Network Working Group
INTERNET-DRAFT
Obsoletes: 4282
Category: Standards Track
<draft-ietf-radext-nai-03.txt>
23 May 2013

DeKok, Alan
FreeRADIUS

The Network Access Identifier
draft-ietf-radext-nai-03

Abstract

In order to provide inter-domain authentication services, it is necessary to have a standardized method that domains can use to identify each others users. This document defines the syntax for the Network Access Identifier (NAI), the user identity submitted by the client prior to accessing network resources. This document is a revised version of RFC 4282 [RFC4282], which addresses issues with international character sets, as well as a number of other corrections to the previous document.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 8, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the

document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

Appendix A - Changes from RFC4282 3
1. Introduction 4
1.1. Terminology 5
1.2. Requirements Language 6
1.3. Purpose 7
1.4. Motivation 7
2. NAI Definition 8
2.1. UTF-8 Syntax and Normalization 8
2.2. Formal Syntax 9
2.3. NAI Length Considerations 9
2.4. Support for Username Privacy 10
2.5. International Character Sets 11
2.6. The Normalization Process 11
2.7. Use in Other Protocols 12
2.8. Routing inside of AAA Systems 13
2.9. Compatibility with Email Usernames 14
2.10. Compatibility with DNS 14
2.11. Realm Construction 15
2.11.1. Historical Practices 16
2.12. Examples 17
3. Security Considerations 17
4. IANA Considerations 18
5. References 18
5.1. Normative References 18
5.2. Informative References 19
Appendix A - Changes from RFC4282 21

1. Introduction

Considerable interest exists for a set of features that fit within the general category of inter-domain authentiction, or "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 [RFC4301].

* Other protocols which are interested in leveraging the users credentials in order to take advantage of an existing authentication framework.

In order to enhance the interoperability of these 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].

When the NAI was defined for network access, it had the side effect of defining an identifier which could be used elsewhere. Some systems which required the use of an identifier did so by leveraging the NAI. This process simplified the management of credentials, by re-using the same credential in multiple situations. We suggest that this re-use is good practice. The alternative is to have protocol-specific identifiers, which increases cost to both user and administrator.

The goal of this document is to define the format of an identifier which can be used in many protocols. A protocol may transport an

encoded version of the NAI (e.g. '.' as %2E). However, the definition of the NAI is protocol independent. We hope to encourage the wide-spread adoption of the NAI as an identifier. This adoption will decrease work required to leverage identification and authentication in other protocols. It will also decrease the complexity of systems for end users and administrators.

We note that this document only suggest that the NAI be used, but does not require it. Many protocols already define their own identifier formats. Some of these are incompatible with the NAI, while others allow the NAI in addition to non-NAI identifiers. This definition of the NAI has no requirements on protocol specifications, implementations, or deployments. We simply suggest that using a standard identifier format is preferable to using multiple incompatible identifier formats.

This document is a revised version of [RFC4282], which originally defined internationalized NAIs. Differences and enhancements compared to that document are listed in Appendix A.

1.1. Terminology

This document frequently uses the following terms:

"Local" or "localized" text

Text which is either in non-UTF-8, or in non-normalized form. The character set, encoding, and locale are (in general) unknown to Authentication, Authorization, and Accounting (AAA) network protocols. The client which "knows" the locale may have a different concept of this text than other AAA entities, which do not know the same locale.

Network Access Identifier

The Network Access Identifier (NAI) is the user identity submitted by the client during network access authentication. 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 email 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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document 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.

We also hope that other protocols can take advantage of the NAI. Many protocols include authentication capabilities. The authentication credentials supplied in those protocols can end up being transported in AAA protocols. It is therefore useful to define a representation of the user credentials which can be shared across multiple protocols.

1.4. Motivation

The changes from [RFC4282] are listed in detail in Appendix A. However, some additional discussion is appropriate to motivate those changes.

The motivation to revise [RFC4282] began with internationalization concerns raised in the context of [EDUROAM]. Section 2.1 of [RFC4282] defines ABNF for realms which limits the realm grammar to English letters, digits, and the hyphen "-" character. The intent appears to have been to encode, compare, and transport realms with the ToASCII operation defined in [RFC5890]. There are a number of problems with this approach:

* The [RFC4282] ABNF is not aligned with internationalization of DNS.

* The requirement in Section 2.1 that realms are ASCII conflicts with the Extensible Authentication Protocol (EAP) and RADIUS, which are both 8-bit clean, and which both recommend the use of UTF-8 for identitifiers.

* Section 2.4 required mappings that are language-specific, and which are nearly impossible for intermediate nodes to perform correctly without information about that language.

* Section 2.4 requires normalization of user names, which may conflict with local system or administrative requirements.

* The recommendations in Section 2.4 for treatment of bidirectional characters have proven to be unworkable.

* The prohibition against use of unassigned code points in Section 2.4 effectively prohibits support for new scripts.

* No Authentication, Authorization, and Accounting (AAA) client, proxy, or server has implemented any of the requirements in [RFC4282] Section 2.4, among other sections.

With international roaming growing in popularity, it is important for these issues to be corrected in order to provide robust and inter-operable network services.

2. NAI Definition

2.1. UTF-8 Syntax and Normalization

UTF-8 characters can be defined in terms of octets using the following ABNF [RFC5234], taken from [RFC3629]:

```
UTF8-xtra-char  =   UTF8-2 / UTF8-3 / UTF8-4

UTF8-2          =   %xC2-DF UTF8-tail

UTF8-3          =   %xE0 %xA0-BF UTF8-tail /
                    %xE1-EC 2(UTF8-tail) /
                    %xED %x80-9F UTF8-tail /
                    %xEE-EF 2(UTF8-tail)

UTF8-4          =   %xF0 %x90-BF 2( UTF8-tail ) /
                    %xF1-F3 3( UTF8-tail ) /
                    %xF4 %x80-8F 2( UTF8-tail )

UTF8-tail       =   %x80-BF
```

These are normatively defined in [RFC3629], but are repeated in this document for reasons of convenience.

See [RFC5198] and section 2.6 of this specification for a discussion of normalization. Strings which are not in Normal Form Composed (NFC) are not valid NAIs and SHOULD NOT be treated as such. Implementations which expect to receive a NAI, but get non-normalised (but otherwise valid) UTF-8 strings instead SHOULD attempt to create a local version of the NAI, which is normalized from the input identifier. This local version can then be used for local processing.

Systems MAY accept user identifiers in forms other than the NAI. This specification does not forbid that practice. It only codifies the format and interpretation of the NAI. Where protocols carry identifiers which are expected to be transported over an AAA protocol, it is RECOMMENDED that the identifiers be in NAI format.

2.2. Formal Syntax

The grammar for the NAI is given below, described in Augmented Backus-Naur Form (ABNF) as documented in [RFC5234].

```
nai             =   utf8-username
nai             =/  "@" utf8-realm
nai             =/  utf8-username "@" utf8-realm

utf8-username   =  dot-string
dot-string      =  string
dot-string      =/ dot-string "." string
string          =  utf8-atext
string          =/ string utf8-atext

utf8-atext      =  ALPHA / DIGIT /
                   "!" / "#" /
                   "$" / "%" /
                   "&" / "'" /
                   "*" / "+" /
                   "-" / "/" /
                   "=" / "?" /
                   "^" / "_" /
                   "`" / "{" /
                   "|" / "}" /
                   "~" /
                   UTF8-xtra-char

utf8-realm      =  1*( label "." ) label

label           =  utf8-rtext *(ldh-str)
ldh-str         =  *( utf8-rtext / "-" ) utf8-rtext
utf8-rtext      =  ALPHA / DIGIT / UTF8-xtra-char
```

2.3. NAI Length Considerations

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

* NAI octet length constraints may impose a more severe constraint on the number of UTF-8 characters.

* 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. However, an NAI transported over Diameter may eventually be translated to RADIUS, in which case the above limitations will apply.

* NAIs may be transported in other protocols. Each protocol can have its own limitations on maximum NAI length.

The above criteria should permit the widest use, and widest possible inter-operability of the NAI.

2.4. Support for Username Privacy

Interpretation of the username part of the NAI depends on the realm in question. Therefore, the utf8-username portion 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. However, current practice is to use the username "anonymous" instead of omitting the username part. This behavior is also permitted.

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.

2.5. 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. Internationalization of the realm portion of the NAI is based on "Internationalized Email Headers" [RFC5335].

In order to ensure a canonical representation, characters of the username portion in an NAI MUST match the ABNF in this specification as well as the requirements specified in [RFC5891]. In practice, these requirements consist of the following item:

* Realms MUST be of the form that can be registered as a Fully Qualified Domain Name (FQDN) within the DNS.

This list is significantly shorter and simpler than the list in Section 2.4 of [RFC4282]. The form suggested in [RFC4282] depended on intermediate nodes performing canonicalizations based on insufficient information, which meant that the form was not canonical. This document instead suggests (Section 2.10) that the realm owner provide a canonical form of the realm, and that all intermediate nodes use that form without modification.

Specifying the realm requirement as above means that the requirements depend on specifications that are referenced here, rather than copied here. This allows the realm definition to be updated when the referenced documents change, without requiring a revision of this specification.

In general, the above requirement means following the requirements specified in [RFC5891].

2.6. The Normalization Process

Conversion to Unicode as well as normalization is expected to be performed by end systems that take "local" text as input. Other AAA systems such as proxies do not have access to locale and character set information that is available to end systems. Therefore, they are typically incapable of converting local input to Unicode.

That is, all processing of NAIs from "local" character sets and locales to UTF-8 is performed by edge systems, prior to the NAIs entering the AAA system. Inside of an AAA system, NAIs are sent over the wire in their canonical form, and this canonical form is used for all NAI and/or realm comparisons.

Copying of localized text into fields that can subsequently be placed into the RADIUS User-Name attribute is problematic. This practice

can result in a AAA proxy encountering non-UTF8 characters within what it expects to be an NAI. An example of this requirement is [RFC3579] Section 2.1, which states:

> the NAS MUST copy the contents of the Type-Data field of the EAP-Response/Identity received from the peer into the User-Name attribute

As a result, AAA proxies expect the contents of the EAP-Response/Identity sent by an EAP supplicant to consist of UTF-8 characters, not localized text. Using localized text in AAA username or identity fields means that realm routing becomes difficult or impossible.

In contrast to [RFC4282] Section 2.4, we expect AAA systems to perform NAI comparisons, matching, and AAA routing based on the NAI as it is received. This specification 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.

For example, much of the common realm routing can be done on the "utf8-realm" portion of NAI, through simple checks for equality. This routing can be done even if the AAA proxy is unaware of internalized domain names. All that is required is for the AAA proxy to be able to enter, store, and compare 8-bit data.

2.7. Use in Other Protocols

As noted earlier, the NAI MAY be used in other, non-AAA protocols. It is RECOMMENDED that the definition given here be used unchanged. Using other definitions for user identifiers may hinder interoperability, along with the users ability to authenticate successfully. It is RECOMMENDED that protocols requiring the use of a user identifier reference this specification, and suggest that the use of an NAI is RECOMMENDED.

We cannot require other protocols to use the NAI for user identifiers. Their needs are unknown, and unknowable. We simply suggest that interoperability and inter-domain authentication is useful, and should be encouraged.

Where a protocol is 8-bit clean, it can likely transport the NAI as-is, without further modification.

Where a protocol is not 8-bit clean, it MUST NOT affect the definition or handling of the NAI. That is, if a protocol escapes the '.' character as "%2E", then the protocol may have an identifier

transported as "fred@example%2Ecom", whereas the NAI for that user is "fred@example.com". Any comparison, validation, or use of the NAI MUST be done on its un-escaped (i.e. utf8-clean) form.

2.8. Routing inside of AAA Systems

Many AAA systems use the "utf8-realm" portion of the NAI to route requests within a AAA proxy network. The semantics of this operation involves a logical AAA routing table, where the "utf8-realm" portion acts as a key, and the values stored in the table are one or more "next hop" AAA servers.

Intermediate nodes MUST use the "utf8-realm" portion of the NAI without modification to perform this lookup. Comparisons between the NAI as given in a AAA packet, and as provisioned in a logical AAA routing table SHOULD be done as a byte-for-byte equality test. As noted earlier, intermediate nodes may not have access to the same locale information as the system which injected the NAI into the AAA routing systems. Therefore, almost all "case insensitive" comparisons will be wrong. Where the "utf8-realm" is entirely ASCII, current systems sometimes perform case-insensitive matching on realms. This practice MAY be continued, as it has been shown to work in practice.

We also note that many existing systems use user identifiers which are similar in format to the NAI, but which are not compliant with this specification. For example, they may use non-NFC form, or they may have multiple "@" characters in the user identifier. Intermediate nodes MAY normalize non-NFC identifiers to NFC, prior to looking up the "utf8-realm" in the logical routing table. Intermediate nodes MUST NOT modify the identifiers that they forward. The data as entered by the user is inviolate.

The "utf8-realm" provisioned in the logical AAA routing table SHOULD be provisioned to the proxy prior to it receiving any AAA traffic. The "utf8-realm" SHOULD be supplied by the "next hop" or "home" system that also supplies the routing information necessary for packets to reach the next hop.

This "next hop" information may be any of, or all of, the following information: IP address; port; RADIUS shared secret; TLS certificate; DNS host name; or instruction to use dyanmic DNS discovery (i.e. look up a record in the "utf8-realm" domain). This list is not exhaustive, and my be extended by future specifications.

It is RECOMMENDED to use the entirety of the "utf8-realm" for the routing decisions. However, systems MAY use a portion of the "utf8-realm" portion, so long as that portion is a valid

"utf8-realm", and that portion is handled as above. For example, routing "fred@example.com" to a "com" destination is forbidden, because "com" is not a valid "utf8-realm". However, routing "fred@sales.example.com" to the "example.com" destination is permissible.

Another reason to forbid the use of a single label (e.g. "fred@sales") is that many systems treat a single label as being a local identifier within their realm. That is, a user logging in as "fred@sales" to a domain "example.com", would be treated as if the NAI was instead "fred@sales.example.com". Permitting the use of a single label would mean changing the interpretation and meaning of a single label, which cannot be done.

2.9. Compatibility with Email 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 [RFC5198], it has been modified for the purposes of Section 2.2. It does not permit quoted text along with "folding" or "non-folding" whitespace that is commonly used in email addresses. As such, the NAI is not necessarily equivalent to usernames used in e-mail.

However, it is a common practice to use email addresses as user identifiers in AAA systems. The ABNF in Section 2.2 is defined to be close to the "utf8-addr-spec" portion of [RFC5335], while still being compatible with [RFC4282].

In contrast to [RFC4282] Section 2.5, we state that the internationalization requirements for NAIs and email addresses are substantially similar. The NAI and email identifiers may be the same, and both need to be entered by the user and/or the operator supplying network access to that user. There is therefore good reason for the internationalization requirements to be similar.

2.10. Compatibility with DNS

The "utf8-realm" portion of the NAI is intended to be compatible with Internationalized Domain Names (IDNs) [RFC5890]. As defined above, the "utf8-realm" portion as transported within an 8-bit clean protocol such as RADIUS and EAP can contain any valid UTF8 character. There is therefore no reason for a NAS to apply the ToAscii function to the "utf8-realm" portion of an NAI, prior to placing the NAI into a RADIUS User-Name attribute. Unlike DNS, the NAI does not make a distinction between A-labels and U-labels. Is instead an IDNA-valid label, as per Section 2.3.2.1 of [RFC5890].

When the realm portion of the NAI is used as the basis for name resolution, it may be necessary to convert internationalized realm names to ASCII using the ToASCII operation defined in [RFC5890]. As noted in [RFC6055] Section 2, resolver Application Programming Interfaces (APIs) are not necessarily DNS-specific, so that the ToASCII operation needs to be applied carefully:

Applications which convert an IDN to A-label form before calling (for example) getaddrinfo() will result in name resolution failures if the Punycode name is directly used in such protocols. Having libraries or protocols to convert from A-labels to the encoding scheme defined by the protocol (e.g., UTF-8) would require changes to APIs and/or servers, which IDNA was intended to avoid.

As a result, applications SHOULD NOT assume that non-ASCII names are resolved using the public DNS and blindly convert them to A-labels without knowledge of what protocol will be selected by the name resolution library.

There is, however, a problem with this approach. A AAA proxy may not have sufficient information in order to perform the ToAscii conversion properly. We therefore RECOMMEND that only the owner of the realm perform the ToAscii conversion. We RECOMMEND that the owner of the realm pre-provision all proxies with the "utf8-realm" portion of the NAI, along with the value returned from passing the "utf8-realm" to the ToAscii function. This key-value pair can then be placed into the logical AAA routing table discussed above. Having only one entity perform the ToAscii function ensures that the result returned by that function are considered as canonical by all other participants in a AAA network.

The paragraph above does not negate all of the benefits of using DNS to automatically discover the location of a "next hop" AAA server. Many AAA proxies require a business or legal relationship prior to routing any traffic. This relationship can be leveraged to bootstrap the DNS information located in the logical AAA routing table.

2.11. Realm Construction

The home realm usually appears in the "utf8-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 intermediate proxies.

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.

2.11.1. Historical Practices

Some systems have historically used NAI modifications with multiple "prefix" and "suffix" decorations to perform explicit routing through multiple proxies inside of a AAA network. This practice is NOT RECOMMENDED for the following reasons:

* Using explicit routing paths is fragile, and is unresponsive to changes in the network due to servers going up or down, or to changing business relationships.

* There is no RADIUS routing protocol, meaning that routing paths have to be communicated "out of band" to all intermediate AAA nodes, and also to all end-user systems (supplicants) expecting to obtain network access.

* Using explicit routing paths requires thousands, if not millions of end-user systems to be updated with new path information when a AAA routing path changes. This adds huge expense for updates that would be better done at only a few AAA systems in the network.

* Manual updates to RADIUS paths are expensive, time-consuming, and prone to error.

* Creating compatible formats for the NAI is difficult when locally-defined "prefixes" and "suffixes" conflict with similar practices elsewhere in the network. These conflicts mean that connecting two networks may be impossible in some cases, as there is no way for packets to be routed properly in a way that meets all requirements at all intermediate proxies.

* Leveraging the DNS name system for realm names establishes a globally unique name space for realms.

In summary, network practices and capabilities have changed significantly since NAIs were first overloaded to define AAA routes through a network. While explicit path routing was once useful, the

time has come for better methods to be used.

2.12. 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
```

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
```

One example given in [RFC4282] is still permitted by the ABNF, but it is NOT RECOMMMENDED because of the use of the ToAscii function to create an ASCII encoding from what is now a valid UTF-8 string.

```
alice@xn--tmonesimerkki-bfbb.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 2.4, 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 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 is therefore 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

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

[RFC3629]
Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5198]
Klensin J., and Padlipsky M., "Unicode Format for Network Interchange", RFC 5198, March 2008

[RFC5234]
Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

[RFC5890]
Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 5890

5.2. Informative References

[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.

[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.

[RFC4282]
Aboba, B. et al., "The Network Access Identifier", RFC 4282, December 2005.

[RFC4301]
Kent, S. and S. Keo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5335]
Y. Abel, Ed., "Internationalized Email Headers", RFC 5335, September 2008.

[EDUROAM]
http://eduroam.org, "eduroam (EDUcational ROAMing)"

[RFC5891]
Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891

[RFC6055]
Thaler, D., et al, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, February 2011.

Acknowledgments

The initial text for this document was [RFC4282], which was then heavily edited. The original authors of [RFC4282] were Bernard Aboba, Mark A. Beadles, Jari Arkko, and Pasi Eronen.

The ABNF validator at http://www.apps.ietf.org/abnf.html was used to verify the syntactic correctness of the ABNF in Section 2.

Appendix A - Changes from RFC4282

This document contains the following updates with respect to the previous NAI definition in RFC 4282 [RFC4282]:

* The formal syntax in Section 2.1 has been updated to forbid non-UTF8 characters. e.g. characters with the "high bit" set.

* The formal syntax in Section 2.1 has been updated to allow UTF-8 in the "realm" portion of the NAI.

* The formal syntax in [RFC4282] Section 2.1 applied to the NAI after it was "internationalized" via the ToAscii function. The contents of the NAI before it was "internationalized" were left indeterminate. This document updates the formal syntax to define an internationalized form of the NAI, and forbids the use of the ToAscii function for NAI "internationalization".

* The grammar for the user and realm portion is based on a combination of the "nai" defined in [RFC4282] Section 2.1, and the "utf8-addr-spec" defined in [RFC5335] Section 4.4.

* All use of the ToAscii function has been moved to normal requirements on DNS implementations when realms are used as the basis for DNS lookups. This involves no changes to the existing DNS infrastructure.

* The discussions on internationalized character sets in Section 2.4 have been updated. The suggestion to use the ToAscii function for realm comparisons has been removed. No AAA system has implemented these suggestions, so this change should have no operational impact.

* The section "Routing inside of AAA Systems" section is new in this document. The concept of a "local AAA routing table" is also new, although it accurately describes the functionality of wide-spread implementations.

* The "Compatibility with EMail Usernames" and "Compatibility with DNS" sections have been revised and updated. We now note that the ToAscii function is suggested to be used only when a realm name is used for DNS lookups, and even then the function is only used by a resolving API on the local system, and even then we recommend that only the home network perform this conversion.

* The "Realm Construction" section has been updated to note that editing of the NAI is NOT RECOMMENDED.

* The "Examples" section has been updated to remove the instance of the IDN being converted to ASCII. This behavior is now forbidden.

Authors' Addresses

Alan DeKok
The FreeRADIUS Server Project

Email: aland@freeradius.org