Configuring an eduroam heartbeat for your visitor network

Updated 25/09/2024

Contents

Introduction - what is an eduroam heartbeat

Ensuring that your eduroam network is functional for eduroam visitors is an important part of being a member of eduroam, after all, if visitors cannot connect to eduroam at your institution, it leads to dissatisfaction not only with your network but also with eduroam itself. Many of our members employ an eduroam service heartbeat as part of their network monitoring. The ideal solution is for the administrator to be alerted should an interruption to the service occur for whatever reason (RADIUS issue, internet connection outage, NRPS issue etc). While a heartbeat is optional, we have found that where an organisation does have a heartbeat, when a problem has arisen the organisation has rapidly become aware and has been able to respond quickly to restore service.

Some monitoring software comes with such functionality as a module that just needs to be enabled and configured, other software needs some extra things to make it work. The ever popular Windows alas generally does not have such functionality, so Windows NPS server administrators are left in the dark about such a vital part of their service. In this article we describe solutions for both Windows and linux platforms.

So how does an eduroam heartbeat work? The eduroam heartbeat is effectively a regular test to see whether a simulated visitor onto your network can reach their home server. The visitor test function of Support Server provides you with a unique test account on 'eduroam.ac.uk' and you can use this together with eapol_test and routine scheduling software to create the heartbeat monitor. Credentials for the visitor simulation test account can be found on your 'Troubleshoot' page of the eduroam Support portal > blue Tests panel > Visiting authentication test box.

How to configure an eduroam heartbeat

Generally, configuring a heartbeat is fairly simple. A script or a batch file running in the background at regular intervals (every 10, 15, 30, 60 minutes perhaps) can provide you with forewarning that something is not quite right with your service.

On Windows, a regular Windows task can write into the event log, which in turn you can interrogate for the specific event to see when the problem started.

On Linux, a bash script that simply executes the eapol_test binary (which is part of the wpa_supplicant packages on most Linux platforms) with a configuration file and which evaluates the outcome of the execution is sufficient, provided you monitor your system logs (syslogs). If you use Nagios or something similar, you can get away with using the excellent rad_eap_test [1] script instead. rad_eap_test accepts the test username, test password, the server IP address and secret on the command-line instead of a configuration file, and it returns a simple error code to indicate success or failure. It requires the following utilities (some of which are installed by default):

```
eapol_test
dig
bc
sed
awk
```

After a successful run, you should see a successful authentication in your RADIUS server logs for the eduroam.ac.uk visited authentication test user.

Preparation

Ideally, you should run this heartbeat on the same server as your RADIUS software. On Windows, this means running the task and the batchfile on the same server that your NPS instance is running on. You can however also run the script from another server if you so prefer, but be aware that the success or failure is written into its event or system log instead.

You must also define a RADIUS client in your RADIUS server software with the IP address from where you run the script as the client address. So if the IP address from where you run the heartbeat script is 192.168.23.45, define a client with the IP 192.168.23.45. Give it a very simple and straight-forward secret. The FreeRADIUS project likes the secret 'testing123' for obvious reasons. You might want to choose a different one.

Note down the secret you just used for the client and the IP address for your RADIUS server. You will need these.

The configuration file for both the below Linux script and the Windows batch file further along is the standard eapol_test configuration. **Note: This configuration assumes that your server understands TLS v1.2. If it does not, adjust the first line by setting the tlsv1_0 or tlsv1_1 options to zero, and also schedule an upgrade for your server, as TLS v1.0 and v1.1 are deprecated and ideally must not be used**:

```
network={
phase1="tls_disable_tlsv1_0=1 tls_disable_tlsv1_1=1
tls_disable_tlsv1_2=0 tls_disable_tlsv1_3=1 peapver=0"
key_mgmt=WPA-EAP
eap=PEAP
identity="<your visiting test username>@eduroam.ac.uk"
anonymous_identity="@eduroam.ac.uk"
password="<your visiting test password>"
phase2="eapauth=MSCHAPV2 mschapv2_retry=0"
}
```

Nb. <Your visiting test username/password> for your organisation can be found on Support Server > Troubleshoot > Tests panel > Visiting authentication test box.

For the sake of the examples below, we shall call this file 'eapol_peap.cfg' and store it in /opt/eduroamHB (on Linux) or C:\eduroamHB (on Windows).

Linux

If you do not use Nagios, or you are unable to use the rad_eap_test script mentioned further up, you can use this instead. This is a very simple bash script that will log to the syslog if there was a failure or not. Call it eduroam_heartbeat_check.sh and store it alongside the config file:

```
#!/bin/sh
# Run a heartbeat
ip=<the IP address for your server>
s=<the secret for the client you created>
bssid=<the BSSID for your eduroam network, in a MAC format like
02:00:00:00:00:01>
# check that eapol_test works
if [[ -x /sbin/eapol_test ]]; then
  if [[ -f /opt/eduroamHB/eapol_peap.cfg ]]; then
    # get the actual output
    i=$(/sbin/eapol_test -c /opt/eduroamHB/eapol_peap.cfg -N
30:s:$bssid:eduroam -N 32:s:eduroamUK-heartbeat -t 5 -r 1 -a $ip -s
$s |tail -1)
    # output is either success or failure
    /bin/logger eduroam Heartbeat: $i
  fi
fi
```

The result of this script is not a success or failure errorcode, but rather a line in the system log with the text 'eduroam Heartbeat: <result>', along with a successful authentication in your RADIUS server logs for the eduroam.ac.uk visited authentication test user.

The assumption of the above script is that you a) run the script on the same server as your RADIUS software, and b) that eapol_test lives in /sbin. **Adjust these locations in the script accordingly.**

This script requires the following utilities (some of which are preinstalled):

```
eapol_test
tail
logger
```

You can run this script by either adding a line into a crontab for your monitoring user, or you can copy the eduroam_heartbeat_check.sh into /etc/cron.hourly for an hourly run. Alternatively, if you prefer a more frequent run, add a file into /etc/cron.d/ with this contents:

```
# Run the ten-minute jobs
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=monitor
*/10 * * * * monitor /opt/eduroamHB/eduroam_heartbeat_check.sh
```

Adjust 'monitor' to your monitoring user if you have one (ideally, you should not run such commands as root as it's not necessary). Restart crond with the 'service crond restart' command.

To test your server's connection to the roaming servers, you can use the same script and replace the server IP and secret with the NRPS of your choice, along with its secret. This effectively skips your server and tries for the NRPS directly instead.

Windows

Windows traditionally does not have a process to monitor NPS. As with the Linux script, you will need a configuration file, and the eapol_test utility, which you can download here [2].

We have built, code-signed and published it for Windows, so after downloading, you can right-click the executable, click Properties, and then examine the code signing certificate. It should be signed around March 21 2021.

As with Linux, you need a configuration file (see above). The extension of the configuration file is not important, but do note down the location and name because you will need it. In the example code, I assume C:\eduroamHB as the location, and thus I'll use C:\eduroamHB\eapol_peap.cfg as name.

To run eapol_test.exe, place it in the same directory as eapol_peap.cfg (or your chosen name). Then, in the same location, create a batchfile with this contents:

```
@echo off
set ROAMING_IP="<the IP address for your server>"
set ROAMING_SECRET="<the secret for the client you created>"
set BSSID="<the BSSID for your eduroam network, in a MAC format like
02:00:00:00:00:01>"
cd C:\eduroamHB
eapol_test.exe -t6 -N 33:x:4f53432d457874656e6465642d49643d31323435 -
N 30:s:%BSSID%:eduroam -N 32:s:eduroamUK-heartbeat -c
eapol_peap.cfg -a %ROAMING_IP% -s %ROAMING_SECRET% |findstr /R
"^SUCCESS$" >nul 2>nul
set MYVAR=%errorlevel%
IF "%MYVAR%" == "0" GOTO EventSuccess
eventcreate /Id 2 /D "eduroam Heartbeat: FAILURE" /T ERROR /L system
```

```
/SO eduroamHeartBeat >nul 2>nul
GOTO End
:EventSuccess
eventcreate /Id 1 /D "eduroam Heartbeat: SUCCESS" /T SUCCESS /L
system /SO eduroamHeartBeat >nul 2>nul
:End
```

As before, adjust the location of the file and the directories in the batchfile accordingly. You can also use this batchfile to test your connection to the roaming servers by adjusting the ROAMING_IP and ROAMING_SECRET settings to the IP of the NRPS of your choice, along with its secret. This will skip your server and test your external connection directly instead.

To create a scheduled task that runs every ten minutes, you should execute this command as an administrator on your server that the batchfile runs from:

```
SCHTASKS /Create /U <domain\user> /P <password> /SC MINUTE /MO 10
/TN eduroamHeartBeat /TR "C:\eduroamHB\eduroamHB.bat" /RU NT
AUTHORITY\SYSTEM
```

Adjust the /U and /P parameters to an admin user and its password. After every run, you should see an event in the 'System' log in the Event Viewer of your server called 'eduroamHeartBeat'. An 'Information' type message means it will be a success message. An 'Error' type message will contain an error. Then check your event log regularly to see whether your external connection is still 'up'.

Links
[1] https://github.com/CESNET/rad_eap_test
[2] https://github.com/janetuk/eapol_test/releases