sensitive items are overwritten or cleared except for documented
2861 attestation material; across power cycles, no authentication credentials or confidential
2862 cryptographic material persist beyond the documented attestation exceptions.
- 2863 • Fail: a sensitive item remains in memory at handoff without documented exception; an
2864 authentication credential or cryptographic material persists across power cycles outside the
2865 documented exceptions.

2866 **Assessment evidence:** inventory of sensitive items; record of memory inspection at handoff; record of
2867 inspection after power cycling.

2868 **6.7 Integrity**

2869 The present clause specifies assessment criteria for the requirements in Clause 5.7.

2870

Table 6.7.1-1: Requirements addressed in the present clause

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-INT-001	Privilege boundaries between boot stages	Cross-stage access attempts for each documented boundary	All
REQ-BM-INT-002	Chain-of-trust verification to root trust anchor	Tests with full valid chains and with injected broken chains	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-003	Halt boot on verification failure	Per-class verification-failure injection; halt observation	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-004	Measurement chain rooted in hardware trust anchor	Measurement capture + expected-chain reproduction; root identity check	MEDIUM, HIGH (if implemented)
REQ-BM-INT-005	Verify update packages before installation	Tests with valid packages, altered packages, and packages modified after transfer	MEDIUM, HIGH
REQ-BM-INT-006	Software-based verification with protected keys	Platform HW availability check; verification-path observation; key-storage inspection	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-007	Configurable approved signing algorithm list	List inspection against Annex K; documented modification test	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-008	Multiple signature algorithms supported	Per-algorithm verification test	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-009	Verification bypass requires authentication	Authorised and unauthorised bypass-invocation tests	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-010	No modification between verification and execution	Concurrent-modification test during the verification window	All
REQ-BM-INT-011	Load into protected memory before verification	Loading-sequence observation; unprotected-path modification test	MEDIUM, HIGH (if HW supports)
REQ-BM-INT-012	Sensitive config: HW-storage or authenticated encryption	Storage inspection; ciphertext non-determinism; freshness/replay test	All (if implemented)
REQ-BM-INT-013	Restore secure defaults on configuration corruption	Corruption-injection per category; restoration outcome; feasibility documentation	All (if implemented)
REQ-BM-INT-014	Authenticate network boot servers by certificate	Valid, no-cert, untrusted-root server-connection tests	MEDIUM, HIGH (if implemented)
REQ-BM-INT-015	Reject invalid/expired/revoked server certs	Server-connection tests for each failure state; revocation-mechanism exercise	MEDIUM, HIGH (if implemented)
REQ-BM-INT-016	Validate network response origin	Authorised / rogue source-response tests; race-condition test	MEDIUM, HIGH (if implemented)
REQ-BM-INT-017	Rollback protection for boot stages and config	Older-version rejection; current/newer acceptance; OEM policy bounds	All (if implemented)
REQ-BM-INT-018	HW-backed anti-rollback counters; signed version metadata	Counter-storage inspection; metadata-signature tests; counter-decrement rejection	HIGH only
REQ-BM-INT-019	Certificate chain validation to trusted root	Submissions of valid chains and chains representing each failure class	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-020	Revocation of mutable keys and certificates	Pre/post-revocation usage; persistence across power cycle; downgrade bypass	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-021	Cryptographic algorithms per Annex K	Per-operation algorithm/key-size/parameter inspection; deprecation-schedule check	All
REQ-BM-INT-022	Cryptographic agility (algorithm migration)	Trust-anchor update; post-update verification; migration-policy test	MEDIUM, HIGH

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-INT-023	Collision/preimage-resistant one-way functions	Per-function inspection against Annex K categorisation	MEDIUM, HIGH (LOW if implemented)
REQ-BM-INT-024	Trust-anchor agility (key replacement)	Trust-anchor update; post-update verification; migration-policy test	MEDIUM, HIGH

2871

2872 **6.7.1 [ACC-BM-INT-001]**

2873 **Assessment objective:** verify that the boot manager enforces privilege boundaries between boot stages
 2874 such that code in one stage cannot obtain privileges assigned to another stage or cross established trust
 2875 boundaries. References REQ-BM-INT-001.

2876 **Assessment preparation:**

- 2877 • Test environment: the boot manager at a point where two or more boot stages execute with
 2878 distinct privileges.
- 2879 • Required information: the manufacturer's specification of privilege boundaries between stages
 2880 (memory regions, privileged instructions, security-sensitive hardware access) and the enforcement
 2881 mechanism for each.
- 2882 • Required tools: test payload executed within a stage attempting cross-stage privilege access.

2883 **Assessment activities:**

- 2884 • For each privilege boundary, execute a test payload in the lower-privileged stage attempting access
 2885 to the higher-privileged stage's memory, instructions, or hardware resources; confirm rejection.
- 2886 • Attempt invocation across a trust boundary (e.g. calling a higher-trust routine with crafted
 2887 arguments) and confirm the enforcement mechanism prevents escalation.

2888 **Assessment verdict:**

- 2889 • Pass: every documented privilege boundary rejects cross-stage access attempts.
- 2890 • Fail: a cross-stage access or trust-boundary crossing succeeds.

2891 **Assessment evidence:** enumeration of privilege boundaries; records of cross-access tests per boundary.

2892 **6.7.2 [ACC-BM-INT-002]**

2893 **Assessment objective:** verify that the boot manager verifies integrity and authenticity of each boot stage
 2894 using approved cryptographic mechanisms and establishes a chain of trust to a root trust anchor before
 2895 transferring control. References REQ-BM-INT-002.

2896 **Assessment preparation:**

- 2897 • Test environment: the boot manager with verified boot active and representative boot-stage
 2898 artefacts.
- 2899 • Required information: the manufacturer's specification of the chain of trust (stages, credentials,
 2900 verification algorithm per stage); the identity of the root trust anchor.
- 2901 • Required tools: signed boot-stage artefacts using trusted and untrusted credentials; verification-
 2902 step observation tool.

2903 **Assessment activities:**

2904 • Execute a full boot with all stages validly signed; confirm each stage is verified using an approved
2905 mechanism per Annex K and that the chain is traced back to the root trust anchor before control
2906 transfer.

2907 • Substitute one stage's credential with an untrusted credential; confirm the chain of trust check
2908 detects the break and halts before that stage's control transfer.

2909 **Assessment verdict:**

2910 • Pass: every stage is verified using an approved mechanism; the chain of trust reaches the root trust
2911 anchor before control transfer; a broken chain halts control transfer.

2912 • Fail: a stage executes without verification; the chain of trust does not reach the root trust anchor; a
2913 broken chain is not detected.

2914 **Assessment evidence:** chain-of-trust specification; observation of verification per stage; record of injection
2915 of broken chains.

2916 **6.7.3 [ACC-BM-INT-003]**

2917 **Assessment objective:** verify that the boot manager prevents boot continuation with components that fail
2918 integrity or authenticity verification. References REQ-BM-INT-003.

2919 **Assessment preparation:**

2920 • Test environment: the boot manager with verified boot active and the ability to substitute
2921 components with documented failure conditions.

2922 • Required information: the manufacturer's enumeration of verification failure classes (invalid
2923 credential, expired credential, revoked credential, tampered component, missing verification data).

2924 • Required tools: per-class component-substitution tool.

2925 **Assessment activities:**

2926 • For each enumerated failure class, substitute a component exhibiting that failure condition and
2927 initiate boot.

2928 • Confirm the boot manager halts prior to executing the failed component.

2929 • Confirm no subsequent boot stage is executed while the failure condition persists.

2930 **Assessment verdict:**

2931 • Pass: each failure class halts boot before execution of the failed component; no subsequent stages
2932 execute.

2933 • Fail: a failed component is executed; a subsequent stage executes despite a prior failure.

2934 **Assessment evidence:** enumeration of failure classes; records of substitution and halt per class.

2935 6.7.4 [ACC-BM-INT-004]

2936 **Assessment objective:** verify that the boot manager computes and records a cryptographic measurement
2937 of each boot stage into hardware-protected storage before transferring control, establishing a measurement
2938 chain rooted in a hardware-based trust anchor. References REQ-BM-INT-004.

2939 **Assessment preparation:**

- 2940 • Test environment: the boot manager with measured boot active on a platform with hardware-
2941 protected measurement storage (e.g. TPM).
- 2942 • Required information: the manufacturer's specification of the measurement scheme (stages
2943 measured, hash algorithm, storage target, hardware root of trust).
- 2944 • Required tools: hardware measurement-storage read tool; external verifier capable of reproducing
2945 the expected measurement chain.

2946 **Assessment activities:**

- 2947 • Execute a boot and read the measurement storage at the transition between stages; confirm that
2948 each stage's measurement is recorded before control is transferred.
- 2949 • Reproduce the expected measurement chain using the external verifier from the documented initial
2950 values; confirm the recorded chain matches.
- 2951 • Confirm the root of the measurement chain is the hardware trust anchor.

2952 **Assessment verdict:**

- 2953 • Pass: every stage measurement is recorded in hardware-protected storage before control transfer;
2954 the chain matches the expected computation; the root is the hardware trust anchor.
- 2955 • Fail: a stage is not measured; a measurement is recorded after control transfer; the chain deviates
2956 from expected computation; the root is not the hardware trust anchor.

2957 **Assessment evidence:** specification of the measurement scheme; capture of measurements per stage;
2958 record of reproduction of the expected chain.

2959 6.7.5 [ACC-BM-INT-005]

2960 **Assessment objective:** verify that the boot manager verifies authenticity and integrity of update packages
2961 before installation, and additionally after network transfer where updates are written to protected storage.
2962 References REQ-BM-INT-005.

2963 **Assessment preparation:**

- 2964 • Test environment: the boot manager in its update-receiving and update-installing configuration;
2965 where applicable, with protected intermediate storage for updates.
- 2966 • Required information: the manufacturer's specification of the verification algorithm applied to
2967 update packages; whether protected intermediate storage is used; the verification point(s) per the
2968 REQ.
- 2969 • Required tools: update-package generation tool producing packages with valid and invalid
2970 authenticity/integrity; network-transfer capture tool.

2971 **Assessment activities:**

- 2972 • Submit a package with valid authenticity and integrity; confirm verification succeeds and
2973 installation proceeds.
- 2974 • Submit a package altered in transit (valid at source, altered after signing); confirm verification
2975 rejects before installation.
- 2976 • Where protected intermediate storage is used, verify the additional verification after network
2977 transfer before the package is committed to that storage; attempt modification between transfer
2978 and storage commit and confirm detection.

2979 **Assessment verdict:**

- 2980 • Pass: valid packages are installed; altered packages are rejected before installation; where
2981 protected storage is used, verification after network transfer catches in-flight alteration.
- 2982 • Fail: an altered package is installed; a post-transfer modification to protected storage is not
2983 detected.

2984 **Assessment evidence:** verification specification; test records per package; record of modification tests after
2985 network transfer.

2986 **6.7.6 [ACC-BM-INT-006]**

2987 **Assessment objective:** verify that, where hardware security components are unavailable, the boot manager
2988 implements software-based cryptographic verification with keys stored in protected storage. References
2989 REQ-BM-INT-006.

2990 **Assessment preparation:**

- 2991 • Test environment: the boot manager operating on a platform where hardware security components
2992 are absent or documented as unavailable.
- 2993 • Required information: the manufacturer's specification of the software-based verification
2994 mechanism and the protected-storage location for the keys.
- 2995 • Required tools: storage inspection tool; software-side verification observation.

2996 **Assessment activities:**

- 2997 • Confirm that hardware security components are unavailable on the platform under assessment;
2998 where available, record the REQ as not applicable per 5.1.3 and terminate.
- 2999 • Where unavailable, observe the software-based verification path is used for verified boot and that
3000 keys are retrieved from the documented protected storage.
- 3001 • Attempt to modify the stored keys through any interface other than the documented authorised
3002 path; confirm rejection.

3003 **Assessment verdict:**

- 3004 • Pass: software-based verification is used in absence of hardware security components; keys reside
3005 in the documented protected storage; unauthorised modification of the keys is rejected.
- 3006 • Fail: software-based verification is absent; keys are in unprotected storage; unauthorised key
3007 modification is accepted.

3008 **Assessment evidence:** record of platform availability; observation of the verification path; inspection of key
3009 storage.

3010 6.7.7 [ACC-BM-INT-007]

3011 **Assessment objective:** verify that the boot manager maintains a configurable list of approved signing
3012 algorithms where the trust anchor is not immutable. References REQ-BM-INT-007.

3013 **Assessment preparation:**

3014 • Test environment: the boot manager with verified boot active, a mutable trust anchor, and
3015 configuration capability exposed.

3016 • Required information: the manufacturer's specification of the list of approved algorithms, its
3017 configuration interface, and the authorisation required to modify it.

3018 • Required tools: configuration tool.

3019 **Assessment activities:**

3020 • Inspect the current approved-algorithm list and confirm each entry is an approved algorithm per
3021 Annex K.

3022 • Apply a documented change to the list (add, remove, or reorder an algorithm) via the configuration
3023 interface and confirm the change is reflected in the boot manager's verification behaviour.

3024 • Attempt to modify the list without the documented authorisation; confirm rejection.

3025 **Assessment verdict:**

3026 • Pass: the approved-algorithm list is configurable through the documented interface with
3027 authorisation; each entry is an approved algorithm; changes take effect; unauthorised changes are
3028 rejected.

3029 • Fail: the list is not configurable; it contains non-approved entries; changes do not take effect;
3030 unauthorised changes are accepted.

3031 **Assessment evidence:** inspection of the current list; test records of documented changes; record of
3032 unauthorised-modification tests.

3033 6.7.8 [ACC-BM-INT-008]

3034 **Assessment objective:** verify that the boot manager, where the trust anchor is not immutable, supports
3035 multiple signature algorithms. References REQ-BM-INT-008.

3036 **Assessment preparation:**

3037 • Test environment: the boot manager configured with verified boot and multiple approved
3038 algorithms active.

3039 • Required information: the manufacturer's list of signature algorithms the boot manager supports
3040 concurrently.

3041 • Required tools: boot-stage artefacts signed with each of the supported algorithms.

3042 **Assessment activities:**

3043 • For each supported algorithm, present a boot-stage artefact signed with that algorithm and confirm
3044 successful verification.

3045 • Where multiple algorithms are configurable concurrently, confirm the boot manager can verify
3046 artefacts using any configured algorithm without prior re-configuration.

3047 **Assessment verdict:**

3048 • Pass: at least two approved signature algorithms are supported and each verifies successfully when
3049 used independently.

3050 • Fail: only one signature algorithm is supported; verification fails for an algorithm declared as
3051 supported.

3052 **Assessment evidence:** list of supported algorithms; verification test records per algorithm.

3053 6.7.9 [ACC-BM-INT-009]

3054 **Assessment objective:** verify that verification bypass requires authentication. References REQ-BM-INT-009.

3055 **Assessment preparation:**

3056 • Test environment: the boot manager with verified boot active and any documented verification-
3057 bypass path exposed.

3058 • Required information: the manufacturer's specification of the verification-bypass path (if any) and
3059 the authentication requirement.

3060 • Required tools: authentication stimulation tool; bypass-invocation tool.

3061 **Assessment activities:**

3062 • Where no verification-bypass path is documented, attempt to invoke bypass via undocumented
3063 paths; confirm rejection and terminate.

3064 • Where a documented bypass path exists, attempt invocation without authentication; confirm
3065 rejection.

3066 • Invoke with the documented authentication; confirm the bypass is granted for the scope defined by
3067 the manufacturer and no wider scope.

3068 **Assessment verdict:**

3069 • Pass: verification bypass is accepted only with the documented authentication; no undocumented
3070 bypass path grants verification bypass.

3071 • Fail: verification bypass is accepted without authentication; an undocumented path grants bypass.

3072 **Assessment evidence:** specification of the bypass path; records of authorised and unauthorised
3073 invocations.

3074 6.7.10 [ACC-BM-INT-010]

3075 **Assessment objective:** verify that the boot manager prevents unauthorised modification between
3076 verification and execution. References REQ-BM-INT-010.

3077 **Assessment preparation:**

- 3078 • Test environment: the boot manager with inspection access to the memory region holding a
3079 verified component between verification and execution.
- 3080 • Required information: the manufacturer's specification of the memory-protection mechanism
3081 applied between verification and execution (e.g. locked page tables, execute-in-place from write-
3082 protected storage).
- 3083 • Required tools: memory-modification attempt tool operating concurrently with the boot manager's
3084 window between verification and execution.

3085 **Assessment activities:**

- 3086 • During the window between a component's verification and its execution, attempt to modify the
3087 component's memory through any accessible interface (DMA, other processor cores where
3088 applicable, peripheral-initiated access).
- 3089 • Confirm the memory-protection mechanism prevents modification, or that the modification is
3090 detected and execution halted.
- 3091 • Verify the protection mechanism remains in effect for the full window (no early release).

3092 **Assessment verdict:**

- 3093 • Pass: modification attempts are prevented or detected for the full verification-to-execution
3094 window; the protection mechanism does not release early.
- 3095 • Fail: modification succeeds during the window without detection; the protection mechanism
3096 releases before execution.

3097 **Assessment evidence:** specification of memory protection; record of concurrent-modification tests.

3098 6.7.11 [ACC-BM-INT-011]

3099 **Assessment objective:** verify that, where the hardware platform supports protected memory, the boot
3100 manager loads components into protected memory before verification. References REQ-BM-INT-011.

3101 **Assessment preparation:**

- 3102 • Test environment: the boot manager on a platform with documented protected-memory capability
3103 (e.g. locked memory regions, enclaves).
- 3104 • Required information: the manufacturer's declaration of the platform's protected-memory support
3105 and the mechanism used.
- 3106 • Required tools: memory-region inspection tool capable of identifying protected regions.

3107 **Assessment activities:**

- 3108 • Confirm the platform supports protected memory as documented; where not supported, record
3109 the REQ as not applicable per 5.1.3 and terminate.
- 3110 • Observe the component-loading sequence; confirm components are placed into the protected
3111 region before verification begins.
- 3112 • Attempt to modify the component content during the verification window through any unprotected
3113 access path; confirm the attempt fails due to the protection.

3114 **Assessment verdict:**

- 3115 • Pass: on supporting platforms, components are loaded into protected memory before verification;
3116 modification attempts through unprotected paths fail.
- 3117 • Fail: on supporting platforms, components are verified before being loaded into protected memory;
3118 unprotected-path modification succeeds.

3119 **Assessment evidence:** record of platform capability; observation of the loading sequence; record of
3120 modification attempts.

3121 **6.7.12 [ACC-BM-INT-012]**

3122 **Assessment objective:** verify that sensitive configuration settings are protected by hardware-protected
3123 storage or by authenticated encryption employing freshness protection and non-deterministic encryption.
3124 References REQ-BM-INT-012.

3125 **Assessment preparation:**

- 3126 • Test environment: the boot manager with configuration capability and sensitive configuration items
3127 populated.
- 3128 • Required information: the manufacturer's enumeration of sensitive configuration settings; the
3129 protection mechanism for each (hardware-protected storage, or authenticated encryption with
3130 freshness and non-deterministic properties); cryptographic parameters listed in Annex K.
- 3131 • Required tools: storage inspection tool; ciphertext analysis tool for entropy and determinism.

3132 **Assessment activities:**

- 3133 • For each sensitive configuration setting, identify the protection mechanism and confirm it matches
3134 the declared category.
- 3135 • Where hardware-protected storage is used, inspect the storage and confirm the setting is not
3136 accessible outside the protected path.
- 3137 • Where authenticated encryption is used, confirm the algorithm is listed in Annex K; confirm the
3138 ciphertext for equal plaintexts differs across encryption events (non-determinism); confirm
3139 freshness is enforced by attempting replay of a prior ciphertext and observing rejection.

3140 **Assessment verdict:**

- 3141 • Pass: every sensitive setting is protected as declared; hardware-protected settings are inaccessible
3142 outside the protected path; encrypted settings are non-deterministic and freshness-protected.
- 3143 • Fail: a sensitive setting lacks the declared protection; ciphertext is deterministic; replay is accepted.

3144 **Assessment evidence:** mapping of settings to protection mechanisms; records of storage inspection;
3145 ciphertext non-determinism test; replay-rejection test.

3146 **6.7.13 [ACC-BM-INT-013]**

3147 **Assessment objective:** verify that the boot manager restores secure default configuration settings when
3148 configuration corruption is detected, where technically feasible. References REQ-BM-INT-013.

3149 **Assessment preparation:**

- 3150 • Test environment: the boot manager with configuration capability and the ability to inject
3151 configuration corruption.
- 3152 • Required information: the manufacturer's specification of the corruption-detection mechanism; the
3153 secure default configuration; the documented technical-feasibility limitations (e.g. immutable-
3154 element restrictions).
- 3155 • Required tools: configuration-corruption injection tool.

3156 **Assessment activities:**

- 3157 • Inject corruption into each configurable category and confirm the boot manager detects the
3158 corruption.
- 3159 • Confirm that, where technically feasible, the boot manager restores the secure default for the
3160 affected category.
- 3161 • For categories documented as outside technical feasibility (e.g. immutable e-fuses), confirm the
3162 documented handling (retain residual-risk disclosure, documented limitation).

3163 **Assessment verdict:**

- 3164 • Pass: corruption is detected in every configurable category; restoration to secure default succeeds
3165 where feasible; documented limitations apply to the documented categories.
- 3166 • Fail: corruption is not detected; restoration fails in a category documented as feasible; a category
3167 claimed infeasible lacks documentation.

3168 **Assessment evidence:** records of corruption injection; records of restoration outcome; record of feasibility
3169 documentation.

3170 6.7.14 [ACC-BM-INT-014]

3171 **Assessment objective:** verify that the boot manager authenticates network boot servers using
3172 cryptographic certificates when boot servers are authorised to launch actions or provide unauthenticated
3173 data or configuration. References REQ-BM-INT-014.

3174 **Assessment preparation:**

- 3175 • Test environment: the boot manager configured for network boot in a controlled environment with
3176 distinct authorised and unauthorised server identities.
- 3177 • Required information: the manufacturer's specification of the server authentication mechanism
3178 (certificate format, trust roots, verification algorithm) and the set of actions/data affected.
- 3179 • Required tools: controlled network boot server capable of presenting valid certificates, invalid
3180 certificates, or no certificates.

3181 **Assessment activities:**

- 3182 • Connect the boot manager to a server presenting a valid trusted certificate; confirm the boot
3183 manager accepts authorised actions and data from the server.
- 3184 • Connect to a server presenting no certificate; confirm the boot manager rejects the session before
3185 any action is launched or data is accepted.
- 3186 • Connect to a server presenting a certificate from an untrusted root; confirm rejection.

3187 **Assessment verdict:**

3188 • Pass: only servers presenting a cryptographically valid trusted certificate can launch actions or
3189 provide data; unauthenticated or untrusted-root servers are rejected.

3190 • Fail: an action is launched or data accepted from a server without a valid trusted certificate.

3191 **Assessment evidence:** specification of server authentication; connection records per scenario.

3192 **6.7.15 [ACC-BM-INT-015]**

3193 **Assessment objective:** verify that the boot manager rejects network boot connections with invalid, expired,
3194 or revoked server certificates. References REQ-BM-INT-015.

3195 **Assessment preparation:**

3196 • Test environment: the boot manager configured for network boot with access to certificate-
3197 validation reference data (CRL, OCSP, or documented equivalent).

3198 • Required information: the manufacturer's specification of the certificate-validity checks applied; the
3199 revocation mechanism.

3200 • Required tools: controlled server capable of presenting certificates in each failure state (invalid
3201 signature, expired, revoked, not-yet-valid, malformed).

3202 **Assessment activities:**

3203 • For each certificate failure state, connect the boot manager to the server and confirm the session is
3204 rejected before any network boot action.

3205 • Confirm revocation status is checked by the documented mechanism (e.g. CRL retrieval, OCSP
3206 query, local revocation store consultation); where offline, confirm the offline mechanism is present.

3207 **Assessment verdict:**

3208 • Pass: each certificate failure state results in connection rejection; the revocation mechanism is
3209 exercised.

3210 • Fail: a failure state is not rejected; the revocation mechanism is absent or bypassable.

3211 **Assessment evidence:** enumeration of failure states; connection records per state; record of exercise of the
3212 revocation mechanism.

3213 **6.7.16 [ACC-BM-INT-016]**

3214 **Assessment objective:** verify that the boot manager validates that network configuration and boot-related
3215 responses originate from authorised infrastructure. References REQ-BM-INT-016.

3216 **Assessment preparation:**

3217 • Test environment: the boot manager configured for network boot in an environment supporting
3218 authorised and rogue response sources.

3219 • Required information: the manufacturer's enumeration of response types (DHCP, PXE offers,
3220 network-provided boot configuration, similar) and the authorisation mechanism applied to each
3221 (e.g. signed responses, authenticated transport, trusted source address enforcement).

- 3222 • Required tools: response injection tools representing authorised and rogue sources.
- 3223 **Assessment activities:**
- 3224 • For each enumerated response type, present a response from an authorised source and confirm
3225 acceptance.
- 3226 • Present a response from a rogue source and confirm rejection.
- 3227 • Where a race condition may exist between authorised and rogue responses, confirm the boot
3228 manager selects the authorised response.
- 3229 **Assessment verdict:**
- 3230 • Pass: each response type accepts authorised sources and rejects rogue sources; race conditions
3231 favour authorised responses.
- 3232 • Fail: a rogue response is accepted for any response type; a race condition causes rogue preference.
- 3233 **Assessment evidence:** enumeration of response types; records of authorised and rogue inputs per type;
3234 record of race-condition tests.
- 3235 **6.7.17 [ACC-BM-INT-017]**
- 3236 **Assessment objective:** verify that the boot manager enforces rollback protection when verifying boot
3237 stages and configuration data. References REQ-BM-INT-017.
- 3238 **Assessment preparation:**
- 3239 • Test environment: the boot manager with update or configuration capability and representative
3240 current and prior-version artefacts.
- 3241 • Required information: the manufacturer’s specification of the rollback-protection mechanism
3242 (version counter, signed version metadata, reference to OEM configurability per 5.3 note).
- 3243 • Required tools: versioned artefact generation tool.
- 3244 **Assessment activities:**
- 3245 • Apply a prior-version boot stage or configuration signed correctly but with an earlier version than
3246 the current; confirm the boot manager rejects the older artefact.
- 3247 • Apply the current or a newer authorised version; confirm acceptance.
- 3248 • Where OEM rollback configurability is documented, confirm the OEM-configurable path remains
3249 bounded by the manufacturer’s policy and does not disable rollback protection for stages where
3250 the policy requires it.
- 3251 **Assessment verdict:**
- 3252 • Pass: prior-version artefacts are rejected; newer authorised versions are accepted; OEM-
3253 configurable relaxation stays within the documented policy.
- 3254 • Fail: a prior-version artefact is accepted; OEM configurability disables protection where the policy
3255 forbids.
- 3256 **Assessment evidence:** test records of versioned artefacts; record of OEM policy inspection.

3257 6.7.18 [ACC-BM-INT-018]

3258 **Assessment objective:** verify that, at HIGH security profile, the boot manager stores anti-rollback counters
3259 in hardware-backed or tamper-evident storage and verifies signed version metadata before accepting
3260 updates. References REQ-BM-INT-018.

3261 **Assessment preparation:**

3262 • Test environment: the boot manager at HIGH security profile with update capability and the anti-
3263 rollback counter storage inspectable.

3264 • Required information: the manufacturer's specification of the counter storage mechanism
3265 (hardware-backed or tamper-evident) and the signed version metadata format.

3266 • Required tools: counter storage inspection tool; version-metadata signing tool producing valid,
3267 altered, and unsigned metadata.

3268 **Assessment activities:**

3269 • Inspect the counter storage and confirm it is hardware-backed or tamper-evident per the declared
3270 mechanism.

3271 • Submit an update package with correctly signed version metadata with version \geq counter; confirm
3272 acceptance.

3273 • Submit an update package with altered metadata (signature invalid or metadata altered after
3274 signing); confirm rejection.

3275 • Submit an update package with unsigned metadata; confirm rejection before any update action.

3276 • Attempt to decrement the counter through any interface; confirm rejection.

3277 **Assessment verdict:**

3278 • Pass: counter storage is hardware-backed or tamper-evident; signed metadata is verified before
3279 acceptance; altered or unsigned metadata is rejected; counter decrement attempts are rejected.

3280 • Fail: counter storage is not hardware-backed or tamper-evident; metadata verification is absent or
3281 late; counter decrement succeeds.

3282 **Assessment evidence:** inspection of counter storage; submission records per metadata variant; record of
3283 counter decrement attempts.

3284 6.7.19 [ACC-BM-INT-019]

3285 **Assessment objective:** verify that certificate chains are validated against a trusted root before certificates
3286 are used for security-relevant operations. References REQ-BM-INT-019.

3287 **Assessment preparation:**

3288 • Test environment: the boot manager with verified boot or measured boot supporting remote
3289 attestation.

3290 • Required information: the manufacturer's trusted-root set; the chain-validation policy (end-entity
3291 to root, validity-period enforcement, revocation, policy constraints).

- 3292 • Required tools: certificate generation tool producing chains rooted in trusted and untrusted roots,
3293 with and without each validation property.

3294 **Assessment activities:**

- 3295 • Present certificates for a security-relevant operation with a valid chain to a trusted root, within
3296 validity period, not revoked, and policy-compliant; confirm acceptance.
- 3297 • Present certificates with the chain broken at each point, including: chain ending at an untrusted
3298 root; expired intermediate; revoked intermediate; policy-violation at any level; confirm rejection at
3299 each failure.

3300 **Assessment verdict:**

- 3301 • Pass: only full chains to a trusted root meeting all validation criteria are accepted; every
3302 documented failure class is rejected.
- 3303 • Fail: a chain failing any validation criterion is accepted.

3304 **Assessment evidence:** record of the trusted-root set; test records for each failure class.

3305 **6.7.20 [ACC-BM-INT-020]**

3306 **Assessment objective:** verify that the boot manager supports revocation of mutable compromised keys and
3307 certificates. References REQ-BM-INT-020.

3308 **Assessment preparation:**

- 3309 • Test environment: the boot manager with verified boot or measured boot supporting remote
3310 attestation, and a mutable trust anchor or certificate store.
- 3311 • Required information: the manufacturer's specification of the revocation mechanism (revocation
3312 database, CRL, OCSP, authenticated configuration update, revocation metadata); the revocation-slot
3313 constraints.
- 3314 • Required tools: revocation-data generation tool.

3315 **Assessment activities:**

- 3316 • Designate a previously-valid key or certificate as compromised and apply revocation through the
3317 documented mechanism.
- 3318 • Attempt to use the revoked key or certificate in a security-relevant operation; confirm rejection
3319 post-revocation where acceptance occurred pre-revocation.
- 3320 • Confirm the revocation state survives power cycle and is not subject to bypass via downgrade or
3321 replay of pre-revocation state.

3322 **Assessment verdict:**

- 3323 • Pass: revocation applied through the documented mechanism causes the previously-valid key or
3324 certificate to be rejected; the revocation state persists and is not bypassable.
- 3325 • Fail: revocation is not enforced after application; the revocation state does not persist; a bypass
3326 path exists.

3327 **Assessment evidence:** specification of the revocation mechanism; usage records before and after
3328 revocation; records of persistence and bypass attempts.

3329 6.7.21 [ACC-BM-INT-021]

3330 **Assessment objective:** verify that the boot manager uses cryptographic algorithms, key sizes, and
3331 parameters in accordance with Annex K. References REQ-BM-INT-021.

3332 **Assessment preparation:**

- 3333 • Test environment: the boot manager with cryptographic operations observable.
- 3334 • Required information: the manufacturer's inventory of cryptographic algorithms, key sizes, and
3335 parameters used by the boot manager; reference to Annex K.
- 3336 • Required tools: inspection tool for cryptographic operations (algorithm, key size, parameters).

3337 **Assessment activities:**

- 3338 • For each cryptographic operation performed by the boot manager, observe and record the
3339 algorithm, key size, and parameters used.
- 3340 • Compare against Annex K; confirm every used combination is listed and within the applicable
3341 deprecation schedule.

3342 **Assessment verdict:**

- 3343 • Pass: every observed algorithm, key size, and parameter combination is listed in Annex K and within
3344 the applicable deprecation schedule.
- 3345 • Fail: any combination used is not listed in Annex K, or has passed its deprecation schedule.

3346 **Assessment evidence:** inventory of cryptographic operations with Annex K cross-reference.

3347 6.7.22 [ACC-BM-INT-022]

3348 **Assessment objective:** verify that the boot manager supports migration between approved cryptographic
3349 algorithms. References REQ-BM-INT-022.

3350 **Assessment preparation:**

- 3351 • Test environment: the boot manager configured to operate with at least one approved
3352 cryptographic algorithm, and capable of operating with at least one alternative approved algorithm.
- 3353 • Required information: the manufacturer's specification of the algorithm-migration mechanism
3354 (configuration metadata, multi-algorithm support, controlled algorithm selection, or boot manager
3355 replacement); the list of approved algorithms supported; reference to Annex K
- 3356 • Required tools: tooling to invoke the migration mechanism as documented by the manufacturer
3357 (e.g. metadata update tool, algorithm selection interface, or the update mechanism where
3358 migration is achieved through boot manager update).

3359 **Assessment activities:**

- 3360 • Confirm that at least one approved algorithm is in active use as the verification algorithm.

- 3361 • Through the documented migration mechanism, transition the boot manager from the current
3362 approved algorithm to a different approved algorithm from Annex K.
- 3363 • Confirm that, after migration, the boot manager performs verification under the new algorithm;
3364 artefacts signed under the new algorithm are accepted; the policy applied to artefacts signed under
3365 the previous algorithm matches documentation (e.g. grace period, immediate rejection,
3366 documented transition).
- 3367 • Where multi-algorithm concurrent support is documented, confirm that artefacts signed under any
3368 active algorithm are verified.

3369 **Assessment verdict:**

- 3370 • Pass: the documented migration mechanism transitions the boot manager between approved
3371 algorithms; verification after migration operates under the new algorithm; transition behaviour
3372 matches the documented policy.
- 3373 • Fail: no migration mechanism exists or is functional; verification after migration fails; transition
3374 behaviour diverges from documentation.

3375 **Assessment evidence:** specification of the algorithm-migration mechanism; records of verification before
3376 and after migration; record of the migration policy; records of multi-algorithm verification where applicable.

3377 **6.7.23 [ACC-BM-INT-023]**

3378 **Assessment objective:** verify that the boot manager uses collision-resistant and preimage-resistant one-way
3379 functions for origin authentication and verification of boot-code provenance. References REQ-BM-INT-023.

3380 **Assessment preparation:**

- 3381 • Test environment: the boot manager with verified boot or measured boot active.
- 3382 • Required information: the manufacturer's specification of the one-way functions used for origin
3383 authentication and provenance verification; reference to Annex K for collision- and preimage-
3384 resistance.
- 3385 • Required tools: inspection tool for the one-way functions invoked during boot.

3386 **Assessment activities:**

- 3387 • Observe the one-way functions invoked during verification and measurement operations; confirm
3388 each is listed in Annex K and categorised as collision-resistant and preimage-resistant.
- 3389 • Confirm that no function used for these purposes has an Annex K deprecation that applies to the
3390 current assessment date.

3391 **Assessment verdict:**

- 3392 • Pass: every one-way function used for origin authentication and provenance verification is Annex K
3393 approved and within its deprecation schedule.
- 3394 • Fail: a function used for these purposes is not Annex K approved or has passed its deprecation
3395 schedule.

3396 **Assessment evidence:** inventory of invoked functions with Annex K cross-reference.

3397 **6.7.24 [ACC-BM-INT-024]**

3398 **Assessment objective:** verify that the boot manager supports replacement of trust anchors used for boot
 3399 verification. References REQ-BM-INT-024.

3400 **Assessment preparation:**

- 3401
- Test environment: the boot manager with update capability and a mutable trust anchor store.
- 3402
- Required information: the manufacturer's specification of the trust-anchor replacement mechanism (e.g. authenticated update path, key provisioning interface, recovery procedure); the list of trust
 3403 anchors and the procedure for replacing each; the documented authority required to authorise
 3404 replacement
 3405
- 3406
- Required tools: the trust-anchor replacement tool or interface as documented by the manufacturer;
 3407 signing material under both the existing and the replacement trust anchor.

3408 **Assessment activities:**

- 3409
- Confirm the trust anchor is mutable and can be replaced through the documented authorised path.
- 3410
- Replace the trust anchor with a new anchor; confirm the boot manager subsequently verifies
 3411 artefacts signed under the new anchor.
- 3412
- Confirm that artefacts signed under the prior anchor are subject to the documented post-
 3413 replacement policy (e.g. immediate rejection, documented transition, configurable grace period).
- 3414
- Attempt trust-anchor replacement through an unauthorised path; confirm rejection.

3415 **Assessment verdict:**

- 3416
- Pass: the trust anchor is replaceable through the documented authorised path; post-replacement
 3417 verification operates under the new anchor; unauthorised replacement attempts are rejected.
- 3418
- Fail: the trust anchor cannot be replaced; post-replacement verification fails; unauthorised
 3419 replacement is accepted.

3420 **Assessment evidence:** record of trust-anchor replacement; verification records before and after
 3421 replacement; record of rejection of unauthorised attempts; documentation of replacement authority.

3422 **6.8 Data minimisation**

3423 The present clause specifies assessment criteria for the requirements in Clause 5.8.

3424 **Table 6.8.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-DM-001	Minimise information disclosure to network	Traffic capture; field-by-field justification; failure-mode capture	MEDIUM, HIGH (if implemented)
REQ-BM-DM-002	No sensitive disclosure; no credential retention	Category search in traffic; handoff-state credential inspection	MEDIUM, HIGH (if implemented)

3425

3426 6.8.1 [ACC-BM-DM-001]

3427 **Assessment objective:** verify that information disclosed by the boot manager to network infrastructure
3428 during boot is limited to what is necessary for boot operations. References REQ-BM-DM-001.

3429 **Assessment preparation:**

- 3430 • Test environment: the boot manager configured for network boot in a controlled network
3431 environment permitting passive traffic capture.
- 3432 • Required information: the manufacturer's documentation of network boot protocols used, the list
3433 of fields populated by the boot manager in each protocol exchange, and the justification for each
3434 field.
- 3435 • Required tools: network traffic analyser; protocol-aware decoder for the network boot protocols in
3436 use (e.g. DHCP, PXE, TFTP, HTTP(S)).

3437 **Assessment activities:**

- 3438 • Capture the complete network traffic generated by the boot manager during a successful network
3439 boot, from first transmission to successful handoff.
- 3440 • Parse each outbound message and enumerate the information fields populated by the boot
3441 manager.
- 3442 • Verify that each populated field is justified by a documented boot operation and that no sensitive
3443 or unnecessary information is included (cross-check with ACC-BM-DM-002 for the sensitive-
3444 information categories).
- 3445 • Repeat the capture under documented failure modes (e.g. no response from server, invalid
3446 response) and confirm that the information disclosed does not expand under those conditions.

3447 **Assessment verdict:**

- 3448 • Pass: every field transmitted by the boot manager is justified by a documented boot operation; no
3449 sensitive or unnecessary information is disclosed under nominal or failure conditions.
- 3450 • Fail: the boot manager transmits information not justified by a documented boot operation, or
3451 discloses sensitive information, in nominal or failure conditions.

3452 **Assessment evidence:** captured network traces; field-by-field justification table; records of failure-mode
3453 capture.

3454 6.8.2 [ACC-BM-DM-002]

3455 **Assessment objective:** verify that the boot manager does not disclose sensitive device information on the
3456 network and that temporary network credentials are not retained beyond their required use. References
3457 REQ-BM-DM-002.

3458 **Assessment preparation:**

- 3459 • Test environment: the boot manager in a controlled network environment supporting traffic
3460 capture and post-boot inspection of the handoff state.
- 3461 • Required information: the manufacturer's enumeration of sensitive device information (at
3462 minimum: firmware version, hardware serial numbers, internal configuration, diagnostic data); the

3463 manufacturer's statement of which temporary credentials (if any) are used during network boot
3464 and their intended lifetime.

3465 • Required tools: network traffic analyser; inspection tool for the boot manager's handoff state
3466 (memory, registers, passed structures as applicable to the platform).

3467 **Assessment activities:**

3468 • Capture the network traffic generated by the boot manager during a complete network boot.

3469 • Search the captured traffic for instances of the documented sensitive-information categories;
3470 confirm none are disclosed.

3471 • Where temporary network credentials (e.g. session tokens, ephemeral keys, bearer tokens) are
3472 used, inspect the handoff state after boot completion and confirm that those credentials are
3473 neither retained in memory nor passed to the operating system.

3474 • Where network boot is executed under failure modes, confirm that sensitive information is not
3475 disclosed as part of error or diagnostic responses.

3476 **Assessment verdict:**

3477 • Pass: none of the documented sensitive-information categories appear in network traffic under
3478 nominal or failure conditions; temporary network credentials are not retained beyond their
3479 required use.

3480 • Fail: any sensitive-information category is disclosed, or temporary network credentials persist in the
3481 handoff state or beyond their intended lifetime.

3482 **Assessment evidence:** list of sensitive-information categories; network trace records with search results;
3483 records of state inspection at handoff for temporary credential retention.

3484 6.9 Availability protection

3485 The present clause specifies assessment criteria for the requirements in Clause 5.9.

3486

Table 6.9.1-1: Requirements addressed in the present clause

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-AP-001	Fallback to known-good configuration	Corruption injection and fallback activation test	All (if implemented)
REQ-BM-AP-002	Auto-recover from update interruptions	Injection for each interruption class; consistent-state check	HIGH only
REQ-BM-AP-003	Timeouts on blocking operations	Delayed-input response measurement per operation	All
REQ-BM-AP-004	Retry and resource limits (sig-verify, network)	Sustained stimulus with retry count and resource measurement	All
REQ-BM-AP-005	No unauthorised bypass of verification	Per-step bypass attempt; bypass-path authorisation test	All
REQ-BM-AP-006	Error detection/correction on critical data	Per-item error injection; correction verification	All
REQ-BM-AP-007	Availability of essential boot code	Withstand-scenario corruption; availability-mechanism protection check	MEDIUM, HIGH (not in immutable memory)
REQ-BM-AP-008	Auth or physical presence in recovery mode	Per-action authorisation test across recovery triggers	All (if implemented)
REQ-BM-AP-009	Handle network boot failures (availability)	Injection for each failure class; documented handling verification	MEDIUM, HIGH (if implemented)
REQ-BM-AP-010	Local boot under network unavailable / DoS	Network-DoS test; local boot independence verification	MEDIUM, HIGH (if implemented)
REQ-BM-AP-011	Logs for security-relevant failures	Per-condition injection and log-capture match	MEDIUM, HIGH (LOW if implemented)
REQ-BM-AP-012	Protect security-relevant logs from tampering	Attempt for each tampering class; detection	MEDIUM, HIGH (LOW if implemented)
REQ-BM-AP-013	Time representations valid beyond 2038	Width inspection; post-2038 input processing; internal-time test	All
REQ-BM-AP-014	Protect recovery mechanism from modification	Per-interface modification and disablement attempts	All (if implemented)
REQ-BM-AP-015	Prevent partial execution on verification fail	Verification-failure injection at multiple stages; execution observation	MEDIUM, HIGH (LOW if implemented)
REQ-BM-AP-016	No security-sensitive ops without crypto services	Per-operation unavailability injection; fallback inspection	All
REQ-BM-AP-017	No security-control bypass on violations/failures	Per-violation injection; retained-control state inspection	All (if implemented)
REQ-BM-AP-018	Preserve recovery capability across updates	Application for each update class; recovery-mechanism exercise	MEDIUM, HIGH (LOW if implemented)

3487

3488 **6.9.1 [ACC-BM-AP-001]**

3489 **Assessment objective:** verify that the boot manager supports fallback to a previous known-good
3490 configuration. References REQ-BM-AP-001.

3491 **Assessment preparation:**

3492 • Test environment: the boot manager with a current configuration and a documented known-good
3493 configuration stored.

3494 • Required information: the manufacturer's specification of the known-good configuration
3495 mechanism, the fallback trigger conditions, and the fallback procedure.

3496 • Required tools: configuration-corruption or configuration-replacement tool; fallback-trigger tool.

3497 **Assessment activities:**

3498 • Corrupt or replace the current configuration in a manner documented as triggering fallback.

3499 • Invoke the fallback trigger and confirm the boot manager boots using the known-good
3500 configuration.

3501 • Confirm the known-good configuration is not corrupted by the trigger or fallback itself.

3502 **Assessment verdict:**

3503 • Pass: fallback returns the boot manager to the known-good configuration; the known-good
3504 configuration remains intact after fallback.

3505 • Fail: fallback does not occur on documented trigger, or the known-good configuration is lost or
3506 corrupted.

3507 **Assessment evidence:** record of the known-good configuration; test records of corruption and fallback.

3508 **6.9.2 [ACC-BM-AP-002]**

3509 **Assessment objective:** verify that, at HIGH security profile, the boot manager automatically recovers from
3510 interruptions during the update process. References REQ-BM-AP-002.

3511 **Assessment preparation:**

3512 • Test environment: the boot manager at HIGH security profile in update-orchestration mode.

3513 • Required information: the manufacturer's enumeration of interruption classes during update
3514 (power loss, transport interruption, intermediate-stage failure); the documented recovery
3515 mechanism for each.

3516 • Required tools: interruption-injection tools for each documented class (e.g. power cut at arbitrary
3517 update stages).

3518 **Assessment activities:**

3519 • For each interruption class, induce the interruption at representative points during an update and
3520 confirm the boot manager automatically recovers to either the pre-update or post-update state
3521 without manual intervention.

3522 • Confirm the resulting state is consistent (no split state with part of the update applied).

3523 • Confirm recovery does not require operations outside the boot manager's control.

3524 **Assessment verdict:**

3525 • Pass: every documented interruption class results in automatic recovery to a consistent state
3526 without manual intervention.

- 3527 • Fail: an interruption results in an inconsistent state, or recovery requires manual intervention or
3528 operations outside the boot manager.

3529 **Assessment evidence:** enumeration of interruption classes; injection records per class; records of state
3530 inspection after interruption.

3531 6.9.3 [ACC-BM-AP-003]

3532 **Assessment objective:** verify that the boot manager enforces timeouts to prevent indefinite blocking during
3533 boot-related operations. References REQ-BM-AP-003.

3534 **Assessment preparation:**

- 3535 • Test environment: the boot manager with control over the response latency of its inputs (network
3536 peers, peripherals, storage).
- 3537 • Required information: the manufacturer's enumeration of boot-related operations subject to
3538 timeout; each timeout value.
- 3539 • Required tools: response-latency control tool (e.g. slow-responding emulator); time measurement.

3540 **Assessment activities:**

- 3541 • For each enumerated operation, delay the input source beyond the documented timeout and
3542 measure the boot manager's response.
- 3543 • Confirm the boot manager terminates the operation at or near the documented timeout and does
3544 not block indefinitely.
- 3545 • Confirm the boot manager's subsequent behaviour is as documented (e.g. fallback, halt, error
3546 state).

3547 **Assessment verdict:**

- 3548 • Pass: every enumerated operation is terminated at its documented timeout; the boot manager
3549 does not block indefinitely on any documented input.
- 3550 • Fail: an operation blocks beyond its documented timeout, or the post-timeout behaviour is
3551 inconsistent with documentation.

3552 **Assessment evidence:** enumeration of operation timeouts; records of delay and response measurements
3553 per operation.

3554 6.9.4 [ACC-BM-AP-004]

3555 **Assessment objective:** verify that the boot manager limits retry attempts and resource consumption for
3556 signature verification and network operations. References REQ-BM-AP-004.

3557 **Assessment preparation:**

- 3558 • Test environment: the boot manager in states exercising signature verification and network
3559 operations.
- 3560 • Required information: the manufacturer's specification of retry-attempt limits and resource-
3561 consumption bounds for each operation class.

- 3562 • Required tools: attempt-stimulation tool (e.g. repeated invalid-signature submissions, repeated
3563 network attempts); resource-monitoring tool.

3564 **Assessment activities:**

- 3565 • Repeatedly submit inputs to signature verification and network-operation paths beyond the
3566 documented retry limit; confirm the boot manager halts retries at the documented limit.
- 3567 • Measure resource consumption during sustained retry stimulus; confirm it remains within the
3568 documented bounds.
- 3569 • Confirm that exceeding the retry limit produces the documented terminal behaviour rather than
3570 unbounded retrying.

3571 **Assessment verdict:**

- 3572 • Pass: retry limits and resource consumption bounds are observed under sustained stimulus;
3573 terminal behaviour matches documentation.
- 3574 • Fail: retries continue beyond the documented limit, or resource consumption exceeds the
3575 documented bounds.

3576 **Assessment evidence:** specification of retry and resource policy; test records with measurements under
3577 sustained stimulus.

3578 **6.9.5 [ACC-BM-AP-005]**

3579 **Assessment objective:** verify that the boot manager prevents unauthorised bypass of security verification
3580 steps during normal operation. References REQ-BM-AP-005.

3581 **Assessment preparation:**

- 3582 • Test environment: the boot manager in normal operational mode; documented debug or in-field
3583 modes disabled or appropriately gated.
- 3584 • Required information: the manufacturer's enumeration of security verification steps; the
3585 documented authorisation mechanism for any permitted bypass (e.g. in-field debug mode).
- 3586 • Required tools: tools to attempt bypass via input manipulation, state manipulation, or invocation of
3587 documented bypass paths without authorisation.

3588 **Assessment activities:**

- 3589 • For each security verification step, attempt to bypass it through input manipulation and via any
3590 documented bypass path without providing the documented authorisation; confirm the step is
3591 executed and the bypass is rejected.
- 3592 • Where a documented bypass path exists (e.g. in-field debug mode), attempt invocation without
3593 authentication and confirm rejection.
- 3594 • Confirm that in normal operational mode no mechanism permits unauthorised bypass.

3595 **Assessment verdict:**

- 3596 • Pass: every security verification step executes in normal operation; documented bypass paths
3597 require authentication; unauthorised bypass attempts are rejected.

- 3598 • Fail: a security verification step can be bypassed without authentication, or normal operational
3599 mode permits unauthorised bypass.

3600 **Assessment evidence:** enumeration of verification steps; records of bypass attempts per step; test records
3601 of bypass-path authorisation.

3602 6.9.6 [ACC-BM-AP-006]

3603 **Assessment objective:** verify that, at MEDIUM or HIGH security profile, the boot manager detects and
3604 corrects errors in critical data using mechanisms supported by the underlying platform. References REQ-
3605 BM-AP-006.

3606 **Assessment preparation:**

- 3607 • Test environment: the boot manager at MEDIUM or HIGH security profile with access to error-
3608 inducing channels on the critical-data storage.
- 3609 • Required information: the manufacturer's enumeration of critical data items; the error-detection
3610 and correction mechanism used for each (e.g. ECC, FEC, cross-copy comparison); the platform
3611 features relied upon.
- 3612 • Required tools: bit-flip injection tool or equivalent for the platform; correction-observation tool.

3613 **Assessment activities:**

- 3614 • For each critical-data item, inject a documented-recoverable error into the stored value.
- 3615 • Confirm the boot manager detects the error during its next access.
- 3616 • Confirm the error is corrected and the data is readable at its expected value.
- 3617 • Inject a documented non-recoverable error and confirm detection plus documented failure
3618 handling.

3619 **Assessment verdict:**

- 3620 • Pass: recoverable errors are detected and corrected; non-recoverable errors are detected and
3621 produce documented failure handling.
- 3622 • Fail: an injected recoverable error is not detected or not corrected; a non-recoverable error is not
3623 detected.

3624 **Assessment evidence:** enumeration of critical data; records of error injection and correction per item.

3625 6.9.7 [ACC-BM-AP-007]

3626 **Assessment objective:** verify that, at MEDIUM or HIGH security profile where the boot manager is not
3627 stored in immutable memory, essential boot code remains available in the presence of storage corruption
3628 or failure. References REQ-BM-AP-007.

3629 **Assessment preparation:**

- 3630 • Test environment: the boot manager at MEDIUM or HIGH security profile on a platform whose
3631 boot-code storage is mutable and accessible for corruption injection.

3632 • Required information: the manufacturer's identification of essential boot code; the availability
 3633 mechanism (e.g. redundant storage, fallback partition); the corruption scenarios the mechanism is
 3634 expected to withstand.

3635 • Required tools: storage-corruption injection tool; boot-execution observation tool.

3636 **Assessment activities:**

3637 • Corrupt the primary copy of the essential boot code within the documented withstand-scenario;
 3638 confirm the boot manager continues boot using the redundant copy or fallback partition.

3639 • Corrupt at the boundary of the documented withstand-scenario (e.g. both copies) and confirm the
 3640 boot manager produces the documented failure indication rather than silently continuing.

3641 • Confirm the availability mechanism itself is protected from unauthorised corruption.

3642 **Assessment verdict:**

3643 • Pass: essential boot code remains available under documented withstand-scenarios; beyond the
 3644 withstand-scenario the boot manager produces documented failure handling; the availability
 3645 mechanism is protected.

3646 • Fail: essential boot code becomes unavailable within the documented withstand-scenario, or the
 3647 availability mechanism is bypassable.

3648 **Assessment evidence:** identification of essential boot code; test records of corruption injection; record of
 3649 inspection of the availability mechanism.

3650 6.9.8 [ACC-BM-AP-008]

3651 **Assessment objective:** verify that sensitive actions performed in recovery mode require authentication or
 3652 physical presence. References REQ-BM-AP-008.

3653 **Assessment preparation:**

3654 • Test environment: the boot manager with recovery capability and recovery mode entry
 3655 controllable.

3656 • Required information: the manufacturer's enumeration of recovery-mode actions classified as
 3657 sensitive; the authentication or physical-presence requirement for each.

3658 • Required tools: recovery-mode entry tool; authentication and physical-presence stimulation tools.

3659 **Assessment activities:**

3660 • Enter recovery mode by each documented trigger (including automatic triggers on boot failure).

3661 • Attempt each sensitive action without authentication or physical presence; confirm rejection.

3662 • Attempt each sensitive action with the documented authentication or physical-presence assertion;
 3663 confirm acceptance.

3664 **Assessment verdict:**

3665 • Pass: every sensitive recovery-mode action requires the documented authentication or physical
 3666 presence; attempts without either are rejected.

- 3667 • Fail: a sensitive recovery-mode action can be performed without the documented authentication or
3668 physical presence.

3669 **Assessment evidence:** enumeration of sensitive actions; authorisation test records per action.

3670 6.9.9 [ACC-BM-AP-009]

3671 **Assessment objective:** verify that the boot manager, where network boot is supported, handles network
3672 boot failures in a manner that maintains boot availability. References REQ-BM-AP-009.

3673 **Assessment preparation:**

- 3674 • Test environment: the boot manager configured for network boot in a controlled environment
3675 permitting network-level failure injection.

- 3676 • Required information: the manufacturer's enumeration of network-boot failure classes; the
3677 documented handling mechanism for each (e.g. timeout, fallback to local boot, backoff retries,
3678 alternative servers).

- 3679 • Required tools: network failure-injection tool (server unavailable, partial response, malformed
3680 response); boot-outcome observation.

3681 **Assessment activities:**

- 3682 • For each documented failure class, induce the failure and observe the boot manager's handling.

- 3683 • Confirm the documented handling mechanism is applied (timeout fires, fallback is entered, backoff
3684 retries proceed, or alternative servers are attempted as documented).

- 3685 • Confirm the boot manager reaches a documented terminal state (successful boot via alternative
3686 path, documented failure indication) rather than hanging indefinitely.

3687 **Assessment verdict:**

- 3688 • Pass: each documented failure class invokes the documented handling mechanism and reaches a
3689 documented terminal state.

- 3690 • Fail: a failure class produces hanging or undocumented behaviour, or the documented handling
3691 mechanism is not invoked.

3692 **Assessment evidence:** enumeration of failure classes; injection and outcome records per class.

3693 6.9.10 [ACC-BM-AP-010]

3694 **Assessment objective:** verify that the boot manager, where network boot is supported, maintains local
3695 boot capability when network services are unavailable or under denial-of-service conditions. References
3696 REQ-BM-AP-010.

3697 **Assessment preparation:**

- 3698 • Test environment: the boot manager configured with network boot and a local boot capability, in a
3699 controlled environment permitting network denial-of-service conditions.

- 3700 • Required information: the manufacturer's specification of the local boot capability, the conditions
3701 under which local boot is entered, and the transition mechanism.

- 3702 • Required tools: network denial-of-service tool (e.g. sustained flooding, total unreachability); boot-
3703 outcome observation.

3704 **Assessment activities:**

- 3705 • Execute the boot manager with the network entirely unavailable; confirm local boot is entered
3706 within the documented transition time and proceeds successfully.

- 3707 • Execute the boot manager under sustained denial-of-service against the network-boot target;
3708 confirm local boot is entered rather than the boot manager being held up indefinitely.

- 3709 • Confirm the local boot code path is not dependent on network-sourced information.

3710 **Assessment verdict:**

- 3711 • Pass: local boot succeeds under total network unavailability and under denial-of-service conditions;
3712 the local path is independent of the network source.

- 3713 • Fail: the boot manager cannot complete boot under the tested conditions, or the local path
3714 depends on the network source.

3715 **Assessment evidence:** enumeration of failure classes; injection and outcome records per class.

3716 **6.9.11 [ACC-BM-AP-011]**

3717 **Assessment objective:** verify that the boot manager generates logs for security-relevant failure conditions.
3718 References REQ-BM-AP-011.

3719 **Assessment preparation:**

- 3720 • Test environment: the boot manager with verified or measured boot and logging capability active.

- 3721 • Required information: the manufacturer's enumeration of security-relevant failure conditions (e.g.
3722 verification failures, integrity violations, authentication failures, unexpected modification of boot
3723 components); the documented log-entry content for each.

- 3724 • Required tools: fault-injection tool for each condition; log capture tool.

3725 **Assessment activities:**

- 3726 • Induce each enumerated security-relevant failure condition.

- 3727 • Capture the log output and confirm that a log entry consistent with the documented content is
3728 generated for each.

- 3729 • Confirm the entry identifies the condition type and the affected component or step.

3730 **Assessment verdict:**

- 3731 • Pass: every enumerated condition produces the documented log entry.

- 3732 • Fail: a condition fails to produce a log entry, or the entry does not identify the condition type or
3733 affected component.

3734 **Assessment evidence:** enumeration of failure conditions; records of injection and log capture per condition.

3735 6.9.12 [ACC-BM-AP-012]

3736 **Assessment objective:** verify that security-relevant logs are protected against tampering. References REQ-
3737 BM-AP-012.

3738 **Assessment preparation:**

- 3739 • Test environment: the boot manager with security-relevant log entries produced and the log
3740 storage inspectable.
- 3741 • Required information: the manufacturer's specification of the log-protection mechanism (e.g.
3742 append-only storage, cryptographic chaining, hardware-protected storage, signed log sequences).
- 3743 • Required tools: direct log-storage access tool; tampering-attempt tool (entry modification, entry
3744 deletion, reordering, insertion).

3745 **Assessment activities:**

- 3746 • Attempt each tampering class (modify, delete, reorder, insert) directly on the log storage.
- 3747 • Confirm the boot manager detects the tampering on subsequent access and reports the violation
3748 through the documented indication.
- 3749 • Where cryptographic chaining or signing is the mechanism, confirm the detection is cryptographic
3750 rather than relying on access control alone.

3751 **Assessment verdict:**

- 3752 • Pass: every tampering class is detected; the protection mechanism functions as documented.
- 3753 • Fail: a tampering class is not detected, or the documented mechanism is absent.

3754 **Assessment evidence:** specification of log protection; records of tampering attempts per class; detection
3755 records.

3756 6.9.13 [ACC-BM-AP-013]

3757 **Assessment objective:** verify that the boot manager uses time representations valid beyond the year 2038
3758 for all security-relevant uses of time. References REQ-BM-AP-013.

3759 **Assessment preparation:**

- 3760 • Test environment: the boot manager with the ability to set internal time or to process inputs
3761 containing time values beyond 2038-01-19 03:14:07 UTC.
- 3762 • Required information: the manufacturer's enumeration of security-relevant uses of time (certificate
3763 validity, internal system time for boot decisions, timestamps in logs); the representation used for
3764 each.
- 3765 • Required tools: time-manipulation tool to set inputs beyond the 2038 boundary; inspection tool for
3766 each time representation.

3767 **Assessment activities:**

- 3768 • For each enumerated use, inspect the representation width and confirm it accommodates values
3769 beyond the 2038 boundary (e.g. 64-bit timestamp).

3770 • Process inputs containing a time value beyond the 2038 boundary (e.g. a certificate with a post-
3771 2038 expiry) and confirm correct handling.

3772 • Set the boot manager's internal time beyond the 2038 boundary where supported and confirm
3773 subsequent operations behave correctly.

3774 **Assessment verdict:**

3775 • Pass: every enumerated representation accommodates post-2038 values; inputs and internal time
3776 beyond the boundary are handled correctly.

3777 • Fail: a representation wraps or produces incorrect behaviour at or after the 2038 boundary.

3778 **Assessment evidence:** enumeration of time uses with inspection of representation; records of input
3779 processing after 2038.

3780 6.9.14 [ACC-BM-AP-014]

3781 **Assessment objective:** verify that recovery mechanisms are protected from unauthorised modification or
3782 disablement. References REQ-BM-AP-014.

3783 **Assessment preparation:**

3784 • Test environment: the boot manager with recovery capability and the recovery-mechanism storage
3785 and configuration accessible.

3786 • Required information: the manufacturer's specification of the recovery mechanism and the
3787 protection applied to its code, data, and enablement state.

3788 • Required tools: modification and disablement attempt tools for each interface exposing the
3789 recovery mechanism.

3790 **Assessment activities:**

3791 • Attempt modification of the recovery mechanism's code and data through each documented
3792 interface without the authorised path; confirm rejection.

3793 • Attempt disablement of the recovery mechanism through each interface without authorised path;
3794 confirm rejection.

3795 • Attempt modification via undocumented or residual interfaces; confirm rejection.

3796 **Assessment verdict:**

3797 • Pass: the recovery mechanism's code, data, and enablement state are protected from modification
3798 or disablement through any interface without the authorised path.

3799 • Fail: the recovery mechanism can be modified or disabled through any interface without the
3800 authorised path.

3801 **Assessment evidence:** specification of the recovery mechanism; records of modification and disablement
3802 attempts per interface.

3803 6.9.15 [ACC-BM-AP-015]

3804 **Assessment objective:** verify that partial execution of boot components is prevented when verification has
3805 not completed successfully. References REQ-BM-AP-015.

3806 **Assessment preparation:**

- 3807 • Test environment: the boot manager with verified boot and the ability to interrupt or fail
- 3808 verification mid-process.
- 3809 • Required information: the manufacturer's specification of the execution-gate mechanism between
- 3810 verification and execution.
- 3811 • Required tools: verification-fault injection tool; execution observation tool.

3812 **Assessment activities:**

- 3813 • Induce verification failure at various stages (before completion, at a boundary, immediately prior to
- 3814 execution) and confirm no portion of the component is executed.
- 3815 • Induce a verification process that does not complete (e.g. crashed verifier) and confirm the boot
- 3816 manager does not proceed to execution.
- 3817 • Inspect the transition between verification and execution for any race or shortcut path.

3818 **Assessment verdict:**

- 3819 • Pass: no partial execution occurs when verification fails or does not complete; no race or shortcut
- 3820 path bypasses the gate.
- 3821 • Fail: partial execution occurs when verification is incomplete or failed; a race or shortcut path exists.

3822 **Assessment evidence:** records of verification-failure injection; records of execution observation; record of
 3823 inspection of the transition path.

3824 **6.9.16 [ACC-BM-AP-016]**

3825 **Assessment objective:** verify that the boot manager does not perform security-sensitive operations when
 3826 required cryptographic components or services are unavailable. References REQ-BM-AP-016.

3827 **Assessment preparation:**

- 3828 • Test environment: the boot manager with the ability to induce unavailability of required
- 3829 cryptographic components or services (e.g. TPM unavailable, cryptographic provider faulted).
- 3830 • Required information: the manufacturer's enumeration of security-sensitive operations and the
- 3831 cryptographic components or services each depends upon.
- 3832 • Required tools: cryptographic-component unavailability injection tool.

3833 **Assessment activities:**

- 3834 • For each security-sensitive operation, make the required cryptographic component or service
- 3835 unavailable and attempt the operation.
- 3836 • Confirm the operation is not performed under the unavailability condition and the boot manager
- 3837 produces documented handling.
- 3838 • Confirm that no fallback uses a weaker substitute that would circumvent the security intent.

3839 **Assessment verdict:**

3840 • Pass: no security-sensitive operation is performed when its required cryptographic component or
3841 service is unavailable; no weaker substitute is used.

3842 • Fail: a security-sensitive operation is performed under unavailability conditions, or a weaker
3843 substitute is used.

3844 **Assessment evidence:** mapping of operations to components; unavailability test records per operation;
3845 record of fallback inspection.

3846 6.9.17 [ACC-BM-AP-017]

3847 **Assessment objective:** verify that, where recovery capability exists, the boot manager does not bypass
3848 security controls when handling security violations or verification failures. References REQ-BM-AP-017.

3849 **Assessment preparation:**

3850 • Test environment: the boot manager with recovery capability and the ability to induce security
3851 violations and verification failures.

3852 • Required information: the manufacturer's enumeration of violation and failure classes; the handling
3853 path for each, including the security controls retained.

3854 • Required tools: violation and failure injection tools; inspection tool for the security-control state
3855 during and after handling.

3856 **Assessment activities:**

3857 • Induce each enumerated violation or failure class and observe the handling path.

3858 • Confirm that the security controls documented as retained are active during and after handling (e.g.
3859 verification is still required before execution; authentication is still required for sensitive actions).

3860 • Confirm that the handling path does not reach a state where security controls are weakened or
3861 bypassed.

3862 **Assessment verdict:**

3863 • Pass: every handling path retains the documented security controls; no handling path weakens or
3864 bypasses security controls.

3865 • Fail: a handling path weakens or bypasses security controls documented as retained.

3866 **Assessment evidence:** enumeration of violation classes with mapping of retained control; injection records
3867 per class with records of control state.

3868 6.9.18 [ACC-BM-AP-018]

3869 **Assessment objective:** verify that recovery capability is preserved across firmware and configuration
3870 updates. References REQ-BM-AP-018.

3871 **Assessment preparation:**

3872 • Test environment: the boot manager with recovery and update capabilities in an environment
3873 supporting applying representative updates.

3874 • Required information: the manufacturer's specification of the recovery-preservation mechanism
3875 across updates; the update scenarios required to preserve recovery capability.

- 3876 • Required tools: update-application tool; recovery-mechanism exercise tool.

3877 **Assessment activities:**

- 3878 • Apply firmware and configuration updates through the documented update paths.
- 3879 • After each update, exercise the recovery mechanism and confirm it is functional.
- 3880 • Include in the test sample a firmware update that alters boot-manager components and a
3881 configuration update that alters recovery-related configuration.

3882 **Assessment verdict:**

- 3883 • Pass: recovery capability functions after every documented update class.
- 3884 • Fail: recovery capability is broken, disabled, or degraded by any documented update class.

3885 **Assessment evidence:** enumeration of update classes; records of recovery exercises per update.

3886 6.10 Impact minimisation

3887 The present clause specifies assessment criteria for the requirements in Clause 5.10.

3888 **Table 6.10.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-IM-001	No harmful/disruptive network behaviour	Traffic capture under nominal and failure modes; loop-topology exercise	MEDIUM, HIGH (if implemented)

3889

3890 6.10.1 [ACC-BM-IM-001]

3891 **Assessment objective:** verify that the boot manager does not generate harmful or disruptive network
3892 behaviour. References REQ-BM-IM-001.

3893 **Assessment preparation:**

- 3894 • Test environment: the boot manager configured for network boot in a controlled network
3895 environment with other hosts and network segments present, permitting passive observation of all
3896 traffic originating from the boot manager.
- 3897 • Required information: the manufacturer's enumeration of network behaviours produced by the
3898 boot manager (broadcast scope, multicast participation, repeated retries, peer-discovery patterns)
3899 and the documented bounds on each.
- 3900 • Required tools: network traffic analyser capable of identifying broadcast storms, loop formation,
3901 and abnormal traffic rates; topology probe able to detect loop conditions.

3902 **Assessment activities:**

- 3903 • Capture the complete network traffic generated by the boot manager across a successful network
3904 boot and across each documented failure-mode scenario.
- 3905 • Analyse the captured traffic for behaviour patterns that would be harmful or disruptive to other
3906 hosts or network segments, at minimum: broadcast rate relative to documented bounds; presence
3907 of loop-inducing frame forwarding; sustained retry patterns beyond documented bounds; traffic
3908 targeted at unintended destinations.

- 3909 • Exercise the boot manager in a network topology where loops could form if frame-forwarding
3910 behaviour were incorrect; confirm no loop is induced.

3911 **Assessment verdict:**

- 3912 • Pass: observed network behaviour stays within the documented bounds; no broadcast storm, loop,
3913 or sustained disruptive pattern is produced under nominal or failure-mode operation.

- 3914 • Fail: the boot manager produces a broadcast storm, induces a network loop, or generates sustained
3915 traffic beyond documented bounds.

3916 **Assessment evidence:** documented bounds on network behaviour; traffic captures under nominal
3917 conditions and under failure modes; record of loop-topology exercises.

3918 6.11 Minimisation of attack surfaces

3919 The present clause specifies assessment criteria for the requirements in Clause 5.11.

3920 **Table 6.11.1-1: Requirements addressed in the present clause**

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-MAS-001	Only necessary resources enabled at handoff	Resource-state capture at handoff moment	All
REQ-BM-MAS-002	Non-essential code excluded from production build	Build configuration inspection; binary indicator search	All
REQ-BM-MAS-003	Non-operational interfaces disabled or protected	Per-interface operational/non-operational classification; protection test	All
REQ-BM-MAS-004	Unused interfaces and protocols removed	Runtime and binary enumeration; operational-purpose mapping	All
REQ-BM-MAS-005	Validate all inputs against expected format	Per-channel well-formed, malformed, fuzz testing	All
REQ-BM-MAS-006	Detect errors; reject non-compliant inputs (network)	Per-operation fault injection; non-compliant input test	MEDIUM, HIGH (if implemented)
REQ-BM-MAS-007	Mechanism to disable non-required config options	Per-option disable/re-enable test via deployment-integrator interface	All (if implemented)

3921

3922 6.11.1 [ACC-BM-MAS-001]

3923 **Assessment objective:** verify that the boot manager, before handing off control to the boot target, ensures
3924 that only the resources necessary for the boot target remain enabled. References REQ-BM-MAS-001.

3925 **Assessment preparation:**

- 3926 • Test environment: the boot manager executed to the handoff point with inspection capability over
3927 the platform resource state at that point (peripherals, memory regions, DMA controllers, interrupt
3928 sources, CPU modes).

- 3929 • Required information: the manufacturer's enumeration of resources enabled during boot; for each,
3930 whether it is required by the boot target at handoff and the rationale.

- 3931 • Required tools: platform-state inspection tool capable of enumerating enabled resources at the
3932 handoff moment.

3933 **Assessment activities:**

- 3934 • Capture the resource state at the handoff point (e.g. by halting immediately prior to handoff, or by
3935 instrumenting the handoff transition).
- 3936 • For each resource enumerated as enabled, confirm it is documented as required by the boot target;
3937 investigate any undocumented enabled resource.
- 3938 • For each resource documented as not required at handoff but enabled during earlier boot stages,
3939 confirm it has been disabled, quiesced, or revoked before handoff.

3940 **Assessment verdict:**

- 3941 • Pass: every resource enabled at handoff is documented as required by the boot target; no resource
3942 enabled earlier in the boot remains enabled at handoff unless documented as required.
- 3943 • Fail: an undocumented resource is enabled at handoff, or a resource documented as not required
3944 remains enabled.

3945 **Assessment evidence:** record of resource enumeration with required/not-required classification; record of
3946 state capture at handoff.

3947 6.11.2 [ACC-BM-MAS-002]

3948 **Assessment objective:** verify that non-essential code is excluded from production builds of the boot
3949 manager. References REQ-BM-MAS-002.

3950 **Assessment preparation:**

- 3951 • Test environment: the production build of the boot manager as placed on the market.
- 3952 • Required information: the manufacturer's build configuration and documented differences
3953 between development, test, and production builds; the list of code paths, modules, or features
3954 categorised as non-essential for production (e.g. debug output, test harnesses, development
3955 shortcuts, example configurations).
- 3956 • Required tools: binary analysis tools capable of identifying string constants, code paths, and linked
3957 modules; where source access is available, build-manifest inspection.

3958 **Assessment activities:**

- 3959 • Inspect the build configuration and confirm that non-essential modules and flags are not enabled
3960 for the production build.
- 3961 • Inspect the binary for indicators of non-essential code (e.g. debug strings, test symbols, unreachable
3962 branches linked only by non-production flags) and investigate any finding.
- 3963 • Where source access is available, confirm that guards around non-essential code exclude it under
3964 the production build configuration.

3965 **Assessment verdict:**

- 3966 • Pass: the production build excludes code categorised as non-essential by the manufacturer; no
3967 indicator of non-essential code is found in the production binary beyond those documented.

- 3968 • Fail: the production build contains modules or code paths documented as non-essential, or binary
3969 analysis surfaces non-essential code not present in the documentation.

3970 **Assessment evidence:** record of build configuration; findings of binary analysis; review of source guards
3971 where applicable.

3972 6.11.3 [ACC-BM-MAS-003]

3973 **Assessment objective:** verify that interfaces and functions not intended for operational use are disabled or
3974 protected. References REQ-BM-MAS-003.

3975 **Assessment preparation:**

- 3976 • Test environment: the boot manager in its operational configuration on a representative platform.
- 3977 • Required information: the manufacturer's enumeration of interfaces and functions exposed by the
3978 boot manager, each classified as operational or non-operational; for non-operational items, the
3979 applied mechanism (disabled, physically protected, authentication-gated, fuse-locked).
- 3980 • Required tools: interface enumeration tools for the platform; interaction tools for each interface
3981 category (e.g. serial debug, JTAG, USB debug, memory-mapped registers).

3982 **Assessment activities:**

- 3983 • Enumerate the interfaces accessible on the operational product and compare against the
3984 documented list.
- 3985 • For each interface classified as non-operational, confirm the documented protection is in effect;
3986 test the protection by attempting to exercise the interface without the documented unlock path.
- 3987 • For each interface classified as operational, confirm it matches the manufacturer's intended
3988 purpose and exposes only documented functions.

3989 **Assessment verdict:**

- 3990 • Pass: every non-operational interface or function is protected by its documented mechanism and
3991 cannot be exercised without the documented unlock path; operational interfaces expose only
3992 documented functions.
- 3993 • Fail: a non-operational interface is reachable without the documented protection; an operational
3994 interface exposes undocumented functions.

3995 **Assessment evidence:** enumeration of interfaces with record of operational classification; record of
3996 protection tests per interface.

3997 6.11.4 [ACC-BM-MAS-004]

3998 **Assessment objective:** verify that unused or non-essential interfaces and protocols have been removed
3999 from the boot manager. References REQ-BM-MAS-004.

4000 **Assessment preparation:**

- 4001 • Test environment: the production build of the boot manager.
- 4002 • Required information: the manufacturer's enumeration of interfaces and protocols supported by
4003 the boot manager; for each, the operational purpose that justifies its inclusion.

- 4004 • Required tools: interface and protocol enumeration tools; binary analysis tools capable of
4005 identifying linked protocol stacks or interface handlers.

4006 **Assessment activities:**

- 4007 • Enumerate the interfaces and protocols exposed at runtime and, via binary analysis, those compiled
4008 into the boot manager.

- 4009 • For each interface or protocol, confirm an operational-purpose justification exists and the protocol
4010 is actively used in documented boot scenarios.

- 4011 • Identify any interface or protocol present in the binary or at runtime without a documented
4012 operational purpose and record it as an unused or non-essential item not removed.

4013 **Assessment verdict:**

- 4014 • Pass: every runtime- or binary-present interface and protocol maps to a documented operational
4015 purpose.

- 4016 • Fail: an interface or protocol is present without a documented operational purpose.

4017 **Assessment evidence:** runtime and binary interface/protocol enumeration; justification mapping record.

4018 6.11.5 [ACC-BM-MAS-005]

4019 **Assessment objective:** verify that the boot manager validates all inputs for conformance with expected
4020 formats. References REQ-BM-MAS-005.

4021 **Assessment preparation:**

- 4022 • Test environment: the boot manager accessible through each documented input channel (e.g.
4023 configuration files, signed artefacts, network inputs, user interfaces, inter-component messages).

- 4024 • Required information: the manufacturer's enumeration of input channels; for each, the expected
4025 input format and the validation mechanism applied.

- 4026 • Required tools: input fuzzing tools appropriate to each channel; controlled-input generation tools.

4027 **Assessment activities:**

- 4028 • For each input channel, submit well-formed inputs and confirm acceptance.

- 4029 • Submit inputs violating the expected format in documented ways (e.g. truncated, oversized,
4030 malformed encoding, out-of-range values); confirm rejection before any security-relevant
4031 processing.

- 4032 • Submit a sample of non-targeted fuzz inputs on each channel and confirm that the boot manager
4033 handles them without entering an undefined state.

4034 **Assessment verdict:**

- 4035 • Pass: well-formed inputs are accepted; documented malformed inputs are rejected before security-
4036 relevant processing; fuzz inputs do not cause undefined states.

- 4037 • Fail: a documented malformed input is processed without rejection, or a fuzz input drives the boot
4038 manager into an undefined state.

4039 **Assessment evidence:** enumeration of input channels; well-formed and malformed test records per
4040 channel; fuzz test summary.

4041 6.11.6 [ACC-BM-MAS-006]

4042 **Assessment objective:** verify that the boot manager, in network boot scenarios, detects errors in critical
4043 operations, rejects non-compliant inputs, and transitions to error handling on faults. References REQ-BM-
4044 MAS-006.

4045 **Assessment preparation:**

- 4046 • Test environment: the boot manager configured for network boot in a controlled environment
4047 where network inputs can be manipulated and faults can be induced.
- 4048 • Required information: the manufacturer's enumeration of critical network-boot operations; for
4049 each, the error conditions, the compliance criteria for inputs, and the documented error-handling
4050 transition.
- 4051 • Required tools: controllable network source; fault-injection tools; state inspection tool for the boot
4052 manager's error-handling path.

4053 **Assessment activities:**

- 4054 • For each enumerated critical operation, induce the documented error condition (e.g. truncated
4055 response, invalid signature in a signed resource, protocol-level fault); confirm detection.
- 4056 • Submit non-compliant inputs (e.g. violating protocol structure, outside value ranges); confirm
4057 rejection before the operation completes.
- 4058 • Confirm that on detection or rejection the boot manager transitions to the documented error-
4059 handling state rather than continuing into a potentially compromised path.

4060 **Assessment verdict:**

- 4061 • Pass: each enumerated error condition is detected; non-compliant inputs are rejected; the
4062 documented error-handling transition is taken on every fault.
- 4063 • Fail: an error condition is not detected; a non-compliant input is processed; the boot manager
4064 continues past a detected fault without the documented error-handling transition.

4065 **Assessment evidence:** enumeration of critical operations; fault-injection and non-compliant input test
4066 records, per operation; records of error-handling transitions.

4067 6.11.7 [ACC-BM-MAS-007]

4068 **Assessment objective:** verify that the boot manager provides a mechanism to disable configuration options
4069 that are not required by the deployment integrator. References REQ-BM-MAS-007.

4070 **Assessment preparation:**

- 4071 • Test environment: the boot manager accessible through its configuration interface.
- 4072 • Required information: the manufacturer's enumeration of configuration options; for each, the
4073 mechanism provided for the deployment integrator to disable or suppress the option; the
4074 documentation directed at deployment integrators explaining the disable mechanism.

- 4075 • Required tools: configuration tool; inspection tool for the boot manager's effective configuration
4076 state.

4077 **Assessment activities:**

- 4078 • For each configuration option that may be disabled per manufacturer documentation, invoke the
4079 disable mechanism and confirm that the option is no longer exposed or active in the boot
4080 manager's effective state.

- 4081 • Confirm that the disable mechanism is documented for the deployment integrator and does not
4082 require manufacturer-restricted access.

- 4083 • Confirm that re-enabling a previously disabled option, where documented, requires the same
4084 mechanism accessible to the deployment integrator.

4085 **Assessment verdict:**

- 4086 • Pass: each documented configuration option can be disabled through the deployment-integrator-
4087 accessible mechanism; disablement takes effect in the boot manager's state; re-enablement where
4088 documented uses the same mechanism.

- 4089 • Fail: a documented configuration option cannot be disabled by the deployment integrator;
4090 disablement does not take effect; re-enablement requires undocumented access.

4091 **Assessment evidence:** enumeration of configuration options with mapping of disable mechanisms; test
4092 records of disable and re-enable per option.

4093 6.12 Exploitation mitigation mechanisms

4094 The requirements addressed by the present clause are specified in other clauses of the present document
4095 and assessed under the corresponding clauses of clause 6:

- 4096 • privilege separation and trust-boundary enforcement: REQ-BM-INT-001 assessed under 6.7;
- 4097 • resource minimisation and DMA restriction at handoff: REQ-BM-MAS-001 assessed under 6.11;
- 4098 • access control on cryptographic key material: REQ-BM-CON-001 assessed under 6.6;
- 4099 • attack-surface reduction: REQ-BM-MAS-002, REQ-BM-MAS-003, REQ-BM-MAS-004, REQ-BM-MAS-
4100 007 assessed under 6.11;
- 4101 • input validation and error handling: REQ-BM-MAS-005, REQ-BM-MAS-006 assessed under 6.11.

4102 6.13 Logging and monitoring

4103 The present clause specifies assessment criteria for the requirements in Clause 5.13.

4104

Table 6.13.1-1: Requirements addressed in the present clause

Requirement	Subject	Assessment summary	Profile applicability
REQ-BM-LOG-001	Tamper-evident recording of boot measurements	Retention-mechanism read at handoff; tamper-alteration test	MEDIUM, HIGH (LOW if verified/measured boot implemented)
REQ-BM-LOG-002	Measurements support remote attestation + freshness	Attestation-format parse; challenge-response; replay rejection	MEDIUM, HIGH (if measured boot implemented)
REQ-BM-LOG-003	Log security-relevant failures and state changes	Fault injection for each event class; log-entry capture	All (if implemented)
REQ-BM-LOG-004	Version information accessible	Query for each consumer role; field-completeness check	All

4105

4106 **6.13.1 [ACC-BM-LOG-001]**

4107 **Assessment objective:** verify that the boot manager records measurements of boot components and
4108 security-critical configuration in a tamper-evident way before handoff. References REQ-BM-LOG-001.

4109 **Assessment preparation:**

- 4110 • Test environment: the boot manager executed to completion with access to the retention
4111 mechanism used for measurements (e.g. platform configuration registers, tamper-proof storage).
- 4112 • Required information: the manufacturer's documentation of the measurement retention
4113 mechanism, the set of components and configuration items measured, and the tamper-evidence
4114 properties claimed for the mechanism.
- 4115 • Required tools: inspection tool for the retention mechanism appropriate to the platform (e.g. TPM
4116 command interface for PCR read, secure-storage inspection utility).

4117 **Assessment activities:**

- 4118 • Execute a complete boot and, immediately before handoff, read the retention mechanism to obtain
4119 the measurement set actually recorded.
- 4120 • Compare the recorded measurements against the documented set of components and
4121 configuration items; confirm the set is complete.
- 4122 • Verify the tamper-evidence property of the retention mechanism by attempting to alter a recorded
4123 measurement via documented and undocumented interfaces; confirm that alteration is either
4124 prevented or produces a detectable indication.
- 4125 • Where retention is volatile (e.g. PCR), confirm that retention across the boot manager's execution
4126 to handoff is sufficient for the assessment objective.

4127 **Assessment verdict:**

- 4128 • Pass: measurements are recorded for all documented components and configuration items before
4129 handoff; the retention mechanism is tamper-evident by inspection and by attempted alteration.
- 4130 • Fail: measurements are missing, incomplete, or recorded after handoff; or the retention mechanism
4131 permits undetected alteration.

4132 **Assessment evidence:** record of retention mechanism inspection; record of measurement-set
4133 completeness; record of tamper-alteration tests.

4134 6.13.2 [ACC-BM-LOG-002]

4135 **Assessment objective:** verify that the boot manager provides measurement records in a format that
4136 supports remote attestation and includes mechanisms to ensure freshness against replay. References REQ-
4137 BM-LOG-002.

4138 **Assessment preparation:**

- 4139 • Test environment: the boot manager configured to emit measurement records to an external
4140 attestation verifier.
- 4141 • Required information: the manufacturer's specification of the measurement record format, the
4142 supported freshness mechanism (e.g. nonces, monotonic counters, timestamps, challenge-
4143 response), and the attestation protocol or format the records conform to.
- 4144 • Required tools: attestation verifier or parser capable of consuming the specified record format; tool
4145 for generating attestation challenges where the freshness mechanism is challenge-based.

4146 **Assessment activities:**

- 4147 • Capture a measurement record produced by the boot manager and parse it using an attestation
4148 verifier implementing the specified format.
- 4149 • Confirm that the parsed record contains at least the fields required for remote attestation:
4150 component identifiers, measurement digests, and the freshness element.
- 4151 • Issue a challenge (or consume a nonce) as supported by the freshness mechanism; verify that the
4152 boot manager's response binds to the challenge or nonce.
- 4153 • Replay a previously valid measurement record to the verifier and confirm that the replay is detected
4154 by the freshness mechanism.

4155 **Assessment verdict:**

- 4156 • Pass: measurement records parse against the specified attestation format; the freshness
4157 mechanism binds records to a fresh element and rejects replayed records.
- 4158 • Fail: records fail to parse; required fields are missing; freshness mechanism absent or fails to detect
4159 replay.

4160 **Assessment evidence:** sample of parsed measurement records; record of challenge-response exchange;
4161 record of replay rejection.

4162 6.13.3 [ACC-BM-LOG-003]

4163 **Assessment objective:** verify that the boot manager produces log indications for security-relevant failures
4164 and state changes. References REQ-BM-LOG-003.

4165 **Assessment preparation:**

- 4166 • Test environment: the boot manager executed under conditions that can be induced to produce
4167 each class of security-relevant event.

- 4168 • Required information: the manufacturer's enumeration of security-relevant failures and state
4169 changes recorded by the boot manager (at minimum: verification failures, authentication failures,
4170 security policy violations, recovery mode activation, execution of unsigned code), the log output
4171 mechanism, and the log record format.
- 4172 • Required tools: fault-injection or stimulation tools for each event class (e.g. corrupted boot image
4173 for verification failure, invalid credentials for authentication failure); log capture tool appropriate to
4174 the log output mechanism.

4175 **Assessment activities:**

- 4176 • For each enumerated event class, induce the event (e.g. present a boot image with an invalid
4177 signature to trigger a verification failure) and capture the boot manager's log output.
- 4178 • Confirm that an identifiable log entry is produced for each induced event, that the entry identifies
4179 the event class, and that the entry is consistent with the documented log record format.
- 4180 • Confirm that log entries are produced at the time the event occurs and are not suppressed under
4181 ongoing failure conditions.

4182 **Assessment verdict:**

- 4183 • Pass: each enumerated event class produces an identifiable log entry matching the documented
4184 format.
- 4185 • Fail: any enumerated event class fails to produce a log entry, produces an indistinguishable entry, or
4186 produces an entry inconsistent with the documented format.

4187 **Assessment evidence:** enumeration of event classes; records of fault-injection tests; captured log entries for
4188 each induced event.

4189 **6.13.4 [ACC-BM-LOG-004]**

4190 **Assessment objective:** verify that the boot manager provides version information accessible to the
4191 operating system, management systems, or authorised entity. References REQ-BM-LOG-004.

4192 **Assessment preparation:**

- 4193 • Test environment: the boot manager executed to completion with the interfaces accessible to
4194 operating system, management system or authorised entity consumers.
- 4195 • Required information: the manufacturer's documentation of the version information exposed
4196 (content and format), the access mechanism (e.g. ACPI table, UEFI variable, runtime service,
4197 platform-specific register), and the identity of the consumer role accessing the information.
- 4198 • Required tools: consumer-role tool for the access mechanism (e.g. operating-system utility reading
4199 the exposed structure).

4200 **Assessment activities:**

- 4201 • Query the boot manager's version information through each documented access mechanism.
- 4202 • Confirm that the returned information contains the documented version fields (e.g. boot manager
4203 version, build identifier, relevant component versions).

- 4204 • Confirm that the access is available to the documented consumers (operating system, management
4205 system, authorised entity) and that no authorisation step beyond those documented is required.

4206 **Assessment verdict:**

- 4207 • Pass: version information is accessible through each documented access mechanism, to each
4208 documented consumer role, and contains the documented fields.
- 4209 • Fail: any documented access mechanism fails to return the version information, or the returned
4210 information is missing required fields, or access requires undocumented authorisation.

4211 **Assessment evidence:** sample query and response through each documented access mechanism; record of
4212 field completeness.

4213 6.14 Data removal and transparency

4214 The requirements addressed by the present clause are specified in other clauses of the present document
4215 and assessed under the corresponding clauses of clause 6:

- 4216 • overwrite of confidential or secret data after use: REQ-BM-CON-007 assessed under 6.6;
4217 • secure erasure of sensitive data at decommissioning: REQ-BM-CON-010 assessed under 6.6.

4218 Transparency toward the user on data processed by the boot manager is not a boot manager product
4219 characteristic; boot managers do not process user personal data in the course of normal boot operations.
4220 No dedicated assessment criteria are defined for transparency.

4221 6.15 Vulnerability handling

4222 Assessment of the requirements specified in CEN/CLC JT013090:2026 (CEN/CLC prEN 40000-1-3) [1] applies.
4223 The present document does not define additional assessment criteria for these requirements.

4224

4225 **Annex A (normative): Relationship between the present**
4226 **document and the essential requirements of EU**
4227 **Regulation 2024/2847**

4228 The present document has been prepared under the Commission's standardisation request C(2025) 618
4229 [i.3] final to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847
4230 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity
4231 requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No
4232 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) [i.1].

4233 Once the present document is cited in the Official Journal of the European Union under that Regulation,
4234 compliance with the normative clauses of the present document given in table A-1 confers, within the limits
4235 of the scope of the present document, a presumption of conformity with the corresponding requirements
4236 of that Regulation and associated EFTA regulations.

4237
4238**Table A-1: Relationship between the present document and the requirements of Regulation (EU) 2024/2847**

Harmonised Standard ETSI EN 304 623					
Requirement				Requirement Conditionality	
No	Description	Requirements of Regulation	Clause(s)	U/C	Condition
Part I: Security Properties of Products with Digital Elements					
1	Products designed with appropriate cybersecurity	Annex I, Part I, (1)	Clause 5	U	
2	No known exploitable vulnerabilities	Annex I, Part I, (2)(a)	5.2	U	
3	Secure by default configuration	Annex I, Part I, (2)(b)	5.3	U	
4	Vulnerabilities addressable through updates	Annex I, Part I, (2)(c)	5.2, 5.4	C	When update capability present (Clause 4.3.3.2)
5	Protection from unauthorised access	Annex I, Part I, (2)(d)	5.5	C	When authentication capability present (Clause 4.3.3.8)
6	Confidentiality protection	Annex I, Part I, (2)(e)	5.6	U	
7	Integrity protection	Annex I, Part I, (2)(f)	5.7	U	
8	Data minimisation	Annex I, Part I, (2)(g)	5.8	U	
9	Availability and resilience	Annex I, Part I, (2)(h)	5.9	U	
10	Minimize impact on other devices	Annex I, Part I, (2)(i)	5.10	U	
11	Reduce attack surfaces	Annex I, Part I, (2)(j)	5.11	U	
12	Mitigate exploitation impact	Annex I, Part I, (2)(k)	5.12, 5.6, 5.7, 5.11	U	
13	Security-relevant recording/monitoring	Annex I, Part I, (2)(l)	5.13	C	When the product design supports recording of security-relevant events (Clause 4.3.3.7)
14	Secure data/settings removal	Annex I, Part I, (2)(m)	5.6	U	
Part II: Vulnerability Handling					
15	Identify and document vulnerabilities	Annex I, Part II, (1)	5.15	U	
16	Address and remediate vulnerabilities	Annex I, Part II, (2)	5.15	C	When update capability present (Clause 4.3.3.2)
17	Regular security testing	Annex I, Part II, (3)	5.15	U	
18	Disclose fixed vulnerabilities	Annex I, Part II, (4)	5.15	U	
19	Coordinated vulnerability disclosure	Annex I, Part II, (5)	5.15	U	
20	Facilitate vulnerability information sharing	Annex I, Part II, (6)	5.15	U	
21	Secure update distribution	Annex I, Part II, (7)	5.4, 5.15	C	When update capability present (Clause 4.3.3.2)
22	Free security updates with advisory	Annex I, Part II, (8)	5.4, 5.15	C	When update capability present (Clause 4.3.3.2)

4239

4240 Requirement:

4241	No	A unique identifier for one row of the table which may be used to identify a
4242		requirement.
4243	Description	A textual reference to the requirement.
4244		Requirements of Regulation
4245		Identification of article(s) defining the requirement in the Regulation.
4246		Clause(s) of the present document
4247		Identification of clause(s) defining the requirement in the present document unless
4248		another document is referenced explicitly.
4249		Requirement conditionality:
4250	U/C	Indicates whether the requirement is unconditionally applicable (U) or is conditional
4251		upon the manufacturer's claimed functionality of the equipment (C)
4252	Condition	Explains the conditions when the requirement is applicable for a requirement which is
4253		classified "conditional"
4254		

4255 Annex B (informative): 4256 Security analysis

4257 This annex documents the security analysis underlying the requirements in clause 5. It identifies assets
4258 (B.1), risk factors (B.2), assumptions (B.3), threats (B.4), and maps risk factors to use cases (B.5) and security
4259 profiles (B.6).

4260 Boot managers face unique challenges: pre-OS execution, establishment of platform trust, and extreme
4261 product diversity from 256-byte ROM code to full UEFI systems. This diversity makes requirements based on
4262 deployment scenario impractical since manufacturers often cannot predict final deployment context.

4263 The standard uses a capability-based model where objective technical features determine which threats are
4264 relevant and which requirements apply. Each product capability has dual security impact: it introduces
4265 attack surface requiring protective measures and enables security mechanisms that mitigate specific
4266 threats. For example, update capability enables vulnerability remediation while requiring protection against
4267 malicious update injection. Requirements address both aspects.

4268 B.1 Assets

4269 B.1.1 A-CRYPT: Cryptographic keys and certificates

4270 Keys and certificates establishing trust in the boot chain. Compromise enables signing malicious code as
4271 legitimate or decrypting protected data.

- 4272 • Root verification keys (platform keys, root certificates).
- 4273 • Signature database.
- 4274 • Device attestation keys.
- 4275 • Symmetric encryption keys.
- 4276 • Revocation data.

4277 NOTE: Platform keys require special attention due to supply chain risks including default keys and key
4278 leakage. See T-SUPPLY-6.

4279 B.1.2 A-ROLLBACK: Rollback protection data

4280 Counters and version information that prevent downgrade to vulnerable versions. Compromise enables
4281 rollback attacks where attackers exploit known patched vulnerabilities.

- 4282 • Version counters (monotonic, non-decreasing).
- 4283 • Minimum version numbers.
- 4284 • Anti-rollback enforcement data.

4285 B.1.3 A-CONFIG: Configuration data

4286 Settings that control boot manager security behaviour. Compromise allows attackers to weaken security
4287 policies or enable unauthorised boot paths.

- 4288 • Security policy settings.
- 4289 • Trust anchor configuration.
- 4290 • Boot device selection and priority.
- 4291 • Network boot credentials (when present).
- 4292 • Recovery mode settings.
- 4293 • Debug interface lockdown state.
- 4294 • Parsed input data (e.g. boot images, logo images, UEFI variables, configuration files).
- 4295 NOTE: Parsed input is attack surface for parser vulnerabilities. See T-INTEGRITY-14.

4296 B.1.4 A-AUTH: Authentication credentials

4297 Credentials controlling access to boot manager configuration and recovery functions. Compromise enables
4298 unauthorised configuration changes and security policy modifications.

- 4299 • Setup/BIOS passwords (hashed).
- 4300 • Administrator passwords (hashed).
- 4301 • Physical presence authentication tokens.
- 4302 • Recovery authentication data.

4303 B.1.5 A-MEASURE: Measurement and attestation data

4304 Records of boot components and configuration for remote verification. Compromise allows attackers to hide
4305 malicious modifications and appear trustworthy during attestation.

- 4306 • Event log.
- 4307 • Measurement metadata (component identities, versions, algorithms used).

4308 B.1.6 A-AUDIT: Audit logs

4309 Records of security-relevant events during boot. Compromise allows attackers to hide evidence of attacks
4310 and prevent detection or forensic investigation.

- 4311 • Verification success/failure events.
- 4312 • Authentication attempts.
- 4313 • Configuration change records.
- 4314 • Security policy violation attempts.
- 4315 • Recovery mode activation events.

4316 B.1.7 A-RUNTIME: Runtime state data

4317 Security state during boot execution. Compromise enables attackers to bypass security checks, manipulate
4318 boot decisions, or enable unlimited authentication attempts.

- 4319 • Security mode state (setup mode, user mode, deployed mode).
- 4320 • Verification completion flags.
- 4321 • Boot attempt counters.
- 4322 • Memory protection configuration.
- 4323 • Boot phase state machine.
- 4324 NOTE: State transitions are attack surface for privilege escalation. See T-INTEGRITY-15.

4325 B.1.8 A-CODE: Boot manager executable code

4326 The boot manager code itself while executing or staged for update. Compromise provides persistent
4327 privileged access and complete control over boot process.

- 4328 • Boot manager code in writable memory during execution.
- 4329 • Runtime services code (if persisting after boot target handoff).
- 4330 • Update staging area contents.

4331 B.1.9 A-UPDATE: Update packages and delivery mechanisms

4332 Update packages during delivery, staging, and installation, plus update tooling.

- 4333 • Update packages in transit.
- 4334 • Staged updates awaiting installation.
- 4335 • Update signature metadata.
- 4336 • Update verification tools.
- 4337 • Update delivery infrastructure trust anchors.

4338 NOTE: Update tools themselves are attack targets (BIOS Disconnect). See T-INTEGRITY-16, T-SUPPLY-7.

4339 B.2 Risk Factors

4340 B.2.1 General

4341 Risk factors characterise the security-relevant aspects of boot manager deployment. They provide the
4342 analytical basis for mapping of the use cases to the security profiles (B.5, B.6). Four risk factors are defined.
4343 RF-SURFACE is determined by the product's implemented capabilities. RF-NET, RF-AVAIL, and RF-IMPACT are
4344 determined by the deployment context described in clause 4.4.

4345 B.2.2 RF-SURFACE: Capability-driven attack surface

4346 The set of capabilities implemented in the boot manager determines the attack surface exposed during
4347 boot. Each additional capability introduces interfaces, protocols, or state that an attacker may target.

4348

4349

Table B.2.2-1: Capability-driven attack surface levels

Level	Description	Examples
Minimal	Immutable code, no configuration or update interfaces	ROM-only, fixed boot path
Standard	Verified boot, update capability, configuration management, logging	Updateable boot manager with local interfaces
Extended	Standard plus network boot, remote configuration, debug interfaces, or measured boot	Full-featured boot manager with network and remote management

4350

4351 B.2.3 RF-NET: Network reachability

4352 Network reachability of the boot manager during boot, update, or configuration operations. Elevated
 4353 network exposure increases remote attack surface before operating system security mechanisms are active.
 4354 See clause 4.4.3 for deployment context.

4355

4356

Table B.2.3-1: Network reachability levels

Level	Description	Examples
Isolated	No network connectivity during boot	Air-gapped, internal storage only
Managed	Network connectivity within managed perimeter	Enterprise with network boot via VPN
Exposed	Direct internet or untrusted network exposure	Edge devices

4357

4358 B.2.4 RF-AVAIL: Operational continuity

4359 The operational continuity expectations for the system in which the boot manager is deployed. Higher
 4360 continuity expectations increase the impact of boot failures and constrain acceptable recovery strategies.
 4361 See clauses 4.4.5 and 4.4.8 for deployment context.

4362

Table B.2.4-1: Operational continuity levels

Level	Description	Examples
Flexible	Downtime acceptable for maintenance and recovery	Development boards, consumer devices
Standard	Normal business availability expectations	Office equipment, personal computing
Critical	Continuous operation required; boot failure causes safety impact or critical service disruption	Medical devices, industrial control, telecommunications infrastructure

4363

4364 B.2.5 RF-IMPACT: Compromise severity

4365 The severity of consequences if the boot manager is compromised, considering data sensitivity, safety
 4366 implications, and downstream effects on other system components. See clauses 4.4.5 and 4.4.8 for
 4367 deployment context.

4368

Table B.2.5-1: Compromise severity levels

Level	Description	Examples
Limited	No sensitive data, no safety impact	Development boards, consumer devices
Significant	Personal or business data, moderate operational disruption	Consumer computing, enterprise workstations
Critical	Safety-critical systems, highly sensitive data, or critical infrastructure	Medical devices, financial infrastructure, government systems

4369

4370 B.3 Assumptions

4371 B.3.1 General

4372 The security analysis in this annex assumes the following conditions hold. These assumptions describe
 4373 properties that the boot manager depends on but cannot itself verify or enforce. Where an assumption
 4374 does not hold, the corresponding threats identified in B.4 are not mitigated by this standard.

4375 B.3.2 Hardware platform assumptions

4376 **AS-HW-1:** The hardware root of trust (immutable boot ROM, e-fuses, or equivalent) is correctly
 4377 implemented and has not been modified after manufacture. [T-PHYS-5, T-PHYS-6, T-PHYS-9]

4378 **AS-HW-2:** Hardware memory protection and privilege separation mechanisms function as specified by the
 4379 platform. [T-PHYS-2]

4380 **AS-HW-3:** Hardware security components (TPM, secure element, HSM) correctly isolate key material and
 4381 resist physical probing at the level claimed by the component. [T-PHYS-3, T-PHYS-4]

4382 **AS-HW-4:** Hardware debug interfaces (JTAG, SWD) are disabled or protected by the platform when claimed.
 4383 [T-PHYS-11]

4384 **AS-HW-5:** The platform provides sufficient entropy for cryptographic key generation. [T-INTEGRITY-12]

4385 **AS-HW-6:** The platform provides hardware mechanisms to resist fault injection (voltage glitching, EM
 4386 interference) when claimed. [T-PHYS-2]

4387 B.3.3 Process assumptions

4388 **AS-PROC-1:** The manufacturing environment provisions correct initial key material and firmware. [T-SUPPLY-
 4389 1]

4390 **AS-PROC-2:** Development toolchains have not been compromised. [T-SUPPLY-3]

4391 **AS-PROC-3:** Update distribution infrastructure is operated securely by the manufacturer. [T-SUPPLY-4, T-
 4392 SUPPLY-5]

4393 **AS-PROC-4:** Update tooling is authentic and has not been tampered with. [T-SUPPLY-7]

4394 B.3.4 Boot target assumptions

4395 **AS-TARGET-1:** The post-handoff stage (operating system, hypervisor, or next-stage loader) implements
 4396 security properties consistent with the declared use case, or another verification mechanism takes over at
 4397 handoff. Applies to all use cases.

4398 **AS-TARGET-2:** For UC-IMM, the post-handoff stage maintains equivalent immutability, or the boot target
4399 itself is immutable. Where this assumption does not hold, the absence of post-deployment remediation
4400 that defines UC-IMM's risk posture extends only to the boot manager and not to the running system.

4401 B.4 Threats

4402 B.4.1 Security property impacts

4403 B.4.1.1 General

4404 Boot managers present unique risks due to their privileged position, persistence, invisibility, and role as
4405 trust foundation:

- 4406 • Privileged position: Operates at highest privilege level with direct hardware access.
- 4407 • Persistence: Survives OS reinstall and traditional security remediation.
- 4408 • Invisibility: Executes below OS visibility and security tools.
- 4409 • Trust establishment: Forms the foundation for all subsequent security decisions

4410 These characteristics mean boot manager compromises have disproportionate impact compared to
4411 application or OS-level compromises. Different capabilities introduce different threat exposures.

4412 B.4.1.2 Confidentiality

4413 The confidential information that the bootloader may have access to include the storage encryption keys,
4414 network access keys, and the user or device identity. This information may be stored in non-volatile storage,
4415 in a trusted hardware element, or retrieved from another system as part of a remote attestation.

4416 If this data is stored in clear text in non-volatile storage, a local attacker may be able to gain access to the
4417 confidential information. This sort of storage is only suggested for use cases where physical access is
4418 protected by other means.

4419 Even if the confidential data is stored in a trusted hardware component, the local bus that connects the
4420 bootloader to the external hardware may be compromised by a local attacker. In some cases, the secrets are
4421 returned to the bootloader in plain text on the bus, putting them at risk of disclosure.

4422 If the secrets are encrypted on the bus, but the measurements from the bootloader to the trusted
4423 hardware component are not, the device may be susceptible to a replay attack leading to disclosure of the
4424 secrets.

4425 If the devices on the local bus are not authenticated in some way, a local attacker might be able to install an
4426 interposer to intercept the measurements and modify them, similar to the replay attack, to reveal the
4427 secrets.

4428 If the attacker can downgrade from verified to measured boot, they should not be able to unseal
4429 confidential data since the configuration measurements will not match, but they might be able to present
4430 an authentic looking authentication screen to the user and trick them into revealing recovery or other
4431 secrets.

4432 During a remote attestation, an attacker may be able to force a protocol downgrade that could lead to a
4433 disclosure of user identity or the returned secrets.

4434 A loss of confidentiality of disk encryption keys could lead to a larger loss of confidentiality of the disk
4435 contents, as well as could be leveraged to a persistent loss of integrity in the operating system.

4436 B.4.1.3 Integrity

4437 Integrity for the bootloader means that the OS that it hands control to is the one that the system's owner
4438 intends, that the system is configured as intended, and any hardware measurements correctly reflect this
4439 configuration. Any attack that compromises integrity can also compromise availability and confidentiality,
4440 and potentially compromise the availability, confidentiality and integrity of the operating system that it
4441 loads.

4442 An attacker that can modify the bootloader configuration can lead to a loss of integrity, such as by changing
4443 the verification keys to load an attacker signed operating system or enabling a debug mode that does not
4444 perform validation.

4445 An attacker that can force the bootloader to execute untrusted code, through misconfiguration or other
4446 means such as TOCTOU, can lead to a loss of integrity.

4447 An attacker might be able to modify measurements when they are being sent to the trusted hardware
4448 element, leading to a failure of integrity of the measurements, which could lead to faked attestations.

4449 An attacker might rollback the system configuration to an older, vulnerable but still signed version that could
4450 be exploited to compromise integrity unless there are rollback protections.

4451 Additionally, some attacks on bootloader integrity can lead to persistence in flash, disk, or other storage,
4452 allowing the attacker to compromise the integrity of not only the current boot, but all future boots as well.

4453 Attacks on bootloader runtime services from the operating system or from a hypervisor guest are also a
4454 potential attack on overall system integrity since they often have elevated privilege levels above the
4455 hypervisor itself. This is another means that an attacker on the bootloader could gain persistence on the
4456 system after the bootloader has passed control to the operating system.

4457 Audit and security event logs may be targeted by attackers to hide evidence of boot compromise. Loss of
4458 audit log integrity prevents forensic investigation and attack detection.

4459 B.4.1.4 Availability

4460 If an attacker changes any of the boot configuration in a verified boot system, the device may no longer
4461 boot and might not be accessible by the user. Depending on when during the boot process the
4462 configuration verification fails, communicating the nature of the failure to the user may be challenging.
4463 System designers should consider recovery flows for re-establishing trust in the device if the bootloader
4464 verification fails.

4465 Corruption of rollback protection, measurement data, or runtime state can cause boot failures. Critical asset
4466 unavailability can render the system unbootable, creating support and recovery challenges.

4467 B.4.1.5 Impact of other devices

4468 Integrity failures during the bootloader process can leave the attacker in a position to interfere with other
4469 devices on the network or future components in the boot chain.

4470 Availability failures could lead to a boot loop where the bootloader continuously tries to authenticate with a
4471 remote server.

4472 B.4.2 Threat identification methodology

4473 This risk analysis uses three interconnected elements:

4474 **Assets (Annex B.1)** Nine categories requiring protection: code, configuration, runtime state, cryptographic
4475 material, measurements, audit logs, authentication credentials, update packages and rollback counters.

4476 **Threats (Annex B.4.3):** Six threat categories: T-INTEGRITY, T-PERSIST, T-PHYS, T-SUPPLY, T-NET, T-AVAIL.

4477 **Product capabilities (clause 4.3):** Technical features determining which threats are relevant and which
4478 security mechanisms are available.

4479 The analysis maps product characteristics to threat exposure:

- 4480 • Which threats become relevant based on technical features.
- 4481 • Which assets are impacted.
- 4482 • Risk implications for each characteristic.

4483 Each product characteristic clause provides:

- 4484 • Relevant threats for each characteristic variant.
- 4485 • Assets affected.
- 4486 • Threat exposure analysis.
- 4487 • Risk implications for manufacturers.

4488 B.4.2.1 Development approach

4489 The threat taxonomy in clause B.4.3 was developed through analysis of documented attacks, security
4490 frameworks, regulatory requirements, and expert review.

4491 B.4.2.2 Threat sources

- 4492 • NIST SP 800-193 [i.6] Platform Firmware Resiliency Guidelines, Appendix A attack scenarios.
- 4493 • Threat frameworks: MITRE EMB3D, CAPEC, CWE firmware patterns.
- 4494 • Documented attacks, for example BlackLotus (CVE-2022-21894), BootHole (CVE-2020-10713),
4495 LogoFAIL (CVE-2023-40238), BootBandit (CVE-2022-34301).
- 4496 • CRA Annex I: Essential cybersecurity requirements for supply chain, updates, vulnerability handling.

4497 B.4.2.3 Categorization rationale

4498 Six threat categories separate attacks by vector and persistence mechanism. Categories reflect attack
4499 mechanisms while acknowledging sophisticated attacks combine multiple vectors.

4500 B.4.2.4 Limitations

4501 Taxonomy represents threats known at standard development.

4502 B.4.3 Threat catalogue

4503 B.4.3.1 Introduction

4504 This clause provides the threat catalogue referenced by requirements in clause 5. Each threat is identified
 4505 by a unique identifier. Threat relevance is determined by capability presence. Threat exposure level is
 4506 determined by risk factor analysis per Annex B.2.

4507 NOTE: The "Mitigated by" column uses the following markers: REQ-BM-ESR-NNN: Requirement(s) that
 4508 mitigate this threat.

4509 Threats marked [HW] or [PROC] are not addressable by boot manager software alone.

4510 B.4.3.2 T-INTEGRITY: Boot integrity attacks

4511 Boot integrity attacks target the authenticity and trustworthiness of boot components, attempting to
 4512 execute unauthorised code, weaken security policies, bypass verification mechanisms, or exploit parsing
 4513 and input validation weaknesses. Threat exposure increases with RF-SURFACE (additional capabilities
 4514 introduce verification and parsing attack surface).

4515 B.4.3.2.1 Code execution attacks

4516 Attacks that achieve execution of unauthorised code during boot.

4517 **Table B.4.3.2.1-1: Code execution attacks**

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-INTEGRITY-1	A-CODE	All products	Malicious boot component substitution	REQ-BM-INT-002, REQ-BM-INT-003
T-INTEGRITY-2	A-CODE	Verified boot	Signature verification bypass	REQ-BM-INT-002, REQ-BM-INT-003, REQ-BM-INT-009
T-INTEGRITY-3	A-CODE, A-UPDATE	Update capability	Malicious update injection	REQ-BM-INT-005, REQ-BM-SU-004, REQ-BM-SU-005
T-INTEGRITY-4	A-RUNTIME, A-CODE	All products	Runtime code modification	REQ-BM-INT-010

4518

4519 B.4.3.2.2 Configuration tampering

4520 Attacks that modify security policies and boot settings to weaken protections.

4521

Table B.4.3.2.2-1: Configuration tampering

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-INTEGRITY-5	A-CONFIG, A-RUNTIME	Configuration capability	Security policy weakening via configuration interface	REQ-BM-AAC-002, REQ-BM-AAC-003, REQ-BM-AAC-004, REQ-BM-INT-009
T-INTEGRITY-6	A-CONFIG	Configuration capability	Boot order manipulation to load attacker media	REQ-BM-AAC-002
T-INTEGRITY-7	A-CONFIG, A-CRYPT	Configuration and update capability	Trust anchor database modification	REQ-BM-AAC-001, REQ-BM-CON-003
T-INTEGRITY-8	A-CONFIG, A-RUNTIME	Runtime update and configuration capability	Configuration TOCTOU exploitation	REQ-BM-INT-010
T-INTEGRITY-9	A-CONFIG	Recovery capability	Recovery configuration abuse	REQ-BM-AP-008, REQ-BM-AP-014

4522

4523 B.4.3.2.3 Cryptographic compromise

4524 Attacks targeting keys, certificates, and cryptographic material.

4525

Table B.4.3.2.3-1: Cryptographic compromise

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-INTEGRITY-10	A-CRYPT	Software-only implementation	Key extraction via software attacks	[HW]
T-INTEGRITY-11	A-CRYPT	Configuration and update capability	Key injection/replacement via interfaces	REQ-BM-AAC-001
T-INTEGRITY-12	A-CRYPT	Verified and measured boot	Weak key generation from insufficient entropy	[HW]
T-INTEGRITY-13	A-CRYPT	Long lifetime	Cryptographic algorithm exploitation	REQ-BM-INT-021, REQ-BM-INT-022

4526

4527 B.4.3.2.4 Parser and input validation attacks

4528

Table B.4.3.2.4-1: Parser and input validation attacks

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-INTEGRITY-14	A-CODE, A-RUNTIME	All products	Boot image and data format parsing vulnerabilities	REQ-BM-MAS-005
T-INTEGRITY-15	A-CODE, A-CONFIG	Network boot	Network protocol parsing vulnerabilities	REQ-BM-MAS-005, REQ-BM-MAS-006
T-INTEGRITY-16	A-CRYPT	Verified boot	Certificate and cryptographic format parsing vulnerabilities	REQ-BM-INT-019
T-INTEGRITY-17	A-RUNTIME	All products	Boot phase state machine manipulation	REQ-BM-INT-001, REQ-BM-MAS-002
T-INTEGRITY-18	A-CRYPT, A-CODE	Long lifetime, verified and measured boot	Cryptographic algorithm deprecation over extended deployment	REQ-BM-INT-021, REQ-BM-INT-022, REQ-BM-AP-013
T-INTEGRITY-19	A-CRYPT	Long lifetime, verified boot	Key compromise accumulation over extended periods	[HW], REQ-BM-INT-020

4529

4530 B.4.3.2.5 Rollback attacks

4531 Attacks that install older, vulnerable versions of firmware or configuration.

4532 **Table B.4.3.2.5-1: Rollback attacks**

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-INTEGRITY-20	A-ROLLBACK, A-CODE	Update capability	Forced version downgrade or anti-rollback bypass	REQ-BM-INT-017, REQ-BM-INT-018, REQ-BM-SBD-004
T-INTEGRITY-21	A-ROLLBACK	Update capability, software-only implementation	Version counter manipulation or reset	[HW], REQ-BM-INT-018
T-INTEGRITY-22	A-CONFIG	Configuration capability	Configuration rollback to insecure settings	REQ-BM-INT-017, REQ-BM-LOG-001

4533

4534 NOTE: Products without update capability are immune to firmware rollback attacks but face elevated
4535 risk from vulnerability accumulation.

4536

4537 B.4.3.3 T-PERSIST: Persistent firmware threats

4538 Persistent firmware threats establish malware or compromise that survives operating system reinstallation,
4539 disk formatting, and traditional security remediation. Threat exposure increases with RF-SURFACE (writable
4540 storage and update mechanisms enable persistence).

4541 **Table B.4.3.3-1: Persistent firmware threats**

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-PERSIST-1	A-CODE	Fixed storage, removable media, external expansion	Bootkit installation in boot manager storage	REQ-BM-INT-002, REQ-BM-INT-003, REQ-BM-INT-006
T-PERSIST-2	A-CODE	Fixed storage, removable media	Boot image tampering in persistent storage	REQ-BM-INT-002, REQ-BM-INT-006
T-PERSIST-3	A-CONFIG	Recovery capability	Recovery mechanism compromise	REQ-BM-AP-008, REQ-BM-AP-014
T-PERSIST-4	A-MEASURE	Measured boot	Attestation data forgery to hide compromise	REQ-BM-LOG-001, REQ-BM-LOG-002
T-PERSIST-5	A-AUDIT	All products	Event log manipulation to hide evidence	REQ-BM-AP-012
T-PERSIST-6	A-CODE, A-CONFIG	Network boot	Network boot infrastructure compromise	REQ-BM-INT-014, REQ-BM-INT-016
T-PERSIST-7	A-RUNTIME, A-AUDIT	All products	Boot resource exhaustion preventing successful boot completion	REQ-BM-AP-003, REQ-BM-AP-004

4542

4543 B.4.3.4 T-PHYS: Physical attacks

4544 Physical attacks require direct hardware access to extract secrets, modify hardware, inject faults, or
4545 manipulate components. Physical security context determines attack feasibility more than boot manager
4546 capabilities. Threat exposure is independent of risk factors; physical security context is an assumption (see
4547 B.3.2).

4548

Table B.4.3.4-1: Physical attacks

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-PHYS-1	A-CRYPT, A-RUNTIME, A-CODE	Boot from external expansion, software-only implementation	DMA attacks via peripheral buses	REQ-BM-MAS-001, REQ-BM-CON-005
T-PHYS-2	A-RUNTIME, A-CODE	Software-only implementation, verified boot	Fault injection (voltage glitching, EM interference)	[HW]
T-PHYS-3	A-CRYPT, A-MEASURE	Hardware security storage	Hardware security component initialization attacks	[HW]
T-PHYS-4	A-CRYPT, A-AUTH	Hardware security storage, verified and measured boot	Side-channel key extraction (power, EM, timing)	[HW]
T-PHYS-5	A-CODE, A-CONFIG, A-CRYPT	Physical access	Hardware tampering and component modification	[HW]
T-PHYS-6	A-CODE, A-CONFIG	Fixed storage boot	Flash memory direct manipulation	[HW]
T-PHYS-7	A-CODE, A-CONFIG	Removable media boot	Removable media substitution	REQ-BM-MAS-001, REQ-BM-INT-003
T-PHYS-8	A-CODE, A-CONFIG	External expansion boot	External storage device compromise	REQ-BM-MAS-001, REQ-BM-INT-003
T-PHYS-9	A-CODE	Internal storage boot	ROM extraction via chip de-packaging	[HW]
T-PHYS-10	A-CODE, A-CRYPT, A-RUNTIME	Debug interface boot	Debug interface exploitation and authentication bypass	REQ-BM-AAC-001, REQ-BM-MAS-002, REQ-BM-MAS-003
T-PHYS-11	A-CRYPT, A-CODE	Debug interface boot	Memory and firmware extraction via debug interfaces	[HW]

4549

4550 **B.4.3.5 T-SUPPLY: Supply chain attacks**

4551 Supply chain attacks compromise boot managers during manufacturing, development, distribution, or
4552 through systemic trust failures. These threats affect all products regardless of capabilities, though update
4553 mechanisms expand the attack surface. Threat exposure increases with RF-SURFACE (update capability
4554 extends the supply chain attack surface).

4555

Table B.4.3.5-1: Supply chain attacks

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-SUPPLY-1	A-CODE, A-CRYPT	All products	Manufacturing environment compromise	[PROC], REQ-BM-INT-002, REQ-BM-SBD-001
T-SUPPLY-2	A-CODE	All products	Third-party component compromise	REQ-BM-SU-006
T-SUPPLY-3	A-CODE	All products	Development tool compromise (compiler attacks)	[PROC]
T-SUPPLY-4	A-CODE	Update capability	Distribution interception	[PROC]
T-SUPPLY-5	A-CODE, A-CRYPT	Update capability	Update infrastructure compromise	[PROC]
T-SUPPLY-6	A-CRYPT	Verified boot	Default or leaked platform keys	REQ-BM-CON-002, REQ-BM-SU-005
T-SUPPLY-7	A-UPDATE	Update capability	Update tooling compromise	[PROC]
T-SUPPLY-8	A-CRYPT	Verified and measured boot	Trust anchor revocation failures	REQ-BM-INT-008, REQ-BM-INT-020

4556

4557 **B.4.3.6 T-NET: Network-based attacks**

4558 Network-based attacks exploit network connectivity during boot, update, or configuration processes,
 4559 providing remote attack surface. These attacks occur before operating system security mechanisms are
 4560 active. Threat exposure increases with RF-NET (network reachability enables remote exploitation before OS
 4561 security is active).

4562

Table B.4.3.6-1: Network based attacks

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-NET-1	A-CODE, A-CONFIG, A-CRYPT	Network update capability	Man-in-the-middle and boot image tampering in transit	REQ-BM-DM-002, REQ-BM-INT-005, REQ-BM-INT-015
T-NET-2	A-CODE, A-CONFIG	Network boot	Rogue DHCP/TFTP/HTTP boot servers	REQ-BM-INT-014, REQ-BM-INT-016, REQ-BM-MAS-004
T-NET-3	A-CODE	Network update capability	Network protocol implementation vulnerabilities	REQ-BM-MAS-004, REQ-BM-MAS-005, REQ-BM-MAS-006
T-NET-4	A-CONFIG, A-AUTH	Remote configuration capability	Remote management interface exploitation	REQ-BM-MAS-003, REQ-BM-CON-006
T-NET-5	A-UPDATE	Network update capability	Update denial of service	REQ-BM-IM-001, REQ-BM-AP-004
T-NET-6	A-CONFIG, A-CRYPT, A-MEASURE	Network boot, Measured boot	Unauthorised disclosure of boot configuration, attestation, or cryptographic material via network	REQ-BM-CON-009

4563

4564 **B.4.3.7 T-AVAIL: Availability and resilience threats**

4565 Availability and resilience threats prevent successful boot completion, cause denial of service, or undermine
 4566 recovery mechanisms that ensure operational continuity. Threat exposure increases with RF-AVAIL (higher

4567 continuity expectations amplify the impact of boot failures) and RF-SURFACE (additional capabilities
4568 introduce failure modes).

4569

Table B.4.3.7-1: Resilience threats

Threat ID	Affected Assets	Introduced by	Description	Mitigated by
T-AVAIL-1	A-CODE, A-CONFIG	Update capability	Update process causes denial of service through resource exhaustion or deadlock	REQ-BM-SU-006, REQ-BM-AP-011, REQ-BM-AP-012
T-AVAIL-2	A-RUNTIME, A-CONFIG	Network boot capability	Network boot resource exhaustion or infrastructure unavailability	REQ-BM-AP-003, REQ-BM-AP-004, REQ-BM-AP-005, REQ-BM-AP-009, REQ-BM-AP-010
T-AVAIL-3	A-CONFIG, A-RUNTIME	Configuration management	Boot configuration corruption preventing successful boot	REQ-BM-AP-001, REQ-BM-AP-013
T-AVAIL-4	A-CODE, A-RUNTIME	Multiple boot sources	Primary boot path failure without fallback mechanism	REQ-BM-AP-002, REQ-BM-AP-007
T-AVAIL-5	A-CONFIG, A-CODE	Recovery capability	Recovery mechanism unavailable or corrupted when needed	REQ-BM-AP-014, REQ-BM-AP-018

4570

4571 NOTE 1: Availability threats often interact with integrity threats (T-INTEGRITY) and persistence threats
4572 (T-PERSIST).

4573 NOTE 2: Many availability protections are reliability features (timeouts, retries, fallbacks) rather than
4574 direct security countermeasures. These requirements support recovery from attacks but are
4575 not mitigations in themselves.

4576 B.5 Mapping of risk factors to use cases

4577 Table B.5-1 maps the risk factors defined in B.2 to the use cases defined in clause 4.8. Values reflect typical
4578 deployments. Actual values for a specific product may vary within the use case.

4579 NOTE 1: The mappings in this clause are informative. They support the rationale for the use case
4580 definitions in clause 4.8 and the security profiles in B.6, but do not in themselves impose
4581 requirements on the boot manager.

4582

Table B.5-1: Risk factor mapping to use cases

Risk factor	UC-IMM	UC-VER	UC-HW
RF-SURFACE	Minimal	Standard	Standard
RF-NET	Isolated	varies	varies
RF-AVAIL	Flexible	varies	varies
RF-IMPACT	Limited	varies	Critical

4583

4584 NOTE 2: "varies" indicates that the risk factor is determined by the deployment context of the device in
4585 which the boot manager is integrated, not by the boot manager product itself.

4586 B.6 Mapping of risk factors to security profiles

4587 B.6.1 General

4588 The security profiles are derived from the risk factor analysis in B.5. This clause documents the rationale for
4589 associating each use case with its security profile.

4590 B.6.2 UC-IMM: LOW

4591 UC-IMM products have minimal attack surface (RF-SURFACE: Minimal).

4592 The absence of update, network boot, and configuration capabilities eliminates entire threat categories (T-
4593 NET, T-AVAIL-1, T-INTEGRITY-3, T-INTEGRITY-20/21/22).

4594 The residual risk is vulnerability accumulation without remediation capability, accepted because the
4595 minimal attack surface limits exploitability. Security profile LOW is appropriate.

4596 B.6.3 UC-VER: MEDIUM

4597 UC-VER products have standard attack surface (RF-SURFACE: Standard).

4598 Verified boot detects and refuses execution of bootkit-installed components covered by T-INTEGRITY-1/2
4599 and T-PERSIST-1/2.

4600 Update capability mitigates long-term vulnerability accumulation but introduces T-INTEGRITY-3 and T-
4601 SUPPLY-4/5/7. Logging enables post-incident analysis.

4602 Key provisioning supports trust anchor lifecycle management. The balance of mitigation and introduced risk
4603 places UC-VER above LOW.

4604 Environmental factors (RF-NET, RF-AVAIL, RF-IMPACT) vary by deployment but do not change the security
4605 profile; they are addressed through functional capability requirements triggered when the corresponding
4606 capability is present. Security profile MEDIUM is appropriate.

4607 B.6.4 UC-HW: HIGH

4608 UC-HW products add hardware-backed key protection to UC-VER capabilities. This addresses T-INTEGRITY-
4609 10 (software key extraction) which UC-VER cannot mitigate.

4610 Hardware security reduces software-only attack surface against the root of trust to physical attacks
4611 requiring specialist equipment.

4612 Typical deployments have critical impact expectations (RF-IMPACT: Critical), while availability expectations
4613 vary by deployment context. The combination of elevated impact and the hardware assurance described
4614 above justifies the highest assessment depth, independent of the availability expectation. Security profile
4615 HIGH is appropriate.

4616 **Annex C (informative):**
4617 **Product Documentation**

4618 **C.1 Introduction**

4619 Table C.1-1 maps the documentation artifacts described in this annex to the requirements they support and
4620 the security profiles for which they are applicable.

4621 **Table C.1-1: Documentation to requirements mapping**

Artifact	Supports	LOW	MEDIUM	HIGH
Threat modelling	REQ-BM-KEV-001 REQ-BM-KEV-002	X	X	X
Boot flow and trust boundaries	REQ-BM-MAS-001	X	X	X
Cryptographic architecture	REQ-BM-CON-001	X	X	X
Memory architecture	REQ-BM-INT-001 REQ-BM-MAS-001	X	X	X
Interface architecture	REQ-BM-MAS-003	X	X	X
Recovery mechanisms	REQ-BM-AP-014		X	X
Isolation boundaries	REQ-BM-INT-001 REQ-BM-MAS-003	X	X	X
Software Bill of Materials (SBOM)	REQ-BM-LOG-004	X	X	X
Measurement process	REQ-BM-LOG-001		X	X
Environmental assumptions	REQ-BM-INT-004	X	X	
Configuration security	REQ-BM-AAC-002		X	X
Update procedures	REQ-BM-SU-004		X	X
Compensating controls	REQ-BM-KEV-003 REQ-BM-SU-003	X	X	X
Physical security	REQ-BM-MAS-003	X	X	X
End-of-life guidance	REQ-BM-CON-007, REQ-BM-CON-010	X	X	X

4622

4623 NOTE: Security testing evidence applies to all profiles (LOW, MEDIUM, HIGH) and is addressed through
4624 the conformity assessment regime in Clause 6. Testing evidence requirements are described in
4625 C.7.

4626 **C.2 Security design documentation**

4627 **Threat modelling:** Documented threat modelling covering all threats in Annex B applicable to the product's
4628 capabilities.

4629 **Boot flow and trust boundaries:** Documentation of the complete boot flow from power-on to handoff,
4630 identifying trust boundaries between stages and the defence-in-depth approach.

4631 **Cryptographic architecture:** Documentation of the cryptographic key hierarchy, storage locations, and trust
4632 model. This includes key provisioning, lifecycle management, and identification of trust anchors.

4633 **Memory and interface architecture:** Documentation of memory layout, protection mechanisms, and
4634 interfaces exposed at each boot stage.

4635 **Recovery mechanisms:** Documentation of recovery and failsafe mechanisms, including recovery triggers,
4636 failsafe boot paths, and user notification.

4637 **Isolation boundaries:** Documentation of isolation boundaries between boot components, including the
4638 mechanisms employed and failure isolation properties.

4639 **Software Bill of Materials (SBOM):** Software bill of materials aligned with the requirements of prEN 40000-
4640 1-3 [1], listing all third-party components with version information.

4641 C.3 Physical security documentation

4642 Documentation of physical security requirements when updates are not supported. This covers physical
4643 access control requirements, tamper detection expectations, and compensating controls for lack of update
4644 capability.

4645 C.4 Integrity mechanism documentation

4646 **Measurement process:** Documentation of the measurement process for validation, including measurement
4647 algorithm, sequence, storage, and validation method.

4648 **Environmental assumptions without hardware security:** Documentation of environmental assumptions for
4649 physical security when hardware security components are not available, including threat model limitations
4650 and deployment restrictions.

4651 C.5 Configuration security documentation

4652 Documentation of security impact for each configuration option, including recommended secure values and
4653 warnings for insecure configurations. Applies to boot managers with configuration capability (all profiles).

4654 C.6 Vulnerability handling documentation

4655 **Update procedures:** Documentation of update procedures, or explicit statement that updates are not
4656 supported with rationale.

4657 **Compensating controls for non-updateable implementations:** Documentation of compensating controls
4658 and hardware replacement as the update mechanism when updates are not supported.

4659 **End-of-life guidance:** End-of-life guidance when vulnerabilities cannot be mitigated for non-updateable
4660 implementations, including decommissioning procedures.

4661 **Vulnerability mitigation scope:** Documentation of which vulnerabilities can be addressed via configuration
4662 versus code updates.

4663 C.7 Security testing evidence

4664 Evidence demonstrating the boot manager has undergone appropriate security validation.

4665 **Signature verification testing:** Evidence of testing valid signatures, invalid signatures, expired signatures,
4666 and revoked signatures.

4667 **Authentication mechanism testing:** Evidence of authentication bypass and privilege escalation testing.

4668 **Rollback protection testing:** Evidence of rollback attempt testing with older valid images and version
4669 counter manipulation.

- 4670 **Secure failure mode testing:** Evidence of testing behaviour under verification failure, resource exhaustion,
4671 and corrupted input.
- 4672 **Input fuzz testing:** Evidence of fuzz testing covering parsers for configuration, images, and certificates.
- 4673 **Memory safety testing:** Evidence of buffer overflow and integer overflow testing, including static and
4674 dynamic analysis.
- 4675 **Code review:** Evidence of security-focused code review.
- 4676 **Architecture review:** Evidence of architecture review against the documented threat model.
- 4677 **Binary analysis:** Evidence of binary analysis of production builds, including verification that debug symbols
4678 and test code are removed.
- 4679 **Supply chain security review:** Evidence of supply chain security review covering third-party components
4680 and build process.
- 4681 **Test coverage metrics:** Documentation of test coverage metrics for code, requirements, and threats.
- 4682 **Vulnerability remediation:** Evidence of remediation of vulnerabilities identified during testing.

4683 **Annex E (informative):**
4684 **Relation to NIST SP 800-193**

4685 **E.1 Purpose**

4686 This annex maps requirements to the Protection/Detection/Recovery (PDR) framework from NIST SP
4687 800-193 Platform Firmware Resiliency Guidelines.

4688 NOTE 1: NIST SP 800-193 [i.6] defines Roots of Trust (RTU, RTD, RTRec) anchored in immutable
4689 hardware. These are out of scope, boot managers consume these roots but do not provide
4690 them.

4691 NOTE 2: "Section" references throughout this annex refer to sections within NIST SP 800-193 [i.6], not
4692 clauses within the present document.

4693 **E.2 Protection (NIST SP 800-193, Section 4.2)**

4694 **E.2.1 Authenticated update mechanism (NIST SP 800-193, Section**
4695 **4.2.1.1)**

4696 **Table E.2.1-1: Authenticated update mechanism**

Requirement	Description
REQ-BM-SU-001	Update mechanisms enabling vulnerability remediation
REQ-BM-INT-005	Verify authenticity and integrity of update packages
REQ-BM-SU-005	Execute update operations atomically
REQ-BM-SU-004	Store update verification keys in hardware security component
REQ-BM-INT-018	Enforce rollback protection
REQ-BM-INT-019	Store anti-rollback counters, verify signed version metadata

4697

4698 **E.2.2 Integrity protection (NIST SP 800-193, Section 4.2.1.2)**

4699 **Table E.2.2-1: Integrity protection**

Requirement	Description
REQ-BM-INT-002	Verify integrity and authenticity using hashes and signatures
REQ-BM-INT-003	Prevent boot with unverified, invalid, expired, or revoked components
REQ-BM-INT-006	Maintain configurable list of approved signatures
REQ-BM-INT-009	Prevent unauthorised modification between verification and execution
REQ-BM-INT-010	Load components into protected memory before verification
REQ-BM-AAC-007	Protect trusted certificate stores from unauthorised modification
REQ-BM-CON-010	Overwrite sensitive data after use
REQ-BM-LOG-001	Record measurements in a tamper-evident manner

4700

4701 E.2.3 Non-bypassability (NIST SP 800-193, Section 4.2.1.3)

4702 **Table E.2.3-1: Non-bypassability**

Requirement	Description
REQ-BM-CON-001	Restrict access to cryptographic key material
REQ-BM-MAS-001	Disable debug interfaces, revoke DMA access before handoff
REQ-BM-INT-001	Enforce privilege boundaries between boot stages
REQ-BM-AAC-001	Require physical presence for key/certificate changes
REQ-BM-AAC-002	Protect boot order and parameters from unauthorised modification
REQ-BM-AAC-003	Require authentication before weakening security defaults
REQ-BM-AAC-005	Verify cryptographic signature on security policy changes
REQ-BM-INT-003	Prevent boot continuation with unverified components
REQ-BM-INT-008	Require authentication before allowing verification bypass
REQ-BM-SBD-001	Enable cryptographic signature validation by default
REQ-BM-SBD-002	No default passwords, backdoors, or undocumented access
REQ-BM-MAS-003	Disable debug interfaces and test modes in production

4703

4704 E.2.4 Protection of critical data (Section 4.2.4)

4705 **Table E.2.4-1: Protection of critical data**

Requirement	Description
REQ-BM-INT-012	Protect sensitive configuration with authenticated encryption
REQ-BM-CON-006	Hash stored credentials

4706

4707 E.3 Detection (NIST SP 800-193, Section 4.3)

4708 E.3.1 Detection of corrupted code (NIST SP 800-193, Section 4.3.1)

4709 **Table E.3.1-1: Detection of corrupted code**

Requirement	Description
REQ-BM-LOG-001	Record measurements in tamper-evident manner.
REQ-BM-LOG-002	Provide measurement logs with freshness mechanisms
REQ-BM-INT-004	Check component provenance and version consistency
REQ-BM-INT-002	Verify boot components against trust anchor before execution

4710

4711 E.3.2 Detection of corrupted critical data (NIST SP 800-193, Section 4.3.2)

4712 **Table E.3.2-1: Detection of corrupted critical data**

Requirement	Description
REQ-BM-INT-013	Restore secure defaults when corruption detected
REQ-BM-LOG-001	Record measurements of boot components and configuration
REQ-BM-LOG-001	Record measurements in a tamper-evident manner
REQ-BM-LOG-003	Indicate security-relevant failures and state changes

4713

4714 E.3.3 Notification

4715

Table E.3.3-1: Notification

Requirement	Description
REQ-BM-SBD-005	Provide user-visible indication when security is reduced
REQ-BM-AAC-006	Indicate when running with modified policies
REQ-BM-AP-018	Indicate errors with sufficient detail
REQ-BM-LOG-002	Provide measurement records in verifiable format
REQ-BM-LOG-003	Indicate security-relevant failures

4716

4717 E.4 Recovery (NIST SP 800-193, Section 4.4)

4718 E.4.1 Recovery of mutable code (NIST SP 800-193, Section 4.4.1)

4719

Table E.4.1-1: Recovery of mutable code

Requirement	Description
REQ-BM-AP-016	Halt boot, maintain integrity on failure
REQ-BM-AP-017	Enter recovery mode on security violations
REQ-BM-AP-003	Support redundant boot paths
REQ-BM-AP-005	Enforce timeouts for all operations
REQ-BM-AP-011	Require authentication for recovery mode
REQ-BM-AP-015	Protect recovery from unauthorised modification
REQ-BM-AP-018	Maintain recovery accessibility, preserve across updates
REQ-BM-INT-022	Support revocation of compromised keys

4720

4721 E.4.2 Recovery of critical data (NIST SP 800-193, Section 4.4.2)

4722

Table E.4.2-1: Recovery of critical data

Requirement	Description
REQ-BM-SBD-006	Support restoration of secure defaults
REQ-BM-CON-009	Cryptographically erase sensitive data during disposal
REQ-BM-CON-010	Indicate successful sanitization completion
REQ-BM-AP-001	Support fallback to known-good configuration
REQ-BM-AP-012	Handle network boot failures with fallback
REQ-BM-LOG-004	Provide version information accessible to OS

4723

4724 E.5 Firmware categories (NIST SP 800-193)

4725 NIST SP 800-193 [i.6] applies PDR to specific firmware categories. The present document applicability:

4726

Table E.5-1: Firmware categories

SP 800-193 [i.6] Category	The present document Applicability
Platform firmware (BIOS/UEFI)	system firmware
Boot firmware	primary focus
Option ROMs	when boot-relevant
SMM firmware	when boot-relevant
Management controller	boot-related functions only
Network controller	network boot only

4727

4728

4729 Annex K (normative):
 4730 Generic requirements and assessment criteria for the
 4731 use of state of the art cryptography V 0.0.7 (2026-04-
 4732 10)

4733 **Editor's note: This Annex K has not been finalized yet by the cross-vertical Task Force.**

4734 **Editor : This normative Annex is intended to provide generic requirements and assessment criteria for the**
 4735 **use of state -of- the- art cryptography to be used in the context of Vertical Standards for the CRA as**
 4736 **common framework.**

4737 The current approach is intended to allow for two approaches in the requirements

4738 i)ENISA ACM catalogue (i.e. concrete Crypto whitelist)

4739 ii) evidence to be provided for the use of further crypto- appropriateness as State-of-the Art

4740 e.g. by referencing in the required documentation

4741 - publicly available national crypto catalogues or

4742 - publications of use case specific or sector specific crypto-algorithm or catalogues

4743 Case (i) :Annex K seems expected by the EC to provide a reference to an explicit list of state -of-the-art
 4744 algorithms **published by ECCG as an accredited body**

4745 Case (ii) For provision of conformity vertical standards are additionally expected to have clearly explicitly
 4746 list any additional algorithms they use, including references to a published catalogue or standard issued by a
 4747 **recognized body."**

4748 - The Footnotes are Editors Notes only designed for integration into the main text of the Vertical Standards
 4749 as provided for input e.g. for the Terms & Definition, Normative References or Bibliography or as additional
 4750 explanation , not to be integrated in the final version

4751 K.1 State of the Art Cryptography (CRY-SOTA)

4752 K.1.1 Requirement

4753 The default configuration for each security mechanism supported by the product shall use (*one or more*) public
 4754 available cryptographic algorithms¹, which are
 4755 (i) listed in the ECCG Agreed Cryptographic Mechanism (ACM) catalogue[1] classified as [CRY-SOTA-listed] **or**
 4756 (ii) suitable and feasible for the corresponding use case, classified as [CRY-SOTA-unlisted]
 4757
 4758

¹ cryptographic algorithm

-sequence of instructions based on mathematical properties to protect confidentiality, integrity or authenticity against attacks.

Note: Cryptographic algorithms describe cryptographic protocols/schemes/constructors/ atomic primitives such as TLS/ Symmetric Entity Authentication Schemes/AES-128 as part of a CMAC/AES-256

4759 [1] ENISA European Cybersecurity Certification Group: “Agreed Cryptographic Mechanisms, vers 2.0” (ACM)
4760 ²

4761 **NOTE 1:** The use of security mechanism as for example

4762 integrity, authentication, access control, secure communication, secure storage and secure update are
4763 described in the main text of this standard.

4764 **NOTE 2** Supporting evidence options for (ii) [CRY-SOTA unlisted], that an algorithm, which is suitable and
4765 feasible for the respective use case, is listed in the related assessment criteria (K.1.2.1) clause.

4766 K.1.2 Assessment criteria

4767 K.1.2.1 Assessment objective

4768 The purpose of this assessment case is to check whether the implemented algorithms for the default configuration are
4769 appropriately identified as CRY SOTA, (see (K 1.1 (i))
4770

4771 K.1.2.1.1 Assessment preparation

4772 ▪ Preconditions for the assessment : If the product has a default configuration, then the default
4773 configuration shall be used for the assessment. Otherwise, the delivery state configuration shall be
4774 used (i.e. the configuration when the product is made available on the market in accordance with
4775 CRA Annex I part 1(2) (b).

4776 K.1.2.1.2 Assessment activities

4777 ▪ For every security mechanism the list of all used algorithms, shall be documented with evidence to
4778 be provided , if they are identified as CRY- SOTA.

4779 **Supporting Evidence:**

4780 ▪ Documentation includes default configuration description (in accordance with K.1.2.1.2) with
4781 references to algorithm listed in (ACM)
4782

4783 Assignment of verdict:

4784 ▪ The verdict PASS shall be assigned if evidence has been provided.
4785 ▪ The verdict FAIL shall be assigned otherwise
4786

4787 **NOTE 3:** Functional correctness and completeness assessment criteria of the documentation can be
4788 specified in accordance with the capabilities set in the specific Vertical Standards .

4789 K.1.2.2 Assessment objective

4790 The purpose of this assessment case is to check that evidence is provided in the documentation that a
4791 suitable and feasible algorithm has been implemented. (see K1.1 (ii))

² The ACM document (ECCG) is available at the ECCG cryptography page at the below link:

- https://certification.enisa.europa.eu/document/download/a845662b-ae0-484e-9191-890c4cfa7aaa_en?filename=ECCG%20Agreed%20Cryptographic%20Mechanisms%20version%202.pdf

ECCG cryptography page: https://certification.enisa.europa.eu/publications/eucc-guidelines-cryptography_en

4792 K.1.2.2.1 Assessment preparation

4793

4794

4795

4796

4797

- Preconditions for the test: If the product has a default configuration, then the default configuration shall be used for the assessment. Otherwise, the delivery state configuration shall be used (i.e. the configuration when the product is made available on the market in accordance with CRA Annex I part 1(2) (b).

4798 K.1.2.2.2 Assessment activities

4799

4800

For every security mechanism and for every used algorithm, and identified as not included in CRY- SOTA, the documentation shall provide evidence

4801

- that this algorithm is suitable and feasible for the respective use case

4802

4803

Supporting Evidence:

4804

4805

4806

4807

4808

4809

4810

4811

4812

4813

4814

4815

4816

4817

1. Identification of the certain algorithm by reference to publicly available referenced algorithm catalogues or publications issued by a recognized body and stable for the last two years as for example
 - a. published national cryptographic catalogues³ or
 - b. sector specific cryptographic algorithm catalogues⁴ or
 - c. publicly available referenced cryptographic algorithms specified in industry standards, that are feasible and suitable for the use case; or
 - d. other publicly available referenced cryptographic algorithms that are feasible and suitable for the use case e.g. have been submitted to peer reviews, security proofs or security analysis by academics, side-channel analysis, tentative of fault attacks and related documents are publicly available
 - e. required for interoperability with recognized legacy system according to the intend purpose of the product in accordance to CRA Act (2024 /2847) (55) and their Guidance on the application (2.5) or

³ e.g. BSI – BSI TR-02102-1, Cryptographic Mechanisms: Recommendations and Key Lengths , Vers. 2026-01, January 31, 2026

⁴ e.g. EPC342-08 /Version 15.0 /Guidelines on cryptographic algorithms usage and key management - PPSSG / 7 March 2025

In addition to general cryptographic catalogues, certain industry sectors maintain their own specialized algorithm catalogues optimized for specific threat models, hardware and/or software architectures or performance constraints. These vertical use-case specific cryptographic algorithm catalogues may serve as supporting evidence when they explicitly recognize algorithms as suitable for the intended use case, e.g. the *ARM Confidential Compute Architecture (CCA) Security Model, Section 12.3.3 (“Memory Encryption”)*, recommends specific algorithms for memory protection in dedicated hardware environments, including: *QARMA-128 with a 256-bit key and AES-128-XEX with two independent 128-bit keys*

4818 2. Supplemental verification , not sufficient for their own: no entry of known exploitable
 4819 vulnerabilities provided in ENISA “European Vulnerability Database”⁵ and the Single Reporting
 4820 Platform (SRP) established for the CRA⁶

4821 Assignment of verdict:

- 4822 ▪ The verdict PASS shall be assigned if respective evidence has been provided,
- 4823 ▪ The verdict FAIL shall be assigned otherwise.

4824
 4825

4826 **EXAMPLE 1:** As example for a national cryptographic algorithm catalogue can serve

- 4827 ▪ A national published Crypto Catalog by the NCCAs⁷ or a national public available
 4828 crypto algorithm catalogue referenced in the ACM, e.g.
 - 4829 • BSI – BSI TR-02102-1, Cryptographic Mechanisms: Recommendations and
 4830 Key Lengths, Vers. 2026-01, January 31, 2026

4831 **EXAMPLE 2:** As examples for a use case sector specific public available cryptographic algorithm catalogue
 4832 can serve

- 4833 ▪ [a] European Payments Council EPC342-08: "Guidelines on cryptographic algorithms
 4834 usage and key management".
- 4835 ▪ [b] ETSI TS 119 312: "Electronic Signatures and Trust Infrastructures (ESI);
 4836 Cryptographic Suites".
- 4837 ▪ [c] 3GPP TS 33.210: “3rd Generation Partnership Project; Technical Specification
 4838 Group Services and System Aspects; Network Domain Security (NDS); IP network
 4839 layer security
 4840

4841 **EXAMPLE 3:** As examples for a use case Industry specific public available cryptographic algorithm catalogue
 4842 can serve

- 4843 ▪ QARMA-128 with a 256-bit key and AES-128-XEX with two independent 128-bit keys
 4844 used e.g. in ARM Confidential Compute Architecture (CCA) Security Model, Section
 4845 12.3.3
 4846

⁵ European Vulnerability Database established pursuant
 to Article 12(2) of Directive (EU) 2022/2555, <https://euvd.enisa.europa.eu/ENISA>

⁶ SRP = , Single Reporting Platform CRA
<https://www.enisa.europa.eu/topics/product-security-and-certification/single-reporting-platform-srp>

⁷ NCCA =
[National Cybersecurity Certification Authorities](https://www.enisa.europa.eu/topics/product-security-and-certification/national-cybersecurity-certification-authorities)

4847 **NOTE 4** : Further Vertical Standard -specific evidence for the appropriateness of the used
4848 cryptography can be added by the respective Vertical Standard in the main text e.g. *description and*
4849 *formal verification or cryptographic security proof or security analysis of a new algorithm*

4850
4851

4852 **NOTE 5**: Formal verification can use mathematical proofs or rigorous methods to prove an
4853 algorithm's correctness, ensuring it meets its formal specification for all valid inputs, unlike testing
4854 which only samples cases.

4855 K.2 Crypto agility

4856 K.2.1 Requirement

4857 Where a security mechanism supported by the product uses in its the default configuration a cryptographic
4858 algorithm that is

4859 (i) listed in the CRY-SOTA catalogue or

4860 (ii) referenced in a crypto catalogue within the context of K.1.1.(ii) in accordance to the Cyber Resilience Act
4861 (CRA)

4862 and that algorithm is expected to be deprecated within the intended lifetime of the product, the product
4863 shall provide a mechanism for updating the cryptographic algorithm or deprecating its usage.

4864 **EXAMPLE 4** : Hybridization serves as a strategy to mitigate the case of deprecation, For instance , if within
4865 a secure storage mechanism, it can involve combination classical asymmetric cryptographic algorithms with
4866 quantum.resistant algorithm through dual encapsulation. This approach ensures data confidentiality data
4867 over a specific future time horizon , assuming that at least one of the employed algorithm remains
4868 uncompromised during this period

4869 **NOTE 6**: To maintain SOTA for cryptographic algorithm within the intended lifetime of the product concepts
4870 to consider are crypto agility additional to the capability of updating cryptographic algorithms on the
4871 product in accordance to Secure Update and Secure Communication mechanism.

4872 **NOTE 7**: The [ACM] listing consist of wo classes of SOTA algorithms; **Legacy mechanisms** with an expiry
4873 date as defined in ACM, and **Recommended mechanisms** with no set expiry date.

4874 **NOTE 8** : For products that cannot have their cryptographic algorithms updated for example if the
4875 implementation or part uses a hardware-based root of trust, it is important to check if the intended lifetime
4876 of the equipment does not exceed the recommended usage lifetime of their cryptographic algorithms state
4877 . Thereby the implementation of an algorithm can include the specific implementation of their parameters
4878 e.g. their key length.

4879 **NOTE 9**: If a component storing the algorithm or corresponding parameters of a main product is replaced
4880 by a new component, the product is considered as a new product according to the New Legislative
4881 Framework Blue Guide, if the replacement provides a substantial modification to the main product

4882 K.2.2 Assessment criteria

4883 K.2.2.1 Assessment objective

4884 The purpose of this assessment case is (the conceptual assessment) whether the product is prepared to update
4885 cryptographic algorithms for the supported security mechanism.

4886

4887 K.2.2.1.1 Assessment preparation

- 4888 ▪ Preconditions for the test: If the product has a default configuration, then the default configuration
 4889 shall be used for the assessment. Otherwise, the delivery state configuration shall be used (i.e. the
 4890 configuration when the product is made available on the market in accordance with CRA Annex I
 4891 part 1(2) (b).

4892 K.2.2.1.2 Assessment activities

- 4893 ▪ For every used algorithm, the life span of the algorithm is documented, as well its property, in case
 4894 of SOTA and if otherwise available, if the algorithm is considered as legacy or recommended
 4895 algorithm.⁸
 4896 ▪ If the life span of the product exceeds the life span of a legacy algorithm, the algorithm is marked as
 4897 updatable by a recommended algorithm in the documentation.
 4898 ▪ If an algorithm is identified as recommended in the documentation , no further action is required
 4899 respective the update of the specific algorithm.

4900 K.2.2.1.3 Supporting Evidence

- 4901 • Description /documentation of the performed test.
 4902 • All test records of the performed test.

4903 K.2.2.1.4 Assignment of verdict

- 4904 ▪ The verdict PASS shall be assigned if respective evidence has been provided,
 4905 ▪ The verdict FAIL shall be assigned otherwise.
 4906

4907

History

Version	Date	Status
V0.0.12	December 2025	Clean-up done by <i>editHelp!</i> E-mail: mailto:edithelp@etsi.org

4908

⁸ defined in [ACM]