EMU Working Group
Internet-Draft
Updates: 5216, 9140, 9190 (if approved)
Intended status: Standards Track
Expires: 7 January 2026

A. DeKok
InkBridge Networks

6 July 2025

The eap.arpa domain and EAP provisioning
draft-ietf-emu-eap-arpa-08

Abstract

This document defines the eap.arpa domain for use only in Network Access Identifiers (NAIs) as a way for Extensible Authentication Protocol (EAP) peers to signal to EAP servers that they wish to obtain limited, and unauthenticated, network access. EAP peers signal which kind of access is required via certain predefined identifiers which use the Network Access Identifier (NAI) format of RFC 7542. A table of identifiers and meanings is defined, which includes entries for RFC 9140.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-emu-eap-arpa/.

Discussion of this document takes place on the EMU Working Group mailing list (mailto:emut@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/emut/. Subscribe at https://www.ietf.org/mailman/listinfo/emut/.

Source for this draft and an issue tracker can be found at https://github.com/freeradius/eap-arpa.git.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In most uses, EAP [RFC3748] requires that the EAP peer have pre-provisioned credentials. Without credentials, the device cannot obtain network access in order to be provisioned with credentials. This limitation creates a bootstrapping problem.

This specification addresses that bootstrapping problem. It creates a framework for predefined "well-known" provisioning credentials, and instantiates that framework for two mechanisms.

Clients can submit these predefined provisioning credentials to a server in order to obtain limited network access. At the same time, servers can know in advance that these credentials are to be used only for provisioning, and avoid granting unrestricted network access to peers which submit these credentials.

The device can either use the EAP channel itself for provisioning, as with TEAP [RFC7170], or the EAP server can give the device access to a limited captive portal such as with [RFC8952]. Once the device is provisioned, it can use those provisioned credentials to obtain full network access.

The predefined provisioning credentials use a generic identity format. Identifiers in this space are generically referred to as "EAP Provisioning Identifiers" (EPI).

Since the identity is predefined and used only for unauthenticated network access, there is little benefit to specifying predefined passwords. Where supported by the underlying EAP method, this specification provides for password-less access. Where passwords are required, the password is defined to be the same as the identity.

2. Terminology

EAP Provisioning Identifier (EPI)

> The EAP Provisioning Identifier (EPI) is defined to be a strict subset of the Network Access Identifier [RFC7542]. The EPI is an NAI which ends with "eap.arpa". The domain or "realm" portion of the NAI is defined in [RFC7542], Section 2.2, which is a more restrictive subset of the domain name conventions specified in RFC1034.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview

A device which has no device-specific credentials can use a predefined provisioning identifier in Network Access Identifier (NAI) format [RFC7542]. The NAI is composed of two portions, the utf8-username, and the utf8-realm domain. For simplicity here, we refer to these as the "username" and "realm" fields.

The realm is chosen to be independent of, and unused by, any existing organization, and thus to be usable by all organizations. The realm needs to be one which is not automatically proxied by any existing Authentication, Authorization, and Accounting (AAA) proxy framework as defined in [RFC7542], Section 3. The realm also needs to be one which does not return results for [RFC7585] dynamic discovery.

This specification does not, however, forbid routing of packets for realms in the "eap.arpa" domain. Instead, it leaves such routing up to individual organizations.

We note that this specification is fully compatible with all known EAP implementations, so it is fail-safe. When presented with a peer wishing to use this specification, existing implementations will return EAP Failure, and will not otherwise misbehave.

3.1. The eap.arpa realm

This document defines the "eap.arpa" domain as being used for provisioning within EAP. A similar domain has previously been used for EAP-NOOB [RFC9140], as "eap-noob.arpa". This document extends that concept, and standardizes the practices surrounding it,

NOTE: the "arpa" domain is controlled by the IAB. Allocation of "eap.arpa" requires agreement from the IAB.

RFC-EDITOR: This text can be updated on publication to indicate that the IAB has approved it.

3.2. The realm field

The NAIs defined by this specification use the [RFC7542] "realm" field to signal the behavior being requested; in particular, the subdomain under eap.arpa allows for different requested methods to be distinguished. The subdomain in the realm field is assigned via the EAP Provisioning Identifier Registry, which is defined in Section 6.2. The subdomain MUST follow the syntax defined in [RFC7542], Section 2.2, which is a more restrictive subset of the domain name conventions specified in RFC1034.

It is RECOMMENDED that the first subdomain of "eap.arpa" use the EAP method name, as defined in the IANA Extensible Authentication Protocol (EAP) Registry, sub-registry "Method Types". However, that registry does not follow the domain name conventions specified in RFC1034, so it is not possible to make a "one-to-one" mapping between the Method Type name and the subdomain.

Where it is not possible to make a direct mapping between the EAP Method Type name due to the EAP Method Type name not matching the [RFC7542], Section 2.2 format, the name used in the realm registry SHOULD be similar enough to allow the average reader to understand which EAP Method Type is being used.

Additional subdomains are permitted in the realm, which permit vendors and Standards Development organizations (SDOs) the ability to self-assign a delegated range of identifiers which do not conflict with other identifiers.

Any realm defined in this registry (e.g. "tls.eap.arpa") also implicitly defines a subdomain "v." (e.g. "v.tls.eap.arpa"). Vendors or SDOs can self-allocate within the "v." subdomain, using domains which they own. For example, An "example.com" company could self-allocate and use the realm "example.com.v.tls.eap.arpa". See Section 6.6 for more discussion of this topic.

3.3. The username field

The username field is dependent on the EAP method being used for provisioning. For example, [RFC9140] uses the username "noob". Other EAP methods MAY omit the username as RECOMMENDED in [RFC7542]. The username of "anonymous" is NOT RECOMMENDED for specifications using this format, even though it is permitted by [RFC7542]. The name "anonymous" is widely used in NAIs today, and we wish to avoid confusion.

The username field is assigned via the EAP Provisioning Identifier Registry which is defined in Section 6.2. The username field MAY be empty, or else hold a fixed value. While [RFC7542] recommends omitting the username portion for user privacy, the names here are defined in public specifications. User privacy is therefore not needed for provisioning identifiers, and the username field can be publicly visible.

3.4. Operation

Having described the format and contents of NAIs in the eap.arpa realm to define the EAP Provisioning Identifier (EPI), we now describe how those EPIs are used by EAP peers and EAP peers to signal provisioning information

3.4.1. EAP Peer

An EAP peer signals that it wishes a certain kind of provisioning by using an EPI, along with an associated EAP method. The meaning of the EPI, and behavior of the peer, are defined by a separate specification. That specification will typically define both the EPI, and the EAP method or methods which are used for provisioning.

The EPI used by the peer MUST be taken from an entry in the "EAP Provisioning Identifiers" registry, and the EAP method used with that NAI MUST match the corresponding EAP method from that same entry.

Where an EAP peer allows local selection of a provisioning method, the choice of EPI is defined by the provisioning method. As a result, the EAP peer MUST NOT have a field which lets the user identifier field be configured directly. Instead the user (or some other process) chooses a provisioning method, and the peer then chooses an EPI which matches that provisioning method.

While EAP peers allow users to enter user identifiers directly for existing EAP methods, they SHOULD NOT check whether those identfiers match any EPI. Any user who enters an identifier which matches an EPI will either get rejected because the server does not support

provisioning, or the user will be placed into a captive portal. There is no security or privacy issues with a user manually entering an EPI as the user identifier.

When all goes well, running EAP with the EPI results in new authentication credentials being provisioned. The peer then drops its network connection, and re-authenticates using the newly provisioned credentials. The user MAY be involved in this process, but in general provisioning results in the EAP peer automatically gaining network access using the provisioned credentials.

There are a number of ways in which provisioning can fail. One way is when the server does not implement the provisioning method. EAP peers therefore MUST track which provisioning methods have been tried, and not repeat the same method to the same EAP server when receiving an EAP Nak. EAP peers MUST rate limit attempts at provisioning, in order to avoid overloading the server. This rate limiting SHOULD include jitter and exponential backoff.

Since there is no way to signal whether the failed provisioning is due to a transient failure on the EAP server, or whether it is due to the EAP server not supporting that provisioning method, EAP peers SHOULD err on the side of long delays between retrying the same provisioning method to an EAP server. EAP peers MAY retry a given provisioning method after a sufficiently long interval that the EAP server might have implemented the provisioning method, e.g., at least a day, and perhaps no more than a month.

Another way for the provisioning method to fail is when the new credentials do not result in network access. It is RECOMMENDED that when peers are provisioned with credentials, that they immediately try to gain network access using those credentials. That process allows errors to be quickly discovered and addressed.

An EAP peer may have been provisioned with temporary credentials or credentials that expire after some period of time (e.g., an X.509 certificate with notAfter date set). It SHOULD therefore attempt to provision new credentials before the current set expires. Unfortunately, any re-provisioning process with EAP will involve the device dropping off from the "full" network, in order to connect to the provisioning network. It is therefore RECOMMENDED that re-provisioning methods be provided which can be used when the device has full network access. See Section 3.6 for additional discussion on this topic.

3.4.2. EAP Servers

An EAP session begins with the server receiving an initial EAP-Request/Identity message. An EAP server supporting this specification MUST examine the identity to see if it uses a realm located under eap.arpa. If so, the identity is an EPI. Processing of all other identities is unchanged by this specification.

If the server receives a malformed EPI, it MUST reply with an EAP Failure, as per [RFC3748], Section 4.2. Otherwise, the EPI is examined to determine which provisioning method is being requested by the peer.

If the server does not recognize the EPI requested by the peer, it MUST reply with an EAP Nak of type zero (0). This reply indicates that the requested provisioning method is not available. The server also MUST reply with a Nak of type zero (0) as per [RFC3748], Section 5.3.1, if the peer proposes an EAP method which is not supported by the server, or is not recognized as being valid for that provisioning method. The peer can then take any remedial action which it determines to be appropriate.

Once the server accepts the provisioning method, it then replies with an EAP method which MUST match the one associated with the EPI. The EAP process then proceeds as per the EAP state machine outlined in [RFC3748].

Implementations MUST treat peers using an EPI as untrusted, and untrustworthy. Once such a peer is authenticated, it MUST be placed into a limited network, such as a captive portal. The limited network MUST NOT permit general network access. Implementations should be aware of methods which bypass simple blocking, such as tunneling data over DNS.

A secure provisioning network is one where only the expected traffic is allowed, and all other traffic is blocked. The alternative of blocking only selected "bad" traffic results in substantial security failures. As most provisioning methods permit unauthenticated devices to gain network access, these methods have a substantial potential for abuse by malicious actors. As a result, the limited network needs to be designed assuming that it will be abused by malicious actoer.

A limited network SHOULD also limit the duration of network access by devices being provisioned. The provisioning process should be fairly quick, and in the order of seconds to tens of seconds in duration. Provisioning times longer than this likely indicate an issue, and it may be useful to block the problematic device from the network.

A limited network SHOULD also limit the amount of data being transferred by devices being provisioned, and SHOULD limit the network services which are available to those devices. The provisioning process generally does not need to download large amounts of data, and similarly does not need access to a large number of services.

Servers SHOULD rate-limit provisioning attempts. A misbehaving peer can be blocked temporarily, or even permanently. Implementations SHOULD limit the total number of peers being provisioned at the same time. We note that there is no requirement to allow all peers to connect without limit. Instead, peers are provisioned at the discretion of the network being accessed, which may permit or deny those devices based on reasons which are not explained to those devices.

Peers MUST rate-limit their provisioning attempts. If provisioning fails, it is likely because provisioning is not available. Retrying provisioning repeatedly in quick succession is not likely to change the server behavior. Instead, it is likely to result in the peer being blocked. The peer SHOULD retry provisioning no more than once every few minutes, and SHOULD perform exponential back-off on its provisioning attempts.

Implementations SHOULD use functionality such as the RADIUS Filter-Id attribute ([RFC2865], Section 5.11) to set packet filters for the peer being provisioned. For ease of administration, the Filter-Id name could simply be the EPI, or a similar name. Such consistency aids with operational considerations when managing complex networks.

3.5. Other Considerations

Implementations MUST NOT permit EAP method negotiation with provisioning credentials. That is, when an EPI is used, any EAP Nak sent by a server MUST contain only EAP method zero (0). When an EAP peer uses an EPI and receives an EAP Nak, any EAP methods given in that Nak MUST be ignored.

While a server may support multiple provisioning methods, there is no way in EAP to negotiate which provisioning method can be used. It is also expected that the provisioning methods will be specific to a particular type of peer device. That is, a given peer is likely to support only one provisioning method.

As a result, there is no need to require a method for negotiating provisioning methods.

3.6. Considerations for Provisioning Specifications

The operational considerations discussed above have a number of impacts on specifications which define provisioning methods.

3.6.1. Negotiation

Specifications which define provisioning for an EAP method SHOULD provide a method-specific process by which implementations can negotiate a mutually acceptable provisioning method.

For the reasons noted above, however, we cannot make this suggestion mandatory. If it is not possible for a provisioning method to define any negotiation, then that limitation should not be a barrier to publishing the specification.

3.6.2. Renewal of Credentials

Where a provisioning method is expected to create credentials which do not expire, the specification SHOULD state this explicitly.

Where credentials expire, it is RECOMMENDED that specifications provide guidance on how the credentials are to be updated. For example, an EAP method could permit re-provisioning to be done as part of a normal EAP authentication, using the currently provisioned credentials.

It is RECOMMENDED that the provisioning methods provide for a method which can be used without affecting network access. A specification could define provisioning endpoints such as Enrollment over Secure Transport (EST) [RFC7030], or Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP) [RFC4210]. The provisioning endpoints could be available both on the provisioning network, and on the provisioned (i.e., normal) network. Such an architecture means that devices can be re-provisioned without losing network access.

3.7. Notes on AAA Routability

When we say that the eap.arpa domain is not routable in an AAA proxy framework, we mean that the domain does not exist, and will never resolve to anything for dynamic discovery as defined in [RFC7585]. In addition, administrators will not have statically configured AAA proxy routes for this domain. Where routes are added for this domain, they will generally be used to implement this specification.

In order to avoid spurious DNS lookups, RADIUS servers supporting [RFC7585] SHOULD perform filtering in the domains which are sent to DNS. Specifically, names in the "eap.arpa" domain SHOULD NOT be looked up in DNS.

4. Background and Rationale

In this section, we provide background on the existing functionality, and describe why it was necessary to define provisioning methods for EAP.

4.1. Review of Existing Functionality

For EAP-TLS, both [RFC5216] Section 2.1.1 and [RFC9190] provide for "peer unauthenticated access". However, those documents define no way for a peer to signal that it is requesting such access. The presumption is that the peer connects with some value for the EAP Identity, but without using a client certificate. The EAP server is then supposed to determine that the peer is requesting unauthenticated access, and take the appropriate steps to limit authorization.

There appears to be no EAP peer or server implementations which support such access, since there is no defined way to perform any of the steps required, i.e., to signal that this access is desired, and then limit access.

Wi-Fi Alliance has defined an unauthenticated EAP-TLS method, using a vendor-specific EAP method as part of HotSpot 2.0r2 [HOTSPOT]. However, there appears to be few deployments of this specification.

EAP-NOOB [RFC9140] takes this process a step further. It defines both a way to signal that provisioning is desired, and also a way to exchange provisioning information within EAP-NOOB. That is, there is no need for the device to obtain limited network access, as all of the provisioning is done inside of the EAP-NOOB protocol.

Tunnel Extensible Authentication Protocol (TEAP) [RFC7170] provides for provisioning via an unauthenticated TLS tunnel. That document provides for a server unauthenticated provisioning mode, but the inner TLS exchange requires that both ends authenticate each other. There are ways to provision a certificate, but the peer must still authenticate itself to the server with pre-existing credentials. As a result, any provisioning method which uses TEAP will have to address this limitation.

4.2. Taxonomy of Provisioning Types

There are two scenarios where provisioning can be done. The first is where provisioning is done within the EAP method, as with EAP-NOOB [RFC9140]. The second is where EAP is used to obtain limited network access (e.g. as with a captive portal). That limited network access is then used to run IP based provisioning over more complex protocols.

4.2.1. Rationale for Provisioning over EAP

It is often useful to do all provisioning inside of EAP, because the EAP / AAA admin does not have control over the network. It is not always possible to define a captive portal where provisioning can be done. As a result, we need to be able to perform provisioning via EAP, and not via some IP protocol.

5. Interaction with EAP Methods

As the provisioning identifier is used within EAP, it necessarily has interactions with, and effects on, the various EAP methods. This section discusses those effects in more detail.

Some EAP methods require shared credentials such as passwords in order to succeed. For example, both EAP-MSCHAPv2 (PEAP) and EAP-PWD [RFC5931] perform cryptographic exchanges where both parties knowing a shared password. Where password-based methods are used, the password SHOULD be the same as the provisioning identifier, as there are few reasons to define a method-specific password.

This requirement also applies to TLS-based EAP methods such as EAP Tunneled Transport Layer Security (EAP-TTLS) and Protected Extensible Authentication Protocol (PEAP). Where the TLS-based EAP method provides for an inner identity and inner authentication method, the credentials used there SHOULD be the provisioning identifier for both the inner identity, and any inner password.

It is RECOMMENDED that provisioning be done via a TLS-based EAP methods. TLS provides for authentication of the EAP server, along with integrity and confidentiality protection for any provisioning data exchanged in the tunnel. Similarly, if provisioning is done in a captive portal outside of EAP, EAP-TLS permits the EAP peer to run a full EAP authentication session while having nothing more than a few certificate authorities (CAs) locally configured.

5.1. High Level Requirements

All provisioning methods which are specified within the eap.arpa domain MUST define a way to authenticate the server. This authentication can happen either at the EAP layer (as with TLS-based EAP methods), or after network access has been granted (if credentials are provisioned over HTTPS).

Where TLS-based EAP methods are used, implementations MUST still validate EAP server certificates in all situations other than provisioning. Where the provisioning method under the "eap.arpa" domain defines that provisioning happen via another protocol such as with HTTPS, the EAP peer MAY skip validating the EAP server certificate.

Whether or not the server certificate is ignored, the peer MUST treat the local network as untrusted. [RFC8952], Section 6.2 has more discussion on this topic.

The ability to not validate the EAP server certificates relaxes the requirements of [RFC5216], Section 5.3 which requires that the server certificate is always validated. . For the provisioning case, it is acceptable in some cases to not validate the EAP server certificate, but only so long as there are other means to authenticate the data which is being provisioned.

However, since the device likely is configured with web CAs (otherwise, the captive portal would also be unauthenticated), provisioning methods could use those CAs within an EAP method in order to allow the peer to authenticate the EAP server. Further discussion of this topic is better suited for the specification(s) which define a particular provisioning method. We do not discuss it here further, other than to say that it is technically possible.

5.2. EAP-TLS

This document defines an identifier "portal@tls.eap.arpa", which allows EAP peers to use unauthenticated EAP-TLS. The purpose of the identifier is to allow EAP peers to signal EAP servers that they wish to obtain a "captive portal" style network access.

This identifier signals the EAP server that the peer wishes to obtain "peer unauthenticated access" as per [RFC5216], Section 2.1.1 and [RFC9190]. Note that peer unauthenticated access MUST provide for authentication of the EAP server, such as with a server certificate. Using TLS-PSK with a well-known PSK value is generally not appropriate, as it would not provide server authentication.

An EAP server which agrees to authenticate this request MUST ensure that the device is placed into a captive portal with limited network access. Implementations SHOULD limit both the total amount of data transferred by devices in the captive portal, and SHOULD also limit the total amount of time a device spends within the captive portal.

This method is an improvement over existing captive portals, which are typically completely unsecured and unauthenticated. Using peer unauthenticated TLS for network access ensures that the EAP server is proven to be authentic. The use of 802.1X ensures that the link between the EAP peer and EAP authenticator (e.g. access point) is also secured.

Further details of the captive portal architecture can be found in [RFC8952]. The captive portal can advertise support for the "eap.arpa" domain via an 802.11u NAI realm.

5.3. EAP-NOOB

It is RECOMMENDED that server implementations of Nimble out-of-band authentication for EAP (EAP-NOOB) accept both identities "noob@eap-noob.arpa" and "@noob.eap.arpa" as synonyms.

It is RECOMMENDED that EAP-NOOB peers use "@noob.eap.arpa" first, and if that does not succeed, use "noob@eap-noob.arpa".

6. IANA Considerations

A number IANA actions are required. There are two registry updates in order to define "eap.arpa". A new registry is created. The "noob@eap-noob.arpa" registry entry is deprecated.

6.1. .arpa updates

There are two updates to the ".arpa" registry.

IANA is also instructed to refuse further allocation requests which are directly within the ".arpa" registry for any functionality related to the EAP protocol. Instead, new allocations related to EAP are to be made within the new "EAP Provisioning Identifiers" registry.

6.1.1. Deprecating eap-noob.arpa

IANA is instructed to update the "eap-noob.arpa" entry as follows.

The USAGE field is updated to add the word DEPRECATED.

The REFERENCE field is updated to add a reference to THIS-DOCUMENT.

6.1.2. Defining eap.arpa

IANA is instructed to update the ".ARPA Zone Management" registry with the following entry:

DOMAIN

eap.arpa

USAGE

For provisioning within the Extensible Authentication Protocol framework.

REFERENCE

THIS-DOCUMENT

IANA is instructed to update the "Special-Use Domain Names" registry as follows:

NAME

eap.arpa

REFERENCE

THIS-DOCUMENT

6.1.2.1. Domain Name Reservation Considerations

This section answers the questions which are required by Section 5 of [RFC6761]. At a high level, these new domain names are used in certain situations in EAP. The domain names are never seen by users, and they do not appear in any networking protocol other than EAP.

1. Users:
   User are not expected to recognize these names as special or use them differently from other domain names. The use of these names in EAP is invisible to end users.

2. Application Software:
   EAP servers and clients are expected to make their software recognize these names as special and treat them differently.
   This document discusses that behavior.
   EAP peers should recognize these names as special, and should

refuse to allow users to enter them in any interface.
EAP servers and RADIUS servers should recognize the "eap.arpa" domain as special, and refuse to do dynamic discovery ([RFC7585]) for it.

3. Name Resolution APIs and Libraries:
Writers of these APIs and libraries are not expected to recognize these names or treat them differently.

4. Caching DNS Servers:
Writers of caching DNS servers are not expected to recognize these names or treat them differently.

5. Authoritative DNS Servers:
Writers of authoritative DNS servers are not expected to recognize these names or treat them differently.

6. DNS Server Operators:
These domain names have minimal impact on DNS server operators. They should never be used in DNS, or in any networking protocol outside of EAP.
Some DNS servers may receive lookups for this domain, if EAP or RADIUS servers are configured to do dynamic discovery for realms as defined in [RFC7585], and where those servers are not updated to ignore the ".arpa" domain. When queried for the "eap.arpa" domain, DNS servers SHOULD return an NXDOMAIN error.
If they try to configure their authoritative DNS as authoritative for this reserved name, compliant name servers do not need to do anything special. They can accept the domain or reject it. Either behavior will have no impact on this specification.

7. DNS Registries/Registrars:
DNS Registries/Registrars should deny requests to register this reserved domain name.

6.2. EAP Provisioning Identifiers Registry

IANA is instructed to add the following new registry to the "Extensible Authentication Protocol (EAP) Registry" group.

Assignments in this registry are done via "Expert Review" as described in [RFC8126] Section 4.5. Guidelines for experts is provided in Section 6.3.

The contents of the registry are as follows.

Title

EAP Provisioning Identifiers

Registration Procedure(s)

Expert review

Reference

THIS-DOCUMENT

Registry

NAI

The Network Access Identifier in [RFC7542] format. Names are stored as DNS A-Labels [RFC5890], Section 2.3.2.1, and are compared via the domain name comparison rules defined in [STD13].

Method Type

The EAP method name, taken from the "Description" field of the EAP "Method Types" registry.

Reference

Reference where this identifier was defined.

6.2.1. Initial Values

The following table gives the initial values for this table.

NAI	Method-Type	Reference
@noob.eap.arpa	EAP-NOOB	[RFC9140] and THIS-DOCUMENT
portal@tls.eap.arpa	EAP-TLS	[RFC9190] and THIS-DOCUMENT

Table 1

6.3. Guidelines for Designated Experts

The following text gives guidelines for Designated Experts who review allocation requests for this registry.

6.3.1. NAIs

The intent is for the NAI to contain both a reference to the EAP Method Type, and a description of the purpose of the NAI. For example, with an EAP Method Type "name", and a purpose "action", the NAI SHOULD be of the form "action@foo.eap.arpa".

The NAI MUST satisfy the requirements of the [RFC7542], Section 2.2 format. The utf8-username portion MAY be empty. The utf8-username portion MUST NOT be "anonymous". The NAI MUST end with "eap.arpa". NAIs with any "v." subdomain MUST NOT be registered, in order to preserve the functionality of that subdomain.

NAIs in the registry SHOULD NOT contain more than one subdomain. NAIs with a leading "v." subdomain MUST NOT be registered. That subdomain is reserved for vendor and SDO extensions.

The subdomain of the NAI field should correspond to the EAP Method Type name. Care should be taken so that the domain name conventions specified in RFC1034 are followed.

The NAIs in this registry are case-insensitive. While [RFC7542] notes that similar identifiers of different case can be considered to be different, for simplicity this registry requires that all entries MUST be lowercase.

Identifiers MUST be unique when compared in a case-insensitive fashion. While [RFC7542] notes that similar identifiers of different case can be considered to be different, this registry is made simpler by requiring case-insensitivity.

Entries in the registry should be short. NAIs defined here will generally be sent in a RADIUS packet in the User-Name attribute ([RFC2865] Section 5.1). That specification recommends that implementations should support User-Names of at least 63 octets. NAI length considerations are further discussed in [RFC7542] Section 2.3, and any allocations in this registry needs to take those limitations into consideration.

Implementations are likely to support a total NAI length of 63 octets. Lengths between 63 and 253 octets may work. Lengths of 254 octets or more will not work with RADIUS [RFC2865].

6.4. Method Type

Values in "Method Type" field of this registry MUST be taken from the IANA EAP Method Types registry or else it MUST be an Expanded Type which usually indicates a vendor specific EAP method.

The EAP Method Type MUST provide an MSK and EMSK as defined in [RFC3748]. Failure to provide these keys means that the method will not be usable within an authentication framework which requires those methods, such as with IEEE 802.1X.

6.5. Designated Experts

The Designated Expert will post a request to the EMU WG mailing list (or a successor designated by the Area Director) for comment and review, including an Internet-Draft or reference to external specification. Before a period of 30 days has passed, the Designated Expert will either approve or deny the registration request and publish a notice of the decision to the EAP Method Update (EMU) WG mailing list or its successor, as well as informing IANA. A denial notice must be justified by an explanation, and in the cases where it is possible, concrete suggestions on how the request can be modified so as to become acceptable should be provided.

6.6. Organization Self Assignment

This registry allows organizations to request allocations from this registry, but explicit allocations are not always required. Any NAI defined in this registry also implicitly defines a subdomain "v.". Organizations can self-allocate in this space, under the "v." subdomain, e.g. "local@example.com.v.tls.eap.arpa".

The purpose of self-assigned realms is for testing, and for future expansion. There are currently no use-cases being envisioned for these realms, but we do not wish to forbid future expansion.

An organization which has registered a Fully Qualified Domain Name (FQDN) within the DNS can use that name within the "v." subdomain.

As DNS registrations can change over time, an organization may stop using a domain at some point. This change is reflected in the DNS, but is unlikely to be reflected in shipped products which use a self-assigned realm. There is no solution to this problem, other than suggesting that organizations using self-assigned realms do not allow their DNS registrations to expire.

It is therefore RECOMMENDED that organizations avoid the use of self-assigned realms. Organizations MAY use self-assigned realms only when no other alternative exists, and when the organization expects to maintain operation for at least the lifetime of the devices which use these realms.

7. Privacy Considerations

The EAP Identity field is generally publicly visible to parties who can observe the EAP traffic. As the names given here are in a public specification, there is no privacy implication to exposing those names within EAP. The entire goal of this specification is in fact to make those names public, so that unknown (and private) parties can publicly (and anonymously) declare what kind of network access they desire.

However, there are many additional privacy concerns around this specification. Most EAP traffic is sent over RADIUS [RFC2865]. The RADIUS Access-Request packets typically contain large amounts of information such as MAC addresses, device location, etc.

This specification does not change RADIUS or EAP, and as such does not change which information is publicly available, or is kept private. Those issues are dealt with in other specifications, such as [I-D.ietf-radext-deprecating-radius].

However, this specification can increase privacy by allowing devices to anonymously obtain network access, and then securely obtain credentials.

The NAIs used here are contained in a public registry, and therefore do not have to follow the username privacy recommendations of [RFC7542], Section 2.4. However, there may be other personally identifying information contained in EAP or AAA packets. This situation is no different from normal EAP authentication, and thus has no additional positive or negative implications for privacy.

8. Security Considerations

This specification defines a framework which permits unknown, anonymous, and unauthenticated devices to request and to obtain network access. As such, it is critical that network operators provide limited access to those devices.

Future specifications which define an NAI within this registry, should give detailed descriptions of what kind of network access is to be provided.

8.1. On-Path Attackers and Impersonation

In most EAP use-cases, the server identity is validated (usually through a certificate), or the EAP method allows the TLS tunnel to be cryptographically bound to the inner application data. For the methods outlined here, the use of public credentials, and/or skipping server validation allows "on-path" attacks to succeed where they would normally fail

EAP peers and servers MUST assume that all data sent over an EAP session is visible to attackers, and can be modified by them.

The methods defined here MUST only be used to bootstrap initial network access. Once a device has been provisioned, it gains network access via the provisioned credentials, and any network access policies can be applied.

8.2. Provisioning is Unauthenticated

We note that this specification allows for unauthenticated EAP peers to obtain network access, however limited. As with any unauthenticated process, it can be abused. Implementations should take care to limit the use of the provisioning network.

Section 3.4.2 describes a number of methods which can be used to secure the provisioning network. In summary:

* allow only traffic which is needed for the current provisioning method. All other traffic should be blocked. Most notable, DNS has been used to exfiltrate network traffic, so DNS recursive resolvers SHOULD NOT be made available on the provisioning network.

* limit the services available on the provisioning network to only those services which are needed for provisioning.

* limit the number of devices which can access the provisioning network at the same time.

* for any one device, rate limit its access the provisioning network.

* for a device which has accessed the provisioning network, limit the total amount of time which it is allowed to remain on the network

* for a device which has accessed the provisioning network, limit the total amount of data which it is allowed to transfer through the network.

9. Acknowledgements

Mohit Sethi provided valuable insight that using subdomains was better and more informative than the original method, which used only the utf8-username portion of the NAI.

The document was further improved with reviews from Ignes Robles and Ben Kaduk.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/rfc/rfc3748>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <https://www.rfc-editor.org/rfc/rfc5216>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/rfc/rfc7542>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9140] Aura, T., Sethi, M., and A. Peltonen, "Nimble Out-of-Band Authentication for EAP (EAP-NOOB)", RFC 9140, DOI 10.17487/RFC9140, December 2021, <https://www.rfc-editor.org/rfc/rfc9140>.

[STD13] Internet Standard 13,
<https://www.rfc-editor.org/info/std13>.
At the time of writing, this STD comprises the following:

Mockapetris, P., "Domain names - concepts and facilities",
STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
<https://www.rfc-editor.org/info/rfc1034>.

Mockapetris, P., "Domain names - implementation and
specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
November 1987, <https://www.rfc-editor.org/info/rfc1035>.

10.2. Informative References

[HOTSPOT] Alliance, W.-F., "Passpoint", n.d.,
<https://www.wi-fi.org/discover-wi-fi/passpoint>.

[I-D.ietf-radext-deprecating-radius]
DeKok, A., "Deprecating Insecure Practices in RADIUS",
Work in Progress, Internet-Draft, draft-ietf-radext-
deprecating-radius-06, 25 May 2025,
<https://datatracker.ietf.org/doc/html/draft-ietf-radext-
deprecating-radius-06>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson,
"Remote Authentication Dial In User Service (RADIUS)",
RFC 2865, DOI 10.17487/RFC2865, June 2000,
<https://www.rfc-editor.org/rfc/rfc2865>.

[RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen,
"Internet X.509 Public Key Infrastructure Certificate
Management Protocol (CMP)", RFC 4210,
DOI 10.17487/RFC4210, September 2005,
<https://www.rfc-editor.org/rfc/rfc4210>.

[RFC5890] Klensin, J., "Internationalized Domain Names for
Applications (IDNA): Definitions and Document Framework",
RFC 5890, DOI 10.17487/RFC5890, August 2010,
<https://www.rfc-editor.org/rfc/rfc5890>.

[RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication
Protocol (EAP) Authentication Using Only a Password",
RFC 5931, DOI 10.17487/RFC5931, August 2010,
<https://www.rfc-editor.org/rfc/rfc5931>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names",
RFC 6761, DOI 10.17487/RFC6761, February 2013,
<https://www.rfc-editor.org/rfc/rfc6761>.

[RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <https://www.rfc-editor.org/rfc/rfc7030>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/rfc/rfc7170>.

[RFC7585] Winter, S. and M. McCauley, "Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS Based on the Network Access Identifier (NAI)", RFC 7585, DOI 10.17487/RFC7585, October 2015, <https://www.rfc-editor.org/rfc/rfc7585>.

[RFC8952] Larose, K., Dolson, D., and H. Liu, "Captive Portal Architecture", RFC 8952, DOI 10.17487/RFC8952, November 2020, <https://www.rfc-editor.org/rfc/rfc8952>.

[RFC9190] Preu Mattsson, J. and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3", RFC 9190, DOI 10.17487/RFC9190, February 2022, <https://www.rfc-editor.org/rfc/rfc9190>.

Author’s Address

Alan DeKok
InkBridge Networks
Email: aland@inkbridgenetworks.com