Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

K. Wierenga
Cisco Systems
S. Winter
RESTENA
T. Wolniewicz
Nicolaus Copernicus University

March 9, 2015

The eduroam architecture for network roaming
draft-wierenga-ietf-eduroam-05.txt

Abstract

This document describes the architecture of the eduroam service for federated (wireless) network access in academia. The combination of IEEE 802.1X, EAP and RADIUS that is used in eduroam provides a secure, scalable and deployable service for roaming network access. The successful deployment of eduroam over the last decade in the educational sector may serve as an example for other sectors, hence this document. In particular the initial architectural and standards choices are described, along with the changes that were prompted by operational experience.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Terminology 3
1.2. Notational Conventions 4
1.3. Design Goals 4
1.4. Solutions that were considered 5
2. Classic Architecture 5
2.1. Authentication 6
2.1.1. IEEE 802.1X 6
2.1.2. EAP 7
2.2. Federation Trust Fabric 9
2.2.1. RADIUS 9
3. Issues with initial Trust Fabric 11
3.1. Server Failure Handling 12
3.2. No error condition signalling 13
3.3. Routing table complexity 14
3.4. UDP Issues 15
3.5. Insufficient payload encryption and EAP server validation 16
4. New Trust Fabric 17
4.1. RADIUS with TLS 18
4.2. Dynamic Discovery 19
4.2.1. Discovery of responsible server 19
4.2.2. Verifying server authorisation 20
4.2.3. Operational Experience 21
4.2.4. Possible Alternatives 21
5. Abuse prevention and incident handling 21
5.1. Incident Handling 22
5.1.1. Blocking users on the SP side 23
5.1.2. Blocking users on the IdP side 24
5.1.3. Communicating account blocking to the end user . . . 25
5.2. Operator Name 25
5.3. Chargeable User Identity 26
6. Privacy Considerations 27
6.1. Collusion of Service Providers 27
6.2. Exposing user credentials 28
6.3. Track location of users 28
7. Security Considerations 28
7.1. Man in the middle and Tunneling Attacks 28
7.1.1. Verification of Server Name not supported 29

7.1.2. Neither Specification of CA nor Server Name checks during bootstrap 29
7.1.3. User does not configure CA or Server Name checks . . 29
7.1.4. Tunneling authentication traffic to obfuscate user origin 30
7.2. Denial of Service Attacks 30
7.2.1. Intentional DoS by malign individuals 31
7.2.2. DoS as a side-effect of expired credentials 31
8. IANA Considerations 32
9. References 32
9.1. Normative References 32
9.2. Informative References 34
Appendix A. Acknowledgments 37
Appendix B. Changes 37
Authors' Addresses 37

1. Introduction

In 2002 the European Research and Education community set out to create a network roaming service for students and employees in academia [eduroam-start]. Now over 10 years later this service has grown to more than 10,000 service locations, serving millions of users on all continents with the exception of Antarctica.

This memo serves to explain the considerations for the design of eduroam as well as to document operational experience and resulting changes that led to IETF standardization effort such as RADIUS over TCP [RFC6613] and RADIUS with TLS [RFC6614] and that promoted alternative uses of RADIUS like in ABFAB [I-D.ietf-abfab-arch]. Whereas the eduroam service is limited to academia, the eduroam architecture can easily be reused in other environments.

First this memo describes the original architecture of eduroam. Then a number of operational problems are presented that surfaced when eduroam gained wide-scale deployment. Lastly, enhancements to the eduroam architecture that mitigate the aforementioned issues are discussed.

1.1. Terminology

This document uses identity management and privacy terminology from [RFC6973]. In particular, this document uses the terms Identity Provider, Service Provider and identity management.

1.2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Note: Also the policy that eduroam participants subscribe to, expresses the requirements for participation in RFC 2119 language.

1.3. Design Goals

The guiding design considerations for eduroam were as follows:

- Unique identification of users at the edge of the network

The access Service Provider (SP) needs to be able to determine whether a user is authorized to use the network resources. Furthermore, in case of abuse of the resources, there is a requirement to be able to identify the user uniquely (with the cooperation of the user’s Identity Provider (IdP) operator).

- Enable (trusted) guest use

In order to enable roaming it should be possible for users of participating institutions to get seamless access to the networks of other institutions.

Note: traffic separation between guest users and normal users is possible (for example through the use of VLANs), and indeed widely used in eduroam.

- Scalable

The infrastructure that is created should scale to a large number of users and organizations without requiring a lot of coordination and other administrative procedures (possibly with the exception of an initial set up). Specifically, it should not be necessary for a user that visits another organization to go through an administrative process.

- Easy to install and use

It should be easy for both organizations and users to participate in the roaming infrastructure as that may otherwise inhibit wide scale adoption. In particular, there should be no or easy client installation and only one-time configuration.

- Secure

An important design criterion has been that there needs to be a security association between the end-user and their Identity Provider, eliminating the possibility of credentials theft. The minimal requirements for security are specified in the eduroam policy and subject to change over time. As an additional protection against user errors and negligence, it should be possible for participating Identity Providers add their own requirements for the quality of authentication of their own users without the need for the infrastructure as a whole to implement the same standard.

- Privacy preserving

The design of the system should provide for user anonymization, i.e. a possibility to hide the user's identity from any third parties, including Service Providers.

- Standards based

In an infrastructure in which many thousands of organizations participate it is obvious that it should be possible to use equipment from different vendors, therefore it is important to build the infrastructure using open standards.

1.4. Solutions that were considered

Three architectures were trialed: one based on the use of VPN-technology (deemed secure but not-scalable), one Web captive-portal based (scalable but not secure) and IEEE 802.1X-based, the latter being the basis of what is now the eduroam architecture. An overview of the candidate architectures and their relative merits can be found in [nrenroaming-select].

The chosen architecture is based on:

- IEEE 802.1X ([dot1X-standard]) as port based authentication framework using
- EAP ([RFC3748]) for integrity and confidentially protected transport of credentials and a
- RADIUS ([RFC2865]) hierarchy as trust fabric.

2. Classic Architecture

Federations, like eduroam, implement essentially two types of direct trust relations (and one indirect). The trust relation between an end-user and the IdP (operated by the home organization of the user) and between the IdP and the SP (in eduroam the operator of the

network at the visited location). In eduroam the trust relation between user and IdP is through mutual authentication. IdPs and SP establish trust through the use of a RADIUS hierarchy.

These two forms of trust relations in turn provide the transitive trust relation that makes the SP trust the user to use its network resources.

2.1. Authentication

Authentication in eduroam is achieved by using a combination of IEEE 802.1X [dot1X-standard] and EAP [RFC4372] (the latter carried over RADIUS for guest access, see below).

2.1.1. IEEE 802.1X

By using the IEEE 802.1X [dot1X-standard] framework for port-based network authentication, organizations that offer network access (SPs) for visiting (and local) eduroam users can make sure that only authorized users get access. The user (or rather the user's supplicant) sends an access request to the authenticator (wireless access point or switch) at the SP, the authenticator forwards the access request to the authentication server of the SP which in turn proxies the request through the RADIUS hierarchy to the authentication server of the user's home organization (the IdP, see below).

Note: The security of the connections between local wireless infrastructure and local RADIUS servers is a part of the local network of each SP, therefore it is out of scope for this document. For completeness it should be stated that security between access points and their controllers is vendor specific, security between controllers (or standalone access points) and local RADIUS servers is based on the typical RADIUS shared secret mechanism.

In order for users to be aware of the availability of the eduroam service, an SP that offers wireless network access MUST broadcast the SSID 'eduroam', unless that conflicts with the SSID of another eduroam SP, in which case an SSID starting with "eduroam-" MAY be used. The downside of the latter is that clients will not automatically connect to that SSID, thus losing the seamless connection experience.

Note: A direct implication of the common eduroam SSID is that the users cannot distinguish between a connection to the home network and a guest network at another eduroam institution (IEEE 802.11-2012 does have the so-called "Interworking" extensions to make that distinction, but these are not widely implemented yet). Furthermore,

without proper server verification users may even be tricked into joining a rogue eduroam network. Therefore, users should be made aware that they should not assume data confidentiality in the eduroam infrastructure.

To protect over-the-air user data confidentiality, IEEE 802.11 wireless networks of eduroam SP's MUST deploy WPA2+AES, and MAY additionally support WPA/TKIP as a courtesy to users of legacy hardware.

2.1.2. EAP

The use of the Extensible Authentication Protocol (EAP) [RFC4372] serves 2 purposes. In the first place a properly chosen EAP-method allows for integrity and confidentiality protected transport of the user credentials to the home organization. Secondly, by having all RADIUS servers transparently proxy access requests regardless of the EAP-method inside the RADIUS packet, the choice of EAP-method is between the 'home' organization of the user and the user, in other words, in principle every authentication form that can be carried inside EAP can be used in eduroam, as long as they adhere to minimal requirements as set forth in the eduroam Service Definition [eduroam-service-definition].

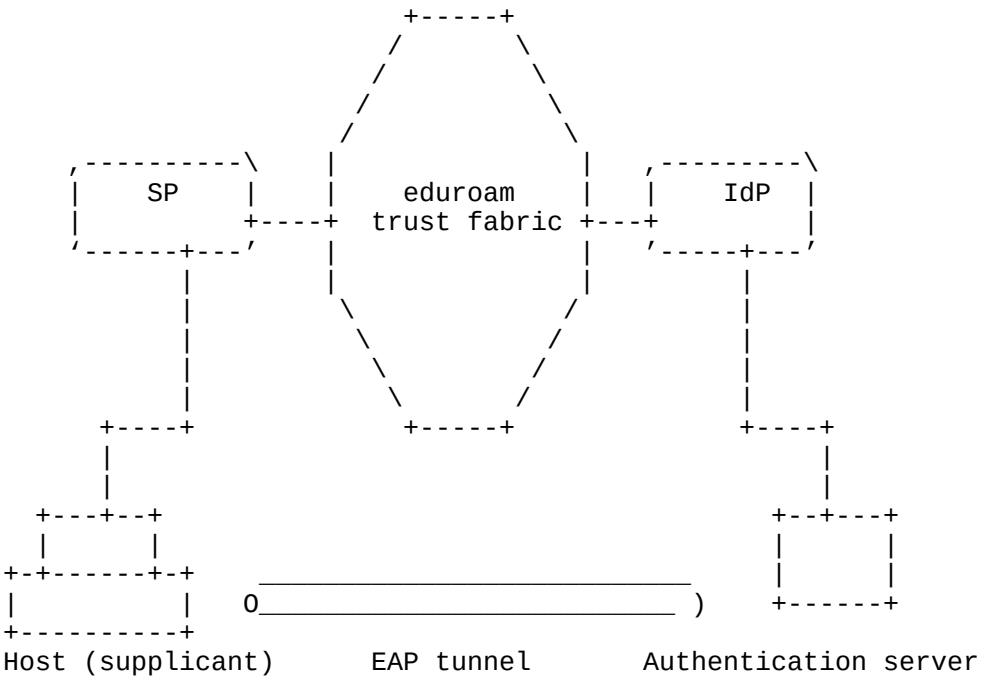

Figure 1: Tunneled EAP

Proxying of access requests is based on the outer identity in the EAP-message. Those outer identities MUST be a valid user identifier with a mandatory realm as per [I-D.ietf-radext-nai], i.e. be of the form something@realm or just @realm, where the realm part is the domain name of the institution that the IdP belongs to. In order to preserve credentials protection, participating organizations MUST deploy EAP-methods that provide mutual authentication. For EAP methods that support outer identity, anonymous outer identities are recommended. Most commonly used in eduroam are the so-called tunneled EAP-methods, that first create a server authenticated TLS tunnel through which the user credentials are transmitted. As depicted in Figure 1, the use of a tunneled EAP-method creates a direct logical connection between the supplicant and the authentication server, even though the actual traffic flows through the RADIUS-hierarchy.

2.2. Federation Trust Fabric

The eduroam federation trust fabric is based on RADIUS. RADIUS trust is based on shared secrets between RADIUS peers. In eduroam any RADIUS message originating from a trusted peer is implicitly assumed to originate from a member of the roaming consortium.

Note: See also the security considerations for a discussion on RADIUS securitythat motivated the work on RADIUS with TLS (RFC6614 [RFC6614]

2.2.1. RADIUS

The eduroam trust fabric consists of a proxy hierarchy of RADIUS servers (organizational, national, global), loosely based on the DNS hierarchy. That is, typically an organizational RADIUS server agrees on a shared secret with a national server and the national server in turn agrees on a shared secret with the root server. Access requests are routed through a chain of RADIUS proxies towards the Identity Provider of the user, and the access accept (or reject) follows the same path back.

Note: In some circumstances there are more levels of RADIUS servers, like for example regional or continental servers, but that doesn’t change the general model. Also, the packet exchange that is described below requires in reality several round-trips.

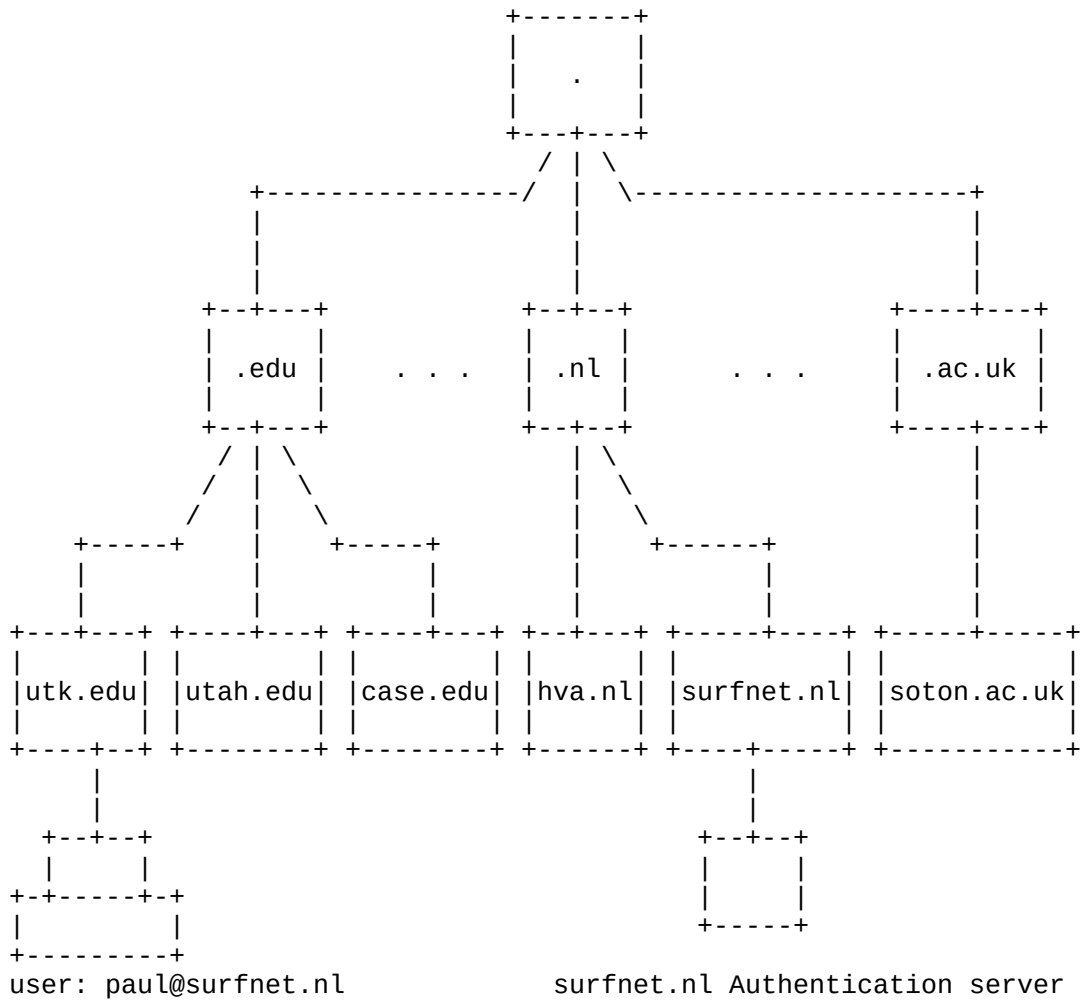

Figure 2: eduroam RADIUS hierarchy

Routing of access requests to the home IdP is done based on the realm part of the outer identity. For example (see: Figure 2), when user paul@surfnet.nl of SURFnet (surfnet.nl) tries to gain wireless network access at the University of Tennessee at Knoxville (utk.edu) the following happens:

- Paul’s supplicant transmits an EAP access request to the Access Point (Authenticator) at UTK with outer identity say anonymous@surfnet.nl

- The Access Point forwards the EAP message to its Authentication Server (the UTK RADIUS server)
- The UTK RADIUS server checks the realm to see if it is a local realm, since it isn't the request is proxied to the .edu RADIUS server
- The .edu RADIUS server verifies the realm, and since it is not a in a .edu subdomain it proxies the request to the root server
- The root RADIUS server proxies the request to the .nl RADIUS server, since the ".nl" domain is known to the root server.
- The .nl RADIUS server proxies the request to the surfnet.nl server, since it knows the SURFnet server.
- The surfnet.nl RADIUS server decapsulates the EAP message and verifies the user credentials, since the user is known to SURFnet.
- The surfnet.nl RADIUS server informs the utk.edu server of the outcome of the authentication request (Access-Accept or Access-Reject) by proxying the outcome through the RADIUS hierarchy in reverse order.
- The UTK RADIUS server instructs the UTK Access Point to either accept or reject access based on the outcome of the authentication.

Note: The depiction of the root RADIUS server is a simplification. In reality the root server is distributed over 3 continents and each maintains a list of the top level realms that a specific root server is responsible for. This also means that, for intercontinental roaming, there is an extra proxy step from one root server to the other. Also, the physical distribution of nodes doesn't need to mirror the logical distribution of nodes. This helps with stability and scalability.

3. Issues with initial Trust Fabric

While the hierarchical RADIUS architecture described in the previous section has served as the basis for eduroam operations for an entire decade, the exponential growth of authentications is expected to lead to, and has in fact in some cases already led to, performance and operations bottlenecks on the aggregation proxies. The following sections describe some of the shortcomings, and the resulting remedies.

3.1. Server Failure Handling

In eduroam, authentication requests for roaming users are statically routed through pre-configured proxies. The number of proxies varies: in a national roaming case, the number of proxies is typically 1 or 2 (some countries deploy regional proxies, which are in turn aggregated by a national proxy); in international roaming, 3 or 4 proxy servers are typically involved (the number may be higher along some routes).

RFC2865 [RFC2865] does not define a failover algorithm. In particular, the failure of a server needs to be deduced from the absence of a reply. Operational experience has shown that this has detrimental effects on the infrastructure and end user experience:

1. Authentication failure: the first user whose authentication path is along a newly-failed server will experience a long delay and possibly timeout

2. Wrongly deduced states: since the proxy chain is longer than 1 hop, a failure further along in the authentication path is indistinguishable from a failure in the next hop.

3. Inability to determine recovery of a server: only a "live" authentication request sent to a server which is believed inoperable can lead to the discovery that the server is in working order again. This issue has been resolved with RFC5997 [RFC5997].

The second point can have significant impact on the operational state of the system in a worst-case scenario: Imagine one realm’s home server being inoperable. A user from that realm is trying to roam internationally and tries to authenticate. The RADIUS server on the hotspot location may assume its own national proxy is down, because it does not reply. That national server, being perfectly alive, in turn will assume that the international aggregation proxy is down; which in turn will believe the home country proxy national server is down. None of these assumptions are true. Worse yet: should any of these servers trigger a failover to a redundant backup RADIUS server, it will still not receive a reply, because the request will still be routed to the same defunct home server. Within a short time, all redundant aggregation proxies might be considered defunct by their preceding hop.

In the absence of proper next-hop state derivation, some interesting concepts have been introduced by eduroam participants; the most noteworthy being a failover logic which considers up/down states not per next-hop RADIUS peer, but instead per realm (See [dead-realm] for details). As of recent, RFC5997 [RFC5997] implementations and

cautious failover parameters make false "downs" unlikely to happen, as long as every hop implements RFC5997. Dead realm detection in that case serves mainly to prevent proxying of large numbers of requests to known dead realms.

3.2. No error condition signalling

The RADIUS protocol lacks signalling of error conditions, and the IEEE 802.1X protocol does not allow to convey extended failure reasons to the end-user's device. For eduroam, this creates issues in a twofold way:

- The home server may have an operational problem, for example its authentication decisions may depend on an external data source such as ActiveDirectory or an SQL server, and the external data source is unavailable. If the RADIUS interface is still functional, there are two options how to reply to an Access-Request which can't be serviced due to such error conditions:

  1. Do Not Reply: the inability to reach a conclusion can be treated by not replying to the request. The upside of this approach is that the end-user's software doesn't come to wrong conclusions and won't give unhelpful hints such as "maybe your password is wrong". The downside is that intermediate proxies may come to wrong conclusions because their downstream RADIUS server isn't responding.

  2. Reply with Reject: in this option, the inability to reach a conclusion is treated like an authentication failure. The upside of this approach is that intermediate proxies maintain a correct view on the reachability state of their RADIUS peer. The downside is that EAP supplicants on end-user devices often react with either false advice ("your password is wrong") or even trigger permanent configuration changes (e.g. the Windows built-in supplicant will delete the credential set from its registry, prompting the user for their password on the next connection attempt). The latter case of Windows is a source of significant helpdesk activity; users may have forgotten their password after initially storing it, but are suddenly prompted again.

  There have been epic discussions in the eduroam community as well as in the IETF RADEXT Working Group as to which of the two approaches is more appropriate; but they were not conclusive.

- Similar considerations as above apply when an intermediate proxy does not receive a reply from a downstream RADIUS server. The proxy may either choose not to reply to the original request,

leading to retries and its upstream peers coming to wrong conclusions about its own availability; or it may decide to reply with Access-Reject to indicate its own liveliness, but again with implications for the end user.

The ability to send Status-Server watchdog requests is only of use after the fact, in case a downstream server doesn't reply (or hasn't been contacted in a long while, so that it's previous working state is stale). The active link-state monitoring of the TCP connection with e.g. RADIUS/TLS (see below) gives a clearer indication whether there is an alive RADIUS peer, but does not solve the defunct backend problem. An explicit ability to send Error-Replies, on the RADIUS (for other RADIUS peer information) and EAP level (for end-user supplicant information), would alleviate these problems but is currently not available.

3.3. Routing table complexity

The aggregation of RADIUS requests based on the structure of the user's realm implies that realms ending with the same top-level domain are routed to the same server; i.e. to a common administrative domain. While this is true for country code Top Level Domains (ccTLDs), which map into national eduroam federations, it is not true for realms residing in generic Top Level Domains (gTLDs). Realms in gTLDs were historically discouraged because the automatic mapping "realm ending" -> "eduroam federation's server" could not be applied. However, with growing demand from eduroam realm administrators, it became necessary to create exception entries in the forwarding rules; such realms need to be mapped on a realm-by-realm basis to their eduroam federations. Example: "kit.edu" (Karsruher Institut fuer Technologie) needs to be routed to the German federation server, whereas "iu.edu" (Indiana University) needs to be routed to the USA federation server.

While the ccTLDs occupy only approx. 50 routing entries in total (and have a upper bound of approx. 200), the potential size of the routing table becomes virtually unlimited if it needs to accomodate all individual entries in .edu, .org, etc.

In addition to that, all these routes need to be synchronised between three international root servers, and the updates need to be applied manually to RADIUS server configuration files. The frequency of the required updates makes this approach fragile and error-prone as the number of entries grows.

3.4. UDP Issues

RADIUS is based on UDP, which was a reasonable choice when its main use was with simple PAP requests which required only exactly one packet exchange in each direction.

When transporting EAP over RADIUS, the EAP conversations requires multiple round-trips; depending on the total payload size, 8-10 round-trips are not uncommon. The loss of a single UDP packet will lead to user-visible delays and might result in servers being marked as dead due to the absence of a reply. The proxy path in eduroam consists of several proxies, all of which introduce a very small packet loss probability; i.e. the more proxies are needed, the higher the failure rate is going to be.

For some EAP types, depending on the exact payload size they carry, RADIUS servers and/or supplicants may choose to fill as much EAP data into a single RADIUS packet as the supplicant’s layer 2 medium allows for, typically 1500 Bytes. In that case, the RADIUS encapsulation around the EAP-Message will add more bytes to the overall RADIUS payload size and in the end exceed the 1500 Byte limit, leading to fragmentation of the UDP datagram on the IP layer. While this is not a problem in theory, practice has shown evidence of misbehaving firewalls which erroneously discard non-first UDP fragments, which ultimately leads to a denial of service for users with such EAP types and that specific configuration.

One EAP type proved to be particularly problematic: EAP-TLS. While it is possible to configure the EAP server to send smaller chunks of EAP payload to the supplicant (e.g. 1200 Bytes, to allow for another 300 Bytes of RADIUS overhead without fragmentation), very often the supplicants which send the client certificate do not expose such a configuration detail to the user. Consequently, when the client certificate is beyond 1500 Bytes in size, the EAP-Message will always make use of the maximum possible layer-2 chunk size, which introduces the fragmentation on the path from EAP peer to EAP server.

Both of the previously mentioned sources of errors (packet loss, fragment discard) lead to significant frustration for the affected users. Operational experience of eduroam shows that such cases are hard to debug since they require coordinated cooperation of all eduroam administrators on the authentication path. For that reason the eduroam community is developing monitoring tools that help to locate fragmentation problems.

Note: For more detailed discussion of these issues please refer to section 1.1 of [RFC6613].

3.5. Insufficient payload encryption and EAP server validation

The RADIUS protocol's design foresaw only the encryption of select RADIUS attributes, most notably User-Password. With EAP methods conforming to the requirements of [RFC4017], the user's credential is not transmitted using the User-Password attribute, and stronger encryption than the one for RADIUS' User-Password is in use (typically TLS).

Still, the use of EAP does not encrypt all personally identifiable details of the user session as some are carried inside clear-text RADIUS attributes. In particular, the user's device can be identified by inspecting the Calling-Station-ID attribute; and the user's location may be derived from observing NAS-IP-Address, NAS-Identifier or Operator-Name attributes. Since these attributes are not encrypted, even IP-layer third parties can harvest the corresponding data. In a worst-case scenario, this enables the creation of mobility profiles. Pervasive passive surveillance using this connection metadata such as the recently uncovered NSA/GCHQ incidents becomes possible by tapping RADIUS traffic from an IP hop near a RADIUS aggregation proxy. While this is possible, the authors are not aware whether this has actually been done.

These profiles are not necessarily linkable to an actual user because EAP allows for the use of anonymous outer identities and protected credential exchanges. However, practical experience has shown that many users neglect to configure their supplicants in a privacy-preserving way or their supplicant doesn't support that. In particular, for EAP-TLS users, the use of EAP-TLS identity protection is not usually implemented and cannot be used. In eduroam, concerned individuals and IdPs which use EAP-TLS are using pseudonymous client certificates to provide for better privacy.

One way out, at least for EAP types involving a username, is to pursue the creation and deployment of pre-configured supplicant configurations which makes all the required settings in user devices prior to their first connection attempt; this depends heavily on the remote configuration possibilities of the supplicants though.

A further threat involves the verification of the EAP server's identity. Even though the cryptographic foundation, TLS tunnels, is sound, there is a weakness in the supplicant configuration: many users do not understand or are willing to invest time into the inspection of server certificates or the installation of a trusted CA. As a result, users may easily be tricked into connecting to an unauthorized EAP server, ultimately leading to a leak of their credentials to that unauthorized third party.

Again, one way out of this particular threat is to pursue the creation and deployment of pre-configured supplicant configurations which makes all the required settings in user devices prior to their first connection attempt.

Note: there are many different and vendor-proprietary ways to pre-configure a device with the necessary EAP parameters (examples include Apple, Inc's "mobileconfig" and Microsoft's "EAPHost" XML schema). Some manufacturers even completely lack any means to distribute EAP configuration data. We believe there is value in defining a common EAP configuration metadata format which could be used across manufacturers, ideally leading to a situation where IEEE 802.1X network end-users merely need to apply this configuration file to configure any of their devices securely with the required connection properties.

Another possible privacy threat involves transport of user-specific attributes in a Reply-Message. If, for example, a RADIUS server sends back a hypothetical RADIUS Vendor-Specific-Attribute "User-Role = Student of Computer Science" (e.g. for consumption of an SP RADIUS server and subsequent assignment into a "student" VLAN), this information would also be visible for third parties and could be added to the mobility profile.

The only way out to mitigate all information leakage to third parties is by protecting the entire RADIUS packet payload so that IP-layer third parties cannot extract privacy-relevant information. RFC2865 RADIUS does not offer this possibility though. This motivated [RFC6614], see below.

4. New Trust Fabric

The operational difficulties with an ever increasing number of participants, as documented in the previous section, have led to a number of changes to the eduroam architecture that in turn have, as mentioned in the introduction, led to standardization effort.

Note: The enhanced architecture components are fully backwards compatible with the existing installed base, and are in fact gradually replacing those parts of it where problems may arise.

Whereas the user authentication using IEEE 802.1X and EAP has remained unchanged (i.e. no need for end-users to change any configurations), the issues as reported above have resulted in a major overhaul of the way EAP messages are transported from the RADIUS server of the SP to that of the IdP and back. The two fundamental changes are the use of TCP instead of UDP and reliance on TLS instead of shared secrets between RADIUS peers.

4.1. RADIUS with TLS

The deficiencies of RADIUS over UDP as described in Section 3.4 warranted a search for a replacement of RFC2865 [RFC2865] for the transport of EAP. By the time this need was understood, the designated successor protocol to RADIUS, Diameter [RFC3588], was already specified by the IETF. However, within the operational constraints of eduroam:

- reasonably cheap to deploy on many administrative domains
- supporting NASREQ Application
- supporting EAP Application
- supporting Diameter Redirect
- supporting validation of authentication requests of the most popular EAP types (EAP-TTLS, PEAP, and EAP-TLS)
- possibility to retrieve these credentials from popular backends such as Microsoft ActiveDirectory, MySQL

no single combination of software could be found. In addition to that, no Wireless Access Points at the disposal of eduroam participants supported Diameter, nor did any of the manufacturers have a roadmap towards Diameter support (that is believed to still be true, more than 10 years later). This led to the open question of lossless translation from RADIUS to Diameter and vice versa; a question not satisfactorily answered by NASREQ.

After monitoring the Diameter implementation landscape for a while, it became clear that a solution with better compatibility and a plausible upgrade path from the existing RADIUS hierarchy was needed. The eduroam community actively engaged in the IETF towards the specification of several enhancements to RADIUS to overcome the limitations mentioned in Section 3. The outcome of this process was [RFC6614] and [I-D.ietf-radext-dynamic-discovery].

With its use of TCP instead of UDP, and with its full packet encryption, while maintaining full packet format compatibility with RADIUS/UDP, RADIUS/TLS [RFC6614] allows to upgrade any given RADIUS link in eduroam without the need of a "flag day".

In a first upgrade phase, the classic eduroam hierarchy (forwarding decision taken by inspecting the realm) remains intact. That way, RADIUS/TLS merely enhances the underlying transport of the RADIUS datagrams. But this already provides some key advantages:

- explicit peer reachability detection using long-lived TCP sessions
- protection of user credentials and all privacy-relevant RADIUS attributes

RADIUS/TLS connections for the static hierarchy could be realised with the TLS-PSK operation mode (which effectively provides a 1:1 replacement for RADIUS/UDP's "shared secrets"), but since this operation mode is not widely supported as of yet, all RADIUS/TLS links in eduroam are secured by TLS with X.509 certificates from a set of accredited CAs.

This first deployment phase does not yet solve the routing table complexity problem (see (Section 3.3); this aspect is covered by introducing dynamic discovery for RADIUS/TLS servers.

4.2. Dynamic Discovery

When introducing peer discovery, two separate issues had to be addressed:

1. How to find the network address of a responsible RADIUS server for a given realm?
2. How to verify that this realm is an authorized eduroam participant?

4.2.1. Discovery of responsible server

Issue 1 can relatively simply be addressed by putting eduroam-specific service discovery information into the global DNS tree. In eduroam this is done by using Naming Authority Pointer (NAPTR) records as per the S-NAPTR specification [RFC3958] with a private-use NAPTR service tag ("x-eduroam:radius.tls"). The usage profile of that NAPTR resource record is that exclusively "S" type delegations are allowed, and that no regular expressions are allowed.

A subsequent lookup of the resulting SRV records will eventually yield hostnames and IP addresses of the authoritative server(s) of a given realm.

Example (wrapped for readability):

```
> dig -t naptr education.example.

;; ANSWER SECTION:
education.example.          43200   IN      NAPTR   100 10 "s"
                                "x-eduroam:radius.tls" ""
                                _radsec._tcp.eduroam.example.


> dig -t srv _radsec._tcp.eduroam.example.

;; ANSWER SECTION:
_radsec._tcp.eduroam.example. 43200  IN      SRV     0 0 2083
                                                tld1.eduroam.example.

> dig -t aaaa tld1.eduroam.example.

;; ANSWER SECTION:
tld1.eduroam.example.       21751  IN      AAAA    2001:db8:1::2
```

Figure 3: SRV record lookup

From the operational experience with this mode of operation, eduroam is pursuing standardisation of this approach for generic AAA use cases. The current radext working group document for this is [I-D.ietf-radext-dynamic-discovery].

Note: It is worth mentioning that this move to a more complex, flexible sytem may make the system as a whole more fragile, as compared to the static set up.

4.2.2. Verifying server authorisation

Any organisation can put "x-eduroam" NAPTR entries into their Domain Name Server, pretending to be eduroam Identity Provider for the corresponding realm. Since eduroam is a service for a heterogeneous, but closed, user group, additional sources of information need to be consulted to verify that a realm with its discovered server is actually an eduroam participant.

The eduroam consortium has chosen to deploy a separate PKI infrastructure which issues certificates only to authorised eduroam Identity Providers and eduroam Service Providers. Since certificates are needed for RADIUS/TLS anyway, it was a straightforward solution to reuse the PKI for that. The PKI fabric allows multiple CAs as trust roots (overseen by a Policy Management Authority), and requires that certificates which were issued to verified eduroam participants are marked with corresponding "X509v3 Policy OID" fields; eduroam

RADIUS servers and clients need to verify the existence of these OIDs in the incoming certificates.

The policies and OIDs can be retrieved from the "eduPKI Trust Profile for eduroam Certificates" ([edupki]).

4.2.3. Operational Experience

The discovery model as described above is currently deployed in approximately 10 countries that participate in eduroam, making more than 100 realms discoverable via their NAPTR records. Experience has shown that the model works and scales as expected; the only drawback being that the additional burden of operating a PKI which is not local to the national eduroam administrators creates significant administrative complexities. Also, the presence of multiple CAs and regular updates of Certificate Revocation Lists makes the operation of RADIUS servers more complex.

4.2.4. Possible Alternatives

There are two alternatives to the above approach which are monitored by the eduroam community:

1. DNSSEC + DANE TLSA records

2. ABFAB Trust Router

For DNSSEC+DANE TLSA, its biggest advantage is that the certificate data itself can be stored in the DNS - possibly obsoleting the PKI infrastructure *if* a new place for the server authorization checks can be found. Its most significant downside is that the DANE specifications only include client-to-server certificate checks, while RADIUS/TLS requires also server-to-client verification.

For the ABFAB Trust Router, the biggest advantage is that it would work without certificates altogether (by negotiating TLS-PSK keys ad-hoc). The downside is that it is currently not formally specified and not as thoroughly understood as any of the other solutions.

5. Abuse prevention and incident handling

Since the eduroam service is a confederation of autonomous networks, there is little justification for transferring accounting information from the Service Provider to any other in general, or in particular to the Identity Provider of the user. Accounting in eduroam is therefore considered to be a local matter of the Service Provider. The eduroam compliance statement ([eduroam-compliance]) in fact specifies that accounting traffic SHOULD NOT be forwarded.

The static routing infrastructure of eduroam acts as a filtering system blocking accounting traffic from misconfigured local RADIUS servers. Proxy servers are configured to terminate accounting request traffic by answering to Accounting-Requests with an Accounting-Response in order to prevent the retransmission of orphaned Accounting-Request messages. With dynamic discovery, Identity Providers which are discoverable via DNS will need to apply these filtering measures themselves. This is an increase in complexity of the Identity Provider RADIUS configuration.

Roaming creates accountability problems, as identified by [RFC4372] (Chargeable User Identity). Since the NAS can only see the (likely anonymous) outer identity of the user, it is impossible to correlate usage with a specific user (who may use multiple devices). A NAS that supports this can request the Chargeable-User-Identity and, if supplied by the authenticating RADIUS server in the Access-Accept message, add this value to corresponding Access-Request packets. While eduroam does not have any charging mechanisms, it may still be desirable to identify traffic originating from one particular user. One of the reasons is to prevent abuse of guest access by users living nearby university campuses. Chargeable User Identity (see below) supplies the perfect answer to this problem, however at the moment of writing, to our knowledge only one hardware vendor (Meru Networks) implements RFC4372 on their Access Points. For all other vendors, requesting the Chargeable-User-Identity attribute needs to happen on the RADIUS server to which the Access Point is connected to. FreeRADIUS supports this ability in the latest distribution, and Radiator can be retrofitted to do the same.

5.1. Incident Handling

10 years of experience with eduroam have not exposed any serious incident. This may be taken as evidence for proper security design as well as suggest that awareness of users that they are identifiable, acts as an effective deterrent. It could of course also mean that eduroam operations lack the proper tools or insight into the actual use and potential abuse of the service. In any case, many of the attack vectors that exist in open networks or networks where access control is based on shared secrets are not present, arguably leading to a much more secure system.

Below a discussion of countermeasures that are taken in eduroam.

The European eduroam policy Service Definition [eduroam-service-definition], as an example, describes incident scenarios and actions to be taken, in this document we present the relevant technical issues.

The initial implementation has been lacking reliable tools for an SP to make it's own decision or for an IdP to introduce a conditional rule applying only to a given SP. The introduction of support for Operator-Name and Chargeable-User-Identity (see below) to eduroam makes both of these scenarios possible.

5.1.1. Blocking users on the SP side

The first action in the case of an incident is to block the user's access to eduroam at the Service Provider. Since the roaming user's true identity is likely hidden behind an anonymous/fake outer identity, the Service Provider can only rely on the realm of the user and his MAC address; if the Identity Provider has already sent a Chargeable-User-Identity (see Section 5.3 for details), then this extra information can be used to identify the user more reliably.

A first attempt at the SP side may be to block based on the MAC address or outer identity. This blocking can be executed before the EAP authentication occurs - typically in the first datagram, acting on the RADIUS attributes EAP-Message/EAP-Response/Identity and Calling-Station-ID. The datagram can either be dropped (supplicant notices a time-out) or replied-to with a RADIUS Access-Reject containing an EAP-Failure. Since malicious users can change both their MAC addresses and the local part of their outer identity between connection attempts, this first attempt is not sufficient to lock out a determined user.

As a second measure, the SP can let the EAP authentication proceed as normal, and verify whether the final Access-Accept response from the RADIUS server contains a Chargeable-User-Identity (CUI). If so, the SP RADIUS server can be configured to turn all future Access- Accepts for this CUI into an Access-Reject/EAP-Failure. This measure is effective and efficient: it locks out exactly the one user which is supposed to be locked out, and has no side-effects on other users, even from the same realm.

If the EAP authentication does not reveal a CUI, the SP can not reliably determine the user in question. The only reliable information to act upon is then the realm portion of the outer identity of the user. The SP will need to resort to blocking the entire realm that the offending user belongs to. This can be done at the EAP-Message/EAP-Repsonse/Identity stage as described above). This is effective, but not efficient: it locks out the user in question, but has a DoS side-effect on all other visiting users from the same realm.

In the absence of a CUI handle, SPs which are not willing to take the drastic step of blocking an entire realm will be forced to contact

the Identity Provider in question and demand that the user be blocked at the IdP side. This involves human interaction between SP and IdP is not possible in real-time.

5.1.2. Blocking users on the IdP side

The IdP has the power to disable a user account altogether, thus blocking this user from accessing eduroam in all sites. If blocking the user is done due a request of an SP (as per the previous section), there may be a more fine-grained possibility to block access to a specific SP - if the SP in question sends the Operator-Name attribute along with his Access-Requests (see Section 5.2 for details).

If the IdP decides to block the user globally, this is typically done by treating the login attempt as if the credentials were wrong: the entire EAP conversation needs to be executed to the point where the true inner identity is revealed (before that, the IdP does not know yet which user is attempting to authenticate). This typically coincides with the point in time where credentials are exchanged. Instead, or in addition to, checking the credential for validity, the Identity Provider also checks whether the user's account is (still) eligible for eduroam use and will return an Access- Reject/EAP-Failure if not.

There may well be cases where opinions between the SP desiring a user lockout and the IdP of the user differ. E.g. an SP might consider massive amounts of up-/downloads with file sharing protocols unacceptable as per local policy, and desire blocking of users that create too much traffic - but the IdP does not take offense on such actions and would not want to lock his user out of eduroam globally because of this one SP's opinion.

In the absence of the Operator-Name attribute, there is no way to apply a login restriction only for a given SP and not eduroam as a whole; eduroam eligibility is an all-or-nothing decision for the IdP.

If the Operator-Name attribute is present, the IdP can use this information to fail the authentication attempt only if the attempt originated from SPs which desire such blocking. Even though the Operator-Name attribute is available from the first RADIUS Access-Request datagram onwards, the EAP authentication needs to be carried out until the true inner identity is known just as in the global blocking case above. The Operator-Name is simply an additional piece of information which the IdP can use to make its decision.

5.1.3. Communicating account blocking to the end user

All the measures above alter the EAP conversation. They either create a premature rejection or timeout at the start of the conversation, or change the outcome of the authentication attempt at the very end of the conversation.

On the supplicant side, these alterations are undistinguishable from an infrastructure failure: a premature rejection or timeout could be due to a RADIUS server being unresponsive, and a rejection at the end of the conversation could be either user error (wrong password) or server error (credential lookup failed). For the supplicant, it is thus difficult to communicate a meaningful error to the user. The newly specified EAP type TEAP, "Tunnel Extensible Authentication Protocol" [RFC7170] has a means to transport fine-grained error reason codes to the supplicant; this has the potential to improve the situation in the future.

The EAP protocol foresees one mechanism to provide such user-interactive communication: the EAP state machine contains states which allow user-visible communication: an extra round of EAP-Request/Notification and the corresponding acknowledgement can be injected before the final EAP-Failure.

However, anecdotal evidence suggests that supplicants typically do not implement this part of the EAP state machine at all. One supplicant is reported to support it, but only logs the contents of the notification in a log file - which is not at all helpful for the end user.

The discovery of reasons and scope of account blocking are thus left as an out-of-band action. The eduroam user support process requires that users with authentication problems contact their Identity Provider as a first measure (via unspecified means, e.g. they could phone their IdP or send an email via a 3G backup link). If the Identity Provider is the one which blocked their access, the user will immediately be informed by them. If the reason for blocking is at the SP side, the Identity Provider will instead inform the user that the account is in working order and that the user needs to contact the SP IT support to get further insight. In that case, that SP-side IT support will notify the users about the reasons for blocking.

5.2. Operator Name

The Operator-Name attribute is defined in [RFC5580] as a means of unique identification of the access site.

The Proxy infrastructure of eduroam makes it impossible for home sites to tell where their users roam to. While this may be seen as a positive aspect enhancing user's privacy, it also makes user support, roaming statistics and blocking offending user's access to eduroam significantly harder.

Sites participating in eduroam are encouraged to add the Operator-Name attribute using the REALM namespace, i.e. sending a realm name under control of the given site.

The introduction of Operator-Name in eduroam has identified one operational problem - the identifier 126 assigned to this attribute has been previously used by some vendors for their specific purposes and has been included in attribute dictionaries of several RADIUS server distributions. Since the syntax of this hijacked attribute had been set to Integer, this introduces a syntax clash with the the RFC definition (Text). Operational tests in eduroam have shown that servers using the Integer syntax for attribute 126 may either truncate the value to 4 octets or even drop the entire RADIUS packet (thus making authentication impossible). The eduroam monitoring and eduroam test tools try to locate problematic sites. [RFC6929] clarifies in Section 2.8 the handling of these packets.

When a Service Provider sends its Operator-Name value, it creates a possibility for the home sites to set up conditional blocking rules, depriving certain users of access to selected sites. Such action will cause much less concern than blocking users from all of eduroam.

In eduroam the Operator Name is also used for the generation of Chargeable User Identity values.

The addition of Operator-Name is a straightforward configuration of the RADIUS server and may be easily introduced on a large scale.

5.3. Chargeable User Identity

The Chargeable-User-Identity (CUI) attribute is defined by RFC4372 [RFC4372] as an answer to accounting problems caused by the use of anonymous identity in some EAP methods. In eduroam the primary use of CUI is in incident handling, but it can also enhance local accounting.

The eduroam policy requires that a given user's CUI generated for requests originating from different sites should be different (to prevent collusion attacks). The eduroam policy thus mandates that a CUI request be accompanied by the Operator-Name attribute, which is used as one of the inputs for the CUI generation algorithm. The Operator-Name requirement is considered to be the "business

requirement" described in Section 2.1 of RFC4372 [RFC4372] and hence conforms to the RFC.

When eduroam started considering using CUI, there were no NAS implementations, therefore the only solution was moving all CUI support to the RADIUS server.

CUI request generation requires only the addition of NUL CUI attributes to outgoing Access-Requests, however the real strength of CUI comes with accounting. Implementation of CUI based accounting in the server requires that the authentication and accounting RADIUS servers used directly by the NAS are actually the same or at least have access to a common source of information. Upon processing of an Access-Accept the authenticating RADIUS server must store the received CUI value together with the device's Calling-Station-Id in a temporary database. Upon receipt of an Accounting-Request, the server needs to update the packet with the CUI value read from the database.

A wide introduction of CUI support in eduroam will significantly simplify incident handling at Service Providers. Introducing local, per-user access restriction will be possible. Visited sites will also be able to notify the home site about the introduction of such a restriction, pointing to the CUI value and thus making it possible for the home site to identify the user. When the user reports the problem at his home support, the reason will be already known.

6. Privacy Considerations

The eduroam architecture has been designed with protection of user credentials in mind as may be clear from the discussion above. However, operational experience has revealed some more subtle points with regards to privacy.

6.1. Collusion of Service Providers

If users use anonymous outer identities, SPs cannot easily collude by linking outer identities to users that are visiting their campus. This poses however problems with remediation of abuse or misconfiguration. It is impossible to find the user that exhibits unwanted behaviour or whose system has been compromised.

For that reason the Chargeable-User-Identity has been introduced in eduroam, constructed so that only the IdP of the user can uniquely identify the user. In order to prevent collusion attacks that CUI is required to be unique per user per Service Provider.

6.2. Exposing user credentials

Through the use of EAP, user credentials are not visible to anyone but the IdP of the user. That is, if a sufficiently secure EAP-method is chosen and EAP is not terminated prematurely.

There is one privacy sensitive user attribute that is necessarily exposed to third parties and that is the realm the user belongs to. Routing in eduroam is based on the realm part of the user identifier, so even though the outer identity in a tunneled EAP-method may be set to an anonymous identifier it MUST contain the realm of the user, and may thus lead to identifying the user if the realm in question contains few users. This is considered a reasonable trade-of between user privacy and usability.

6.3. Track location of users

Due to the fact that access requests (potentially) travel through a number of proxy RADIUS servers, the home IdP of the user typically cannot tell where a user roams to.

The introduction of Operator-Name and dynamic lookups (i.e. direct connections between IdP and SP) however, give the home IdP insight into the location of the user.

7. Security Considerations

This section addresses only security considerations associated with the use of eduroam. For considerations relating to IEEE 802.1X, RADIUS and EAP in general, the reader is referred to the respective specification and to other literature.

7.1. Man in the middle and Tunneling Attacks

The security of user credentials in eduroam ultimately lies within the EAP server verification during the EAP conversation. Therefore, the eduroam policy mandates that only EAP types capable of mutual authentication are allowed in the infrastructure, and requires that IdPs publish all information that is required to uniquely identify the server (i.e. usually the EAP server’s CA certificate and its Common Name or subjectAltName:dNSName).

While this in principle makes Man-in-the-middle attacks impossible, practice has shown that several attack vectors exist nonetheless. Most of these deficiencies are due to implementation shortcomings in EAP supplicants. Examples:

7.1.1. Verification of Server Name not supported

Some supplicants only allow to specify which CA issues the EAP server certificate; it's name is not checked. As a result, any entity that is able to get a server certificate from the same CA can create its own EAP server and trick the end user to submit his credentials to that fake server.

As a mitigation to that problem, eduroam Operations suggests the use of a private CA which exclusively issues certificates to the organisation's EAP servers. In that case, no other entity will get a certificate from the CA and the above supplicant shortcoming does not present a security threat any more.

7.1.2. Neither Specification of CA nor Server Name checks during bootstrap

Some supplicants allow for insecure bootstrapping in that they allow to simply select a network and accept the incoming server certificate, identified by its fingerprint. The certificate is then saved as trusted for later re-connection attempts. If users are near a fake hotspot during initial provisioning, they may be tricked to submit their credentials to a fake server; and furthermore will be branded to trust only that fake server in the future.

eduroam Identity Providers are advised to provide their users with complete documentation for setup of their supplicants without the shortcut of insecure bootstrapping. In addition, eduroam Operations has created a tool which makes correct, complete and secure settings on many supplicants: eduroam CAT ([eduroam-cat]).

7.1.3. User does not configure CA or Server Name checks

Unless automatic provisioning tools such as eduroam CAT are used, it is cumbersome for users to initially configure an EAP supplicant securely. User Inferfaces of supplicants often invite the users to take shortcuts ("Don't check server certificate") which are easier to setup or hide important security settings in badly accessible sub-menus. Such shortcuts or security parameter ommissions make the user subject to man-in-the-middle attacks.

eduroam IdPs are advised to educate their users regarding the necessary steps towards a secure setup. eduroam Research and Development is in touch with supplicant developers to improve their User Interfaces.

7.1.4. Tunneling authentication traffic to obfuscate user origin

There is no link between the EAP outer ("anonymous") identity and the EAP inner ("real") identity. In particular, they can both contain a realm name, and the realms need not be identical. It is possible to craft packets with an outer identity of user@RealmB, and an inner identity of user@realmA. With the eduroam request routing, a Service Provider would assume that the user is from realmB and send the request there. The server at realm B inspects the inner user name, and if proxying is not explicitly disabled for tunneled request content, may decide to send the tunneled EAP payload to realmA, where the user authenticates. A CUI value would likely be generated by the server at realmB, even though this is not its user.

Users can craft such packets to make their identification harder; usually, the eduroam SP would assume the troublesome user to originate from realmB and demand there that the user be blocked. The operator of realmB however has no control over the user, and can only trace back the user to his real origin if logging of proxied requests is also enabled for EAP tunnel data.

eduroam Identity Providers are advised to explicitly disable proxying on the parts of their RADIUS server configuration which processes EAP tunnel data.

7.2. Denial of Service Attacks

Since eduroam’s roaming infrastructure is based on IP and RADIUS, it suffers from the usual DoS attack vectors that apply to these protocols.

The eduroam hotspots are susceptible to typical attacks on consumer edge networks, such as rogue RA, rogue DHCP servers, and others. Notably, eduroam hotspots are more robust against malign users’ DHCP pool exhaustion than typical open or "captive portal" hotspots, because a DHCP address is only leased after a successful authentication, which reduces the pool of possible attackers to eduroam account holders (as opposed to the general public). Furthermore, attacks involving ARP spoofing or ARP flooding are also reduced to authenticated users, because an attacker needs to be in possession of a valid WPA2 session key to be able to send traffic on the network.

This section does not discuss standard threats to consumer edge networks and IP networks in general. The following sections describe attack vectors specific to eduroam.

7.2.1. Intentional DoS by malign individuals

The eduroam infrastructure is more robust against Distributed DoS attacks than typical services which are reachable on the internet because triggering authentication traffic can only be done when physically being in proximity of an eduroam hotspot (be it a wired IEEE 802.1X enabled socket or a Wi-Fi Access Point).

However, when being in the vicinity, it is easy to craft authentication attempts that traverse the entire international eduroam infrastructure; an attacker merely needs to choose a realm from another world region than his physical location to trigger Access-Requests which need to be processed by the SP, then SP-side national, then world region, then target world region, then target national, then target IdP server. So long as the realm actually exists, this will be followed by an entire EAP conversation on that path. Not having actual credentials, the request will ultimately be rejected; but it consumed processing power and bandwidth across the entire infrastructure, possibly affecting all international authentication traffic.

EAP is a lock-step protocol. A single attacker at an eduroam hotspot can only execute one EAP conversation after another, and is thus rate-limited by round-trip times of the RADIUS chain.

Currently eduroam processes several hundred thousands of successful international roaming authentications per day (and, incidentally, approximately 1.5 times as many Access-Rejects). With the requirement of physical proximity, and the rate-limiting induced by EAP’s lock-step nature, it requires a significant amount of attackers and a time-coordinated attack to produce significant load. So far eduroam Operations has not yet observed critical load conditions which could reasonably be attributed to such an attack.

The introduction of dynamic discovery further eases this problem, as authentications will then not traverse all infrastructure servers, removing the world-region aggregation servers as obvious bottlenecks. Any attack would then be limited between an SP and IdP directly.

7.2.2. DoS as a side-effect of expired credentials

In eduroam Operations it is observed that a significant portion of (failed) eduroam authentications is due to user accounts which were once valid, but have in the meantime been de-provisioned (e.g. if a student has left the university after graduation). Configured eduroam accounts are often retained on the user devices, and when in the vicinity of an eduroam hotspot, the user device’s operating system will attempt to connect to this network.

As operation of eduroam continues, the amount of devices with left-over configurations is growing, effectively creating a pool of devices which produce unwanted network traffic whenever they can.

Up until recently, this problem did not emerge with much prominence, because there is also a natural shrinking of that pool of devices due to users finally de-commissioning their old computing hardware.

As of recent, particularly smartphones are programmed to make use of cloud storage and online backup mechanisms which save most, or all, configuration details of the device with a third-party. When renewing their personal computing hardware, users can restore the old settings onto the new device. It has been observed that expired eduroam accounts can survive perpetually on user devices that way. If this trend continues, it can be pictured that an always-growing pool of devices will clog up eduroam infrastructure with doomed-to-fail authentication requests.

There is not currently a useful remedy to this problem, other than instructing users to manually delete their configuration in due time. Possible approaches to this problem are:

- Creating a culture of device provisioning where the provisioning profile contains a "ValidUntil" property, after which the configuration needs to be re-validated or disabled. This requires a data format for provisioning as well as implementation support.

- Improvements to supplicant software so that it maintains state over failed authentications. E.g. if a previously known-working configuration failed to authenticate consistently for 30 calendar days, it should be considered stale and be disabled.

8. IANA Considerations

There are no IANA Considerations

9. References

9.1. Normative References

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

[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.

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

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4372] Adrangi, F., Lior, A., Korhonen, J., and J. Loughney, "Chargeable User Identity", RFC 4372, January 2006.

[RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, January 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5580] Tschofenig, H., Adrangi, F., Jones, M., Lior, A., and B. Aboba, "Carrying Location Objects in RADIUS and Diameter", RFC 5580, August 2009.

[RFC5997] DeKok, A., "Use of Status-Server Packets in the Remote Authentication Dial In User Service (RADIUS) Protocol", RFC 5997, August 2010.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6613] DeKok, A., "RADIUS over TCP", RFC 6613, May 2012.

[RFC6614] Winter, S., McCauley, M., Venaas, S., and K. Wierenga, "Transport Layer Security (TLS) Encryption for RADIUS", RFC 6614, May 2012.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

9.2. Informative References

[I-D.ietf-abfab-arch]
Howlett, J., Hartman, S., Tschofenig, H., Lear, E., and J. Schaad, "Application Bridging for Federated Access Beyond Web (ABFAB) Architecture", draft-ietf-abfab-arch-13 (work in progress), July 2014.

[I-D.ietf-radext-dtls]
DeKok, A., "DTLS as a Transport Layer for RADIUS", draft-ietf-radext-dtls-13 (work in progress), July 2014.

[I-D.ietf-radext-dynamic-discovery]
Winter, S. and M. McCauley, "NAI-based Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS", draft-ietf-radext-dynamic-discovery-13 (work in progress), March 2015.

[I-D.ietf-radext-nai]
DeKok, A., "The Network Access Identifier", draft-ietf-radext-nai-15 (work in progress), December 2014.

[MD5-attacks]
Black, J., Cochran, M., and T. Highland, "A Study of the MD5 Attacks: Insights and Improvements", October 2006, <http://www.springerlink.com/content/40867l85727r7084/>.

[RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.

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

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC4017] Stanley, D., Walker, J., and B. Aboba, "Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs", RFC 4017, March 2005.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[RFC6421] Nelson, D., "Crypto-Agility Requirements for Remote Authentication Dial-In User Service (RADIUS)", RFC 6421, November 2011.

[RFC6929] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", RFC 6929, April 2013.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, May 2014.

[dead-realm] Tomasek, J., "Dead-realm marking feature for Radiator RADIUS servers", 2006, <http://wiki.eduroam.cz/dead-realm/docs/dead-realm.html>.

[dot1X-standard] IEEE, "IEEE std 802.1X-2010", February 2010, <http://standards.ieee.org/getieee802/download/802.1X-2010.pdf>.

[edupki] Delivery of Advanced Network Technology to Europe, "eduPKI", 2012, <https://www.edupki.org/edupki-pma/edupki-trust-profiles/>.

[eduroam-cat] Delivery of Advanced Network Technology to Europe, "eduroam CAT", 2012, <https://cat.eduroam.org>.

[eduroam-compliance] Trans-European Research and Education Networking Association, "eduroam compliance statement", 2011, <http://www.eduroam.org/downloads/docs/eduroam_Compliance_Statement_v1_0.pdf>.

[eduroam-homepage]
Trans-European Research and Education Networking Association, "eduroam Homepage", 2007, <http://www.eduroam.org/>.

[eduroam-policy]
Delivery of Advanced Network Technology to Europe, "European Confederation eduroam policy", 2011, <http://www.eduroam.org/downloads/docs/GN3-12-194_eduroam-policy-%20for-signing_ver2%204_18052012.pdf>.

[eduroam-service-definition]
Delivery of Advanced Network Technology to Europe, "European eduroam policy Service Definition", 2011, <https://www.eduroam.org/downloads/docs/GN3-12-192_eduroam-policy-service-definition_ver28_26072012.pdf>.

[eduroam-start]
Wierenga, K., "Initial proposal for what is now called eduroam", 2002, <http://www.terena.org/activities/tf-mobility/start-of-eduroam.pdf>.

[geant]
Geant Association, "Geant Association", 2008, <http://www.terena.org/>.

[geant2]
Delivery of Advanced Network Technology to Europe, "European Commission Information Society and Media: GEANT2", 2008, <http://www.geant2.net/>.

[nrenroaming-select]
Trans-European Research and Education Networking Association, "Preliminary selection for inter-NREN roaming", 2003, <http://www.terena.org/activities/tf-mobility/deliverables/delG/DelG-final.pdf/>.

[radsec-whitepaper]
Open System Consultants, "RadSec - a secure, reliable RADIUS Protocol", May 2005, <http://www.open.com.au/radiator/radsec-whitepaper.pdf>.

[radsecproxy-impl]
Venaas, S., "radsecproxy Project Homepage", 2007, <http://software.uninett.no/radsecproxy/>.

Appendix A. Acknowledgments

The authors would like to thank the participants in the Geant Association Task Force on Mobility and Network Middleware as well as the Geant project for their reviews and contributions. Special thanks go to Jim Schaad for doing an excellent review of the first version and to him and Alan de Kok for additional reviews.

The eduroam trademark is registered by TERENA.

Appendix B. Changes

This section to be removed prior to publication.

- 00 Initial Revision
- 01 Added Dynamic Discovery, addressed review comments
- 02 Cosmetic changes
- 03 Even More Cosmetic Changes
- 04 Included review comments from Jim Schaad
- 05 Included review comments Jim Schaad and Alan deKok

Authors' Addresses

Klaas Wierenga
Cisco Systems
Haarlerbergweg 13-19
Amsterdam 1101 CH
The Netherlands

Phone: +31 20 357 1752
Email: klaas@cisco.com

Stefan Winter
Fondation RESTENA
6, rue Richard Coudenhove-Kalergi
Luxembourg 1359
Luxembourg

Phone: +352 424409 1
Fax: +352 422473
Email: stefan.winter@restena.lu
URI: http://www.restena.lu.

Tomasz Wolniewicz
Nicolaus Copernicus University
pl. Rapackiego 1
Torun
Poland

Phone: +48-56-611-2750
Fax: +48-56-622-1850
Email: twoln@umk.pl
URI: http://www.home.umk.pl/~twoln/