

configurations for Member organization

15 August 2019

Minseok Jang (msjang@kisti.re.kr)

Researcher

KREONET Center / Div. of National Supercomputing

Korea Institute of Science and Technology Information (KISTI)



Institutional Technical Pre-Requisite

- Identity Management
 - LDAP, Active Directory, IMAP
 - REALM supported
 - Test Account
- WiFi Infrastructure
 - Coverage Map
 - IEEE 802.1x
- Network Access
 - ISP, if not NREN
 - AUP(Acceptable Use Policy)
- IT Support
 - HelpDesk

Institutional Wireless Infrastructure

- 802.1x capable
- Broadcast of 'eduroam' SSID
 - Whole word must be in lower case
 - Issue of overlapping hotspots
 - e.g. eduroam-uniX
 - Potential of IEEE 802.11u to alleviate issue
- Wireless coverage across campuses
 - Availability of coverage maps
- Encryption support (WPA2-Enterprise required) Tunneled EAP protocol
 - Choice of inner-authentication depending on identity store
- Limitations of WiFi Supplicants on various platforms

Institutional RADIUS Server (IRS) 1/2

Requirements for proxied RADIUS requests

- RADIUS attributes
 - Filtering to release only required attributes
 - Generate Chargeable-User-Identity (CUI) for IdP role
 - Include Operator-Name (ON) for SP role
- Terminate (i.e. don't proxy) accounting requests
- Configure bogus realm black-listing
 - E.g. *.3gppnetwork.org
- Reject request with badly formatted username (e.g. invalid characters)
- Use Status-Server to handle non-responsive servers

Institutional RADIUS Server (IRS) 2/2

- Deployment considerations
 - Time synchronization (NTP)
 - Server redundancy (high-availability deployment)
 - Load balancing
- RADIUS request logging
 - Trust model - ability to trace a network access to a real user
 - Traceability of users achieved by logging
 - Attributes to be logged
 - DateTimeStamp, User-Name (outer EAP identity) i.e. realm, Chargeable-User-Identifier, NAS-IP-Address (IP address of adjacent RADIUS client, i.e. NAS or proxy server), Operator-Name (identifier for institution providing network access), Calling-Station-ID (MAC address of user device), authentication response (Accept or Reject)

Today's Session

Overview

- Install Prerequisite for Trainee's PC
- Install IRS using ansible
- Test Local and Remote Account
- Add More Test Users
- Add a Realm to BlackHole Routing
- Check Client Secret
- Check NRS Settings in Client and Proxy Configs
- Compare Configs
- Compare Local and Remote Auth Flow

(For Left Time)

- Add Blackhole Routing
- ...

Install Prerequisite for Trainee's PC

- Terminal SW
 - [MobaXterm](#) for Windows
 - [iTerm](#) for OSX
 - Basic Terminal for Linux
- [WireShark](#) for All OS
 - Packet Capturing and Analyzing Tool
 - (In this session) RADIUS packets captured by tcpdump, downloaded to PC, will be opened with WireShark
- Code Compare and Merge Tool; Meld for All OS
 - [Windows : download and install](#)
 - [OSX : download and install](#)
 - Linux : intall using YUM or APT
- TextEditor ([Atom](#), [Sublime](#), ...) for All OS
 - (Mandatory for Windows) NotePad in Windows cannot open Linux config file correctly

NOTICE / IMPORTANT

This Session for IRS (Institutional RADIUS Server).

Connect to IRS, NOT NRS !

Practice the following slides in IRS !

Install IRS using ansible 1/2

NOTICE

- eduroam-imap-playbook
 - Ansible scripts for eduroam deployment (Install, Config, Test)
 - [Authored by SAFIER](#), [Modified by KROENET](#) under [MIT License](#)
 - Presented at TNC18 - [eduroam for GSuite users? Yes you can!](#)
- Check Linux Version - Ubuntu 18.04

```
# sudo cat /etc/issue  
Ubuntu 18.04 LTS
```

- Install GIT, Ansible

```
sudo apt update;  
sudo apt -y install git ansible
```

- Clone Ansible Script

```
git clone https://github.com/msjang/eduroam-imap-playbook
```

Install IRS using ansible 2/2

- IRS Settings - Change Ansible Group Variable

Change the following file; [group_vars/all](#). It is used to create IRS configs.

```
cd eduroam-imap-playbook;  
vi group_vars/all
```

```
# The upstream eduroam federation-level RADIUS servers  
# These are for sample, not working; get yours from your NRO  
eduroam_flr_servers:  
  - hostname: eduroam.kr  
    ip: 138.44.179.8  
    port: 1812  
    secret: secret_key_between_nro_and_irs  
  
# Your realm for eduroam (usually your primary DNS name)  
radius_realm: xep.kr  
  
# Details of test account(s) to create within your realm  
radius_local_users:  
  - username: eduroam  
    password: xep2018  
  
...
```

- Install IRS - Run Ansible

```
ansible-playbook -i inventories/development site.yml
```

Test Local and Remote Account

- rad_eap_test

```
# rad_eap_test
Parameters :
-H <address> - Address of radius server
-P <port> - Port of radius server
...
```

- Test Local Account

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam \
-e TTLS -2 PAP -u eduroam@xeap.kr -p xeap2018
access-accept; 0
```

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam \
-e TTLS -2 PAP -u eduroam@xeap.kr -p xeap2019
access-reject; 0
```

- Test Remote Account

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam \
-e TTLS -2 PAP -u eduroam@xeap.au -p xeap2018
access-accept; 0
```

Add More Test Users

- Add Test User to Config File; ID: bob, PW: hello

```
sudo vi /etc/freeradius/3.0/users
```

```
...
# You can include local (non-PAM) users like this:
# icecold      Realm == "kisti.re.kr", Cleartext-Password := "snowwhite"
# otheruser    Realm == "kisti.re.kr", Cleartext-Password := "swordfish", Expiration := "25 May 2099"

eduroam       Realm == "kisti.re.kr", Cleartext-Password := "xeap2018"
bob           Realm == "kisti.re.kr", Cleartext-Password := "hello"

# This catches all remaining users and sends them to PAM
DEFAULT      Virtual-Server == eduroam-inner-tunnel, Pam-Auth := "pam-imap-radius", Auth-Type := PAM
...
```

- Restart FreeRadius

```
sudo service freeradius restart
```

- Test New User

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam \
  -e TTLS -2 PAP -u bob@xeap.kr -p hello
access-accept; 0
```

Add a Realm to BlackHole Routing

- Add a Realm to BlackList; xead.au

```
sudo vi /etc/freeradius/3.0/proxy.conf
```

```
...
# blackhole routing - EAP-SIM/MN0s
realm "~\\.3gppnetwork\\.org$" {
    nostrip
}

# realm xead.au {
    nostrip
}

...
```

- Restart FreeRadius

```
sudo service freeradius restart
```

- Test New User

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam \
-e TTLS -2 PAP -u eduroam@xeap.au -p xead2018
access-reject; 0
```

Check Client Secret

- Secret, `testing123`, is used for RADIUS test in `localhost`
Secret is used for each RADIUS communications between two entities.

```
# rad_eap_test -H localhost -P 1812 -S testing123 -m WPA-EAP -s eduroam ...
```

- Secret is defined in `clients.conf` for each client addresses.

```
function PRINT(){ grep -v '#' $1 | sed -e '/^ *$/d'; };  
PRINT /etc/freeradius/3.0/clients.conf
```

```
# PRINT /etc/freeradius/3.0/clients.conf  
client localhost {  
    ipaddr = 127.0.0.1  
    proto = *  
    secret = testing123  
    require_message_authenticator = no  
    limit {  
        max_connections = 16  
        lifetime = 0  
        idle_timeout = 30  
    }  
}  
client localhost_ipv6 {  
    ipv6addr = ::1  
    secret = testing123  
}  
$INCLUDE clients-eduroam-flrs.conf
```

Check NRS Settings in Client and Proxy Configs

- CASE 1 : USER → IRS → NRS → ...
 - RADIUS Client : IRS
 - RADIUS Server : NRS
- CASE 2 : USER → ... → NRS → IRS
 - RADIUS Client : NRS
 - RADIUS Server : IRS
- Settings for NRS can be found both clients.conf and proxy.conf in IRS

```
# PRINT /etc/freeradius/3.0/clients-eduroam-flrs.conf
...
client NRS {
    ipaddr = 150.183.96.51
    secret = MySharedSecret
    require_message_authenticator = yes
    shortname = NRS
    nastype = other
    virtual_server = eduroam
}
...
```

```
# PRINT /etc/freeradius/3.0/proxy.conf
...
home_server NRS {
    type = auth+acct
    ipaddr = 150.183.96.51
    secret = MySharedSecret
    port = 1812
    require_message_authenticator = yes
    status_check = status-server
}
...
home_server_pool EDUROAM {
    type = fail-over
    home_server = NRS
    ...
}
...
realm "~.+$" {
    pool = EDUROAM
    nostrip
}
...
```


Compare Configs 1/4

- Check **Running** and **Original (backed up)** Configs

```
# sudo ls /etc/freeradius/  
3.0      backup_3.0
```

- ZIP config

```
sudo tar zcf /home/kisti/config.tgz /etc/freeradius/
```

- Download config.tgz to Your PC

- (Windows) Use MobaXTerm
- (Linux, OSX)

```
YOUR_PC# scp <ID>@<SERVER_IP>:/home/kisti/config.tgz .
```

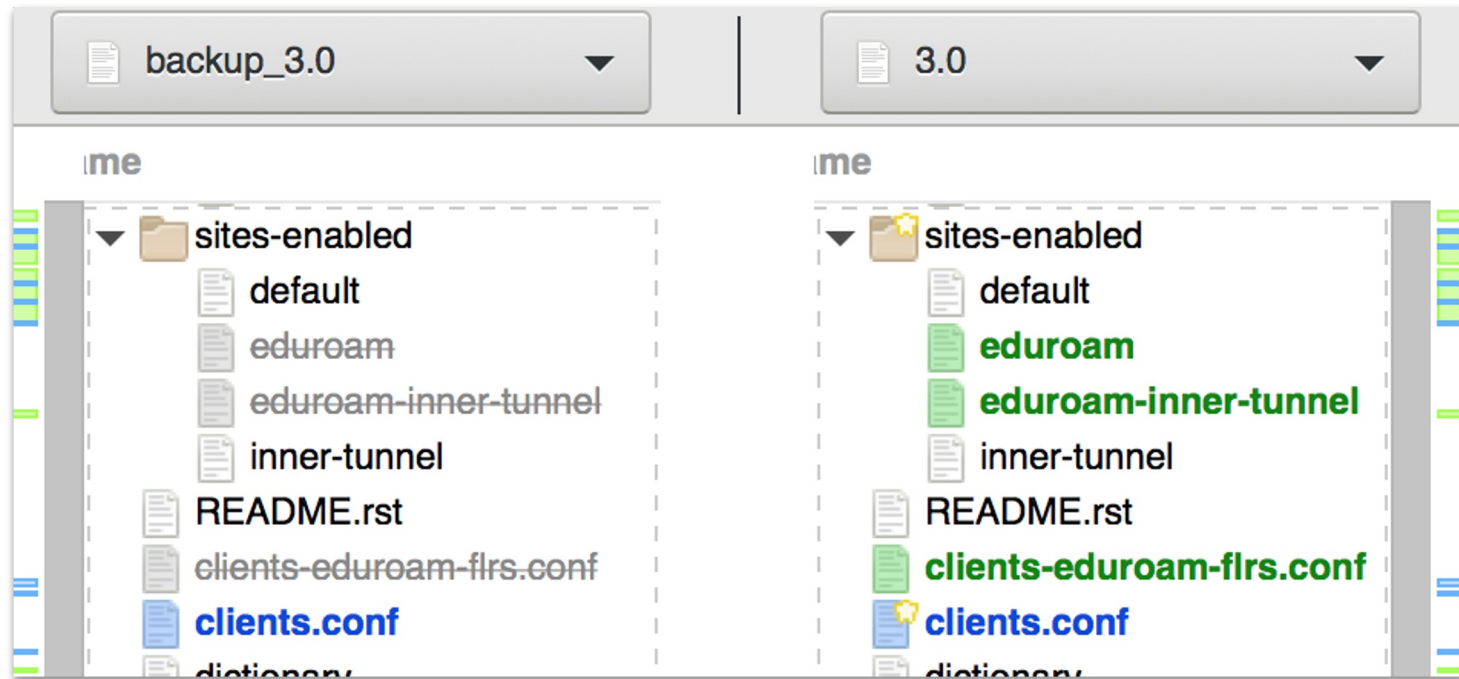
- UnZIP configs.tgz

- (Windows) Use Default UnArchiver Program, or [BandiZip](#)
- (Linux, OSX)

```
YOUR_PC# tar xzf config.tgz
```

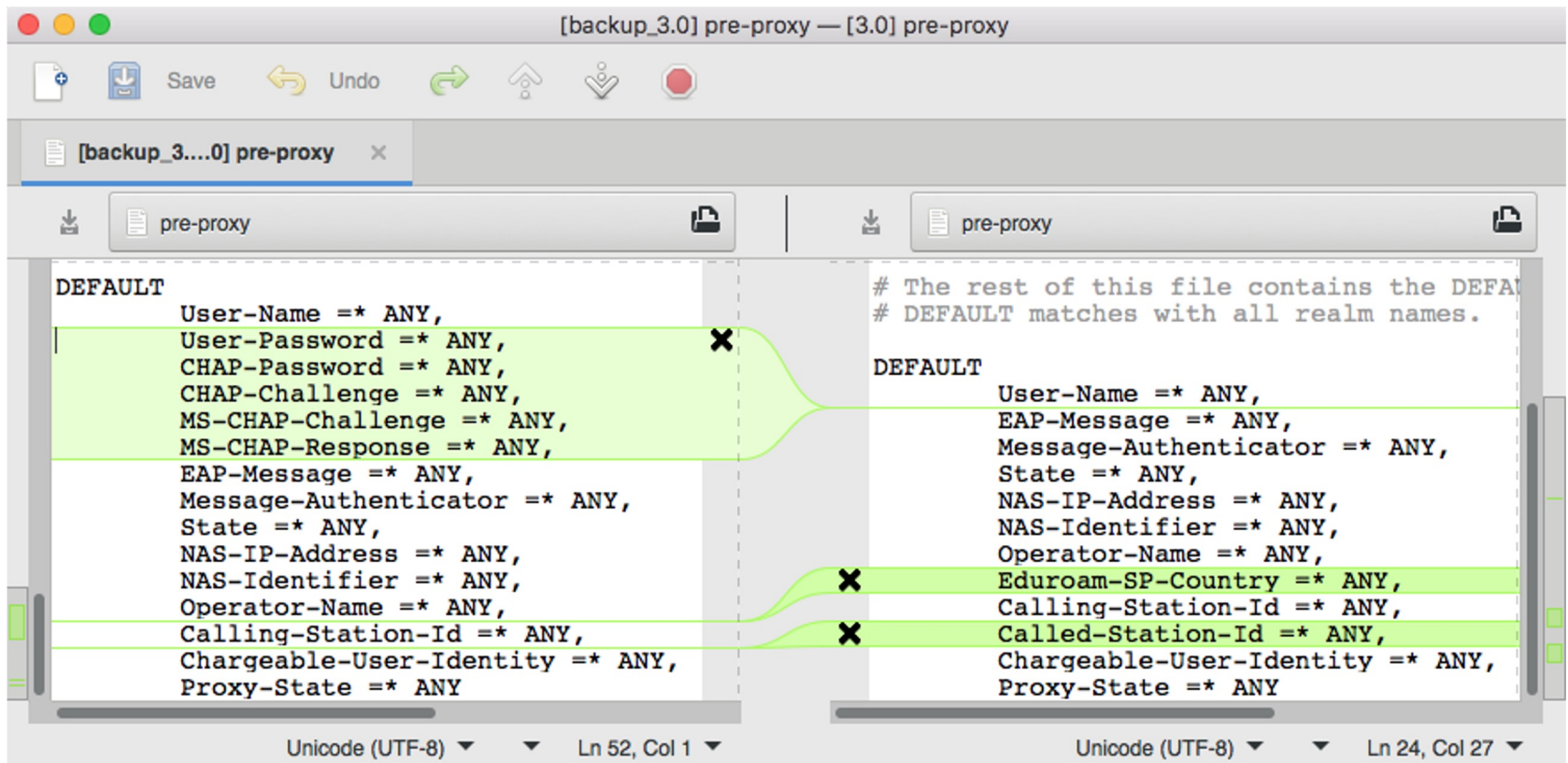
Compare Configs 2/4

- Compare Directories; Original vs Current



Compare Configs 3/4

- Compare pre-proxy; Original vs Current
 - /etc/./mods-config/attr_filter/pre-proxy



Compare Configs 4/4

- Compare proxy.conf; Ansible Template vs Current
 - Before compare, download and unarchive [eduroam-imap-playbook](#) to your PC

```
---
# The upstream eduroam federation-level RADII
# These are for sample, not working; get your
eduroam_flr_servers:
- hostname: flr1-example.kreonet.net
  ip: 150.183.96.51
  port: 1812
  secret: MySharedSecret
- hostname: flr2-example.kreonet.net
  ip: 150.183.96.52
  port: 1812
  secret: MySharedSecret

# Your realm for eduroam (usually your primary)
radius_realm: kisti.re.kr

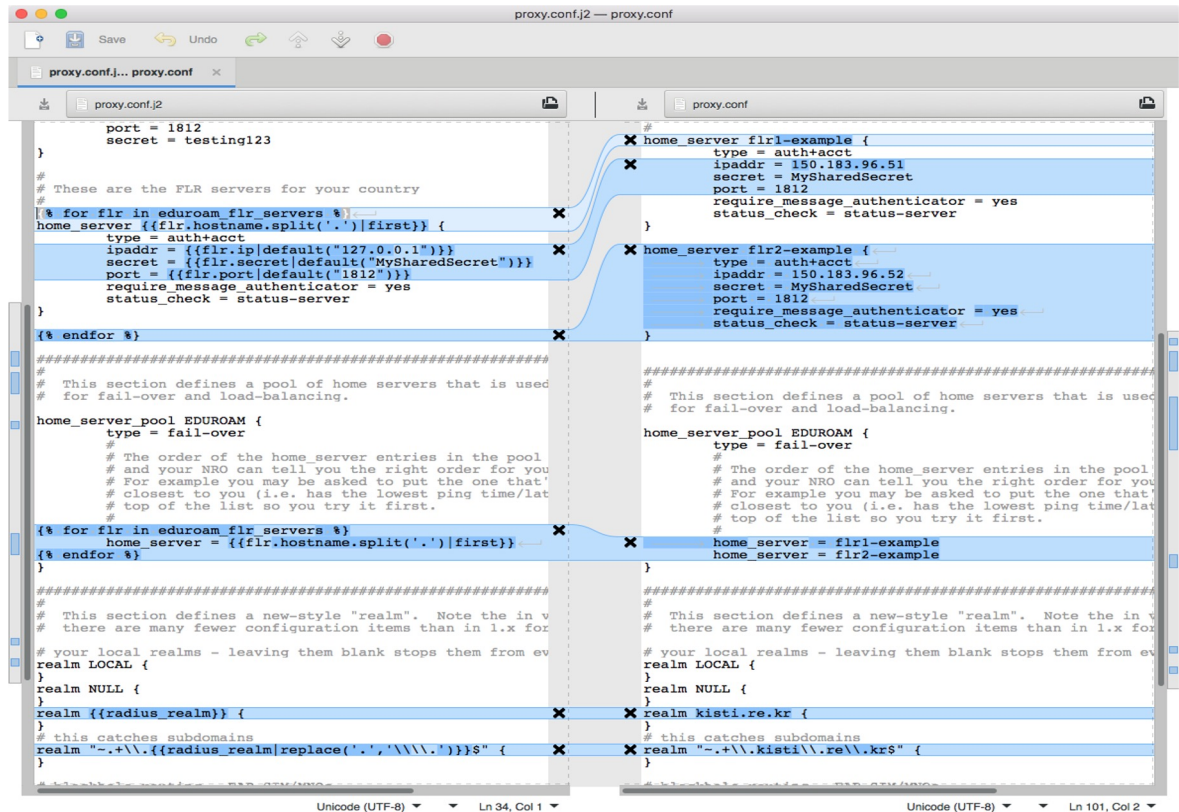
# Details of test account(s) to create within
radius_local_users:
- username: eduroam
  password: xcap2018

# Details of your IMAP server
imap_server: mail.kisti.re.kr
imap_port: 993
imap_ssl: yes

# If your IMAP server requires a realm/domain
imap_realm: "{{radius_realm}}"

# How long (in seconds) to cache credentials
imap_cache: 600

# The name of the PAM service to use for both
pam_service_name: pam-ldap-radius
```



ANSIBLE/group_vars/all



ANSIBLE/.../proxy.conf.j2



/etc/freeradius/3.0/proxy.conf

Compare Configs (advanced)

- Find Modified Configs in Terminal

```
sudo -s;  
cd /etc/freeradius/;  
diff -qr backup_3.0/ 3.0/ | grep -v Only | awk '{ print $4 }'
```

```
# diff -qr backup_3.0/ 3.0/ | grep -v Only | awk '{ print $4 }'  
3.0/certs/ca.cnf  
3.0/certs/client.cnf  
3.0/certs/inner-server.cnf  
3.0/certs/server.cnf  
3.0/certs/xpextensions  
3.0/clients.conf  
3.0/mods-config/attr_filter/pre-proxy  
3.0/mods-config/files/authorize  
3.0/mods-enabled/eap  
3.0/policy.d/cui  
3.0/proxy.conf  
3.0/radiusd.conf  
3.0/users
```

- Find Added Configs in Terminal

```
diff -qr backup_3.0 3.0 | grep "Only in 3.0" | grep -v certs | sed -e 's#Only in ##g; s#: #/#g'
```

```
# diff -qr backup_3.0 3.0 | grep "Only in 3.0" | grep -v certs | sed -e 's#Only in ##g; s#: #/#g'  
3.0/clients-eduroam-flrs.conf  
3.0/mods-available/f_ticks  
3.0/mods-enabled/f_ticks  
3.0/mods-enabled/pam  
3.0/sites-available/eduroam  
3.0/sites-available/eduroam-inner-tunnel  
3.0/sites-enabled/eduroam  
3.0/sites-enabled/eduroam-inner-tunnel
```

Compare Local and Remote Authentication Flow 1/5

Local Auth Test with Logging and Packet Capture (Test eduroam@xeap.kr in xeap.kr)

- Prepare 3 ssh terminals connected to your IRS (xeap.kr)
- Type The following Commands
 - Press **Ctrl + C** after test to shutdown process in terminal 1,2

```
sudo tcpdump -i any port 1812 -w internal-test.pcap -vv
```

TERMINAL 1

```
# tcpdump -i any port 1812 -w internal-test.pcap -vv
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
Got 0
Got 14
^C
14 packets captured
```

```
sudo service freeradius stop;
```

TERMINAL 2

```
sudo LD_PRELOAD=/lib/x86_64-linux-gnu/libpam.so.0 freeradius -X | tee internal-test.log
```

```
# LD_PRELOAD=/lib/x86_64-linux-gnu/libpam.so.0 freeradius -X | tee internal-test.log
FreeRADIUS Version 3.0.16
...
(0) Received Access-Request Id 0 from 127.0.0.1:41323 to 127.0.0.1:1812 length 155
(0) User-Name = "eduroam@xeap.kr"
...
(0) suffix: Checking for suffix after "@"
(0) suffix: Looking up realm "xeap.kr" for User-Name = "eduroam@xeap.kr"
(0) suffix: Found realm "xeap.kr"
(0) suffix: Adding Stripped-User-Name = "eduroam"
(0) suffix: Adding Realm = "xeap.kr"
(0) suffix: Authentication realm is LOCAL
...
(0) Sent Access-Accept Id 6 from 127.0.0.1:1812 to 127.0.0.1:41323 length 0
...
```

```
rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -m WPA-EAP -s eduroam -e TTLS -2 PAP \
-u eduroam@xeap.kr -p xeap2018
```

TERMINAL 3

```
# rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u eduroam@xeap.kr -p xeap2018 -m WPA-EAP -s eduroam -e TTLS -2 PAP
access-accept; 0
```

Compare Local and Remote Authentication Flow 2/5

Remote Auth Test with Logging and Packet Capture (Test eduroam@xeap.au in xeap.kr)

- Prepare 3 ssh terminals connected to your IRS (xeap.kr)
- Type The following Commands
 - Press **Ctrl + C** after test to shutdown process in terminal 1,2

```
sudo tcpdump -i any port 1812 -w external-test.pcap -vv
```

TERMINAL 1

```
# tcpdump -i any port 1812 -w external-test.pcap -vv
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 262144 bytes
Got 0
Got 14
^C
14 packets captured
```

```
sudo service freeradius stop;
```

TERMINAL 2

```
sudo LD_PRELOAD=/lib/x86_64-linux-gnu/libpam.so.0 freeradius -X | tee external-test.log
```

```
# LD_PRELOAD=/lib/x86_64-linux-gnu/libpam.so.0 freeradius -X | tee external-test.log
FreeRADIUS Version 3.0.16
...
(0) Received Access-Request Id 0 from 127.0.0.1:39427 to 127.0.0.1:1812 length 155
(0) User-Name = "eduroam@xeap.au"
...
(0) suffix: Checking for suffix after "@"
(0) suffix: Looking up realm "xeap.au" for User-Name = "eduroam@xeap.au"
(0) suffix: Found realm "~.+$"
(0) suffix: Adding Realm = "xeap.au"
(0) suffix: Proxying request from user eduroam@xeap.au to realm ~.+$
...
(0) Sent Access-Accept Id 6 from 127.0.0.1:1812 to 127.0.0.1:41323 length 0
...
```

```
rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -m WPA-EAP -s eduroam -e TTLS -2 PAP
-u eduroam@xeap.au -p xeap2018
```

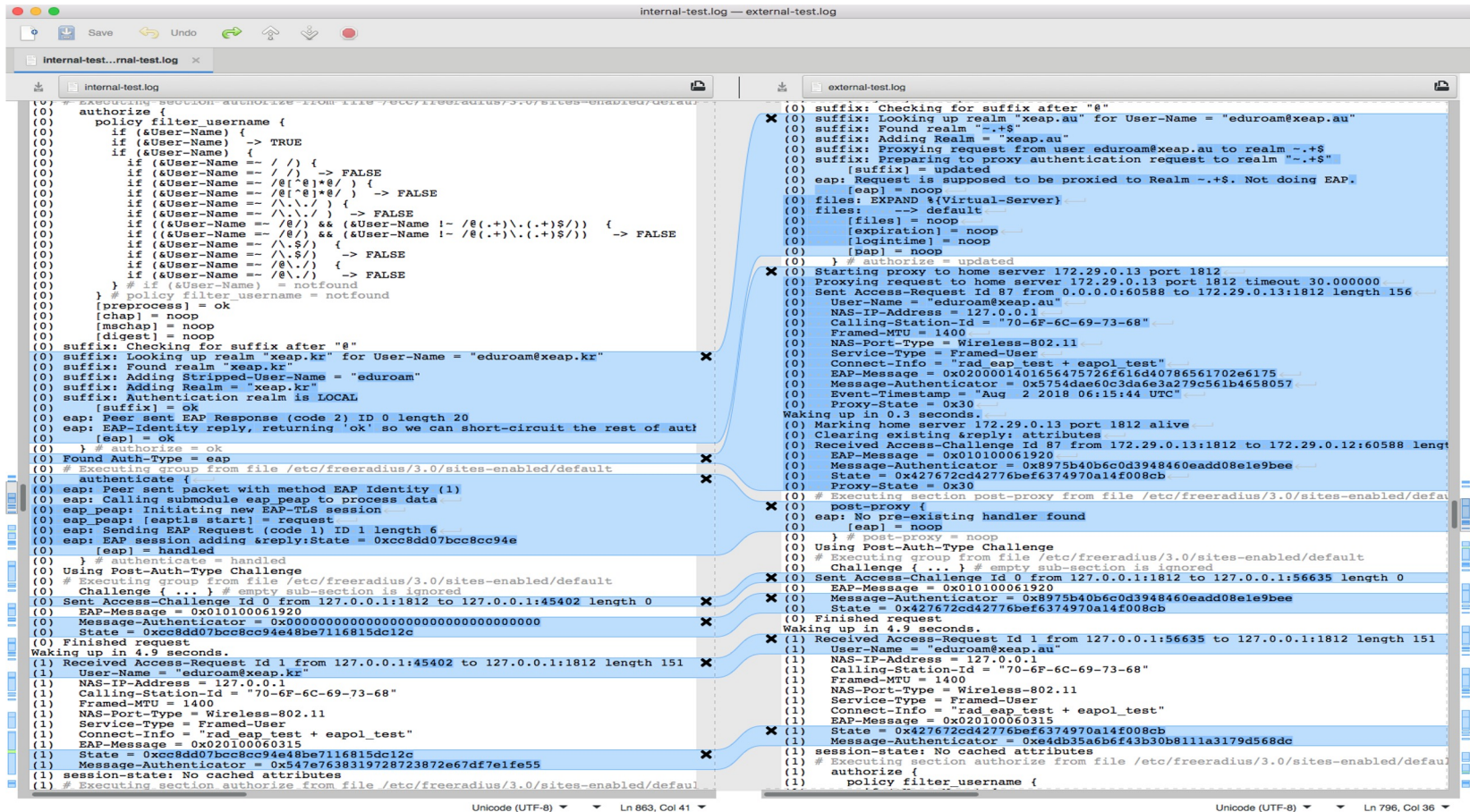
TERMINAL 3

```
# rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u eduroam@xeap.kr -p xeap2018 -m WPA-EAP -s eduroam -e TTLS -2 PAP
access-accept; 0
```

Compare Local and Remote Authentication Flow 3/5

Compare FreeRadius Logs between Local and Remote Authentication

- Before compare, all logs and pcap files should be copied to your PC



Compare Local and Remote Authentication Flow 4/5

Open Radius Packets from Local Authentication with WireShark

internal-test.pcap

Apply a display filter ... <%%/> Expression... +

No.	Time	Source	Protocol	Destination	Length	Info
1	0.000000	127.0.0.1	RADIUS	127.0.0.1	191	Access-Request id=0
2	0.001230	127.0.0.1	RADIUS	127.0.0.1	108	Access-Challenge id=0
3	0.001363	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=1
4	0.002301	127.0.0.1	RADIUS	127.0.0.1	108	Access-Challenge id=1
5	0.002756	127.0.0.1	RADIUS	127.0.0.1	367	Access-Request id=2
6	0.008321	127.0.0.1	RADIUS	127.0.0.1	1124	Access-Challenge id=2
7	0.008621	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=3
8	0.009021	127.0.0.1	RADIUS	127.0.0.1	1124	Access-Challenge id=3
9	0.009326	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=4
10	0.009718	127.0.0.1	RADIUS	127.0.0.1	961	Access-Challenge id=4
11	0.012210	127.0.0.1	RADIUS	127.0.0.1	321	Access-Request id=5
12	0.012955	127.0.0.1	RADIUS	127.0.0.1	163	Access-Challenge id=5
13	0.013345	127.0.0.1	RADIUS	127.0.0.1	272	Access-Request id=6
14	0.014500	127.0.0.1	RADIUS	127.0.0.1	204	Access-Accept id=6

Frame 14: 204 bytes on wire (1632 bits), 204 bytes captured (1632 bits)
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: 1812, Dst Port: 45402
RADIUS Protocol
Code: Access-Accept (2)
Packet identifier: 0x6 (6)
Length: 160
Authenticator: 58dc55db4f243e41cb6f6843ace7b85d
[\[This is a response to a request in frame 13\]](#)
[Time from request: 0.001155000 seconds]
Attribute Value Pairs
AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
AVP: t=EAP-Message(79) l=6 Last Segment[1]
Type: 79
Length: 6
EAP fragment: 03060004
Extensible Authentication Protocol
AVP: t=Message-Authenticator(80) l=18 val=d2504b73755e2f26bce40c516bb72907

```
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 08 00 .....  
0010 45 00 00 bc b0 2a 00 00 40 11 cc 04 7f 00 00 01 E.....*...@.....  
0020 7f 00 00 01 07 14 b1 5a 00 a8 fe bb 02 06 00 a0 .....Z.....  
0030 58 dc 55 db 4f 24 3e 41 cb 6f 68 43 ac e7 b8 5d X·U·0$>A·ohC···]  
0040 1a 3a 00 00 01 37 11 34 83 da 7d 05 08 fb 69 24 ·:···7·4 ·}···i$  
0050 2b e0 49 2d 8c d0 72 f9 5a 71 d0 67 6a d3 f3 25 +·I···r·Zq·gj·%  
0060 9b 82 f9 c7 be 3c fe 8e 93 79 c1 99 37 4b 0d 75 .....<···y···7K·u  
0070 b0 84 e4 ee 6a cb a5 84 8c fd 1a 3a 00 00 01 37 ····j·········7  
0080 10 34 8c 38 0e 65 09 c7 15 ba be fa 5d f8 44 c6 ·4·8·e·······]D·
```

Frame (204 bytes) Reassembled EAP (4 bytes)

internal-test.pcap Packets: 14 · Displayed: 14 (100.0%) · Marked: 7 (50.0%) Profile: Default

Compare Local and Remote Authentication Flow 5/5

Open Radius Packets from Remote Authentication with WireShark

The screenshot displays the Wireshark interface with a list of captured packets and a detailed view of the selected packet (Frame 28). The packet list shows a sequence of RADIUS messages between 127.0.0.1 and 172.29.0.13, including Access-Request, Access-Challenge, and Access-Accept packets. The selected packet (Frame 28) is an Access-Accept message with ID 6, containing an EAP message with a Message-Authenticator AVP.

No.	Time	Source	Protocol	Destination	Length	Info
1	0.000000	127.0.0.1	RADIUS	127.0.0.1	191	Access-Request id=0
2	0.000614	172.29.0.12	RADIUS	172.29.0.13	200	Access-Request id=87
3	0.001110	172.29.0.13	RADIUS	172.29.0.12	111	Access-Challenge id=87
4	0.001295	127.0.0.1	RADIUS	127.0.0.1	108	Access-Challenge id=0
5	0.001444	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=1
6	0.001848	172.29.0.12	RADIUS	172.29.0.13	204	Access-Request id=3
7	0.002131	172.29.0.13	RADIUS	172.29.0.12	111	Access-Challenge id=3
8	0.002283	127.0.0.1	RADIUS	127.0.0.1	108	Access-Challenge id=1
9	0.002648	127.0.0.1	RADIUS	127.0.0.1	367	Access-Request id=2
10	0.003032	172.29.0.12	RADIUS	172.29.0.13	376	Access-Request id=247
11	0.005879	172.29.0.13	RADIUS	172.29.0.12	1127	Access-Challenge id=247
12	0.006059	127.0.0.1	RADIUS	127.0.0.1	1124	Access-Challenge id=2
13	0.006311	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=3
14	0.006672	172.29.0.12	RADIUS	172.29.0.13	204	Access-Request id=48
15	0.006916	172.29.0.13	RADIUS	172.29.0.12	1127	Access-Challenge id=48
16	0.007079	127.0.0.1	RADIUS	127.0.0.1	1124	Access-Challenge id=3
17	0.007330	127.0.0.1	RADIUS	127.0.0.1	195	Access-Request id=4
18	0.007701	172.29.0.12	RADIUS	172.29.0.13	204	Access-Request id=213
19	0.007921	172.29.0.13	RADIUS	172.29.0.12	964	Access-Challenge id=213
20	0.008077	127.0.0.1	RADIUS	127.0.0.1	961	Access-Challenge id=4
21	0.010078	127.0.0.1	RADIUS	127.0.0.1	321	Access-Request id=5
22	0.010454	172.29.0.12	RADIUS	172.29.0.13	330	Access-Request id=145
23	0.010950	172.29.0.13	RADIUS	172.29.0.12	166	Access-Challenge id=145
24	0.011102	127.0.0.1	RADIUS	127.0.0.1	163	Access-Challenge id=5
25	0.011412	127.0.0.1	RADIUS	127.0.0.1	272	Access-Request id=6
26	0.011810	172.29.0.12	RADIUS	172.29.0.13	281	Access-Request id=141
27	0.012357	172.29.0.13	RADIUS	172.29.0.12	207	Access-Accept id=141
28	0.012592	127.0.0.1	RADIUS	127.0.0.1	204	Access-Accept id=6

Frame 28: 204 bytes on wire (1632 bits), 204 bytes captured (1632 bits)
Linux cooked capture
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
User Datagram Protocol, Src Port: 1812, Dst Port: 56635
RADIUS Protocol
Code: Access-Accept (2)
Packet identifier: 0x6 (6)
Length: 160
Authenticator: b01d966641321c5727676a884adee4bb
[\[This is a response to a request in frame 25\]](#)
[Time from request: 0.001180000 seconds]
Attribute Value Pairs
AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
AVP: t=Vendor-Specific(26) l=58 vnd=Microsoft(311)
AVP: t=EAP-Message(79) l=6 Last Segment[1]
Type: 79
Length: 6
EAP fragment: 03060004
Extensible Authentication Protocol
AVP: t=Message-Authenticator(80) l=18 val=66544c1f61e1da6a2c1494bdc5950588

```
0000 00 00 03 04 00 06 00 00 00 00 00 00 00 00 08 00 .....  
0010 45 00 00 bc 6e 1a 00 00 40 11 0e 15 7f 00 00 01 E...n...@.....  
0020 7f 00 00 01 07 14 dd 3b 00 a8 fe bb 02 06 00 a0 .....;  
0030 b0 1d 96 66 41 32 1c 57 27 67 6a 88 4a de e4 bb ...fA2-W'gj.J...  
0040 1a 3a 00 00 01 37 11 34 81 54 e0 d9 45 e3 4d 78 ...:~7.4.T.E.Mx  
0050 aa 01 c6 e5 df a5 ce b1 70 ee ae 21 7f 52 17 9a .....p!~R...  
0060 fb bd 8f 05 a0 39 e5 59 3a 52 de 05 ab 1e 4f 26 .....9.Y:R...0&  
0070 a2 ab 5b 90 b8 4f 11 4a e1 15 1a 3a 00 00 01 37 ...[~0.J...:~7  
0080 10 34 88 19 1a 3a 15 5a 25 8c 67 b3 fb 3c 04 12 .4...:Z%g.<..
```

Q&A

Thank you

Ask More Questions to:

- Minseok Jang msjang@kisti.re.kr

For Left Time

What would be improved of IRS?

1. Improve Logging config
 - Ref. [Logging Guide @ FreeRadius Wiki](#)
2. Add Lock-Out feature
 - Lock-out : Disable account after X auth fails in Y seconds
 - [Anyone can Brute-Force on eduroam to find out one's password](#)
 - Ref1. [MS Windows Server has Lock-Out Feature in NPS server](#)
 - Ref2. [LockOut Guide @ FreeRadius Wiki](#)
 - Ref3. [SQL Guide @ FreeRadius Wiki](#)
3. Improve Blackhole Routing using inner-tunnel
 - Ref. [Rejecting auth from a specific realm @ FreeRadius Mailing List, 2009](#)
4. Use Commercial SSL Server Certificate on RADIUS server
 - Some Institute use Comercial SSL Server Certificate to improve their Security
 - Comercial Server Cert can be applied similar way in the following reference;
Ref. [Use Let's Encrypt Certificates with FreeRADIUS](#)
5. Examples on CUI (Chargeable User Identity)
 - Ref1. [Chargeable User Identity for eduroam: with FreeRADIUS implementation guide @ Jisc](#)
 - Ref2. [Configuring an eduroam FreeRADIUS 3.0 server@ diamond.ac.uk](#)
 - Ref3. [Chargeable User Identity, RFC4372](#)
6. [Hardening TLS for WLAN 802.1X Authentication](#)

Feedback & Request

Discuss among forks on the following issues

- Goods and Lacks on this session
- Request for Next XeAP Workshop

Retrospection on XeAP 2018 Session 3

- Huge tech divergence between trainees
 - Novice for Linux ~ current eduroam operator
- Many trainees practice commands for IRS in NRS