Network Working Group
Request for Comments: 4282
Obsoletes: 2486
Category: Standards Track

B. Aboba
Microsoft
M. Beadles
ENDFORCE
J. Arkko
Ericsson
P. Eronen
Nokia

December 2005

The Network Access Identifier

Status of This Memo

Copyright Notice

Abstract

In order to provide roaming services, it is necessary to have a standardized method for identifying users. This document defines the syntax for the Network Access Identifier (NAI), the user identity submitted by the client during network authentication. "Roaming" may be loosely defined as the ability to use any one of multiple Internet Service Providers (ISPs), while maintaining a formal, customer-vendor relationship with only one. Examples of where roaming capabilities might be required include ISP "confederations" and ISP-provided corporate network access support. This document is a revised version of RFC 2486, which originally defined NAIs. Enhancements include international character set and privacy support, as well as a number of corrections to the original RFC.

Table of Contents

1. Introduction 2
1.1. Terminology 3
1.2. Requirements Language 4
1.3. Purpose 4
2. NAI Definition 4
2.1. Formal Syntax 4
2.2. NAI Length Considerations 6
2.3. Support for Username Privacy 6
2.4. International Character Sets 7
2.5. Compatibility with E-Mail Usernames 8
2.6. Compatibility with DNS 8
2.7. Realm Construction 8
2.8. Examples 10
3. Security Considerations 10
4. IANA Considerations 11
5. References 11
5.1. Normative References 11
5.2. Informative References 12
Appendix A. Changes from RFC 2486 14
Appendix B. Acknowledgements 14

1. Introduction

Considerable interest exists for a set of features that fit within the general category of "roaming capability" for network access, including dialup Internet users, Virtual Private Network (VPN) usage, wireless LAN authentication, and other applications. Interested parties have included the following:

- Regional Internet Service Providers (ISPs) operating within a particular state or province, looking to combine their efforts with those of other regional providers to offer dialup service over a wider area.

- National ISPs wishing to combine their operations with those of one or more ISPs in another nation to offer more comprehensive dialup service in a group of countries or on a continent.

- Wireless LAN hotspots providing service to one or more ISPs.

- Businesses desiring to offer their employees a comprehensive package of dialup services on a global basis. Those services may include Internet access as well as secure access to corporate intranets via a VPN, enabled by tunneling protocols such as the

Point-to-Point Tunneling Protocol (PPTP) [RFC2637], the Layer 2 Forwarding (L2F) protocol [RFC2341], the Layer 2 Tunneling Protocol (L2TP) [RFC2661], and the IPsec tunnel mode [RFC2401].

In order to enhance the interoperability of roaming services, it is necessary to have a standardized method for identifying users. This document defines syntax for the Network Access Identifier (NAI). Examples of implementations that use the NAI, and descriptions of its semantics, can be found in [RFC2194].

This document is a revised version of RFC 2486 [RFC2486], which originally defined NAIs. Differences and enhancements compared to RFC 2486 are listed in Appendix A.

1.1. Terminology

This document frequently uses the following terms:

Network Access Identifier

The Network Access Identifier (NAI) is the user identity submitted by the client during network access authentication. In roaming, the purpose of the NAI is to identify the user as well as to assist in the routing of the authentication request. Please note that the NAI may not necessarily be the same as the user's e-mail address or the user identity submitted in an application layer authentication.

Network Access Server

The Network Access Server (NAS) is the device that clients connect to in order to get access to the network. In PPTP terminology, this is referred to as the PPTP Access Concentrator (PAC), and in L2TP terminology, it is referred to as the L2TP Access Concentrator (LAC). In IEEE 802.11, it is referred to as an Access Point.

Roaming Capability

Roaming capability can be loosely defined as the ability to use any one of multiple Internet Service Providers (ISPs), while maintaining a formal, customer-vendor relationship with only one. Examples of cases where roaming capability might be required include ISP "confederations" and ISP-provided corporate network access support.

Tunneling Service

A tunneling service is any network service enabled by tunneling protocols such as PPTP, L2F, L2TP, and IPsec tunnel mode. One example of a tunneling service is secure access to corporate intranets via a Virtual Private Network (VPN).

1.2. Requirements Language

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

1.3. Purpose

As described in [RFC2194], there are a number of providers offering network access services, and the number of Internet Service Providers involved in roaming consortia is increasing rapidly.

In order to be able to offer roaming capability, one of the requirements is to be able to identify the user's home authentication server. For use in roaming, this function is accomplished via the Network Access Identifier (NAI) submitted by the user to the NAS in the initial network authentication. It is also expected that NASes will use the NAI as part of the process of opening a new tunnel, in order to determine the tunnel endpoint.

2. NAI Definition

2.1. Formal Syntax

The grammar for the NAI is given below, described in Augmented Backus-Naur Form (ABNF) as documented in [RFC4234]. The grammar for the username is based on [RFC0821], and the grammar for the realm is an updated version of [RFC1035].

```
nai         =  username
nai         =/ "@" realm
nai         =/ username "@" realm

username    =  dot-string
dot-string  =  string
dot-string  =/ dot-string "." string
string      =  char
string      =/ string char
char        =  c
char        =/ "\" x
```

```
   c           =  %x21     ; '!'              allowed
                           ; '"'              not allowed
   c           =/ %x23     ; '#'              allowed
   c           =/ %x24     ; '$'              allowed
   c           =/ %x25     ; '%'              allowed
   c           =/ %x26     ; '&'              allowed
   c           =/ %x27     ; '''              allowed
                           ; '(', ')'         not allowed
   c           =/ %x2A     ; '*'              allowed
   c           =/ %x2B     ; '+'              allowed
                           ; ','              not allowed
   c           =/ %x2D     ; '-'              allowed
                           ; '.'              not allowed
   c           =/ %x2F     ; '/'              allowed
   c           =/ %x30-39  ; '0'-'9'          allowed
                           ; ';', ':', '<'    not allowed
   c           =/ %x3D     ; '='              allowed
                           ; '>'              not allowed
   c           =/ %x3F     ; '?'              allowed
                           ; '@'              not allowed
   c           =/ %x41-5a  ; 'A'-'Z'          allowed
                           ; '[', '\', ']'    not allowed
   c           =/ %x5E     ; '^'              allowed
   c           =/ %x5F     ; '_'              allowed
   c           =/ %x60     ; '`'              allowed
   c           =/ %x61-7A  ; 'a'-'z'          allowed
   c           =/ %x7B     ; '{'              allowed
   c           =/ %x7C     ; '|'              allowed
   c           =/ %x7D     ; '}'              allowed
   c           =/ %x7E     ; '~'              allowed
                           ; DEL              not allowed
   c           =/ %x80-FF  ; UTF-8-Octet      allowed (not in RFC 2486)
                           ; Where UTF-8-octet is any octet in the
                           ; multi-octet UTF-8 representation of a
                           ; unicode codepoint above %x7F.
                           ; Note that c must also satisfy rules in
                           ; Section 2.4, including, for instance,
                           ; checking that no prohibited output is
                           ; used (see also Section 2.3 of
                           ; [RFC4013]).
   x           =  %x00-FF  ; all 128 ASCII characters, no exception;
                           ; as well as all UTF-8-octets as defined
                           ; above (this was not allowed in
                           ; RFC 2486).  Note that x must nevertheless
                           ; again satisfy the Section 2.4 rules.

   realm       =  1*( label "." ) label
   label       =  let-dig *(ldh-str)
```

```
ldh-str     =  *( alpha / digit / "-" ) let-dig
let-dig     =  alpha / digit
alpha       =  %x41-5A  ; 'A'-'Z'
alpha       =/ %x61-7A  ; 'a'-'z'
digit       =  %x30-39  ; '0'-'9'
```

2.2. NAI Length Considerations

Devices handling NAIs MUST support an NAI length of at least 72 octets. Support for an NAI length of 253 octets is RECOMMENDED. However, the following implementation issues should be considered:

- NAIs are often transported in the User-Name attribute of the Remote Authentication Dial-In User Service (RADIUS) protocol. Unfortunately, RFC 2865 [RFC2865], Section 5.1, states that "the ability to handle at least 63 octets is recommended." As a result, it may not be possible to transfer NAIs beyond 63 octets through all devices. In addition, since only a single User-Name attribute may be included in a RADIUS message and the maximum attribute length is 253 octets; RADIUS is unable to support NAI lengths beyond 253 octets.

- NAIs can also be transported in the User-Name attribute of Diameter [RFC3588], which supports content lengths up to 2^24 - 9 octets. As a result, NAIs processed only by Diameter nodes can be very long. Unfortunately, an NAI transported over Diameter may eventually be translated to RADIUS, in which case the above limitations apply.

2.3. Support for Username Privacy

Interpretation of the username part of the NAI depends on the realm in question. Therefore, the "username" part SHOULD be treated as opaque data when processed by nodes that are not a part of the authoritative domain (in the sense of Section 4) for that realm.

In some situations, NAIs are used together with a separate authentication method that can transfer the username part in a more secure manner to increase privacy. In this case, NAIs MAY be provided in an abbreviated form by omitting the username part. Omitting the username part is RECOMMENDED over using a fixed username part, such as "anonymous", since it provides an unambiguous way to determine whether the username is intended to uniquely identify a single user.

For roaming purposes, it is typically necessary to locate the appropriate backend authentication server for the given NAI before the authentication conversation can proceed. As a result, the realm

portion is typically required in order for the authentication exchange to be routed to the appropriate server.

2.4. International Character Sets

This specification allows both international usernames and realms. International usernames are based on the use of Unicode characters, encoded as UTF-8 and processed with a certain algorithm to ensure a canonical representation. Internationalization of the realm portion of the NAI is based on "Internationalizing Domain Names in Applications (IDNA)" [RFC3490].

In order to ensure a canonical representation, characters of the username portion in an NAI MUST fulfill the ABNF in this specification as well as the requirements specified in [RFC4013]. These requirements consist of the following:

- Mapping requirements, as specified in Section 2.1 of [RFC4013]. Mapping consists of mapping certain characters to others (such as SPACE) in order to increase the likelihood of correctly performed comparisons.

- Normalization requirements, as specified in Section 2.2 of [RFC4013], are also designed to assist in comparisons.

- Prohibited output. Certain characters are not permitted in correctly formed strings that follow Section 2.3 of [RFC4013]. Ensuring that NAIs conform to their ABNF is not sufficient; it is also necessary to ensure that they do not contain prohibited output.

- Bidirectional characters are handled as specified in Section 2.4 of [RFC4013].

- Unassigned code points are specified in Section 2.5 of [RFC4013]. The use of unassigned code points is prohibited.

The mapping, normalization, and bidirectional character processing MUST be performed by end systems that take international text as input. In a network access setting, such systems are typically the client and the Authentication, Authorization, and Accounting (AAA) server. NAIs are sent over the wire in their canonical form, and tasks such as normalization do not typically need to be performed by nodes that just pass NAIs around or receive them from the network. End systems MUST also perform checking for prohibited output and unassigned code points. Other systems MAY perform such checks, when they know that a particular data item is an NAI.

The realm name is an "IDN-unaware domain name slot" as defined in [RFC3490]. That is, it can contain only ASCII characters. An implementation MAY support Internationalized Domain Names (IDNs) using the ToASCII operation; see [RFC3490] for more information.

The responsibility for the conversion of internationalized domain names to ASCII is left for the end systems, such as network access clients and AAA servers. Similarly, we expect domain name comparisons, matching, resolution, and AAA routing to be performed on the ASCII versions of the internationalized domain names. This provides a canonical representation, ensures that intermediate systems such as AAA proxies do not need to perform translations, and can be expected to work through systems that are unaware of international character sets.

2.5. Compatibility with E-Mail Usernames

As proposed in this document, the Network Access Identifier is of the form user@realm. Please note that while the user portion of the NAI is based on the BNF described in [RFC0821], it has been extended for internationalization support as well as for purposes of Section 2.7, and is not necessarily compatible with the usernames used in e-mail. Note also that the internationalization requirements for NAIs and e-mail addresses are different, since the former need to be typed in only by the user himself and his own operator, not by others.

2.6. Compatibility with DNS

The BNF of the realm portion allows the realm to begin with a digit, which is not permitted by the BNF described in [RFC1035]. This change was made to reflect current practice; although not permitted by the BNF described in [RFC1035], Fully Qualified Domain Names (FQDNs) such as 3com.com are commonly used and accepted by current software.

2.7. Realm Construction

NAIs are used, among other purposes, for routing AAA transactions to the user's home realm. Usually, the home realm appears in the realm portion of the NAI, but in some cases a different realm can be used. This may be useful, for instance, when the home realm is reachable only via another mediating realm.

Such usage may prevent interoperability unless the parties involved have a mutual agreement that the usage is allowed. In particular, NAIs MUST NOT use a different realm than the home realm unless the sender has explicit knowledge that (a) the specified other realm is available and (b) the other realm supports such usage. The sender

may determine the fulfillment of these conditions through a database, dynamic discovery, or other means not specified here. Note that the first condition is affected by roaming, as the availability of the other realm may depend on the user's location or the desired application.

The use of the home realm MUST be the default unless otherwise configured.

Where these conditions are fulfilled, an NAI such as

    user@homerealm.example.net

MAY be represented as in

    homerealm.example.net!user@otherrealm.example.net

In this case, the part before the (non-escaped) '!' MUST be a realm name as defined in the ABNF in Section 2.1. This realm name is an "IDN-unaware domain name slot", just like the realm name after the "@" character; see Section 2.4 for details. When receiving such an NAI, the other realm MUST convert the format back to "user@homerealm.example.net" when passing the NAI forward, as well as applying appropriate AAA routing for the transaction.

The conversion process may apply also recursively. That is, after the conversion, the result may still have one or more '!' characters in the username. For instance, the NAI

    other2.example.net!home.example.net!user@other1.example.net

would first be converted in other1.example.net to

    home.example.net!user@other2.example.net

and then at other2.example.net finally to

    user@homerealm.example.net

Note that the syntax described in this section is optional and is not a part of the ABNF. The '!' character may appear in the username portion of an NAI for other purposes as well, and in those cases, the rules outlined here do not apply; the interpretation of the username is up to an agreement between the identified user and the realm given after the '@' character.

2.8. Examples

Examples of valid Network Access Identifiers include the following:

```
bob
joe@example.com
fred@foo-9.example.com
jack@3rd.depts.example.com
fred.smith@example.com
fred_smith@example.com
fred$@example.com
fred=?#$&*+-/^smith@example.com
nancy@eng.example.net
eng.example.net!nancy@example.net
eng%nancy@example.net
@privatecorp.example.net
\(user\)@example.net
alice@xn--tmonesimerkki-bfbb.example.net
```

The last example uses an IDN converted into an ASCII representation.

Examples of invalid Network Access Identifiers include the following:

```
fred@example
fred@example_9.com
fred@example.net@example.net
fred.@example.net
eng:nancy@example.net
eng;nancy@example.net
(user)@example.net
<nancy>@example.net
```

3. Security Considerations

Since an NAI reveals the home affiliation of a user, it may assist an attacker in further probing the username space. Typically, this problem is of most concern in protocols that transmit the username in clear-text across the Internet, such as in RADIUS, described in [RFC2865] and [RFC2866]. In order to prevent snooping of the username, protocols may use confidentiality services provided by protocols transporting them, such as RADIUS protected by IPsec [RFC3579] or Diameter protected by TLS [RFC3588].

This specification adds the possibility of hiding the username part in the NAI, by omitting it. As discussed in Section 2.3, this is possible only when NAIs are used together with a separate authentication method that can transfer the username in a secure manner. In some cases, application-specific privacy mechanism have

also been used with NAIs. For instance, some Extensible Authentication Protocol (EAP) methods apply method-specific pseudonyms in the username part of the NAI [RFC3748]. While neither of these approaches can protect the realm part, their advantage over transport protection is that privacy of the username is protected, even through intermediate nodes such as NASes.

4. IANA Considerations

In order to avoid creating any new administrative procedures, administration of the NAI realm namespace piggybacks on the administration of the DNS namespace.

NAI realm names are required to be unique, and the rights to use a given NAI realm for roaming purposes are obtained coincident with acquiring the rights to use a particular Fully Qualified Domain Name (FQDN). Those wishing to use an NAI realm name should first acquire the rights to use the corresponding FQDN. Using an NAI realm without ownership of the corresponding FQDN creates the possibility of conflict and therefore is to be discouraged.

Note that the use of an FQDN as the realm name does not require use of the DNS for location of the authentication server. While Diameter [RFC3588] supports the use of DNS for location of authentication servers, existing RADIUS implementations typically use proxy configuration files in order to locate authentication servers within a domain and perform authentication routing. The implementations described in [RFC2194] did not use DNS for location of the authentication server within a domain. Similarly, existing implementations have not found a need for dynamic routing protocols or propagation of global routing information. Note also that there is no requirement that the NAI represent a valid email address.

5. References

5.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

5.2. Informative References

[RFC0821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[RFC2194] Aboba, B., Lu, J., Alsop, J., Ding, J., and W. Wang, "Review of Roaming Implementations", RFC 2194, September 1997.

[RFC2341] Valencia, A., Littlewood, M., and T. Kolar, "Cisco Layer Two Forwarding (Protocol) "L2F"", RFC 2341, May 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2486] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

[RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and G. Zorn, "Point-to-Point Tunneling Protocol", RFC 2637, July 1999.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[RFC3579] Aboba, B. and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)", RFC 3579, September 2003.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[netsel-problem] Arkko, J. and B. Aboba, "Network Discovery and Selection Problem", Work in Progress, October 2005.

Appendix A. Changes from RFC 2486

This document contains the following updates with respect to the original NAI definition in RFC 2486 [RFC2486]:

- International character set support has been added for both usernames and realms. Note that this implies character codes 128 - 255 may be used in the username portion, which may be unacceptable to nodes that only support RFC 2486. Many devices already allow this behaviour, however.

- Username privacy support has been added. Note that NAIs without a username (for privacy) may not be acceptable to RFC 2486-compliant nodes. Many devices already allow this behaviour, however.

- A recommendation to support NAI length of at least 253 octets has been added, and compatibility considerations among NAI lengths in this specification and various AAA protocols are discussed. Note that long NAIs may not be acceptable to RFC 2486-compliant nodes.

- The mediating network syntax and its implications have been fully described and not given only as an example. Note that this syntax is not intended to be a full solution to network discovery and selection needs as defined in [netsel-problem]. Rather, it is intended as a clarification of RFC 2486.

  However, as discussed in Section 2.7, this specification requires that this syntax be applied only when there is explicit knowledge that the peer system supports such syntax.

- The realm BNF entry definition has been changed to avoid an error (infinite recursion) in the original specification.

- Several clarifications and improvements have been incorporated into the ABNF specification for NAIs.

Appendix B. Acknowledgements

Thanks to Glen Zorn for many useful discussions of this problem space, and to Farid Adrangi for suggesting the representation of mediating networks in NAIs. Jonathan Rosenberg reported the BNF error. Dale Worley suggested clarifications of the x and special BNF entries. Arne Norefors reported the length differences between RFC 2486 and RFC 2865. Paul Hoffman helped with the international character set issues. Kalle Tammela, Stefaan De Cnodder, Nagi Jonnala, Bert Wijnen, Blair Bullock, Yoshihiro Ohba, Ignacio Goyret, John Loughney, Henrik Levkowetz, Ted Hardie, Bill Fenner, Sam Hartman, and Richard Perlman provided many useful comments on this

document. The ABNF validator at http://www.apps.ietf.org/abnf.html was used to verify the syntactic correctness of the ABNF in Section 2.1.

Authors' Addresses

Bernard Aboba
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

EMail: bernarda@microsoft.com

Mark A. Beadles
ENDFORCE
565 Metro Place South Suite 300
Dublin OH 43017
USA

EMail: mbeadles@endforce.com

Jari Arkko
Ericsson
Jorvas 02420
Finland

EMail: jari.arkko@ericsson.com

Pasi Eronen
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland

EMail: pasi.eronen@nokia.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.