Chargeable User Identity for eduroam: with FreeRADIUS implementation guide

Author: Scott Armitage

Updated 10/06/2019; 14/07/2021

Chargeable User Identity (CUI) is a single unique identifier for a given user visiting a remote regardless of the outer identity utilised or which device is used to log in. In effect the CUI is an obscured version of the user's real username. This document provides an introduction to CUI and builds on Scott's pre-Networkshop eduroam Forum presentation on Chargeable User Identity in eduroam https://jisc365.sharepoint.com/:p:/s/PublicDocumentLinks/EU_rF9VqLKxCloR50CahOboBHgghIj-O4qfivPy-GJ7oFQ?e=VogA60 [1] ***It also provides instructions for implementing CUI in FreeRADIUS 2. It is hoped that guidance for Radiator deployments will be added in due course. CUI is not supported by Microsoft NPS nor the retired Cisco Secure ACS. For further reading, see RFC 4372.***

How is the CUI useful?

In the world of eduroam (and 802.1X in general) when a user visits another site, the only information the a site has to identify the user is their outer identity and their calling station id (MAC address). Whilst this information is useful (when coupled with authentication event timestamp and IP address) without liaison with the Home site it is insufficient to provide a definitive identifier for the individual. This presents difficulties for the enforcement of AUP and the production of accurate stats.

CUI can help with accountability. If you have a visitor who breaks your AUP and you decide to ban them, how do you enforce this ban? You can't ban the outer identity because i) the user can change that to whatever they like and ii) the same outer identity could be used by multiple visitors (e.g. anonymous@camford.ac.uk [2]) so you may end up blocking other innocent users. If you ban the CSI the user could just login on another device or spoof their CSI. Also, the MAC address may change as the result of the privacy features of the newest operating systems. By having a CUI you can definitively identify the user and block them based on the CUI. Additionally, if you do have an AUP issue when contacting the home site a CUI provides a more reliable link to the user than a CSI which could be spoofed.

CUI can help with measuring the usage of your eduroam service. With only outer identity and CSI, it is difficult to determine how many visitors a site has. For example you could have 10 users all using eduroam with the outer identity anonymous@eduroam.ac.uk [3]. But is that really 10 different users or is it a single user on 10 different devices, or 5 users with 2 devices each etc? CUI solves this as each unique user has a different CUI (and ***the CUI is the same across all of the user's different devices***

).

How do I request a CUI for a user (Visited site)?

If you require a CUI for a user, you simply attach RADIUS attribute 89 (Chargeable-User-Identity) with a null value to the Access-Request.

The Home site authenticating the request should, upon receipt of a NUL Chargeable-User-Identity, generate the CUI value and return it in subsequent replies. This all falls down however if the Home organisation (IdP) has a RADIUS service which doesn't support CUI or hasn’t configured their RADIUS server to respond to CUI requests - you won’t get back a CUI.

If you do get back a CUI for a user, you must then include this in your internal accounting packets and not modify the value.

How do I generate a CUI for a user (Home site)?

If you receive a CUI request RFC 4372 says you should respond with the CUI for the user being authenticated. Whilst the RFC doesn’t specify the method, a CUI must be a transformation of the username. Therefore the recommended method for eduroam is to MD5 hash the username together with a salt(*) and the visited site Operator Name.

Ideally the Visited site ORPS should send their Operator Name (Attribute 126) together with the CUI Request, however in cases where this is absent, the eduroam(UK) NRPSs will inject Operator-Name on behalf of the Visited site using the value in the Identifier field in the Organisation setting panel on the Configure page on Support server.

What does eduroam say about CUI?

The eduroam(UK) Technical Specification states that Visited organisations SHOULD request Chargeable-User-Identity (CUI) in Access-Request packets forwarded to the NRPS if CUI is supported by the ORPS. Home organisations SHOULD respond with a Chargeable-User-Identity (CUI) attribute in an Access-Accept, if the Home RADIUS server supports CUI, where CUI is solicited in the authentication request from the Visited organisation, as described in RFC 4372.

Moreover, a Chargeable-User-Identity response may only be generated by the Home organisation on the condition that the Access-Request from the Visited site contains a non-empty Operator-Name attribute. The value of Chargeable-User-Identity attribute returned in the response MUST have a constant value for each individual user and Operator-Name value. The value of the Chargeable-User-Identity attribute MUST be generated in such a way so as to ensure that the matching of this value to the actual user identity is possible only by the Home site.

(*) 'Salting' is a way of making passwords etc. more secure by adding a random string of characters before the MD5 hash is calculated, which makes it harder to reverse (the longer the random string, the harder you make it).

Implementing CUI with FreeRADIUS 2.2.0

WARNING - These settings should be tested on a suitable test/dev server before implmentation into a live working eduroam service.

Step 1 - Add a salt for generating CUIs to policy.conf

In the policy.conf in /etc/raddb find the CUI section and add a salt for your site. This value should be a long random string which is the same across all of your sites radius servers and should not change over time.

```
        #

        #  The following policies are for the Chargeable-User-
Identity
        #  (CUI) configuration.
        #
        #  The policies below can be called as just 'cui' (not
        #  cui.authorize etc..)  from the various config sections.
        #


        #
        #  cui_hash_key definition
        #  This key serves the purpose of protecting CUI values
against
        #  dictionary attacks, therefore should be chosen as a
"random"
        #  string and kept secret.
        #
        cui_hash_key = "exampleString1234-CHANGE-ME"
```

Step 2 - Add CUI required Flag to policy.conf

In the policy.conf file, after the cui_hash_key, add a new variable called cui_require_operator_name and set it to 1.

```
        #
        # cui_require_operator_name switch
        # If this is set to nonzero value then CUI will only be added
        # when a non-empty Operator-Name value is present in the
request
        #
        cui_require_operator_name = 1
```

Step 3 - Add CUI pre-proxy section to policy.conf

In the policy.conf file, after the cui.authorize section, add a section which will send the a NUL CUI value when proxying (e.g. to the National RADIUS Proxy servers).

```
        #
```

```
        #  Before proxing an Access-Request to a remote server, a
NUL CUI
        #  attribute should be added, unless it is already present
in the request.
        #
        cui.pre-proxy {
                if ("%{Packet-Type}" == Access-Request ) {
                        update proxy-request {
                                Chargeable-User-Identity = '\\000'
                        }
                }
        }
```

Step 4 - Change CUI post-auth section in policy.conf

In policy.conf locate the cui.post-auth section and replace it with the following code:

```
#
#  Add a CUI attribute based on the User-Name, and a secret key
#  known only to this server.
#  For EAP-TTLS and EAP-PEAP methods
#  use_tunneled_reply parameter MUST be set to yes
#
cui.post-auth {
        if (FreeRadius-Proxied-To == 127.0.0.1) {
                if (outer.request:Chargeable-User-Identity && \
                    (outer.request:Operator-Name ||
!("${policy.cui_require_operator_name}"))) {
                        update reply {
                                Chargeable-User-
Identity:="%{md5:${policy.cui_hash_key}%{User-
Name}%{outer.request:Operator-Name:-}}"
                        }
                }
        }
        else {
                if (!("%{control:Proxy-To-Realm}") && \
                    Chargeable-User-Identity && \
                    !(reply:Chargeable-User-Identity) && \
                    (Operator-Name ||
!("${policy.cui_require_operator_name}")) ) {
                        update reply {
                                Chargeable-User-
Identity="%{md5:${policy.cui_hash_key}%{User-Name}%{%{Operator-
Name}:-}}"
                        }
                }
                update reply {
                        User-Name-="%{reply:User-Name}"
                }
                #
                #  The section below will store a CUI for the User
in the DB.
                #  You need to configure the cuisql module and your
database for this to work.
```

```
                #  If your NAS-es can do CUI based accounting
themselves
                #  or you do not care about accounting, comment out
the three lines below.
                #
                #if (reply:Chargeable-User-Identity) {
                #       cuisql
                #}
        }
}
```

Step 5 - Requesting CUI when proxying to eduroam National Roaming Proxy Servers

In the pre-proxy section of your sites-enabled server which handles local reqeusts, call cui. This will call the relevant (pre-proxy) section in the policy.conf

e.g. in /etc/raddb/sites-enabled/camford

```
#
#  When the server decides to proxy a request to a home server,
#  the proxied request is first passed through the pre-proxy
#  stage.  This stage can re-write the request, or decide to
#  cancel the proxy.
#
#  Only a few modules currently have this method.
#
pre-proxy {
#       attr_rewrite

        cui
```

Step 6 - Replying with CUI to incoming requests from eduroam National Roaming Proxy Servers

In the post-auth section of your sites-enabled server which handles the inner-tunnel for requests from the NRPS, call cui. This will call the relevant (post-auth) section in the policy.conf

N.B. Many sites use the same inner-tunnel server (usuall called inner-tunnel) for both request from eduroam NRPS and local radius clients (e.g. wireless controllers)

e.g. in /etc/raddb/sites-enabled/eduroam-inner-tunnel

```
post-auth {

        cui
```

Step 7 - Annoucing availability of CUI to servers sending incoming authentication requests

To annouce to other servers the availablity of CUI from your server you can send back a NUL CUI when responding to an Access-Request (which doesn't contain a NUL CUI itself). This is

done by uncommenting to the cui in the authorize section of the relevant sites-enabled server.

e.g. in /etc/raddb/sites-enabled/eduroam

```
authorize {

        cui
```

Step 8 - Logging CUI Values

In the modules directory create a new linelog module called eduroam_log. This will be used to syslog (to the localmachine) the authentication details.

e.g. /etc/raddb/modules/eduroam_log

```
linelog eduroam_log {
filename = syslog
        format = ""
        reference = "eduroam_log.%{%{reply:Packet-Type}:-format}"
edruoam_log {
      Access-Accept = "eduroam-auth#ORG=%{request:Realm}#USER=%{User-
Name}#CSI=%{Calling-Station-Id}#NAS=%{Called-Station-
Id}#CUI=%{reply:Chargeable-User-Identity}#RESULT=OK#"
             Access-Reject = "eduroam-
auth#ORG=%{request:Realm}#USER=%{User-Name}#CSI=%{Calling-Station-
Id}#NAS=%{Called-Station-Id}#CUI=%{reply:Chargeable-User-
Identity}#RESULT=FAIL#"
        }
}
```

Then call the eduroam_log module in the relevant places. Call it in the post_auth sections of your sites-enabled servers. This should be in both your server for handling authentications coming from local radius clients and the server handling authentications coming from the eduroam NRPS. Note this may need to be called in the inner-tunnel for home users (to log the real username) and in the outer-tunnel for visitors (which aren't authenticated by your radius server so don't appear in the inner-tunnel). e.g.

/etc/raddb/sites-enabled/inner-tunnel

```
post-auth {

        cui

        Post-Auth-Type REJECT {
                attr_filter.access_reject
        }


        #
        # Syslog the login details
        #
        eduroam_log
```

/etc/raddb/sites-enabled/camford

```
post-auth {

        #
        # Syslog the login details
        #
        if ( request:Realm != "local") {
                eduroam_log
        }
```

Source URL: https://community.jisc.ac.uk/library/janet-services-documentation/chargeable-user-identity-eduroam-freeradius-implementation

Links
[1] https://jisc365.sharepoint.com/:p:/s/PublicDocumentLinks/EU_rF9VqLKxCloR50CahOboBHgghIj-O4qfivPy-GJ7oFQ?e=VogA60
[2] mailto:anonymous@camford.ac.uk
[3] mailto:anonymous@eduroam.ac.uk