radiator-flr

Radiator

Radiator is perhaps the most popular server software in eduroam federations. The config file and examples below assume deployment on a UNIX-like platform, such as Linux or FreeBSD. Radiator can also be used on Windows; in which case you will have to adapt some path names etc.

> **Use of IP addresses in this document**
> The IPv4 and IPv6 addresses below are in the IETF "documentation" prefix ranges - you will need to adapt the addresses for your production use.

Version information

This section of the document was created and is verified to work with at least

- Radiator 4.7
- Net::SSLeay 1.37 [prerelease]
- Perl 5.10

It is usually safe to assume that newer versions of these programs work as well.

Net::SSLeay 1.37 is the minimum required version for the RADIUS/TLS parts of the config to work completely: the version is needed for the TLS_PolicyOID configuration parameter to work (which is needed for RADIUS/TLS server authorisation checks).

With currently only one CA exclusively issuing eduroam server certificates, the TLS_PolicyOID check is not essential right now.

It is thus also safe to use version 1.36 (and commenting out the configuration lines regarding TLS_PolicyOID). You should upgrade to 1.37 as soon as it is publicly released and re-enable the parameter in the configuration.

Installation

Sample config file

This is the complete sample config. The contents are explained below.

Base configuration / logging / F-Ticks

Radiator expects the configuration to be in file */etc/radiator/radius.cfg*.

The parameter LogDir defines the directory in which start-up logs and PID file reside. DbDir defines the path to Radiator's data files, such as dictionaries.

```
LogDir    /var/log/radiator
DbDir     /usr/share/radiator
```

Throughout the configuration file, you may want to use DNS names instead of IP addresses. For RADIUS/TLS with dynamic discovery, it is even required to use DNS. The configuration for DNS is as follows (replace the IP addresses with your own):

```
<Resolver>
        Nameservers     198.51.100.254
        Nameservers     2001:db8:100::254
        NAPTR-Pattern x-eduroam:(radius)\.(tls)
        DirectAddressLookup 0
        # Debug
</Resolver>
```

The logs during normal operation are defined separately in <Log> stanzas. The verbosity of logging depends on the Trace level in the configuration: Trace 3 logs are recommended for normal operation, while Trace 4 logs provide verbosity for debugging, if needed. You can define several <Log> instances with different destinations. Let's define logging to syslog with verbosity level 3, and logging to a file for debugging purposes with verbosity level 4. We also define that the log file name changes on a daily basis to enable easy deletion of old files:

```
<Log SYSLOG>
      Facility      local7
      Identifier    log-syslog
      Trace         3
</Log>

<Log FILE>
      Filename      /var/log/radiator/radiator.%Y%m%d.log
      Identifier    log-file
      Trace         4
</Log>
```

You can also log authentication events in one line per authentication separately. The eduroam statistics system, F-Ticks, makes use of that feature. The F-Ticks logging facility is defined as follows:

```
<AuthLog SYSLOG>
        Identifier TICKS
        LogSuccess 1
        LogFailure 1
        LogSock udp
        LogHost 198.51.100.253
        SuccessFormat
F-TICKS/eduroam/1.0#REALM=%R#VISCOUNTRY=%{eduroam-SP-Country}#VISINST=%{Op
erator-Name}#CSI=%{Calling-Station-Id}#RESULT=OK#
        FailureFormat
F-TICKS/eduroam/1.0#REALM=%R#VISCOUNTRY=%{eduroam-SP-Country}#VISINST=%{Op
erator-Name}#CSI=%{Calling-Station-Id}#RESULT=FAIL#
</AuthLog>
```

Here, you need to adapt LogHost to the eduroam F-Ticks logging server (whose address you'll receive from eduroam operations), and the attribute marked with read. Its contents will become clearer later in the configuration file. Note: on some versions of Sys::Syslog and Radiator, you may need to reply "udp" with "inet".

If you monitor your national infrastructure, you will probably have automatic authentications happening which are triggered by your monitoring. F-Ticks can automatically separate these from real-world traffic and keep it out of the statistics. For that to work, you will have to use a value for Calling-Station-ID in your monitoring requests which begins with 22-44-66.

Next, the ports Radiator will use to listen for Authentication and Accounting requests must be defined. The port numbers 1812 and 1813 were assigned to the RADIUS protocol by IANA. Note: Exceptionally, you may come across very old RADIUS equipment which uses non-standard ports 1645 and 1646. Please see the Radiator documentation how to handle these, or consider upgrading the corresponding equipment.

```
AuthPort    1812
AcctPort    1813
```

Client definition

In the client section, all possible peers from which the FLR server is going to accept requests, are listed. I.e. it includes all eduroam SPs in the federation and the uplink to the other federations (in Europe, to the ETLR servers).

For RADIUS, individual clients with their IP address have to be listed and a "secret" has to be assigned to them. As this secret is the only thing that protects the communication between the RADIUS servers from eavesdropping, it must be cryptographically strong (suggested: exactly 16 characters) and well protected.

The clients should also be tagged with the attribute Operator-Name. which takes the format "1<domainname>", and for F-Ticks classification reasons, also with the country the eduroam SP is located in (or UNKNOWN for clients whose geographic location isn't known).

Example: you have an eduroam SP which operates on the address 203.0.113.5 and have negotiated the shared secret "adf7856asdcvxb5p" with it. The SP is based in Antarctica, and uses the domain name "foo.aq". You want it to show up in log files as "icecold-radius".

```
# my eduroam SP in Antarctica

<Client 203.0.113.5>
      Secret                    adf7856asdcvxb5p
      Identifier                icecold-radius
      AddToRequestIfNotExist
Operator-Name=1foo.aq,eduroam-SP-Country=AQ
      RequireMessageAuthenticator
</Client>
```

Note: the Operator-Name attribute has the character "1" preceding the domain name. This is intentional and required as per the corresponding RFC. Please always prepend the character "one" to the domain names of the operator.

The clients for your uplink to ETLRs will look similar to the following. Note they are tagged with Country=UNKNOWN because requests coming from these countries can originate from all over the world (they connect **all other federations**). For the same reason, it also does not make sense to set the Operator-Name attribute.

```
<Client etlr1.eduroam.org>
      IdenticalClients          etlr2.eduroam.org
      Secret                    (as negotiated with eduroam OT)
      Identifier                etlr1.eduroam.org
      AddToRequestIfNotExist    eduroam-SP-Country=UNKNOWN
      RequireMessageAuthenticator
</Client>
```

Two additional clients are useful: one client for localhost, which can be used for local debugging purposes (and which doesn't need a strong secret); and the client which used for European FLR monitoring (negotiate the actual client address eduroam OT) at

```
<Client 192.0.2.1>
        Secret                  (as negotiated with eduroam OT)
        Identifier              Monitoring-ETLR
        AddToRequestIfNotExist eduroam-SP-Country=NONE
        RequireMessageAuthenticator
</Client>

<Client localhost>
        Secret                    mysecret
        DupInterval               0
        AddToRequestIfNotExist    eduroam-SP-Country=NONE
        RequireMessageAuthenticator
</Client>
```

Note: all the Identifier names in the configuration need to be unique, and should be meaningful to you, the server operator.

Finally, to enable RADIUS/TLS clients to communicate with your server, you need an additional section for RADIUS/TLS like the following. Replace your server's IP address(es) and paths to the certificate files as necessary - please refer to the "Certificates" section for details on how to obtain and manage RADIUS/TLS certificates.

```
<ServerRADSEC>
        Port                    2083
        BindAddress             198.51.100.252, ipv6:2001:db8:1::26
        Secret                  radsec
        Protocol                tcp
        UseTLS
        TLS_CAPath              /etc/radiator/certs/CAs/current/
        TLS_CertificateFile     /etc/radiator/certs/server.pem
        TLS_CertificateType     PEM
        TLS_PrivateKeyFile      /etc/radiator/certs/server.key
        TLS_PolicyOID           1.3.6.1.4.1.25178.3.1.1
        TLS_RequireClientCert
        Identifier              RadSec
        AddToRequest            eduroam-SP-Country=UNKNOWN
</ServerRADSEC>
```

Request forwarding

Your eduroam IdPs

eduroam authentication requests are routed based on the User-Name attribute in the request. Radiator will extract the *realm* from the User-Name attribute. Radiator uses <Handler> definitions for routing decisions. Even though routing may seem straight-forward since it is based on a single string, it is unfortunately easy to introduce routing loops. Therefore, special care should be taken to prevent this. There are several approaches to that. The one presented here involves regular expressions. The following example shows these, based on the hyptothetical eduroam IdP realm "foo.aq" in Antarctica, and one authoritative RADIUS server for this realm. That same IdP is also an SP and could originate requests. The handler will then look like the following:

```
<Handler Realm=/^foo\.aq$/i,Client-Identifier=/^(?!icecold-radius$)/>
      <AuthBy RADIUS>
               DisableMTUDiscovery
               RetryTimeout          3
               Retries               1
               FailureBackoffTime    300 # (adjust after own needs)
               UseExtendedIds
               <Host 203.0.113.54>
                        AuthPort           1812
                        AcctPort           1813
                        Secret             xxxxxxxxxxxxxxxxxxxxxxxx
               </Host>
      </AuthBy>
      AuthLog TICKS
      AuthLog defaultAuthLog
</Handler>
```

Note the regular expression: it matches only exactly "foo.aq" - not "barfoo.aq" or "foo.aqx". It also contains a safety measure: since the FLR operator can make the link that the realm "foo.aq" is colocated with a eduroam SP whose Client Identifier is "icecold-radius", it can spot that there must be an error if requests for the realm "foo.aq" leave the server in question. Therefore, the Handler clause will only match if the Client-Identifier is NOT "icecold-radius".

If the eduroam IdP provides multiple servers for resiliency reasons, you can specify this in the Handler as well. Please consult the Radiator manual for further details.

Handlers are evaluated in-order, so you should list all known eduroam IdPs one after another in one big block.

You should also add several "catch-all" realms for unknown realms. They are listed below.

Handling empty realms

Empty realms means User-Name requests that do not carry the @... suffix. In a well-behaved eduroam IdP, empty realms should not reach the FLR server (they would be discarded by the IdP already), but if they do, this following realm definition will catch them and reject the request. A reply will be added to the rejected requests explaining the reason for rejection. Replace <TLD> with the federation top-level domain you are authoritative for.

```
<Handler Realm=/^$/>
      AccountingHandled
      <AuthBy INTERNAL>
           DefaultResult REJECT
     RejectReason Misconfigured client: empty realm! Rejected by <TLD>.
   </AuthBy>
      RejectHasReason
      AuthLog defaultAuthLog
</Handler>
```

Unknown realms in the own federation

As the FLR server, your server needs to provide authoritative answers for all possible realms under your TLD. This means that all unknown realms need to be rejected by your server. Failure to do so may lead to routing loops!

Add the following stanza (after your Handler sections for valid realms!) to catch and reject all unknown realms that end in your own TLD (obviously replacing the term TLD with your top-level domain):

```
<Handler Realm=/.*\.tld$/i>
      AccountingHandled
      <AuthBy INTERNAL>
           DefaultResult REJECT
     RejectReason Misconfigured supplicant or downstream server: uses
non-existing realm in <TLD> federation!
   </AuthBy>
   RejectHasReason
      AuthLog TICKS
      AuthLog defaultAuthLog
</Handler>
```

Other known-bad realms

In general, no further second-guessing of incoming realm names should be done. New federations join eduroam every once in a while, and some connected IdPs may reside under "surprising" TLDs (such as .com). That is not a reason to hard-codedly reject all these realms.

However, there are some few well-known, bad, realms that can safely be filtered. The following entry is such an example. For all other realms, please consult the eduroam OT before applying any rejection rules.

One such invalid realm is seen quite often due to supplicant misconfiguration: myabc.com (this is the default realm in an unconfigured Intel PRO/Set Wireless supplicant). The following stanza rejects this realm with an appropriate error message and blindly acknowledges all Accounting requests.

```
<Handler Realm=/myabc\.com$/i>
      AccountingHandled
      <AuthBy INTERNAL>
           DefaultResult REJECT
     RejectReason Misconfigured client: default realm of Intel PRO/Wireless
supplicant!  Rejected by <TLD>.
   </AuthBy>
   RejectHasReason
      AuthLog TICKS
      AuthLog defaultAuthLog
</Handler>
```

Realms from other federations

This is the last Handler rule: it forwards all requests that haven't matched any previous Handler and determines the routing destination. It will first attempt to discover whether there is a direct RADIUS/TLS soute to the destination realm's server, and if not, route the request to the ETLRs.

```
<Handler User-Name = /\@/>
        <AuthBy DNSROAM>
                Port                2083
                Protocol            radsec
                Transport           tcp
                UseTLS              1
                Secret              radsec
                ReconnectTimeout    1
                NoreplyTimeout      5
                ConnectOnDemand
                TLS_CAPath          /etc/radiator/certs/CAs/current/
                TLS_CertificateFile /etc/radiator/certs/server.pem
                TLS_CertificateType PEM
                TLS_PrivateKeyFile  /etc/radiator/certs/server.key
                TLS_PolicyOID       .1.3.6.1.4.1.25178.3.1.2
                TLS_ExpectedPeerName CN=.*
                <Route>
                        Realm DEFAULT
                        Address etlr1.eduroam.org
                        Port 2083
                        Transport tcp
                        Protocol radsec
                </Route>
        </AuthBy>
        AuthLog TICKS
</Handler>
```

Replace your paths to the certificate files as necessary - please refer to the "Certificates" section for details on how to obtain and manage RADIUS/TLS certificates.

Goodies

Local logging of auths in one line

It is useful to log each authentication locally, with more detail than is needed for F-Ticks. We suggest using the following log definition – it generates one single line of log output per authentication, which is very parser-friendly if logs need to be evaluated later:

```
<AuthLog SYSLOG>
        Identifier              defaultAuthLog
        Facility                local7
        LogIdent                radiator
        FailureFormat           Access-Reject for %u
(User-Name=%{Reply:User-Name}) at Proxy=%c
(CSI=%{Calling-Station-Id}NAS=%{NAS-Identifier}/%N)
        SuccessFormat           Access-Accept for %u
(User-Name=%{Reply:User-Name}) at Proxy=%c
(CSI=%{Calling-Station-Id}NAS=%{NAS-Identifier}/%N)
EAP=%{HexAddress:EAP-Message}
        LogSuccess              1
        LogFailure              1
</AuthLog>
```

SNMP

You may want to configure SNMP access to your server. SNMP allows remote monitoring of activity on a RADIUS server with tools such as RADAR from OSC (http://www.open.com.au/radar/index.html), or drawing simple graphs of activity by rgraph from CESNET (http://www.eduroam.cz/rgraph/).

```
<SNMPAgent>
      ROCommunity xxxxxxxxxxxxxxxxxxxxxxxx
      Managers        localhost 127.0.0.1
</SNMPAgent>
```

Caveats

The previous sections have referenced two specific RADIUS attributes, "Operator-Name" and "eduroam-SP-Country". In Radiator 4.7, these attributes aren't shipped by default and need to be registered in the server's so-called "dictionary".

Operator-Name, and a few more attributes, is defined in the IETF document RFC5580. The definitions in there are canonical, but they clash with the dictionary that's shipped with Radiator, so you will have to remove a few bogus entries, and then add the correct definitions. Please open the file "dictionary" and make the following edits:

Delete the following bogus entries near line 230 of the dictionary file:

```
ATTRIBUTE       Ascend-Route-Preference         126     integer
ATTRIBUTE       Tunneling-Protocol              127     integer
ATTRIBUTE       Ascend-Shared-Profile-Enable    128     integer
ATTRIBUTE       Ascend-Primary-Home-Agent       129     string
ATTRIBUTE       Ascend-Secondary-Home-Agent     130     string
ATTRIBUTE       Ascend-Dialout-Allowed          131     integer
ATTRIBUTE       Ascend-Client-Gateway           132     ipaddr
```

Replace them with the following definitions:

```
ATTRIBUTE       Operator-Name                   126     string
ATTRIBUTE       Location-Information            127     string
ATTRIBUTE       Location-Data                   128     string
ATTRIBUTE       Basic-Location-Policy-Rules     129     string
ATTRIBUTE       Extended-Location-Policy-Rules  130     string
ATTRIBUTE       Location-Capable                131     integer
ATTRIBUTE       Requested-Location-Info         132     integer
```

The attribute eduroam-SP-Country is a custom extension, a so-called "vendor-specific" attribute. It is registered under the namespace of TERENA. Please add the following definition at the end of the dictionary file if you use a version of Radiator BEFORE 4.9 with the patchset of 04 April 2012. For newer versions of Radiator, this attribute is already shipped by default with the server and you do not have to change anything.

```
# TERENA VSAs
#
VENDOR      TERENA      25178
VENDORATTR  25178       eduroam-SP-Country              10      string
```