RADEXT Working Group
INTERNET-DRAFT
Obsoletes: 4282
Category: Standards Track
<draft-ietf-radext-nai-15.txt>
17 December 2014

DeKok, Alan
FreeRADIUS

The Network Access Identifier
draft-ietf-radext-nai-15

Abstract

In order to provide inter-domain authentication services, it is necessary to have a standardized method that domains can use to identify each other’s users. This document defines the syntax for the Network Access Identifier (NAI), the user identifier submitted by the client prior to accessing resources. This document is a revised version of RFC 4282, 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 June 17, 2015.

Copyright Notice

Table of Contents

1. Introduction

Considerable interest exists for a set of features that fit within the general category of inter-domain authentication, or "roaming capability" for network access, including dialup Internet users, Virtual Private Network (VPN) usage, wireless LAN authentication, and other applications.

By "inter-domain authentication", this document refers to situations where a user has authentication credentials at one "home" domain, but is able to present them at a second "visited" domain to access certain services at the visited domain. The two domains generally have a pre-existing relationship, so that the credentials can be passed from the visited domain to the home domain for verification. The home domain typically responds with a permit / deny response, which may also include authorization parameters which the visited domain is expected to enforce on the user.

That is, the "roaming" scenario involves a user visiting, or "roaming" to a non-home domain, and requesting the use of services at that visted domain.

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.

* Telecommunications companies who wish to combine their operations with those of one or more companies in another areas or nations, in order to offer more comprehensive network access service in areas where there is no native service. e.g. In another country.

* 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 in non-AAA systems. Some non-AAA systems defined identifiers which were compatible with the NAI, and deployments used the NAI. This process simplified the management of credentials, by re-using the same credential in multiple situations. Protocols that re-use the same credential or the same identifier format can benefit from this management simplicity. The alternative is to have protocol-specific credentials or identifier formats, which increases cost to both the user and the administrator.

There are privacy implications to using one identifier across multiple protocols. See Section 2.7 and Section 4 for further discussion of this topic.

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. The goal of this document is to encourage the wide-spread adoption of the NAI format. This adoption will decrease work required to leverage identification and authentication in other protocols. It will also decrease the complexity of non-AAA systems for end users and administrators.

This document only suggests that the NAI format be used, but does not require such use. 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. The definition of the NAI in this document has no requirements on protocol specifications, implementations, or deployments.

However, this document suggests that using one standard identifier format is preferable to using multiple incompatible identifier formats. Where identifiers need to be used in new protocols and/or specifications, it is RECOMMENDED that the format of the NAI be used. That is, the interpretation of the identifier is context-specific, while the format of the identifier remains the same. These issues are discussed in more detail in Section 2.8, below.

The recommendation for a standard identifier format is not a recommendation that each user have one universal identifier. In contrast, this document allows for the use of multiple identifiers,

and recommends the use of anonymous identifiers where those identifiers are publicly visible.

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 a common format for user identifiers submitted by a client during authentication. The purpose of the NAI is to allow a user to be associated with an account name, as well as to assist in the routing of the authentication request across multiple domains. Please note that the NAI may not necessarily be the same as the user’s email address or the user identifier 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.

Normalization or Canonicalization

These terms are defined in [RFC6365] Section 4. Those definitions are incorporated here by reference.

Locale

This term is defined in [RFC6365] Section 8. Those definitions are incorporated here by reference.

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", "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 essentially all Internet Service Providers are involved in roaming consortia.

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.

This document suggests that other protocols can take advantage of the NAI format. Many protocols include authentication capabilities, including defining their own identifier formats. These identifiers can then end up being transported in AAA protocols, so that the originating protocols can leverage AAA for user authentication. There is therefore a need for a definition of a user identifier which can be used in multiple protocols.

While the NAI is defined herein, it should be noted that existing protocols and deployments do not always use it. AAA systems MUST therefore be able to handle user identifiers which are not in the NAI format. The process by which that is done is outside of the scope of this document.

Non-AAA systems can 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. This document cannot change existing protocols or practices. It can, however, suggest that using a consistent form for a user identifier is of a benefit to the community.

This document does not make any protocol-specific definitions for an identifier format, and it does not make changes to any existing protocol. Instead, it defines a protocol-independent form for the NAI. It is hoped that the NAI is a user identifier which can be used in multiple protocols.

Using a common identifier format implifies protocols requiring authentication, as they no longer need to specify protocol-specific format for user identifiers. It increases security, as it multiple identifier formats allow attackers to make contradictory claims without being detected (see Section 4.2 for further discussion of this topic). It simplifies deployments, as a user can have one identifier in multiple contexts, which allows them to be uniquely

identified, so long as that identifier is itself protected against unauthorized access.

In short, having a standard is better than having no standard at all.

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 Punycode [RFC3492] encoding form as described in [RFC5891] There are a number of problems with this approach:

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

* The requirement in [RFC4282] Section 2.1 that realms are ASCII conflicts with the Extensible Authentication Protocol (EAP) defined in [RFC3748], and RADIUS, which are both 8-bit clean, and which both recommend the use of UTF-8 for identitifiers.

* [RFC4282] 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.

* [RFC4282] Section 2.4 requires normalization of user names, which may conflict with local system or administrative requirements.

* The recommendations in RFC4282] Section 2.4 for treatment of bidirectional characters have proven to be unworkable.

* The prohibition against use of unassigned code points in RFC4282] 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.

Furthermore, this document was motivated by a desire to codify existing practice related to the use of the NAI format and to encourage widespread use of the format.

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 which instead receive 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. This local version of the identifier MUST NOT be used outside of the local context.

Where protocols carry identifiers which are expected to be transported over an AAA protocol, it is RECOMMENDED that the identifiers be in NAI format. Where the identifiers are not in the NAI format, it is up to the AAA systems to discover this, and to process them. This document does not suggest how that is done. However, existing practice indicates that it is possible.

As internationalized domain names become more widely used, existing practices are likely to become inadequate. This document therefore defines the NAI, which is a user identifier format that can correctly deal with internationalized identifiers.

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 *("." string)
string          = 1*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 [RFC6733], 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 home domain for that realm.

That is, the only domain which is capable of interpreting the meaning of the utf8-username portion of the NAI is the home domain. Any third-party domains cannot form any conclusions about the utf8-username, and cannot decode it into sub-fields. For example, it may be used as "firstname.lastname", or it may be entirely digits, or it may be a random hex identifier. There is simply no way (and no reason) for any other domain to interpret the utf8-username field as having any meaning whatsoever.

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 including a fixed username part is ambiguous as to whether or not the NAI refers to a single user. However, current practice is to use the username "anonymous" instead of omitting the username part. This behavior is also permitted.

The most common use-case of omitting or obfuscating the username part

is with TLS-based EAP methods such as TTLS [RFC5281]. Those methods allow for an "outer" identifier, which is typically an anonymous "@realm". This outer identifier allows the authentication request to be routed from a visited domain to a home domain. At the same time, the username part is kept confidential from the visited network. The protocol provides for an "inner" authentication exchange, in which a full identifier is used to authenticate a user.

That scenario offers the best of both worlds. An anonymous NAI can be used to route authentication to the home domain, and the home domain has sufficient information to identify and authenticate users.

However, some protocols do not support authenticate methods which allow for "inner" and "outer" exchanges. Those protocols are limited to using an identifier which is publicly visible. It is therefore RECOMMENDED that such protocols use ephemeral identifiers. We recognize that this practice is not currently used, and will likely be difficult to implement.

Similarly to the anonymous user, there may be situations where portions of the realm are sensitive. For those situations, it is RECOMMENDED that the sensitive portion of the realm also be omitted. e.g. To use "@example.com" instead of "@sensitive.example.com", or "anonymous@sensitive.example.com". The home domain is authoritative for users in all subdomains, and can (if necessary) route the authentication request to the appropriate subsystem within the home domain.

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, authentication routing is impossible unless the realm portion is available, and in a well-known format.

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 username portion of the NAI is based on the "Internationalized Email Headers" [RFC6532] extensions to the "local-part" portion of email addresses [RFC5322].

In order to ensure a canonical representation, characters of the realm 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.

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.

One caveat on the above recommendation is the issues noted in [RFC6912]. That document notes that there are additional restrictions around DNS registration which forbid some code points from being valid in a DNS U-label. These restrictions cannot be expressed algorithmically.

For this specification, that caveat means the following. Realms not matching the above ABNF are not valid NAIs. However, some realms which do match the ABNF are still invalid NAIs. That is, matching the ABNF is a necessary, but not sufficient, requirement for an NAI.

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

2.6. The Normalization Process

Conversion to Unicode as well as normalization SHOULD be performed by edge systems (e.g. laptops, desktops, smart phones, etc.) that take "local" text as input. These edge systems are best suited to determine the users intent, and can best convert from "local" text to a normalized form.

Other AAA systems such as proxies do not have access to locale and character set information that is available to edge systems. Therefore, they may not always be able to convert local input to Unicode.

That is, all processing of NAIs from "local" character sets and locales to UTF-8 SHOULD be 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, AAA systems are now expected 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 AAA systems such as proxies are not required to perform translations, and can be expected to work through AAA systems that are unaware of international character sets.

In an ideal world, the following requirements would be widely implemented:

* Edge systems using "localized" text SHOULD normalize the NAI prior to it being used as an identifier in an authentication protocol.

* AAA systems SHOULD NOT normalize the NAI, as they may not have sufficient information to perform the normalization.

There are issues with this approach, however.

2.6.1. Issues with the Normalization Process

The requirements in the preceding section are not implemented today. For example, most EAP implementations use a user identifier which is passed to them from some other local system. This identifier is treated as an opaque blob, and is placed as-is into the EAP Identity field. Any subsequent system which receives that identifier is assumed to be able to understand and process it.

This opaque blob unfortunately can contain localized text, which means that the AAA systems have to process that text.

These limitations have the following theoretical and practical implications.

* edge systems used today generally do not normalize the NAI

* Therefore AAA systems SHOUD attempt to normalize the NAI

The suggestion in the above sentence contradicts the suggestion in the previous section. This is the reality of imperfect protocols.

Where the user identifier can be normalized, or determined to be in normal form, the normal form MUST be used as the NAI. In all other circumstances, the user identifier MUST NOT be treated as an NAI. That data is still, however, a user identifier. AAA systems MUST NOT fail authentication simply because the user identifier is not an NAI.

That is, when the realm portion of the NAI is not recognized by an AAA server, it SHOULD try to normalize the NAI into NFC form. That normalized form can then be used to see if the realm matches a known realm. If no match is found, the original form of the NAI SHOULD be used in all subsequent processing.

The AAA server may also convert realms to punycode, and perform all realm comparisons on the resulting punycode strings. This conversion follows the recommendations above, but may have different operational effects and failure modes.

2.7. Use in Other Protocols

As noted earlier, the NAI format can 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 use the NAI format.

This document cannot require other protocols to use the NAI format for user identifiers. Their needs are unknown, and at this time unknowable. This document suggests 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 cannot transport the NAI as-is. Instead, this document presumes that a protocol-specific transport layer takes care of encoding the NAI on input to the protocol, and decoding it when the NAI exits the protocol. The encoded or escaped version of the NAI is not a valid NAI, and MUST NOT be presented to the AAA system.

For example, HTTP carries user identifiers, but escapes the '.' character as "%2E" (among others). When HTTP is used to transport the NAI "fred@example.com", the data as transported will be in the form "fred@example%2Ecom". That data exists only within HTTP, and has no relevance to any AAA system.

Any comparison, validation, or use of the NAI MUST be done on its un-escaped (i.e. utf8-clean) form.

2.8. Using the NAI format for other identifiers

As discussed in Section 1, above, is RECOMMENDED that the NAI format be used as the standard format for user identifiers. This section discusses that use in more detail.

It is often useful to create new identifiers for use in specific contexts. These identifiers may have a number of different properties, most of which are unimportant to this document. The goal of this document is to create identifiers which are to be in a well-known format, and to have namespaces. The NAI format fits these requirements.

One example of such use is the "private user identity", which is an identifier defined by the 3rd-Generation Partnership Project (3GPP). That identifier is used to uniquely identify the user to the network. The identifier is used for authorization, authentication, accounting, administation, etc. The "private user identity" is globally unique, and is defined by the home network operator. The format of the identifier is explicitly the NAI, as stated by Section 13.3 of [3GPP]:

> The private user identity shall take the form of an NAI, and shall have the form username@realm as specified in clause 2.1 of IETF RFC 4282

For 3GPP, the "username" portion is a unique identifier which is derived from device-specific information. The "realm" portion is composed of information about the home network, followed by the base string "3gppnetwork.org". e.g. 234150999999999@ims.mnc015.mcc234.3gppnetwork.org.

This format as defiend by 3GPP ensures that the identifier is globally unique, as it is based off of the "3gppnetwork.org" domain. It ensures that the "realm" portion is specific to a particular home network (or organization), via the "ims.mnc015.mcc234" prefix to the realm. Finally, it ensures that the "username" portion follows a well-known format.

This document suggests that the NAI format be used for all new specifications and/or protocols where a user identifier is required. Where the username portions need to be created with subfields, a well-known and documented method as has been done with 3GPP is preferred to ad-hoc methods.

3. 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. 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 can be wrong. Where the "utf8-realm" is entirely ASCII, current AAA systems sometimes perform case-insensitive matching on realms. This method MAY be continued, as it has been shown to work in practice.

Many existing non-AAA systems have 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 SHOULD 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, AAA 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 non-AAA 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.

3.1. 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 "Internet Message Format" [RFC5322] "local-part" portion of an email address as extended by "Internationalized Email Headers" [RFC6532], 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 "addr-spec" portion of [RFC5322] as extended by [RFC6532], while still being compatible with [RFC4282].

In contrast to [RFC4282] Section 2.5, this document 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.

3.2. 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 convert the "utf8-realm" portion of an NAI into Punycode encoding form [RFC3492] prior to placing the NAI into a RADIUS User-Name attribute.

The NAI does not make a distinction between A-labels and U-labels, as

those are terms specific to DNS. It is instead an IDNA-valid label, as per the first item in Section 2.3.2.1 of [RFC5890]. As noted in that section, the term "IDNA-valid label" encompasses both of the terms A-label and U-label.

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 Punycode [RFC3492] encoding form as described in [RFC5891]. As noted in [RFC6055] Section 2, resolver Application Programming Interfaces (APIs) are not necessarily DNS-specific, so conversion to Punycode needs to be done 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 resolvable using the public DNS and blindly convert them to A-labels without knowledge of what protocol will be selected by the name resolution library.

3.3. 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.

3.3.1. Historical Practices

Some AAA systems have historically used NAI modifications with multiple "prefix" and "suffix" decorations to perform explicit routing through multiple proxies inside of a AAA network.

In RADIUS based environment, the use of decorated NAI 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 edge systems (e.g. supplicants) expecting to obtain network access.

* Using explicit routing paths requires thousands, if not millions of edge 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 manually managed explicit path routing was once useful, the time has come for better methods to be used.

Notwithstanding the above recommendations, the above practice is widely used for Diameter routing [RFC5729]. The routes described there are managed automatically, for both credential provisioning and routing updates. Those routes also exist within a particular framework (typically 3G), where membership is controlled and system behavior is standardized. There are no known issues with using

explicit routing in such an environment.

However, if decorated identifiers are used, such as:

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

Then the part before the (non-escaped) '!' MUST be a "utf8-realm" as defined in the ABNF in Section 2.2. When receiving such an identifier, the "otherrealm.example.net" system MUST convert the identifier to "user@homerealm.example.org" before forwarding the request. The forwarding system MUST then apply normal AAA routing for the transaction, based on the updated identifier.

3.4. 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
```

An additional valid NAI is the following, given as a hex string, as this document can only contain ASCII characters.

```
626f 6240 ceb4 cebf ceba ceb9 cebc ceae 2e63 6f6d
```

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 Punycode [RFC3492] encoding form for what is now a valid UTF-8 string.

alice@xn--tmonesimerkki-bfbb.example.net

4. 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 [RFC6733].

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.

4.1. Correlation of Identities over Time and Protocols

The recommendations in Section 2.7 and Section 2.8 for using the NAI in other protocols has implications for privacy. Any attacker who is capable of observing traffic containing the NAI can track the user, and correlate his activity across time and across multiple protocols. The authentication credentials therefore SHOULD be transported over channels which permit private communications, or multiple identifiers SHOULD be used, so that user tracking is impossible.

It is RECOMMENDED that user privacy be enhanced by configuring multiple identifiers for one user. These identifiers can be changed over time, in order to make user tracking more difficult for a malicous observer. However, provisioning and management of the identifiers may be difficult in to do in practice, which is likely why multiple identifiers are rarely used today.

4.2. Multiple Identifiers

Section 1.3 states that multiple identifier formats allow attackers to make contradictory claims without being detected. This statement

deserves further discussion.

Section 2.4 discussed "inner" and "outer" identifiers in the context of TTLS [RFC5281]. A close reading of that specification shows there is no requirement that the inner and outer identifiers be in any way related. That is, it is perfectly valid to use "@example.com" for an outer identifier, and "user@example.org" as an inner identifier. The authentication request will then be routed to "example.com", which will likely be unable to authenticate "user@example.org".

Even worse, a misconfiguration of "example.com" means that it may in turn proxy the inner authentication request to the "example.org" domain. Such cross-domain authentication is highly problematic, and there are few good reasons to allow it.

It is therefore RECOMMENDED that systems which permit anonymous "outer" identifiers require that the "inner" domain be the same as, or a sub-domain of the "outer" domain. An authentication request using disparate realms is a security violation, and the request SHOULD be rejected.

The situation gets worse when multiple protocols are involved. The TTLS protocol permits MS-CHAP [RFC2433] to be carried inside of the TLS tunnel. MS-CHAP defines its own identifier which is encapsulated inside of the MS-CHAP exchange. That identifier is not required to be any particular format, is not required to be in UTF-8, and in practice, can be one of many unknown character sets. There is no way in practice to determine which character set was used for that identifier.

The result is that the "outer" EAP Identity carried by TTLS is likely to not even share the same character set as the "inner" identifier used by MS-CHAP. The two identifiers are entirely independent, and fundamentally incomparable.

Such protocol design is NOT RECOMMENDED.

5. Administration of Names

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. Administrators MUST NOT

publicly use an NAI realm without first owning the corresponding FQDN. Private use of unowned NAI realms within an administative domain is allowed, though it is RECOMMENDED that example names be used, such as "example.com".

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 [RFC6733] 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.

6. IANA Considerations

This document has no actions for IANA.

7. References

7.1. 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, August 2010

[RFC5891]
Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010

[RFC6365]
Hoffman, P., and Klensin, J., "Terminology Used in Internationalization in the IETF", RFC 6365, September 2011

7.2. 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.

[RFC2433]
Zorn G., and Cobb, S. "Microsoft PPP CHAP Extensions", RFC 2433, October 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.

[RFC3492]
Costello, A., "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

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

[RFC5281]
Funk, P., and Blake-Wilson, S., "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.

[RFC5322]
Resnick, P. (Ed), "Internet Message Format", RFC 5322, October 2008.

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

[RFC5729]
Korhohen, J. (Ed) et. al., "Clarifications on the Routing of Diameter Requests Based on the Username and the Realm", RFC 5729, December 2009

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

[RFC6532]
Yang, A., et al, "Internationalized Email Headers", RFC 6532, February 2012.

[RFC6733]
V. Fajardo, Ed., et al, "Diameter Base Protocol", RFC 6733, October 2012.

[RFC6912]
Sullivan, A., et al, "Principles for Unicode Code Point Inclusion in Labels in the DNS", RFC 6912, April 2013.

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

[3GPP]
3GPP, "TS 23.003 Numbering, addressing, and Identification (Release 12)", July 2014,

ftp://ftp.3gpp.org/Specs/archive/23_series/23.003/.

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. The Punycode transformation 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 it is recommended 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