Blocking non-realm usernames from your eduroam SSID

Blocking non-realm usernames from your eduroam SSID

Contents

Introduction

The eduroam network globally requires the use of usernames in the so-called Network Access Identifier (NAI) format. This format requires usernames to conform to a userID@realm [1] syntax.

The purpose of this format is that the realm element of the format to the sign of the @-sign is used to route the usernames to the appropriate organisation's servers to be authenticated. The lack of a realm element effectively requires the national or the organisational proxy to reject the request because it is known that such a request cannot be routed.

One of the biggest issues that the eduroam network in the UK faces is the proliferation of non-realm usernames, i.e. usernames that do not conform to the NAI format. Examples are below:

fred.bloggs
DOMAIN\fredbloggs
realm!fredbloggs
U0123456
S01234567
host/hostname.ds.university.ac.uk

As can be seen by the examples, none of these contains the crucial @-sign, and thus should not be sent upstream. Permitting your users to connect to eduroam on campus or at their student accomodation handled by your own RADIUS servers using a non-realm username will only lead to a bad user experience when your users doing so attempt to roam off-site (like at other institutions in your location or even abroad) because the realm-less username cannot be routed.

This means that you should reject any requests not containing a realm element outright to enforce the NAI format, even if they are your own users. This should effectively drive the user

to the helpdesk or the student help pages, and your eduroam documentation should explain clearly why the NAI format is necessary. With appropriate tools and settings (such as the eduroam CAT and the geteduroam app together), enforce the correct format from the start and reduce future calls to the helpdesk from off-site roamers (or roamers at student accomodation where eduroam is provided by someone like Glide or ASK4).

For EAP-TLS (certificate-based) authentications, the certificate subject should either contain the commonName (CN) portion of the subject in a NAI format, e.g. ‘fred.bloggs@university.ac.uk [2]’ or ‘hostname@ds.university.ac.uk [3]’, or the appropriate tools provisioning such certificates onto users’ devices must be configured to provide an identity that is in the correct NAI format. The former is preferred.

Instructions

Most RADIUS server software will support a regular expression (regex) in its configuration. Notably, FreeRADIUS, Radiator, Radsecproxy, Cisco Identity Services Engine (ISE) and Aruba ClearPass do this in either their configuration language, their policy, or their rules. The aforementioned RADIUS products/packages also allow for negative matches, i.e. their configuration language, policies or rules allow the negation of a regex match.

The notable exception to the above is Microsoft’s NPS server on Windows. While NPS allows the use of regex conditions in its Connection Request and Network Policy elements in its configuration, NPS does **not** allow for a negative match, and instead a negative lookahead regex syntax needs to be used.

The following instructions will contain rough descriptions of FreeRADIUS, Radiator, and Radsecproxy configuration blocks, which can also be applied to Cisco ISE and Aruba ClearPass as appropriate. The instructions will also contain appropriate instructions for NPS.

FreeRADIUS

FreeRADIUS 3.x contains a policy called ‘filter_username’ in the ‘policy.d/filter’ file in the main FreeRADIUS configuration directory. While this policy checks for the most common errors, it does not check for the absence of a realm element. Instead, you may wish to amend the filter file to add this:

```
filter_username_nai {
    if (&User-Name) {
        #
        #  must have exactly 1 @ (filter_username catches multiples)
        #  e.g. user@site.com
        #
        if (&User-Name !~ /.*@.*/ ) {
            update request {
                &Module-Failure-Message += 'Rejected: No @ in User-
Name'
            }
            reject
        }
        #  must match the 'official' regular expression
        #  e.g. user@site.com
        #
        if (&User-Name !~ /@{1}[-a-zA-Z0-9_]+(\.[-a-zA-Z0-9_]+)+$/ )
```

```
{
            update request {
                &Module-Failure-Message += 'Rejected: Invalid
eduroam username format'
            }
            reject
        }
    }
}
```

Then, in the authorize section(s) of the site file(s) that handle(s) your eduroam SSID authentication, add ‘filter_username_nai’ below an existing (and possibly commented-out) line for ‘filter_username’.

This will reject any non-realm usernames outright with the message that no @-sign exists. Because the existing ‘filter_username’ filter, if not commented out, most likely already deals with things such as multiple consecutive dots or other invalid realm syntax, adding the ‘filter_username_nai’ policy afterwards should enforce the correct format.

You may wish to wrap the line into an if-statement that checks the Called-Station-Id RADIUS attribute (which usually should contain the SSID at the end in a ‘<MAC-Address>:SSID’ format) to qualify that the filter should only run when the SSID is ‘eduroam’ like so:

```
if (&Called-Station-Id =~ /.*\:eduroam$/) {
    filter_username_nai
}
```

Adjust this condition as appropriate for your local style/format, or for the appropriate WLAN equipment attribute that contains the SSID name.

Radsecproxy

radsecproxy uses a very simple format in its configuration, but it does not support multiple conditions, so a set of simple conditions that applies to all requests has to do:

```
realm /@{1}[-a-zA-Z0-9_]+(\.[-a-zA-Z0-9_]+)+$/ {
    server roaming0.ja.net
    server roaming1.ja.net
    server roaming2.ja.net
}

realm * {
    replymessage "Rejected: No @ in User-Name"
}
```

The first condition will specifically only send requests that match the particular format to the eduroam roaming servers, while the condition following will catch all others and reject the requests with the appropriate user message. You can insert any other conditions in between these two realm entries to handle other specific conditions.

Radiator

The Radiator commercial product uses a similar format as radsecproxy, but it does match multiple conditions, a simple format to try is this:

```
<Handler Realm = /@{1}[-a-zA-Z0-9_]+(\.[-a-zA-Z0-9_]+)+$/, Called-
Station-Id = /\:eduroam$/ >
    Identifier eduroam-visitors
    AuthLog AUTHLOG
    AccountingHandled
    AuthBy Proxy-To-eduroamUK
</Handler>

<Handler User-Name = /[^\@]/, Called-Station-Id = /\:eduroam$/ >
    AuthLog AUTHLOG
    <AuthBy INTERNAL>
        AuthResult REJECT
        RejectReason Rejected: No @ in User-Name
    </AuthBy>
</Handler>
```

The first handler will specifically only send requests that match the particular format and where the SSID is eduroam to the eduroam roaming servers, while the handler following will catch any requests for the eduroam SSID where the username does not contain an @-sign and reject the requests with the appropriate user message.

You can insert any other handlers (either Handler or Realm stanzas) in between these two entries to handle other specific conditions. If you don’t need to match the Called-Station-Id, you can simplify the first handler to this:

```
<Realm = /@{1}[-a-zA-Z0-9_]+(\.[-a-zA-Z0-9_]+)+$/ >
```

Cisco ISE and Aruba ClearPass

Cisco ISE and Aruba ClearPass allow you to specify conditions within their rule sets to use matching and non-matching conditions.

For both, the rulesets are evaluated from top to bottom in the sequence, so you should have at least one sequence that forwards visitor traffic to the eduroam NRPS **after** your own domains are evaluated (your domains are usually are first in the sequence). In this instance, you should use a regular expression like this to forward traffic to the service or RADIUS server group:

```
RADIUS attribute User-Name MATCHES (?i)@{1}[-a-zA-Z0-9_]+(\.[-a-zA-
Z0-9_]+)+$
```

If you also need to limit this to the eduroam network, you should add the following condition too:

```
RADIUS attribute Called-Station-Id MATCHES \:eduroam$
```

For rejection of any realm-less users attempting to authenticate over the eduroam SSID, you should probably do the following matches:

```
RADIUS attribute User-Name MATCHES (?i)[^\@]
REJECT
```

Again, should this only be required on the eduroam SSID, use the above Called-Station-Id match.

For Cisco and Aruba equipment, other internal equipment attributes may contain the SSID name, in which case you may want to adjust the Called-Station-Id match to the appropriate internal equipment attribute instead.

NOTE: Aruba ClearPass does case-sensitive matches for regular expressions. This includes matches for your own realms. In this case, please turn your matches into case-**insensitive** matches by adding **(?i)** in front of the regular expression. This reduces accidental loops that pass realm requests from your own users to us when the username is not entirely in lower-case.

Microsoft NPS

In NPS, a default Connection Request Policy called 'Use Windows authentication for all users' exists.

If you installed NPS onto a server (virtual or physical) dedicated only to eduroam and/or govroam, **disable** or **delete** that policy.

If you installed NPS onto a shared server (i.e. you also use it for other purposes, like running other Wi-Fi networks such as a corporate network), you should probably use a match on the RADIUS attribute containing your SSID names (likely to be Called-Station-Id) to ensure that the Connection Request Policy only triggers when the SSID does **not** match ‘eduroam’.

Modify the 'Use Windows…' policy and add a condition by setting the below regular expression matching on the Called-Station-Id:

```
^((?!eduroam).)+$
```

This should check if the Called-Station-Id contains the word 'eduroam' and skip the policy. You should check the event log (in the Server Manager, after selecting 'Informational' events in the Events configuration box) that any realm-less users are rejected if they are attempting to connect to eduroam. If your WLAN equipment uses a different attribute to provide the SSID name, you can use that attribute instead of Called-Station-Id.

Additionally, for those who are using the eduroam CAT system to configure their users and who have enabled user anonymity, or users of Android devices with version 13 or later where the username is set to ‘anonymous’ or ‘anonymous@your [4]-realm’, and who find that authentications fail despite being correct, we suggest adding the user 'anonymous' to your Active Directory but disabling it (so that it merely exists but cannot authenticate).

---

realm-usernames-your-eduroam-ssid

Links
[1] mailto:userID@realm
[2] mailto:fred.bloggs@university.ac.uk
[3] mailto:hostname@ds.university.ac.uk
[4] mailto:anonymous@your