05-11-2009

# Deliverable DJ3.1.1: RadSec Standardisation and Definition of eduroam Extensions

**Deliverable DJ3.1.1**

| | |
|---|---|
| Contractual Date: | 30-10-2009 |
| Actual Date: | 05-11-2009 |
| Grant Agreement No.: | 238875 |
| Activity: | JRA3 |
| Task Item: | T1 |
| Nature of Deliverable: | R |
| Dissemination Level: | PU |
| Lead Partner: | RESTENA |
| Document Code: | GN3-09-213 |
| **Authors:** | S. Winter (RESTENA), T, Wolniewicz (PSNC), I. Thomson (DANTE) |

**Abstract**

This Deliverable describes two items of work started in GN2 JRA5 Task 1 that are now being progressed in GN3 JRA3 Task 1; the standardisation of RadSec (a new transport for the RADIUS protocol) and an extension to eduroam that enables more fine-grained control over logged-on users for hotspot operators; the Chargeable-User-Identity attribute

# Table of Contents

# Tables

# Figures

# Executive Summary

This deliverable provides a comprehensive overview of two developments started by JRA3 Task 1 “Roaming Development” in GN2 that are being carried over into GN3:

- The standardisation of RadSec, a new transport for the RADIUS protocol (RADIUS is one of the cornerstones of the eduroam architecture).
- A prospect extension to eduroam that enables more fine-grained control over logged-on users for hotspot operators; the Chargeable-User-Identity attribute.

In GN2, JRA5 identified several shortcomings of the currently deployed RADIUS protocol, and determined that protocol changes are necessary for eduroam operations. This discovery started the work on RadSec, and triggered a far-reaching campaign to introduce the protocol changes world-wide through the Internet Engineering Task Force (IETF).

The work on Chargeable-User-Identity originated when some hotspot operators experienced a perceived need to block guest users due to them abusing the system. The eduroam architecture currently does not provide sufficient control to pinpoint an individual.

This Deliverable describes the concepts behind these two projects, what has been achieved so far and what the plans for them are in the future.

# 1 Introduction

In GN3, JRA3 Task 1 “Roaming Development” serves as the R&D function for the European eduroam confederation service in SA3 Task 2. The current development efforts for eduroam in GN3 are a continuation of the efforts of GN2 JRA5 WI 1. As a result, there are a number of carry-over items from GN2 that are being continued in this task.

This deliverable provides a comprehensive overview over two of these developments:

- The standardisation of RadSec, a new transport for the RADIUS protocol (RADIUS is one of the cornerstones of the eduroam architecture).
- A prospect extension to eduroam that enables more fine-grained control over logged-on users for hotspot operators; the Chargeable-User-Identity attribute.

The background of the RadSec work is that GN2 JRA5 identified several shortcomings of the currently deployed RADIUS protocol, and determined that protocol changes are necessary for eduroam operations (and would probably be beneficial beyond eduroam). This decision triggered a far-reaching campaign to introduce the protocol changes world-wide through the Internet Engineering Task Force (IETF). Section 2 “Securing the RADIUS Protocol: RADIUS over TLS (RadSec)” of this deliverable explains the history, current state and future outlook of the standardisation efforts.

The work on Chargeable-User-Identity (CUI) originated when some hotspot operators experienced a need to block guest users due to them abusing the system. The eduroam architecture currently does not provide sufficient control to pinpoint an individual; the user could change their computing device's hardware (“MAC”) address and use an anonymous outer identity to obfuscate their identity to a hotspot operator. Even though there is ultimately a defined way of identifying them via out-of-band communication, an in-band persistent but opaque identifier that can be uniquely attributed to a user whenever they use this hotspot would be desirable.

The RADIUS attribute Chargeable-User-Identifier (CUI) can fulfil this requirement, while still preserving the user's privacy. Section 3 “Recognising Users on Re-entry: Chargeable-User-Identity” of this deliverable details the efforts undertaken to introduce CUI into the eduroam architecture.

# 2 Securing the RADIUS Protocol: RADIUS over TLS (RadSec)

In GN2, Joint Research Activity 5 (JRA5) identified a need for an advanced protocol to transmit eduroam authentication requests. After evaluating many potential candidates for the role [DJ514, Chapter 3], a prototype proprietary implementation, RadSec, was selected.

RadSec implements RADIUS over TCP (Transmission Control Protocol) and TLS (Transport Layer Security). This helps to make the transmission of authentication requests with RADIUS both more reliable (by using the proven underlying transport, TCP) and more secure (by encrypting the entire RADIUS payload with TLS). This work is being progressed in GN3 JRA3 Task 1.

Efforts are under way to promote RadSec to become an industry standard. Doing this requires:

- Formal specification of the protocol by an authoritative standards-defining organisation (the IETF; Internet Engineering Task Force).Demonstration of the existence of multiple implementations.
- If possible, a demonstration of actual uptake in the industry.
- Ensuring that the protocol does not have any IPR issues.

Sections 2.1.1 to 2.2.5 describe the efforts that have been undertaken so far to achieve RadSec standardisation. Section 2.2.5.1 describes possible migration paths from RADIUS to RadSec.

The technical specification  itself is being published at the IETF and under this organisation’s change control; therefore, it is not repeated here. Links to the specifications in question can be found at the end of section 2.1.1.

## 2.1 IETF Protocol Standardisation

### 2.1.1 History

When GN2 JRA5 decided to investigate a deployment of RadSec in eduroam in 2007, the relevant IETF charters (particularly of the RADIUS Extensions working group [RADEXTENSIONS]) forbade work on extensions for RADIUS, and instead exclusively promoted the use of a successor protocol; “Diameter”. Since Diameter was not fit for purpose in eduroam, JRA5 instead decided to pursue standardisation anyway.

The first draft [RADSEC-INITIAL] was submitted into the IETF process on 29 June 2007, and presented at the 69th IETF Meeting. It included:

- The TCP transport component of RadSec.
- The TLS encryption component of RadSec. A stub of dynamic discovery using DNS.

After intensive discussions concerning the benefits of RadSec over Diameter, the RADIUS Extensions working group decided to re-charter and include the topic of RADIUS over TCP and TLS in their scope of work on 09 July 2008.

At that time, it was also decided to separate the sub-topics of TCP transport and TLS encryption into two different documents. The TCP transport document was volunteered to be crafted by Alan DeKok, maintainer of the Open Source product FreeRADIUS, due to his then-pending engagement to implement RadSec in FreeRADIUS for the GEANT consortium (see section Implementation - FreeRADIUS), so that his expected implementation experience could be included in the document.

In parallel to the initial RadSec discussions, Alan DeKok also started work on another new transport for RADIUS, designed for short-haul situations (i.e. RADIUS communication locally within a campus, where packet losses can be expected to occur next to never) where TCP is a less ideal transport; RADIUS over DTLS (“Datagram Transport Layer Security” – encrypted transfer of datagrams). This work is happening in parallel to the RadSec effort and is monitored by JRA3 T1.

Towards the end of the GN2 project, JRA5 discovered several flaws in the protocols comprising eduroam which were unrelated to RadSec, but had a collateral impact. This was the problematic treating of internationalised domain names in the protocol combination of IEEE 802.1X, RADIUS and EAP (Note: This topic will be described in more depth in another deliverable, DJ3.1.2,1 “Roaming developments”). The impact on the RadSec specification is that the working group draft contained portions relating to dynamic server discovery via DNS, which relies on a thoroughly defined treatment of internationalised domain names. A release of the RadSec specification could only occur when the internationalisation issue was tackled, which would introduce an unspecified (probably long) delay. To prevent that from happening, the dynamic server discovery portions were split out into a separate document, which is to be released later when the internationalisation issue is solved.

The timeline is shown in Table 2.1:

| Date | RADIUS over TLS ("RadSec") | RADIUS over TCP | RADIUS over DTLS | Dynamic Discovery |
|---|---|---|---|---|
| 26/02/07 | -/- | -/- | Individual Submission v0 (13) | -/- |
| 29/06/07 | Independent Submission v0 (1) | -/- | | -/- |
| 08/02/08 | Independent Submission v1 (2) | -/- | | -/- |
| **09/07/08** | **– Working Group Re-Chartered –** | | | |
| 17/06/08 | Working Group Draft v0 (3) | -/- | | -/- |
| 03/07/08 | | Individual Submission v0 | | -/- |
| 22/08/08 | Working Group Draft v1 (8) | | | -/- |
| 24/10/08 | Working Group Draft v2 (9) | | | -/- |
| 02/11/08 | | Individual Submission v1 | | -/- |
| 11/12/08 | | Working Group Draft (4), (5) | | -/- |
| 16/12/08 | | Working Group Draft (6) | | -/- |
| 11/02/09 | Working Group Draft v3 (10) | | | -/- |
| 27/02/09 | | | | Individual Submission v0 (15) |
| 01/03/09 | | Working Group Draft (7) | | |
| 06/03/09 | Working Group Draft v4 (11) | | | |
| 09/06/09 | | | Individual Submission v1 (14) | |
| 02/07/09 | | | | Working Group Draft v0 (16) |
| 13/07/09 | Working Group Draft v5 (12) | | | Working Group Draft v1 (17) |
| | | | | |

Table 2.1: Timeline of IETF contributions for new RADIUS transports and server discovery.

The following links refer to the various documents relevant to this timeline:

http://tools.ietf.org/html/draft-winter-radsec-00

http://tools.ietf.org/html/draft-winter-radsec-01

http://tools.ietf.org/html/draft-ietf-radext-radsec-00

http://tools.ietf.org/html/draft-ietf-radext-tcp-transport-00

http://tools.ietf.org/html/draft-ietf-radext-tcp-transport-01

http://tools.ietf.org/html/draft-ietf-radext-tcp-transport-02

http://tools.ietf.org/html/draft-ietf-radext-tcp-transport-03

http://tools.ietf.org/html/draft-ietf-radext-radsec-01

http://tools.ietf.org/html/draft-ietf-radext-radsec-02

http://tools.ietf.org/html/draft-ietf-radext-radsec-03

http://tools.ietf.org/html/draft-ietf-radext-radsec-04
http://tools.ietf.org/html/draft-ietf-radext-radsec-05
http://tools.ietf.org/html/draft-dekok-radext-dtls-00
http://tools.ietf.org/html/draft-dekok-radext-dtls-01
http://tools.ietf.org/html/draft-winter-dynamic-discovery-00
http://tools.ietf.org/html/draft-ietf-radext-dynamic-discovery-00
http://tools.ietf.org/html/draft-ietf-radext-dynamic-discovery-01

### 2.1.2 Current State of Standardisation

The lifecycle of IETF Internet Drafts consists of multiple stages. The exact procedures are defined in [ref:http://www.rfc-editor.org/rfc/bcp/bcp9.txt], but a brief overview is given here:

1. Documents usually get crafted in an initial version from one or more authors and are submitted as an **Individual Submission**, which is presented to the appropriate IETF working group.
2. If the working group decides to adopt the draft as a work item, it is put onto one out of four document tracks:
   - Informational (FYI).
   - Best Current Practice (BCP).
   - Experimental Standard (EXP)
   - Standards Track (STD).

   It then becomes a **working group draft**.
3. After the working group has worked on the document so that it is near completion, the group will issue a **Working Group Last Call (WGLC)**. In this phase every active participant is asked to provide comments prior to publication. It is common practice that most of the substantial comments about a draft happen not during the preceding working group discussions, but during WGLC.
4. When the document completes WGLC without comments, it is sent out to every IETF participant for comments. This is the **IETF Last Call (IETF LC)**. In this phase, all members of the IETF are asked for their comments regarding this specification. In particular, members of the Internet Engineering Steering Group (IESG) will review the specification and if necessary provide comments which need to be addressed prior to publication.
5. When the Last Calls have been completed and no further comments are received, the document is handed into the **RFC Editor Queue**. The RFC Editor provides final editorial changes, and makes sure that the document that is to be published does not any normative references to work-in-progress.
6. The RFC Editor will **release** the specification as soon as all prerequisites (normative references) have been issued themselves.

The status of the individual documents we are concerned with here is as follows:

#### RADIUS over TCP

This document aims for EXP status and is currently in Working Group Last Call in the RADIUS Extensions Working Group, and is on its sixth revision. The call lasts until 2 October, 2009, and has pending comments. It will be revised by the author after the WGLC.

#### RADIUS over TCP/TLS

This document aims for EXP status and has recently completed a Working Group Last Call in the RADIUS Extensions Working Group with its fourth revision, and received comments. It will be revised and a new WGLC will be initiated.

#### RADIUS over DTLS

This document is currently an Individual Submission by Alan DeKok and aims for EXP status. A working group consensus call of the RADIUS Extensions Working Group on the adoption of the document is currently open.

#### Dynamic Server Discovery for RADIUS

This document has recently been adopted as a working group item of the RADIUS Extensions working group and is in its second revision. It is currently being worked on.

### 2.1.3 Outlook

Since the drafts mentioned before are either already on the charter of the RADIUS Extensions Working Group or about to be adopted, the probability of all of them being standardised eventually is very high. It is difficult though to estimate how long the process takes, since incoming comments during Last Calls happen unpredictably.

## 2.2 Implementation

Having at least two interoperable implementations of an IETF draft is a prerequisite to becoming a standard. GN2 and GN3 have influenced some implementations of RadSec. In addition, several vendors have started implementing the draft. This section provides an overview of the current state of implementations.

### 2.2.1 Implementations Influenced by GÉANT Development

GN2 JRA5 and later GN3 JRA3 T1 have begun two programmes to develop production-quality RadSec implementations. This section describes these efforts in detail.

#### 2.2.1.1 *radsecproxy*

Radsecproxy [radsecproxy] is being developed in GN2 and GN3 by the NORDUnet consortium member. Specifically, Uninett is developing and maintaining the code base.

In the beginning, radsecproxy was envisaged to implement exclusively classical RADIUS over UDP and the RADIUS over TCP/TLS (“RadSec”) portion, which is reflected in the name. As the specifications around RadSec were split up and new transport models were specified, radsecproxy followed these developments and now implements the following RADIUS transports:

- RADIUS over UDP.
- RADIUS over UDP/DTLS.
- RADIUS over TCP.
- RADIUS over TCP/TLS.

radsecproxy can translate RADIUS payload packets from each transport to each other. It also contains a generic method for dynamic peer discovery (by allowing an external script to be called for a request), which makes it a  good testing ground for various dynamic discovery techniques (including using DNS for discovery, or a Meta Data Server (MDS) system as is being developed and deployed in GN3 SA3).

radsecproxy is written in the programming language C. It has a very small footprint, meaning it requires low CPU and memory resources to run. This makes radsecproxy an excellent tool to put on small devices (like Wireless Access Points) to supplement them with RadSec outbound connectivity. Several Linux distributions for embedded devices already include radsecproxy by default in their distribution repositories.

The current version of radsecproxy is 1.3.1, released on 22 July 2009.

#### 2.2.1.2 *FreeRADIUS*

FreeRADIUS [FREERADIUS:] is probably the most popular RADIUS server in use by eduroam Identity providers (IdPs) and Service providers (SPs). It was therefore seen as being of importance to introduce RadSec support into FreeRADIUS to enable community uptake within eduroam. According to a FreeRADIUS self-assessment, this server is also the most popular RADIUS server in the industry, so introducing RadSec support here for general release would have a far-reaching impact in the industry.

First negotiations with the maintainer and main author of FreeRADIUS (Alan DeKok) started end of 2007. It was envisaged to either create a TERENA mini-project, funded by interested NRENs, or issue a third-party contract to the developer under the GÉANT regime, using in-project funding. By March 2008, it was decided to use a GEANT (GN2) third-party contract for the work.

FreeRADIUS is licensed under the GNU General Public License, Version 2. This license seeks to impose restrictions on work that is based upon existing GPL work, by stating that derivative works need to be made available under the exact same license. This made the acceptance of GPLv2 as the license for the work a prerequisite for the delivery of work by the contractor.

The GN2 consortium did not have a licensing policy at that point in time, and thus the contract negotiations excluded a concrete licensing decision; licensing was to be decided by the consortium during the run time of the third-party contract. A first version of an IPR policy was issued in document [IPR] on 12 March 2009. This IPR policy stated that open source licensing principles should be applied to work done in the project. This statement was deemed as not concrete enough though, and did not lead to a clearance of the meanwhile delivered parts of the code for GPL release.

GN3 management ultimately decided on this particular case and has given a clearance for the results to be licensed under GPL on 8 September 2009. Consequently, the first parts of the code (RADIUS over TCP) have already entered the public source repository of FreeRADIUS and are to be released in version 2.1.8, with the rest of the work to follow later as they are coded.

### 2.2.2 Independent Implementations

The value of a standard is, to a large extent, measured by its uptake in the industry. While it may be necessary to fund particular implementations (as GÉANT did in the above-mentioned projects), it is ultimately a goal that independent implementors see the value and implement a standard on their own for the benefit of their customers.

The following sections cannot provide a complete picture of the independent uptake of the set of standards around RadSec, as enterprises are not obliged to report back to GN3 JRA3 on their development work. Therefore, the following list of implementations is only indicative of the take-up of the standard in the IT industry so far.

#### 2.2.2.1 *Radiator*

The Radiator implementation [RADIATOR] is the original RadSec implementation, and is the one upon which the standardisation efforts were built. Therefore, as it is the initial reference implementation, it has a huge deployment base, with the code being used in production at numerous sites.

However, the features in Radiator are not in complete synchrony with the current IETF drafts. This is because the interoperability of Radiator's deployed base is paramount, and experimental intermediate changes to the operational model might cause problems. Therefore, the code is still at the original version of RadSec as was initially released in Radiator.

The code base includes support for RADIUS over UDP, over TCP, and over TCP/TLS. There is also a DNS based server discovery mechanism similar to the one described in the respective IETF draft.

The creators of the software have committed to update the code to a standards-compliant implementation when the final version of the standard has been released.

#### 2.2.2.2 *LCOS*

LCOS (Lancom Operating System) is shipped with the networking gear of Lancom Systems GmbH [LCOS] in Germany. LCOS powers all Access Points, Routers and All-in-One devices in the vendor's product portfolio.

The vendor has integrated support for RadSec in LCOS very early in the standards process. This implementation is, as of yet, the only known closed source and commercial implementation of RadSec in

Wireless LAN Access Points. Since the introduction of the feature, the quality of the implementation has been improved over time, but it is not always on par with the current drafts of the IETF standardisation process.

Some of the All-in-One devices feature a variety of backhaul network connections (for example into 3G data networks). Using such a 3G backhaul and a RadSec connection to the eduroam infrastructure, it is possible to create an autonomous eduroam hotspot even in the absence of a physical network uplink, making an existing mains power connection the only prerequisite for establishing an eduroam cell. This could be used, for example, for "field" deployments of eduroam to enable researchers to have their usual eduroam network connection outside of static university campus setups. A prototype of this, dubbed "eduroam-in-a-powerplug", is available.

#### 2.2.2.3 *Jradius*

Jradius [JRADIUS] is a software component that enables Java applications for RADIUS support. It is most applicable for Java-based web pages that require a user to log in to use a service. Jradius is also often used as a plugin to RADIUS authentication servers to supplement these servers with extended RADIUS functionality which the base product does not offer.

RadSec support for jradius was announced on 25 August 2009 by its author, David Bird.

### 2.2.3 Deployment

It is not possible to gauge the amount of deployment of RadSec on a global level; only the emergence of new implementations can indicate the actual customer demand for this upcoming standard.

Even in eduroam, giving absolute numbers is an impossible task due to the federated nature of the eduroam infrastructure; federations can use RadSec connections to interconnect their own IdPs and SPs without notifying a central authority. However, several countries have reported on a voluntary basis. These reports seem to indicate that RadSec is superseding home-grown solutions (RADIUS over IPSec tunnels, for example) and in some countries makes up the majority of institutional interconnections.

When it comes to inter-federation deployment within eduroam, there is a central authority to issue RadSec TLS certificates, as the current operational model of RadSec in eduroam uses X.509 certificates from one specific Certification Authority, the eduGAIN CA (see GN2 DS5.4.1, Chapter 3.1 for the current operational model). However many of the countries that deploy RadSec within their federation have not started using inter-federation uplinks. This is probably due to the fact the above-mentioned operational model is only an interim one and may be superseded soon.

### 2.2.4 IPR Situation

RadSec started as a feature of a commercial product called "Radiator" by Open Systems Consultants (OSC). As such, a claim on software patents appeared possible. The company made no such claims, and actively contributed in the creation of the IETF standard document. The IETF standards process contains precautions against retroactive filing of IPR claims; if a contributor knows about IPR in the document but fails to mention it, the claims become void. Due to these provisions, it is safe to assume that the protocol is and will remain unencumbered by IPR claims.

At the 75th IETF meeting in Stockholm, reservations were raised about the name “RadSec” itself (this is how the protocol was first called in the original Radiator implementation), since this may be trademarked as a brand name of OSC. As a way out, the actual Standard is going to be called “RADIUS over TCP/TLS” with no mention of the RadSec brand name.

## 2.2.5 Operations and Management Considerations

Apart from the core standardisation work, operational challenges also need to be considered. One question is how to upgrade a production-level service (like eduroam) from RADIUS to RadSec. Another question is how the new protocol can be monitored so that outages are detected and fixed swiftly. The following two sections give an overview of these questions.

### 2.2.5.1 *Upgrading from RADIUS to RadSec*

This section details ways to upgrade production systems in the static eduroam hierarchy to use RadSec instead of RADIUS. The section does not describe how to enable IdPs and SPs for dynamic server discovery, since the operational model of dynamic discovery in eduroam is not decided yet.

eduroam is a distributed and decentralised service. In particular, the RADIUS connections between any of the involved entities (Identity Providers, Service Providers, intermediate proxies) are directly negotiated between any two such entities at their discretion, and are independent of each other. Also there is no requirement for a “flag day” global update, which mans that any IdP can change its transport and connect to its federation proxy as soon as both are updated and configured to understand the RadSec protocol. This is also true for SPs.

The receiving side of the RADIUS peer can be updated asynchronously, and individual connections of IdPs can be changed individually. This enables a smooth transition:

- IdP and federation proxy use RADIUS.
- IdP gets configured to accept incoming RadSec connections; IdP and federation proxy still use RADIUS.
- Federation proxy gets configured to send requests to IdP via RadSec; IdP and federation proxy use RadSec.

Likewise for SP to proxy connections:

- SP and federation proxy use RADIUS.
- Proxy gets configured to accept incoming RadSec connections; SP and federation proxy still use RADIUS.
- SP gets configured to send requests to federation proxy via RadSec; SP and federation proxy use RadSec.
- The update for the federation proxy in step one needs to be done only once, and is then valid for all SPs.

None of the steps in both of the above scenarios are time-critical, and they do not need to be synchronised at both ends.

Changing the uplink from a federation proxy to the European Top-Level RADIUS Servers (ETLR) is a two-phase process; both of the procedures above need to be carried out as the connection between a federation-level server and the ETLR is bidirectional (i.e. the federation-level proxy acts both on the IdP-side as well as on the SP-side). Both procedures can be executed consecutively and do not need to happen in a synchronised manner.

#### 2.2.5.2 *Monitoring*

Monitoring a RadSec connection requires a monitoring tool that connects to a RadSec server instance and verifies that it can process user authentication requests correctly.

Such a tool does not exist yet, and so monitoring RadSec connections has not been introduced in the eduroam infrastructure.

As a part of the FreeRADIUS RadSec contract outcome, the command-line tool radclient from FreeRADIUS will be retrofitted with RadSec capabilities, and it is expected that this tool can fill the gap to introduce at least basic monitoring: it can trigger non-EAP authentication requests only. The full EAP based monitoring as is currently deployed by the eduroam operations team for RADIUS can likely not be achieved with this tool. Another possible option is to use Jradius and write a Java application for monitoring around it.

Finally, it is possible to use a RADIUS client like eapol_test and set up an instance of radsecproxy in front of it to do the conversion.

These options are being investigated by the eduroam operations team in SA3-T2.

# 3 Recognising Users on Re-entry: Chargeable-User-Identity

## 3.1 Problem Description

Certain EAP methods commonly used within the eduroam service consist of two layers:

1. The outer layer serves mainly the routing purposes and contains a User-Name RADIUS attribute, which is called the *outer identity.*
2. The inner layer contains the private authentication data and is readable only to the user's machine and his/her home server.

The SP can only access the outer identity. This mechanism allows users to hide their true identifier by providing, as the outer identity, a value that is called *anonymous identity*. The anonymous identity is typically of the form ***anonymous@realm*** or simply ***@realm***, but the user may also provide any_id@realm (where any_id can be arbitrarily chosen and may happen to be identical to another user's true identifier).

Since the SP can only rely on the outer identity, it has no means of recognising a user on re-entry. The same outer identifier may be utilised by many users, but also the same user may re-appear under a different outer identifier. MAC addresses can be used to identify devices, but these addresses can be changed by users (if the user desires, the change be done before each network authentication).

The ability to use the anonymous outer identity was always one of eduroam's strong points. The fact that the user can be held accountable for their actions does not imply that the user should be forced to release their true identity at every visited wireless network. Not having to handle a user’s private data is also an advantage for SPs, lowering the effort required to run a guest network. However, this feature of eduroam also causes a major problem. With the current eduroam design there are only two effective ways to block a guest user from accessing the network:

1. The SP may block the entire realm (all users from a given IdP).
2. The IdP may block the user access rights for all of eduroam.

Both of these approaches are too coarse for normal use. eduroam needs a method where a user re-appearing at a given network is recognised. When the SP is able to recognise re-appearing users, it not only has means to control (and swiftly limit) access, but may also build usage accounting for the purpose of applying usage quota.

## 3.2 Review of the Current Status

### 3.2.1 Summary of GN2 Study

A solution to the problem was suggested by GN2 JRA5, and that was to take advantage of an existing standard (RFC 4372 – Chargeable User Identity) [RFCCUI]. While the original motivation behind RFC 4372 was to utilise it for charging roaming users, the method of providing an opaque, semi-persistent user identifier is quite universal and fits eduroam needs.

The RFC 4372 defines one attribute, Changeable-User-Identity (CUI). This is the means of passing an identifier within the Access-Accept RADIUS packet. It also defines how this attribute should be utilised. RFC 4372 recognises the need of a business agreement between roaming partners as the basis of the CUI support, but does not state how such an agreement should be realised. The CUI value can only be sent in response to the CUI request; an Access-Request RADIUS packet containing a CUI attribute. Such a business agreement for eduroam could state that such Access-Request messages that request a CUI also contain an additional attribute carrying an identifier of the SP.

Since the CUI has been defined primarily for accounting, RFC 4372 focuses on implementing the CUI requesting in networking equipment directly interacting with the user (a NAS; wireless access point or controller, Ethernet switch, etc.). We are not aware of any existing implementations of CUI in NASes, but for eduroam it is quite sufficient to have an implementation in RADIUS servers on the SP side.

The proposed extension of the eduroam service is to introduce the following scenario:

- During authentication, an Access-Request RADIUS packet is sent from the SP to the IdP. In addition to the usual RADIUS attributes, the packet should contain the CUI attribute with a NUL value (this is the CUI request) and another attribute carrying an identifier of the SP.
- The IdP uses the received SP identifier and the real user identifier as inputs to a hashing algorithm and produces an opaque user identifier, which is sent back as the CUI value in the Access-Accept RADIUS packet.
- The SP will receive the same CUI value every time a given user authenticates. Different SPs will receive different CUI values for the same user.

Figure 3.1 gives a graphical representation of this extension:

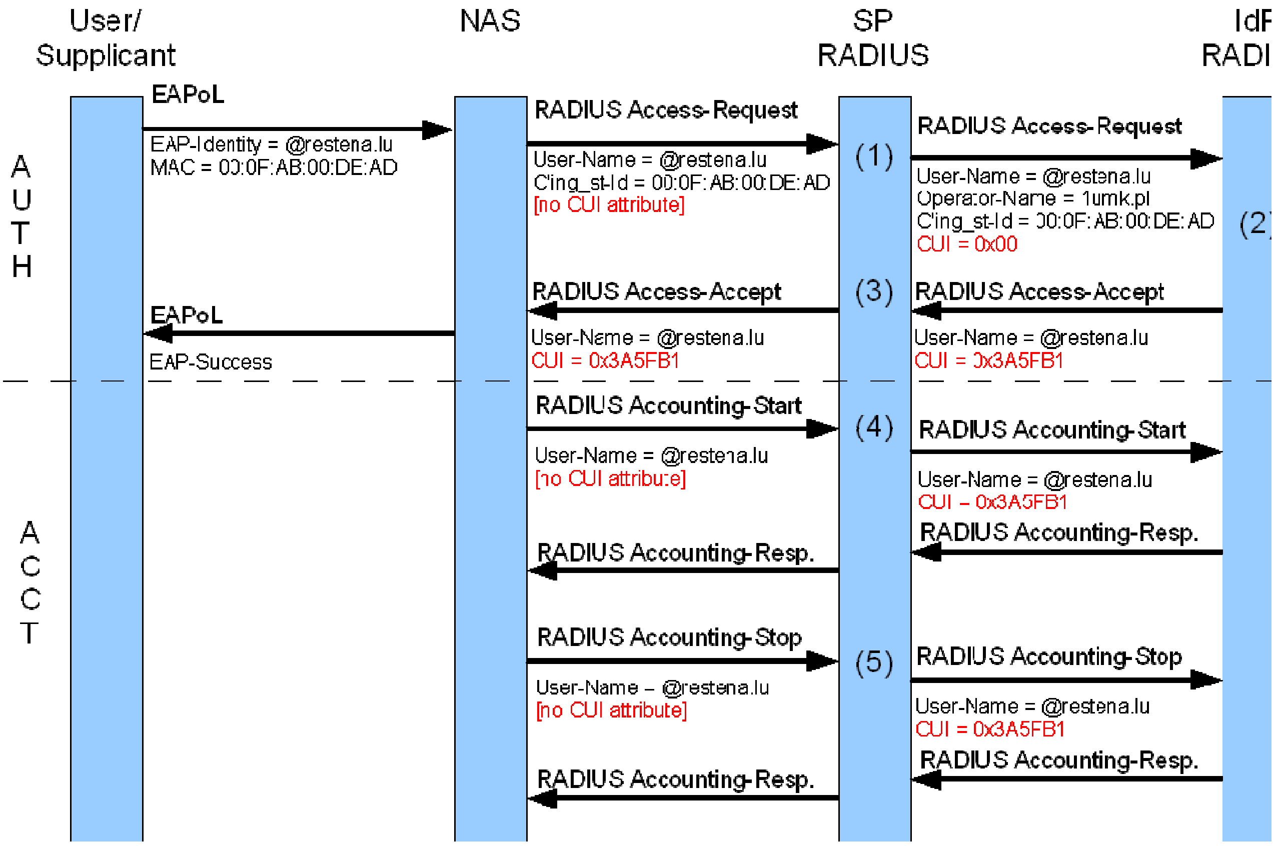

Figure 3.1: CUI extension for eduroam

The implementation must also support accounting. In the scenario described by RFC 4372, the CUI value is received by the NAS, which is also the source of the accounting information. Therefore it is easy for the NAS to correlate the accounting session with the received CUI value and add this value to Accounting-Request packets sent to the RADIUS server. If the implementation is to be done entirely on the RADIUS server, then it is up to this server to do the correlation between received accounting packets and former authentication of the user.

### 3.2.2 Reference Implementation – FreeRADIUS

A proof of concept implementation of CUI support has been carried out by the PIONIER team within the GN2 project. The most popular RADIUS sever software, FreeRADIUS, was used. The implementation was influenced by input from the FreeRADIUS main author, Alan DeKok, and reached its final version in the early stages of the GN3 project. It is currently available in the form of a git fork of the main FreeRADIUS tree and is also distributed to GN3 partners as a patch on FreeRADIUS distributions. This implementation has been used in the production network of Nicolaus Copernicus University, Torun, Poland for several months. It appears stable and satisfies the goals.

The implementation consists of the basic CUI support and an extension done especially for eduroam (using the NAS-Identifier attribute to carry the SP identifier). It also supports accounting by keeping a temporary database of authentication records, which are then matched to corresponding accounting packets.

The implementation did not require any changes to the core of the FreeRADIUS system and is done entirely through configuration files.

### 3.2.3 Testing Tool (eapol_test)

While it is possible to test the CUI support during normal eduroam authentication, it is also very useful to have a testing tool that can simulate a CUI requesting side and display the response of the authenticating server. It is quite standard to use eapol_test (an element of the wpa_supplicant package [WPA]) for both testing and monitoring. This client has also been extended so that it can now test all aspects of the eduroam CUI implementation. The modifications have been accepted by the wpa_supplicant author and added to the distribution.

eapol_test is now capable of sending arbitrary RADIUS attributes in Access-Request packets, therefore it is possible to test if a RADIUS server responds to CUI requests correctly. It is also possible to supplement the authentication process with accounting packets, and test that the accounting support works correctly.

## 3.3 Case Study

Most of the examples provided below originated from everyday experience with eduroam networks. They are not theoretical, but describe real operational needs.

### 3.3.1 Handling of Network Incidents

When an incident requiring immediate action is observed, and is of such a nature that it is imperative that the offending user cannot re-connect, then the SP administrator currently must block the entire offending realm, this way affecting all users from a given IdP.  The connectivity for the offending realm will be restored only after the IdP administrator has identified the guilty user and has blocked the account. Such an action could have quite serious side effects, for example if it occurred during a conference.

eduroam is a world-wide system and it is natural to expect that cultural and political differences will lead to cases where something seen as an offence in one country will not be considered as such in the home institution of a user (IdP). Accessing some politically unpopular network sites can be one example. Since the decision of the IdP to block user's access to eduroam will deny access to the entire service, in certain cases it may be difficult to reach an agreement between the SP and the IdP. On the other hand, the SP should have full powers to enforce its local regulations.

When the SP is capable to recognise the user on re-entry, it is also able to set up individual blocking rules.

### 3.3.2 Identifying and Reacting to the Overuse of Guest Access

It has been observed that eduroam connectivity may be overused, for instance by transferring huge amounts of data or setting up permanent links from residencies. If this is done by guests and not by local users, then the SP has no means to individually measure the traffic for those users. If users are abusing the network on purpose, then it is very likely that they will try to avoid detection by using anonymous identities and frequently changing the MAC address of their devices. Possessing a unique user handle is invaluable in such cases.

It should be noted that this kind of abuse is most likely to happen between institutions which are geographically close (for instance a student of institution A lives in the range of the network of institution B). If the SP suspects such a problem it may come into contact with the IdP in question and agree to use CUI. The proposed technology can be applied and used on a local scale even if it is not yet commonly adopted within the entire eduroam service.

### 3.3.3 Collecting Guest Usage Statistics

Institutions providing eduroam guest access are usually interested in collecting usage statistics (number of guests, amount of data transferred, etc.). Currently the only numbers that can be measured are the number of observed MAC addresses and the number of authentications. It is becoming quite common that users have a few wireless devices connected to the network at the same time. Certain users may be changing MAC addresses of their devices, and a single user who does that on a large scale may totally obscure usage statistics. With CUI, counting users can be precise, and collecting network usage statistics also becomes possible.

### 3.3.4 Providing Additional Services to Guest Users

Since the identification of a user comes from a trusted source, it may be used to grant certain users additional rights in the visited network. If a research guest is to spend a long time at an institution and the only local service required is printing or access to a local network, then instead of the normal practice of creating a local account it may be easier to create a local rule reacting to the CUI value and assigning the user to a privileged VLAN.

There are some obvious complications with this scenario as the CUI value for a concrete user must be identified, but nevertheless this usage of CUI should also be pointed out.

## 3.4 Privacy Considerations

### 3.4.1 eduPersonTargetedId Analogy

The notion of an opaque, persistent identifier is quite standard. eduPerson Object Class Specification [eduPersonOCS] defines eduPersonTargetedId attribute as:

*A persistent, non-reassigned, privacy-preserving identifier for a principal shared between a pair of coordinating entities, denoted by the SAML 2 architectural overview as identity provider and service provider (or a group of service providers). An identity provider uses the appropriate value of this attribute when communicating with a particular service provider or group of service providers, and does not reveal that value to any other service provider except in limited circumstances.*

The implementation used in eduroam should be treated as completely analogous and should be guided by similar rules.

### 3.4.2 Safeguarding Against the Creation of Behavioural User Profiles

The value of the CUI should be different for every SP. This is to stop the collection of data from several SPs and use it to extract information about a users' mobility. Another important argument is that if a globally persistent identifier was used, and if it was accidentally leaked together with the true user's identity, then the user's mobility history would become easily traceable.

It is, therefore, quite obvious that when generating the CUI value, some input representing a given SP must be used. It has been decided to adopt the Operator-Name attribute defined in RFC 5580 [RFC 5580], and send a DNS domain name of the SP. In the case of a direct RadSec connection between the SP and the IdP, instead of the Operator-Name the IdP may choose to use information from the SP RadSec certificate.

RFC 4372 states that normally CUI support should depend on the existence of a business agreement between the roaming partners, and that in the case where the business requirements are not met, the IdP server may refrain from sending the CUI attribute in Access-Accept. The eduroam policy can serve as such a business agreement and can require that the Operator-Name be sent in CUI requests.

### 3.4.3 Dictionary Attack

The CUI implementation must be resistant to dictionary attack. It is assumed that the hashing algorithm used to produce the CUI value may be known. The Operator-Name value is known to the SP. Therefore the IdP must use its own secret salt value as yet another input to the algorithm. Since this value must persist over a long time, it should be impossible to guess and properly secured.

## 3.5 GN3 Roadmap

### 3.5.1 Testing

The implementation prepared by the PIONIER team should be tested using the following scenario:

1. FreeRADIUS extensions must be installed at all testing institutions. A dedicated testing server is recommended, to avoid any disruptions of the production service.

2. Testing should be done both with the eapol_test from wpa_supplicant 0.6.9 (or later) and with normal network authentication through an access-point.
3. All supported EAP methods should be tested.
4. It must be confirmed that:
   a. The SP side of the server sends Access-Request packets containing the NUL CUI value and the Operator-Name filled with the proper value.
   b. A CUI is sent in Access-Accept only when both a CUI and Operator-Name attributes are present in the request.
   c. Different CUI values (for the same user) are returned when the SP identifier is changed.
   d. The expected CUI value for a given user is received (this can only be done on in local tests, when all input value to the hashing algorithm are known to the tester).
5. The accounting logs should be checked to confirm that they contain the proper CUI values.

A test report will be prepared and, if necessary, modifications to the software will be made. The report will be included in deliverable DJ3.1.2,1 – "Roaming developments" in project month 19.

### 3.5.2 Additional Implementation

FreeRADIUS is, by far, the most popular software in eduroam, and is utilised by a vast majority of SPs and IdPs. Therefore, the CUI implementation in FreeRADIUS will already open eduroam for a wide CUI deployment. Radiator from Open System Consultants is another RADIUS server, which has been supported in eduroam from the start. Therefore it is natural to choose it for the second implementation.

Due to license limitations, the initial implementation in Radiator will be done through so-called hooks. This approach is similar to the one utilised in FreeRADIUS.

## 3.6 Deployment

### 3.6.1 Recommendations

The following recommendations are planned:

- Recommendation for individual eduroam institutions: This will be a short description of how institutions can cooperate using CUI, in particular to handle guest network overuse cases. This recommendation will also describe how CUI in FreeRADIUS can be used for better local accounting.
- Recommendations for the eduroam service: A set of recommendations for how CUI should be incorporated into the production service in both the hierarchical and dynamic peer discovery models, what should be added to the eduroam policy, and what additional security precautions need to be taken.
- Informational/Experimental RFC at the IETF: Recommendation for RADIUS implementors: This will describe how CUI is used in eduroam, and what the RADIUS implementation needs to cover to be useful in the eduroam service. The IETF Internet-Draft will be prepared and submitted soon.

### 3.6.2 Monitoring

The eduroam proxy structure can be used to monitor how the CUI is used within eduroam. SPs sending CUI requests and IdPs sending CUI responses should be counted.

It would be very interesting to test IdPs for responding to CUI requests, but unfortunately, for security reasons, most of eduroam server monitoring is done without actual IdP authentication taking place.

# 4 Conclusions

This document gives a summary of two of the key activities undertaken in GN3 JRA3 T1, the RadSec standardisation effort and the preparation of deployment of Chargeable-User-Identifier (CUI).

The RadSec efforts have gone beyond the initial expectations in GN2. Back then, it was hoped to be a single-shot action and that it would be an achievement to pass one document into the IETF process. Over time, the effort has widened in scope, gained a lot of attention in the IETF and is generally recognised as a worthwhile topic to work on. The original one-document approach has been turned in a cohesive document series, and has in the process raised the visibility of eduroam significantly. The developments so far can be considered a significant success. However, work is not finished as the standardisation effort is still underway. Progress on RadSec standardisation will be tracked in further deliverables of GN3 JRA3 T1.

Chargeable-User-Identifier (CUI) existed as an IETF standard but was not readily available in popular RADIUS server implementations. Plus, the specification itself did not consider user privacy to a sufficient extent. The efforts undertaken in GN3 JRA3 T1 have expanded the CUI concept to work in conjunction with another specification, RFC5580, to provide better privacy support. It is expected that these extensions to CUI will be published in an IETF Experimental Standard. The JRA3 team has implemented and tested the CUI attribute, including these extensions, for two of the most popular RADIUS servers in eduroam; FreeRADIUS and Radiator. This prepares the path for a wide-spread introduction of CUI into eduroam. The activities undertaken so far in this area are considered successful.

For both of these eduroam extensions it needs to be noted that JRA3 T1 can only provide options and guidance for eduroam operations. Actual deployment decisions in the production eduroam infrastructure need to be taken by GN3 SA3 T2. JRA3 will present its results to this group in when ready.

# References

| | |
|---|---|
| **[BCP 78]** | http://www.rfc-editor.org/rfc/bcp/bcp78.txt |
| **[eduPersonOCS]** | http://middleware.internet2.edu/eduperson/docs/internet2-mace-dir-eduperson-200806.html |
| **[FREERADIUS]** | http://www.freeradius.org/ |
| **[IPR]** | http://intranet.geant2.net/server/show/conMediaFile.8538 |
| **[JRADIUS]** | http://www.coova.org/JRadius |
| **[LCOS]** | http://www.lancom.de/ |
| [**radextensions**] | http://www.ietf.org/proceedings/69/radext.html |
| **[RADIATOR]** | :http://www.open.com.au/ |
| **[radsecproxy]** | http://software.uninett.no/radsecproxy/ |
| **[RFC 5580]** | http://www.ietf.org/rfc/rfc5580.txt |
| **[RFCCUI]** | http://www.ietf.org/rfc/rfc4372.txt |
| **[WPA]** | http://hostap.epitest.fi/wpa_supplicant/ |
| **[DJ514]** | http://www.eduroam.org/downloads/docs/GN2-06-137v5-Deliverable_DJ5-1-4_Inter-NREN_Roaming_Technical_Specification_20060908164149.pdf |
| **[RADSEC-INITIAL]** | http://tools.ietf.org/html/draft-winter-radsec-00 |

# Glossary

| | |
|---|---|
| BCP | Best Current Practice |
| CPU | Central Processing Unit |
| CUI | Chargeable User Identity |
| DNS | Domain Name System |
| DTLS | Datagram Transport Layer Security |
| ETLR | European Top-Level RADIUS Servers |
| EXP | Experimental Standard |
| FYI | For Your Information |
| IdP | Identity provider |
| IESG | Internet Engineering Steering Group |
| IETF | Internet Engineering Task Force |
| IETF LC | IETF Last Call |
| IPR | Intellectual Property Rights |
| IPSec | Internet Protocol Security |
| LCOS | Lancom Operating System |
| MDS | Meta Data Server |
| OSC | Open Systems Consultants |
| SP | Service Provider |
| STD | Standards Track |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| WGLC | Working Group Last Call |