คู่มือการติดตั้ง Radius server สำหรับบริการ eduroam

บทนำ

ขั้นตอนการติดตั้ง จะประกอบด้วย 3 ขั้นตอนหลัก กับ 2 ทางเลือก ประกอบด้วย

1. การติดตั้งและทดสอบพื้นฐาน
   สามารถทำให้ Radius server ทำงานได้ด้วยตัวเอง ใช้บัญชีผู้ใช้ที่มีอยู่ในไฟล์ของโปรแกรม
2. การเลือกใช้ฐานข้อมูลบัญชีผู้ใช้
   เป็นการกำหนดให้ Radius server ใช้บัญชีผู้ใช้จากฐานข้อมูลภายนอก ประกอบด้วย 2 ทางเลือก
   - การติดตั้งโดยมี LDAP Server เป็นฐานข้อมูลบัญชีผู้ใช้
   - การติดตั้งโดยมี Microsoft Active Directory เป็นฐานข้อมูลบัญชีผู้ใช้
3. การติดตั้งใช้งานร่วมกับ eduroam-TH

วิธีการติดตั้ง เป็นการแนะนำคำสั่งในการดำเนินการอย่างเป็นลำดับ พร้อมตัวอย่างคำสั่งที่ตรงกับสภาพแวดล้อมของเครื่องมากที่สุด เช่น การติดตั้งแพคเกจ การแก้ไขไฟล์ การทดสอบการทำงาน เป็นต้น โดยคุณสมบัติของโปรแกรมเกือบทั้งหมด เป็นการนำไฟล์สำเร็จรูปที่ผ่านการปรับรูปแบบเพื่อไม่ให้ซ้ำกับไฟล์คุณสมบัติเดิมของโปรแกรม นำมาติดตั้ง ดำเนินการแก้ไขเนื้อหาในไฟล์ให้เหมาะสม และใช้งาน

เพื่อให้การติดตั้งมีความถูกต้องและสามารถทำงานได้อย่างไม่มีข้อผิดพลาด จำเป็นต้องดำเนินการตามลำดับขั้นโดยละเอียด ยกเว้นการเลือกใช้ฐานข้อมูลบัญชีผู้ใช้ที่สามารถเลือกได้อย่างใดอย่างหนึ่ง

หัวข้อ การตรวจวิเคราะห์และตรวจสอบการทำงานของ Radius server เป็นส่วนของการแนะนำการปรับแต่งคุณสมบัติ เพื่อให้ Radius server ทำงานที่แตกต่างหรือเพิ่มเติมจากการติดตั้งนี้ รวมถึงการตรวจสอบกิจกรรมที่เกิดขึ้นที่บันทึกไว้ในไฟล์ Log

หัวข้อ การติดตั้ง Wireless Controller หรือ Anonymous Access Point ร่วมกับ Radius server แนะนำวิธีการกำหนดคุณสมบัติของ Radius server และอุปกรณ์ WLC หรือ AP ให้ทำงานร่วมกัน

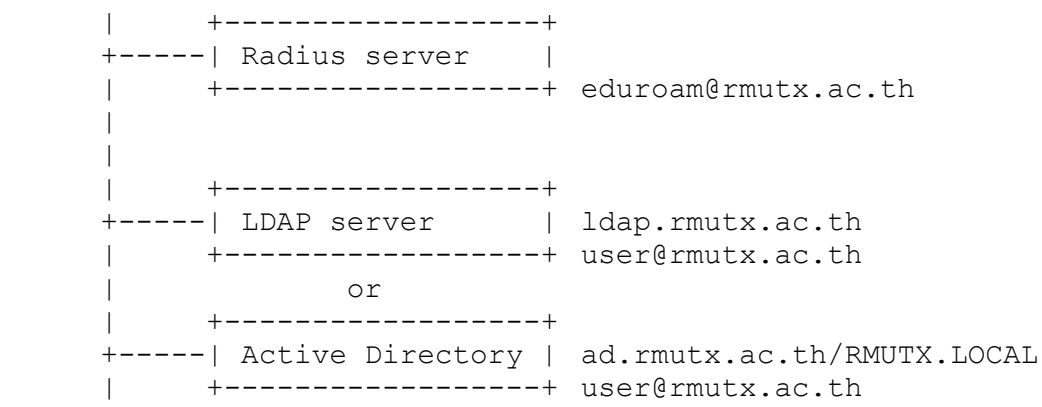
ลักษณะเด่นของการกำหนดคุณสมบัติในการติดตั้งนี้คือ สามารถเปิดโอกาสให้สมาชิกภายในองค์กรใช้บริการเครือข่ายภายในองค์กรได้ ประโยชน์ก็เพื่อให้สมาชิกดำเนินการกำหนดคุณสมบัติการเชื่อมต่อจากเครือข่ายภายในให้สำเร็จก่อน แก้ปัญหาให้เสร็จก่อน จากนั้นจึงจะไปใช้บริการจากผู้ให้บริการอื่นได้ทันที

รุ่นของระบบปฏิบัติการ และโปรแกรม freeradius ที่ใช้ทดลองติดตั้ง

คำสั่งและไฟล์จะอ้างอิงตามระบบปฏิบัติการและโปรแกรม freeradius จึงควรเลือกใช้คำสั่งอย่างถูกต้อง ดังนี้

- Debian 8.2 + freeradius-2.2.5
- CentOS 7.0 + freeradius-3.0.4

โครงสร้างเครือข่ายประกอบการติดตั้ง



-

การติดตั้งและทดสอบขั้นพื้นฐาน

เป็นการติดตั้งและกำหนดคุณสมบัติพื้นฐานให้ Radius server สามารถทำงานได้ด้วยตัวเอง ประกอบด้วยการติดตั้งโปรแกรม ติดตั้งแพคเกจพื้นฐาน ติดตั้งแพคเกจสนับสนุน ติดตั้งโปรแกรมสำหรับทดสอบ แก้ไขคุณสมบัติพื้นฐาน และทดสอบการทำงาน โดยการทดสอบจะนำข้อมูลผู้ใช้แบบไฟล์ข้อความที่มีอยู่ไฟล์ user-eduroam.conf มาใช้งาน

1. อัพเกรดแพคเกจล่าสุดและแพคเกจพื้นฐาน

Debian

```
apt-get update
apt-get upgrade -y
```

อาจต้อง reboot

```
apt-get install ntp -y
```

CentOS

```
yum install epel-release -y
yum update -y
```

อาจต้อง reboot

```
yum install ntp -y
```

2. ติดตั้งแพคเกจ freeradius และแพคเกจสนับสนุน

Debian (freeradius-2.2.5)

```
apt-get install freeradius -y
apt-get install easy-rsa -y
apt-get install wget -y
```

CentOS (freeradius-3.0.4)

```
yum install freeradius freeradius-utils -y
yum install easy-rsa -y
yum install wget -y
```

3. ดาวน์โหลดและคอมไพล์เครื่องมือสำหรับทดสอบ

Debian

```
apt-get install gcc make libssl-dev -y
```

CentOS

```
yum install gcc openssl-devel -y
```

Debian

```
cd /etc/freeradius
```

CentOS

```
cd /etc/raddb
```

Debian & CentOS

```
wget http://www.rmuti.ac.th/user/prakai/p/freeradius-test-
            tool.tar.gz
tar vxfz freeradius-test-tool.tar.gz
cd tool/wpa_supplicant-2.5/wpa_supplicant
cp defconfig .config
vi .config
```

```
CONFIG_EAPOL_TEST=y

#CONFIG_DRIVER_NL80211=y
```

```
make eapol_test
cp eapol_test ../../bin
```

ref: http://deployingradius.com/scripts/eapol_test

4. ดาวน์โหลดชุดไฟล์คุณสมบัติสำเร็จรูป

Debian

```
cd /etc/freeradius
wget http://www.rmuti.ac.th/user/prakai/p/freeradius-2-
eduroam.tar.gz
```

CentOS

```
cd /etc/raddb
wget http://www.rmuti.ac.th/user/prakai/p/freeradius-3-
eduroam.tar.gz
```

5. แตกไฟล์คุณสมบัติสำเร็จรูป

Debian

```
tar vxfz freeradius-2-eduroam.tar.gz
```

CentOS

```
tar vxfz freeradius-3-eduroam.tar.gz
```

รายการไฟล์

- `radiusd-eduroam.conf`
  ไฟล์คุณสมบัติหลัก ใช้ประกอบการแก้ไขไฟล์ radiusd.conf
- `sites-available/eduroam`
  ไฟล์เริ่มต้นสำหรับการเปิดใช้ไซต์ eduroam
- `sites-available/eduroam-inner-tunnel`
  ไฟล์เริ่มต้นสำหรับการเปิดใช้ไซต์ eduroam-inner-tunnel
- `sites-available/eduroam-status`
  ไฟล์เริ่มต้นสำหรับเปิดใช้โมดูล radius status

- `clients-eduroam.conf`
  ไฟล์รายการเครื่องลูกข่ายเพื่อกำหนดให้ UniNet Radius เป็น client ในกรณีการทำหน้าที่เป็น IdP
- `proxy-eduroam.conf`
  ไฟล์คุณสมบัติเพื่อกำหนดให้ UniNet Radius เป็น home server ในกรณีการทำหน้าที่เป็น SP
  ไฟล์คุณสมบัติสำหรับการ
- `eap-eduroam.conf`
  ไฟล์คุณสมบัติการตั้งค่าการทำงานของ EAP
- `eduroam-realm-checks.conf`
  สคริปต์การคัดกรองบัญชีผิดปกติที่ผ่านการรวบรวมแล้ว
- `modules/files-eduroam (Debian)`
  `mods-available/files-eduroam (CentOS)`
  ไฟล์โมดูลที่ใช้อ้างถึงบัญชีผู้ใช้จากไฟล์ข้อความ (user-eduroam)
- `users-eduroam`
  ไฟล์บัญชีผู้ใช้แบบไฟล์ข้อความ สำหรับทดสอบโดยไม่ต้องใช้บัญชีผู้ใช้จากฐานข้อมูล
- `modules/ldap-eduroam (Debian)`
  `mods-available/ldap-eduroam (CentOS)`
  ไฟล์โมดูลที่ใช้อ้างถึงบัญชีผู้ใช้จากฐานข้อมูล LDAP
- `modules/mschap-eduroam (Debian)`
  `mods-available/mschap-eduroam (CentOS)`
  ไฟล์โมดูลที่ใช้อ้างถึงบัญชีผู้ใช้จากฐานข้อมูล Active Directory
- `ldap.attrmap-eduroam`
- `dictionary-eduroam`
- `acct_users-eduroam`
- `preproxy_users-eduroam`

6. แก้ไขไฟล์ radiusd.conf

โดยปรับแก้เฉพาะจุดโดยเทียบจากไฟล์ radiusd-eduroam.conf

Debian

```
vi radiusd.conf
```

```
# Change some configurations in radiusd.conf as show below

# PROXY CONFIGURATION
#
proxy_requests  = yes
$INCLUDE proxy.conf
#  eduroam
$INCLUDE proxy-eduroam.conf

# CLIENTS CONFIGURATION
#
$INCLUDE clients.conf
#  eduroam
$INCLUDE clients-eduroam.conf

modules {
```

```
# Debian
        $INCLUDE ${confdir}/modules/

        #  Extensible Authentication Protocol
        #
        #  For all EAP related authentications.
        #  Now in another file, because it is very large.
        #
        #$INCLUDE eap.conf
        #  eduroam
        $INCLUDE eap-eduroam.conf
}
```

CentOS

vi radiusd.conf

```
# Change some configurations in radiusd.conf as show below

# PROXY CONFIGURATION
#
proxy_requests  = yes
$INCLUDE proxy.conf
#  eduroam
$INCLUDE proxy-eduroam.conf

# CLIENTS CONFIGURATION
#
$INCLUDE clients.conf
#  eduroam
$INCLUDE clients-eduroam.conf

modules {

# CentOS
        $INCLUDE mods-enabled/

        #  Extensible Authentication Protocol
        #
        #$INCLUDE eap.conf
        #  eduroam
        $INCLUDE eap-eduroam.conf

}
```

7. แก้ไขไฟล์ proxy-eduroam.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

vi proxy-eduroam.conf

```
#
#  realm for local service
#
```

```
realm rmutx.ac.th {
        auth_pool = localhost
}
```

8. แก้ไขไฟล์ sites- available/eduroam

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi sites-available/eduroam
```

```
authorize {

      # Change realm to be LOCAL for local user
      if( ("%{Realm}" =~ /rmutx.ac.th$$/) ) {
            update control {
                  Proxy-To-Realm := LOCAL
            }
      }

}

pre-proxy {

         # Update Operator-Name to IdP
         update proxy-request {
                 Operator-Name := "1rmutx.ac.th"
         }

}
```

9. เปิดใช้งานโมดูลชุดใหม่

CentOS

```
cd mods-enabled
ln -s ../ mods-available/files-eduroam
cd ..
```

10. ยกเลิกไซต์เดิม และเปิดใช้ไซต์ใหม่

Debian & CentOS

```
cd sites-enabled
rm -f default
rm -f inner-tunnel
ln -s ../sites-available/eduroam
ln -s ../sites-available/eduroam-inner-tunnel
ln -s ../sites-available/eduroam-status
cd ..
```

11. สร้างไฟล์ Certificate

Debian & CentOS

```
cd certs
```

Debian

```
rm *
cp /usr/share/doc/freeradius/examples/certs/* .
```

Debian & CentOS

```
vi ca.cnf
```

```
[ CA_default ]
:
default_days            = 3650
:

[certificate_authority]
countryName             = TH
stateOrProvinceName     = Bangkok
localityName            = -
organizationName        = RMUTx
emailAddress            = noc@rmutx.ac.th
commonName              = "RMUTx Wi-Fi Certificate Authority"
```

```
vi server.cnf
```

```
[ CA_default ]
:
default_days            = 3650
:

[server]
countryName             = TH
stateOrProvinceName     = Bangkok
localityName            = -
organizationName        = RMUTx
emailAddress            = noc@rmutx.ac.th
commonName              = "RMUTx Wi-Fi Certificate"
```

```
vi client.cnf
```

```
[ CA_default ]
:
default_days            = 3650
:

[client]
countryName             = TH
stateOrProvinceName     = Bangkok
localityName            = -
organizationName        = RMUTx
emailAddress            = noc@rmutx.ac.th
commonName              = wifi@rmutx.ac.th
```

```
./bootstrap

cd ..
```

12. ทดสอบการทำงานแบบพื้นฐาน

หน้าจอที่ 1

Debian

```
service freeradius stop
freeradius -X
(stop debuging with CTRL+C)
```

CentOS

```
service radiusd stop
radiusd -X
(stop debuging with CTRL+C)
```

หน้าจอที่ 2

Debian

```
cd /etc/freeradius/tool
```

CentOS

```
cd /etc/raddb/tool
```

Debian & CentOS

```
./rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u
'eduroam@rmutx.ac.th' -p 'TESTING-PASSWORD' -v -m IEEE8021X -s
eduroam -e PEAP -2 MD5
```

```
access-accept; 0
RADIUS message: code=2 (Access-Accept) identifier=8 length=187
   Attribute 27 (Session-Timeout) length=6
      Value: 600
   Attribute 1 (User-Name) length=21
      Value: 'eduroam@rmutx.ac.th'
   Attribute 79 (EAP-Message) length=6
      Value: 03080004
   Attribute 80 (Message-Authenticator) length=18
      Value: 6668fe5c30e59946dc91ad7200c0a810
```

การติดตั้งโดยมี LDAP Server เป็นฐานข้อมูลบัญชีผู้ใช้

เป็นการติดตั้งและกำหนดคุณสมบัติพื้นฐานให้ Radius server สามารถทำงานกับ LDAP server (OpenLDAP) เพื่อใช้บัญชีผู้ใช้จากฐานข้อมูล LDAP ข้อมูลบัญชีผู้ใช้ควรมีการเก็บรหัสผ่านในรูปแบบ NT/LM Hash (NT-Password, LM-Password)

การทำงานของ Radius server จะติดต่อโดยตรงไปยัง LDAP server ผ่านโมดูลที่มีอยู่ใน Radius server

13. โครงสร้างข้อมูลใน LDAP Server

โครงสร้างหลักโดยย่อของข้อมูลผู้ใช้ใน LDAP Server

```
dn: dc=rmutx,dc=ac,dc=th
objectClass: top
objectClass: organization
dc: rmutx

dn: ou=People,dc=rmutx,dc=ac,dc=th
ou: People
objectClass: top
objectClass: organizationalUnit

dn: uid=user,ou=People,dc=rmutx,dc=ac,dc=th
cn: User
sn: User
objectClass: top
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: sambaSamAccount
uid: user
uidNumber: 1001
gidNumber: 1000
loginShell: /bin/bash
gecos: User User
description: User User
shadowLastChange: 1422293189
shadowMax: 99999
mail: user@rmutx.ac.th
displayName: User User
sambaAcctFlags: [U          ]
sambaHomeDrive: H:
sambaLMPassword: C8DFD5AC0546E95DFF17365FAF1FFE89
sambaNTPassword: 2C47AA9B5AC02360473ECE87B6800920
sambaSID: S-1-5-21-2593467306-2038934869-1345228925-3001
sambaPrimaryGroupSID: S-1-5-21-2593467306-2038934869-1345228925-
3004
userPassword:: e1NTSEF9Y0F1dXBVNURlbVFhakxxaDFSU2VVTHl5Wi9NQ1dlSXM=
```

14. ติดตั้งแพคเกจ freeradius-ldap

Debian (freeradius-2.2.5)

```
apt-get install freeradius-ldap -y
```

CentOS (freeradius-3.0.4)

```
yum install freeradius-ldap -y
```

15. แก้ไขไฟล์ sites-available/eduroam-inner-tunnel

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi sites-available/eduroam-inner-tunnel
```

```
authorize {

        group {
                #  Read the 'users-eduroam' file
                files-eduroam {
                        # return if match
                        ok = return
                }
                #
                # for LDAP
                ldap-eduroam {
                        # return if match
                        ok = return
                }
                # for Active Directory
                #mschap-eduroam {
                #        # return if match
                #        ok = return
                #}
        }

}

authenticate {

        Auth-Type PAP {
                pap
        }

        # for LDAP
        Auth-Type MS-CHAP {
                mschap
        }

        # for Active Directory
        #Auth-Type MS-CHAP {
        #        mschap-eduroam
        #}

        eap-eduroam

}
```

16. แก้ไขไฟล์ (modules or mods-available)/ldap-eduroam

โดยปรับแก้ทุกจุดให้ถูกต้อง สัมพันธ์กับ LDAP server

Debian

vi modules/ldap-eduroam

```
ldap ldap-eduroam {
        server = "your-ldap-server-host"

        #identity = "cn=admin,dc=rmutx,dc=ac,dc=th"
        #password = mypass

        basedn = "o=rmutx,dc=ac,dc=th "

        #base_filter = "(objectclass=radiusprofile)"

        # password_attribute = userPassword
        password_attribute = sambaNTPassword

}
```

CentOS

vi mods-available/ldap-eduroam

```
ldap ldap-eduroam {
        server = "your-ldap-server-host"
#       port = 389

#       identity = "cn=admin,dc=rmutx,dc=ac,dc=th"
#       password = mypass

        base_dn = "dc=rmutx,dc=ac,dc=th"

        update {
                control:Password-With-Header     +=
'userPassword'
                control:NT-Password              :=
'sambaNTPassword'
                control:LM-Password              :=
'sambaLMPassword'
        }

}
```

Debian & CentOS

(modules or mods-available)/ldap-eduroam

```
ldap ldap-eduroam {
        server = "your-ldap-server-host"

        basedn = "o=rmutx,dc=ac,dc=th "
```

```
        filter = "(uid=%{%{Stripped-User-Name}:-%{User-
Name}})"

        #base_filter = "(objectclass=radiusprofile)"

        #access_attr = "dialAccess"

        # password_attribute = userPassword
        password_attribute = sambaNTPassword

}
```

17. เปิดใช้โมดูล ldap-eduroam

CentOS

```
cd mods-enabled
ln -s ../mods-available/ldap-eduroam
cd ..
```

18. เปลี่ยนสิทธิ์หรือเจ้าของของไฟล์

Debian

```
chgrp -R freerad *
```

CentOS

```
chgrp -R radiusd *
```

19. ทดสอบการทำงานด้วยผู้ใช้จาก LDAP Server

หน้าจอที่ 1

Debian

```
service freeradius stop
freeradius -X
(stop debuging with CTRL+C)
```

CentOS

```
service radiusd stop
radiusd -X
(stop debuging with CTRL+C)
```

หน้าจอที่ 2

Debian

```
cd /etc/freeradius/tool
```

CentOS

```
cd /etc/raddb/tool
```

Debian & CentOS

```
./rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u
'user@rmutx.ac.th' -p 'Abcd1234' -v -m IEEE8021X -s eduroam -e PEAP
-2 MSCHAPV2
```

```
access-accept; 0
RADIUS message: code=2 (Access-Accept) identifier=8 length=187
   Attribute 27 (Session-Timeout) length=6
      Value: 600
   Attribute 1 (User-Name) length=21
      Value: 'user@rmutx.ac.th'
   Attribute 79 (EAP-Message) length=6
      Value: 03080004
   Attribute 80 (Message-Authenticator) length=18
      Value: 4f334b7622ec20537163ac31c1926d84
```

การติดตั้งโดยมี Microsoft Active Directory เป็นฐานข้อมูลบัญชีผู้ใช้

เป็นการติดตั้งและกำหนดคุณสมบัติพื้นฐานให้ Radius server สามารถทำงานร่วมกับ Microsoft Active Directory เพื่อตรวจสอบผู้ใช้จากบัญชีผู้ใช้ใน Active Directory

การทำงานของ Radius server จะตรวจสอบตัวตนของผู้ใช้ผ่านโปรแกรมภายนอก คือ samba หรือ winbind จึงจำเป็นต้องกำหนดคุณสมบัติของ samba หรือ winbind ให้สามารถติดต่อกับ Active Directory เสียก่อน

20. แก้ไขไฟล์ /etc/resolv.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi /etc/resolv.conf
```

```
nameserver <dc_server_address>
nameserver <other_dns_server>
```

21. แก้ไขไฟล์ /etc/hosts.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi /etc/hosts.conf
```

```
<dc_server_address>    dc.rmutx.local
```

22. ติดตั้งแพคเกจสนับสนุนเกี่ยวกับ samba, krb5 และ winbind

Debian

```
apt-get install samba winbind krb5-user krb5-config -y
```

```
Default Kerberos version 5 realm:
        RMUTX.LOCAL

Kerberos servers for your realm:
        ad.rmutx.local

Administrative server for your Kerberos realm:
        ad.rmutx.local
```

CentOS

```
yum install samba samba-winbind-clients krb5-workstation -y
```

23. แก้ไขไฟล์ /etc/samba/smb.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi /etc/samba/smb.conf
```

```
[global]

        workgroup = RMUTX
        security = ADS
        realm = RMUTX.LOCAL
        encrypt passwords = yes
        client use spnego = yes

        idmap config *:backend = tdb
        idmap config *:range = 2000-9999
        idmap config RMUTX:backend = ad
        idmap config RMUTX:schema_mode = rfc2307
        idmap config RMUTX:range = 10000-99999

        winbind nss info = rfc2307
        winbind trusted domains only = no
        winbind use default domain = yes
        winbind enum users  = yes
        winbind enum groups = yes
        winbind refresh tickets = Yes

}
```

24. แก้ไขไฟล์ /etc/krb5.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi /etc/krb5.conf
```

```
[libdefaults]
        default_realm = RMUTX.LOCAL
        dns_lookup_realm = true
        dns_lookup_kdc = false
        forwardable = true

[realms]
        RMUTX.LOCAL = {
                kdc = ad.rmutx.local
                admin_server = ad.rmutx.local
        }

[domain_realm]
        .rmutx.local = RMUTX.LOCAL
        rmutx.local = RMUTX.LOCAL
```

25. รีสตาร์ตโปรแกรม samba และ winbind

Debian

```
/etc/init.d/samba restart
/etc/init.d/winbind restart
```

CentOS

```
service smb restart
service winbind restart
```

26. Join เครื่อง Radius server ไปเป็นสมาชิกของ Active Directory Domain

Debian & CentOS

```
net ads join -U Administrator
```

```
Enter Administrator's password: <Administrator's password>
Using short domain name -- RMUTX
Joined 'YOUR-RADIUS-SERVER' to dns domain 'RMUTX.LOCAL'
```

27. ทดสอบผลการ Join เครื่อง Radius server ไปเป็นสมาชิกของ Active Directory Domain

Debian & CentOS

```
wbinfo -u
```

```
administrator
user
and other users
```

28. ทดสอบใช้บัญชีผู้ใช้จาก Active Directory

Debian & CentOS

```
/usr/bin/ntlm_auth --domain=RMUTX.LOCAL --username=user --
password=Asdf1234
```

```
NT_STATUS_OK: Success (0x0)
```

29. เพิ่มสิทธิ์ให้ผู้ใช้ที่รันโพรเซส Radius server เข้าในกลุ่มของผู้ใช้ที่รันโปรเซส winbind

Debian

```
chown root:winbindd_priv /var/lib/samba/winbindd_privileged
usermod -a -G winbindd_priv freerad
```

CentOS

```
chown root: wbpriv /var/run/samba/winbindd
usermod -a -G wbpriv radiusd
```

30. แก้ไขไฟล์ (modules or mods-available)/mschap-eduroam

โดยปรับแก้เฉพาะจุด

Debian

```
vi modules/mschap-eduroam
```

```
mschap mschap-eduroam {
        use_mppe = yes

        require_encryption = yes
        require_strong = yes

        with_ntdomain_hack = yes
```

```
        ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --
domain=RMUTX --username=%{Stripped-User-Name} --
challenge=%{mschap:Challenge:-00} --nt-response=%{mschap:NT-
Response:-00}"

#       ntlm_auth_timeout = 10

#       use_open_directory = yes
#       allow_retry = yes
#       retry_msg = "Re-enter (or reset) the password"
}
```

CentOS

vi mods-available/mschap-eduroam

```
mschap mschap-eduroam {
        use_mppe = yes

        require_encryption = yes
        require_strong = yes

        with_ntdomain_hack = yes

        ntlm_auth = "/usr/bin/ntlm_auth --request-nt-key --
domain=RMUTX --username=%{Stripped-User-Name} --
challenge=%{mschap:Challenge:-00} --nt-response=%{mschap:NT-
Response:-00}"

#       ntlm_auth_timeout = 10

        passchange {
#               ntlm_auth = "/usr/bin/ntlm_auth --helper-
protocol=ntlm-change-password-1"
#               ntlm_auth_username = "username: %{mschap:User-
Name}"
#               ntlm_auth_domain = "nt-domain: %{mschap:NT-
Domain}"
        }
#       use_open_directory = yes
#       allow_retry = yes
#       retry_msg = "Re-enter (or reset) the password"
}
```

31. แก้ไขไฟล์ sites-available/eduroam-inner-tunnel

โดยปรับแก้เฉพาะจุด

Debian & CentOS

vi sites-available/eduroam-inner-tunnel

```
authorize {

        group {
```

```
                        #  Read the 'users-eduroam' file
                        files-eduroam {
                                # return if match
                                ok = return
                        }
                        #
                        # for LDAP
                        #ldap-eduroam {
                        #       # return if match
                        #       ok = return
                        #}
                        # for Active Directory
                        mschap-eduroam {
                                # return if match
                                ok = return
                        }
                }

        }

        authenticate {

                Auth-Type PAP {
                        pap
                }

                # for LDAP
                #Auth-Type MS-CHAP {
                #        mschap
                #}

                # for Active Directory
                Auth-Type MS-CHAP {
                        mschap-eduroam
                }

                eap-eduroam

        }
```

32. เปิดใช้โมดูล mschap-eduroam

CentOS

```
cd mods-enabled
ln -s ../mods-available/mschap-eduroam
cd ..
```

33. เปลี่ยนสิทธิ์หรือเจ้าของของไฟล์

Debian

```
chgrp -R freerad *
```

CentOS

```
chgrp -R radiusd *
```

34. ทดสอบการทำงานด้วยผู้ใช้จาก Active Directory

หน้าจอที่ 1

Debian

```
service freeradius stop
freeradius -X
(stop debuging with CTRL+C)
```

CentOS

```
service radiusd stop
radiusd -X
(stop debuging with CTRL+C)
```

หน้าจอที่ 2

Debian

```
cd /etc/freeradius/tool
```

CentOS

```
cd /etc/raddb/tool
```

Debian & CentOS

```
./rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u
'user@rmutx.ac.th' -p 'Abcd1234' -v -m IEEE8021X -s eduroam -e PEAP
-2 MSCHAPV2
```

```
access-accept; 0
RADIUS message: code=2 (Access-Accept) identifier=8 length=187
   Attribute 27 (Session-Timeout) length=6
      Value: 600
   Attribute 1 (User-Name) length=21
      Value: 'user@rmutx.ac.th'
   Attribute 79 (EAP-Message) length=6
      Value: 03080004
   Attribute 80 (Message-Authenticator) length=18
      Value: 4f334b7622ec20537163ac31c1926d84
```

การติดตั้งใช้งานร่วมกับ eduroam-TH

35. แก้ไขไฟล์ proxy-eduroam.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi proxy-eduroam.conf
```

```
#
#  realm for local service
#
realm rmutx.ac.th {
        auth_pool = localhost
}

home_server eduroam-NRO-a {
        type = auth+acct
        ipaddr = 202.28.112.6
        port = 1812
        secret = XXXXXXXXXXXXXXXX
        #src_ipaddr = xxx.xxx.xxx.xxx
        status_check = status-server
        require_message_authenticator = yes
}
```

36. แก้ไขไฟล์ clients-eduroam.conf

โดยปรับแก้เฉพาะจุด

Debian & CentOS

```
vi clients-eduroam.conf
```

```
#
#  eduroam server (NRO)
#

client eduroam-NRO-a {
        secret = XXXXXXXXXXXXXXXX
        ipaddr = 202.28.112.6 #UniNet
        require_message_authenticator = no
        shortname = eduroam-NRO
        #virtual_server = eduroam
}
```

37. ทดสอบการทำงานด้วยผู้ใช้ eduroam จาก IdP อื่น

หน้าจอที่ 1

Debian

```
service freeradius stop
freeradius -X
(stop debuging with CTRL+C)
```

CentOS

```
service radiusd stop
radiusd -X
(stop debuging with CTRL+C)
```

หน้าจอที่ 2

Debian

```
cd /etc/freeradius/tool
```

CentOS

```
cd /etc/raddb/tool
```

Debian & CentOS

```
./rad_eap_test -H 127.0.0.1 -P 1812 -S testing123 -u
'xxxx@rmutA.ac.th' -p 'xoooox' -v -m IEEE8021X -s eduroam -e PEAP -
2 MSCHAPV2
```

```
access-accept; 0
RADIUS message: code=2 (Access-Accept) identifier=8 length=187
   Attribute 27 (Session-Timeout) length=6
      Value: 600
   Attribute 1 (User-Name) length=21
      Value: 'xxxx@rmutA.ac.th'
   Attribute 79 (EAP-Message) length=6
      Value: 03080004
   Attribute 80 (Message-Authenticator) length=18
      Value: 4f334b7622ec20537163ac31c1926d84
```

การติดตั้ง Wireless Controller หรือ Anonymous Access Point ร่วมกับ Radius server

Radius server: แก้ไขไฟล์ clients.conf

เพิ่ม IP address หรือเครือข่ายของ Anonymous Access Point

Debian & CentOS

```
vi clients.conf
```

```
client <ip_or_network_of_access_point> {
      secret      = testing123
      shortname   = my_access_point
}

client 172.16.11.8 {
      secret      = secret_for_172_16_11_8
      shortname   = ap_172_16_11_8
}

client 192.168.0.0/24 {
      secret      = secret_for_net_192_168_0_0_24
      shortname   = ap_in_net_192_168_0_0_24
}
```

Cisco Wireless Controller

1. Add/Edit RADIUS profile

SECURITY > AAA > RADIUS > Authentication > [New...] or Edit

```
Server IP Address(Ipv4/Ipv6): <radius_server_ip_address>
Shared Secret Format: ASCII
Shared Secret: <secret_shared_with _radius_server>
Confirm Shared Secret: <secret_shared_with _radius_server>
Key Wrap: [ ]
Port Number: 1812
Server Status: Enabled
Network User: [/] Enable
```

SECURITY > AAA > RADIUS > Accounting > [New...] or Edit

```
Server IP Address(Ipv4/Ipv6): <radius_server_ip_address>
Shared Secret Format: ASCII
Shared Secret: <secret_shared_with _radius_server>
Confirm Shared Secret: <secret_shared_with _radius_server>
Port Number: 1813
Server Status: Enabled
Network User: [/] Enable
```

2. Add/Edit Wireless LAN profile

WLANs > WLANs > WLANs > [Create new...] or Edit

```
Type: [WLAN]
```

```
Profile Name: <wlan_profile>
SSID: <wlan_ssid>
```

WLANs > WLANs > WLANs > [wlan_profile] > Security > Layer 2

```
Layer 2 Security: WPA+WPA2

WPA+WPA2 Parameters
      WPA Policy: [ ]
      WPA2 Policy-AES: [/]

Authentication Key Management
      802.1X: [/] Enable
```

WLANs > WLANs > WLANs > [wlan_profile] > Security > AAA Server

```
            Authentication Servers      Accounting Servers
              [/] Enabled                 [/] Enabled
Server 1    [auth_radius_ip:port]       [acct_radius_ip:port]

Radius Server Accounting
      Interim Update: [/]
```

การตรวจวิเคราะห์และตรวจสอบการทำงานของ Radius server

การทำงานของ Radius server นั้น จะมีการรับข้อมูลการร้องขอการเข้าถึง (Access-Request) จากภายนอก และส่งต่อเป็นลำดับขั้นการทำงานตามลำดับที่ประกาศไว้ในไฟล์คุณสมบัติ โดยลำดับขั้นสำคัญจะอยู่ในไฟล์ไซต์ที่ประกาศใช้ประกอบด้วยไฟล์ sites-enabled/eduroam และไฟล์ sites-enabled/eduroam-inner-tunnel

เมื่อ Radius server ได้รับการร้องขอ จะนำข้อมูลการร้องขอเข้าไปประมวลผลตามขั้นตอนในไฟล์ sites-enabled/eduroam เป็นไฟล์แรก และอาจส่งต่อไปยังการประมวลผลภายในในไฟล์ sites-enabled/eduroam-inner-tunnel หรือส่งต่อไปยัง Radius server เครื่องถัดไป

1. การเขียนภาษา unlang ใช้ใน Radius server

ผู้ใช้สามารถเขียนภาษา unlang เพื่อประมวลผลข้อมูลและตัดสินใจการทำงานได้ เช่น เขียนเพื่อการตรวจสอบรูปแบบบัญชีผู้ใช้ให้เหมาะสม หรือเป็นไปตามกฎของการใช้บริการ eduroam เป็นต้น

รูปแบบของภาษา unlang จะใกล้เคียงกับภาษา C สามารถเขียนให้มีการตรวจสอบค่าหรือตัวแปร กำหนดเส้นทางการทำงานตามรูปแบบของภาษาโปรแกรม และกำหนดผลการทำงาน สามารถเขียนภาษา unlang ได้ในส่วนการประมวลข้อมูล เช่น authrize {}, authenticate {} เป็นต้น

ตัวแปรของภาษา unlang จะเป็นตัวแปรภายใน ไม่สามารถประกาศขึ้นเองได้ ตัวแปรที่เกิดขึ้น จะขึ้นกับ 3 ส่วนคือ ส่วนของการทำงานของโมดูล จากการกำหนดเป็น Attribute ในไฟล์ dictionary และสิ่งที่ลูกข่ายส่งเข้ามาขณะร้องขอบริการ

การกำหนดค่าให้ตัวแปร ใช้ใน section ชื่อ update ใน 3 ตำแหน่ง control, request และ response ตัวอย่าง เช่น

```
update request {
      User-Name := “login_name”
}
update control {
      Proxy-To-Realm := “LOCAL”
}
update response {
      Operator-Name := “1abc.ac.th”
}
```

การอ้างถึงตัวแปร ใช้รูปแบบ %{Variable-Name} เช่น ไม่ต้องดำเนินการใน section ใดๆ เช่น

```
if( ("%{Realm}" =~ /rmutx.ac.th$$/) ) {
      reject
}
```

ตัวกระทำในภาษา unlang มีเช่นเดียวกับโปรแกรมภาษา C แต่มีความยืดหยุ่นกว่า เช่น

<u>การเปรียบเทียบ</u>

```
(!foo)          Negation
```

```
(foo || bar)     Or
(foo && bar)     And
(foo == bar)     Equal
(foo != bar)     Not equal
(foo =~ bar)     Regular expression (match)
(foo !~ bar)     Negate regular expression (not match)
(foo < bar)      Less than
(foo > bar)      More than
```

การกำหนดค่า

```
foo = "value"    Add the attribute to the list, if and only if an
                 attribute of the same name is not already present
                 in that list.
foo := "value"   Add the attribute to the list. If any attribute of
                 the same name is already present in that list, its
                 value is replaced with the value of the current
                 attribute.
foo += "value"   Add the attribute to the tail of the list, even if
                 attributes of the same name are already present in
                 the list. When the right hand side of the
                 expression resolves to multiple values, it means
                 add all values to the tail of the list.
```

ตัวอย่างตัวแปรที่มักมีการอ้างถึง สามารถดูได้จากการรันโปรแกรมแบบ Debug เช่น

การร้องขอ (Request)

```
Received Access-Request Id 0 from 127.0.0.1:59868 to 127.0.0.1:1812
        User-Name = 'user@rmutx.ac.th'
        NAS-IP-Address = 127.0.0.1
        Calling-Station-Id = '70-6F-6C-69-73-68'
        Framed-MTU = 1400
        NAS-Port-Type = Wireless-802.11
```

การตอบกลับ (Response)

```
Sending Access-Challenge Id 0 from 127.0.0.1:1812 to ...
        EAP-Message = 0x010100061920
        Message-Authenticator = 0x00000000000000000000000000000000
Sending Access-Accept Id 9 from 127.0.0.1:1812 to 127.0.0.1:59868
        User-Name := 'user@rmutx.ac.th'
        EAP-Message = 0x03090004
        Message-Authenticator = 0x00000000000000000000000000000000
```

การกำหนดเส้นทางการไหลของโปรแกรม สามารถใช้การกระทำแบบเลือกทางพื้นฐาน คือ if else elseif ได้ เช่น

```
if( ("%{Realm}" =~ /rmutx.ac.th$$/) ) {
      update control {
            Proxy-To-Realm := LOCAL
      }
}
else {
      update request {
            Realm := "eduroam"
      }
}
```

2. การคัดกรองบัญชีผู้ใช้ที่ไม่เหมาะสม

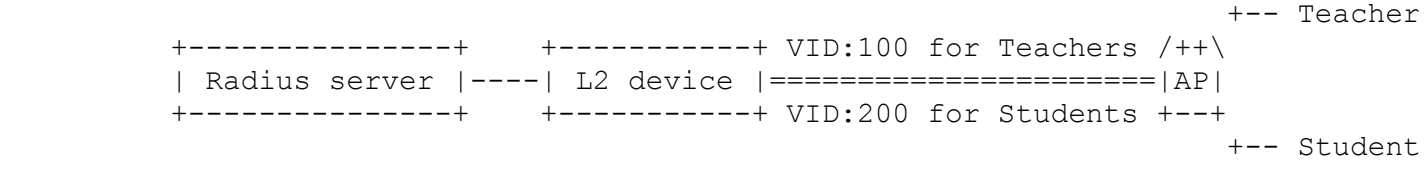
เพื่อคัดกรองบัญชีที่ผิดปกติ จำเป็นต้องเขียนภาษา unlange เพิ่มเข้าไปในไซต์ ตัวอย่างชื่อบัญชีที่ไม่เหมาะสม คือ บัญชีที่ไม่มี realm หรือไม่มี @xxxx หรือบัญชีที่เกิดจากการทำงานโดยอัตโนมัติของบางระบบปฏิบัติการ เช่น 3gppnetwork.org เป็นต้น

ในการติดตั้งนี้ ได้มีการเขียนภาษา unlang เพื่อคัดกรองบัญชีที่ไม่เหมาะสมตามที่ได้รวบรวมไว้แล้ว ไว้ในไฟล์ eduroam-realm-checks.conf และได้นำไฟล์นี้ไปประกอบเป็นส่วนหนึ่งของไฟล์ไซต์ sites-enabled/eduroam

```
authorize {
      $INCLUDE ${confdir}/eduroam-realm-checks.conf
}
```

3. การกำหนดเครือข่ายให้เหมาะสมกับผู้ใช้ที่ต่างกัน

หากต้องการผู้ใช้ต่างการถูกทำให้เชื่อมต่อเข้ากับเครือข่ายที่ต่างกัน สามารถทำได้โดยการส่งข้อมูลหมายเลข VLAN จาก Radius server ไปยัง Wireless Controller (WLC) หรือ Access Point (AP) ได้ ทั้งนี้ ที่ WLC หรือ AP จะต้องประกาศ VLAN ด้วยหมายเลขที่ตรงกับที่ตอบกลับโดย Radius server ตัวอย่างเช่น ต้องการแยกระหว่างอาจารย์ (User-Name: txxxxxx) กับนักศึกษา (User-Name: sxxxxxx) ให้ใช้เครือข่ายที่ต่างกันดังผังเครือข่าย



sites-enabled/eduroam

```
      post-auth {
            update reply {
                  Tunnel-Type := "VLAN"
                  Tunnel-Medium-Type := "IEEE-802"
            }
            if( "%{User-Name}" =~ /^t*/ ) {
                  update reply {
                        Tunnel-Private-Group-Id := 100
                  }
            }
            elseif( "%{User-Name}" =~ /^s*/ ) {
                  update reply {
                        Tunnel-Private-Group-Id := 200
                  }
            }
            else {
            }
      }
```

4. การดูกิจกรรมการทำงานของโปรแกรมโดยละเอียด (Full debuging)

การตรวจสอบการทำงานของโปรแกรม Radius server ว่าทำงานอย่างถูกต้องหรือไม่นั้น วิธีที่ดีที่สุดคือการสั่งรันโปรแกรมแบบ full debuging โปรแกรมจะพิมพ์ผลการทำงาน หรือกิจกรรมที่เกิดขึ้นโดยละเอียดออกทางจอภาพ ในเครื่องหนึ่งเครื่องจะสามารถรันโปรแกรม Radius server ได้เพียงหนึ่งโปรแกรม ดังนั้น หากจะรันโปรแกรมแบบ full debuging จะต้องปิดโปรแกรมเดิมก่อน และสิ้นสุดด้วยการพิมพ์ CTRL+C การดำเนินการเป็นดังนี้

Debian

```
service freeradius stop
freeradius -X
```

CentOS

```
service radiusd stop
radiusd -X
```

5. การบันทึกกิจกรรมใน Log

คุณสมบัติเกี่ยวกับการบันทึกกิจกรรมการทำงานที่กำหนดไว้การติดตั้งนี้ ใช้ไฟล์โมดูลเดิม และมีตำแหน่งการบันทึกตามค่าดั้งเดิมของ Radius server ประกอบด้วย

`sites-enabled/eduroam`

```
authorize {
      # get request from local user and NRO (as IdP and SP)
      # config: ${configdir}/(modules or mods-enabled)/detail.log
      # log: ${logdir}/radacct/<client_ip>/auth-detail-<date>
      auth_log
}
accounting {
      # accouting request from local user and NRO (as IdP and SP)
      # config: ${configdir}/(modules or mods-enabled)/detail
      # log: ${logdir}/radacct/<client_ip>/detail-<date>
      detail
}
post-auth {
      # get result after authentication process (as IdP)
      # config: ${configdir}/(modules or mods-enabled)/detail.log
      # log: ${logdir}/radacct/<client_ip>/reply-detail-<date>
      reply_log
}
pre-proxy {
      # process and forward request to NRO (as SP)
      # config: ${configdir}/(modules or mods-enabled)/detail.log
      # log: ${logdir}/radacct/<client_ip>/pre-proxy-detail-<date>
      pre_proxy_log
}
post-proxy {
      # get response from NRO (as SP)
      # config: ${configdir}/(modules or mods-enabled)/detail.log
      # log: ${logdir}/radacct/<client_ip>/post-proxy-detail-<date>
      post_proxy_log
}
```

ตัวอย่างเนื้อหาในไฟล์ auth-detail

```
Fri Oct 23 22:39:14 2015
        Packet-Type = Access-Request
        User-Name = "eduroam@rmutx.ac.th"
        NAS-IP-Address = 127.0.0.1
        Calling-Station-Id = "70-6F-6C-69-73-68"
        Stripped-User-Name = "eduroam"
        NAS-Port-Type = Wireless-802.11
        Realm = "rmutx.ac.th"
```

ตัวอย่างเนื้อหาในไฟล์ reply-detail

```
Sat Oct 24 02:01:00 2015
        Packet-Type = Access-Accept
        Session-Timeout = 600
        User-Name = "eduroam@rmutx.ac.th"
```

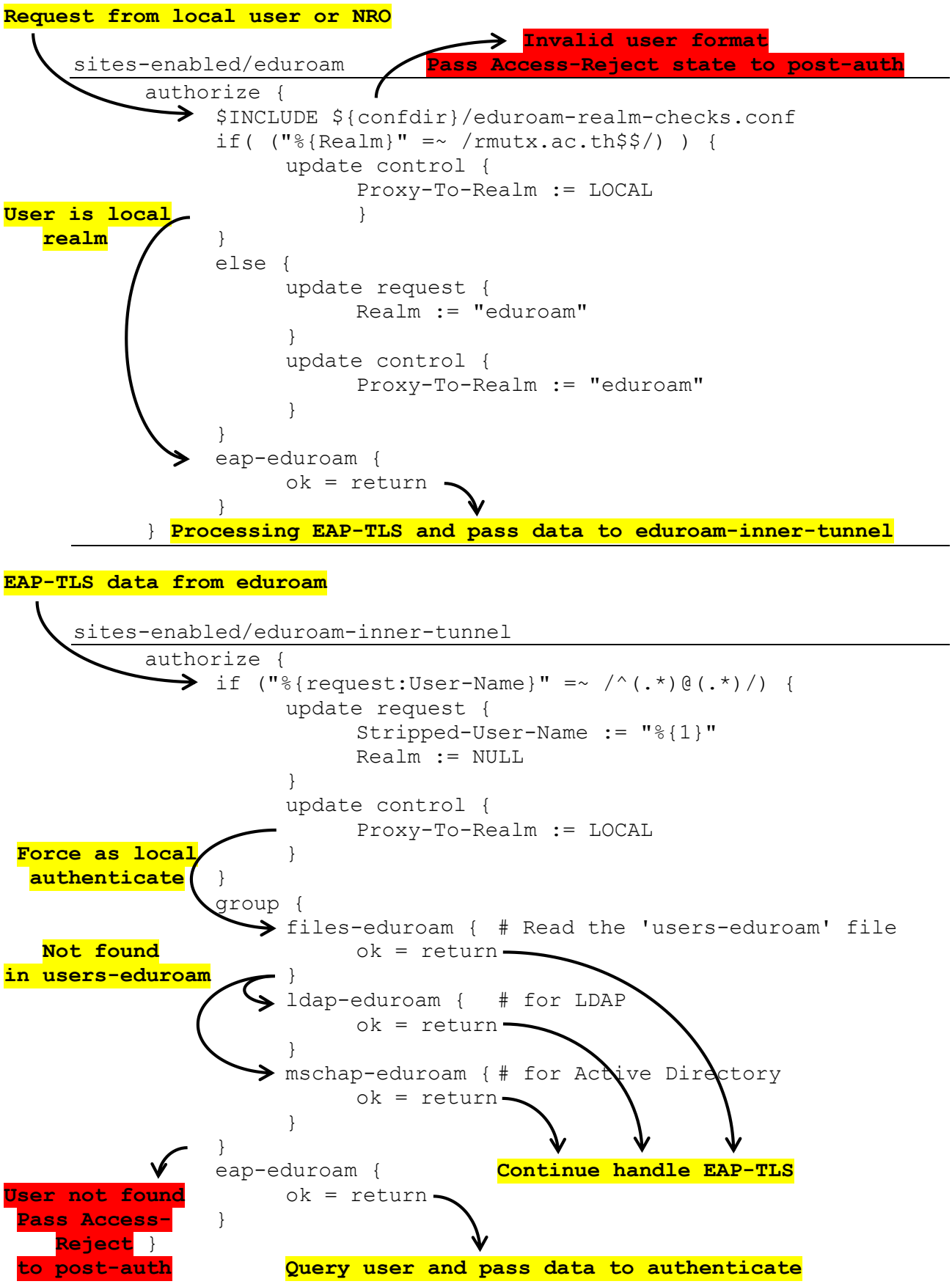
ตัวอย่างเนื้อหาในไฟล์ pre-proxy-detail

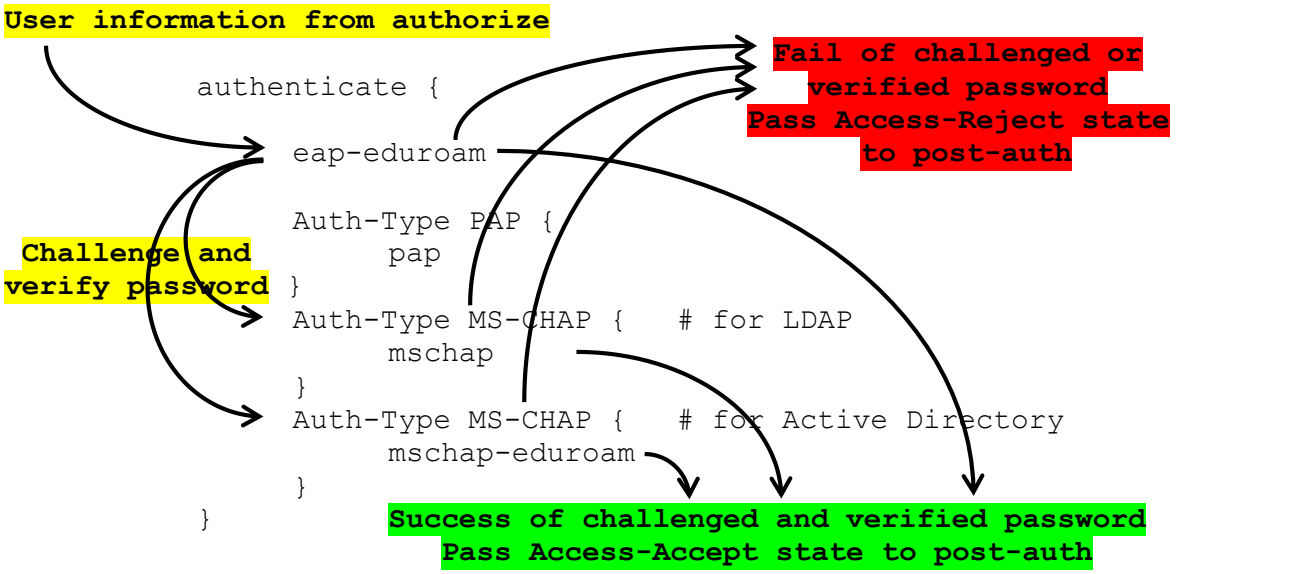
```
Sat Oct 24 00:05:49 2015
        Packet-Type = Access-Request
        User-Name = "eduroam@rmutx.ac.th"
        NAS-IP-Address = 127.0.0.1
        Calling-Station-Id = "70-6F-6C-69-73-68"
        Realm = "eduroam"
        Proxy-State = 0x30
```

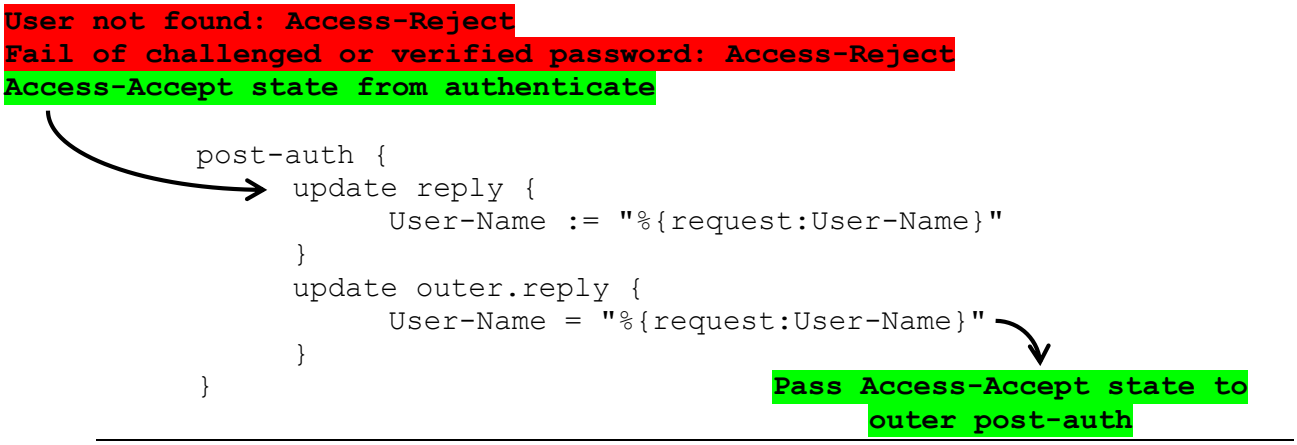
ตัวอย่างเนื้อหาในไฟล์ post-proxy-detail

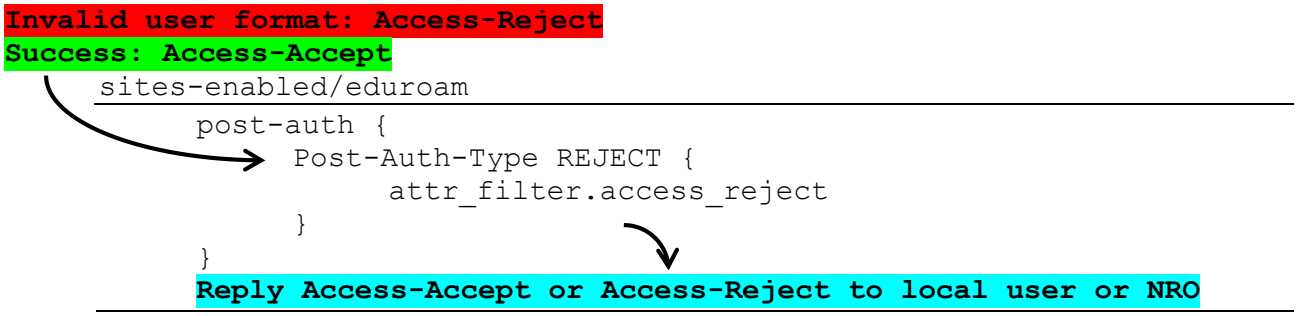
```
Mon Oct 26 15:33:43 2015
        Packet-Type = Access-Accept
        Session-Timeout = 600
        User-Name = "eduroam@rmutx.ac.th"
        Proxy-State = 0x39
```

6. ลำดับการทำงานของคุณสมบัติกรณีทำหน้าที่เป็น IdP (Access-Request)

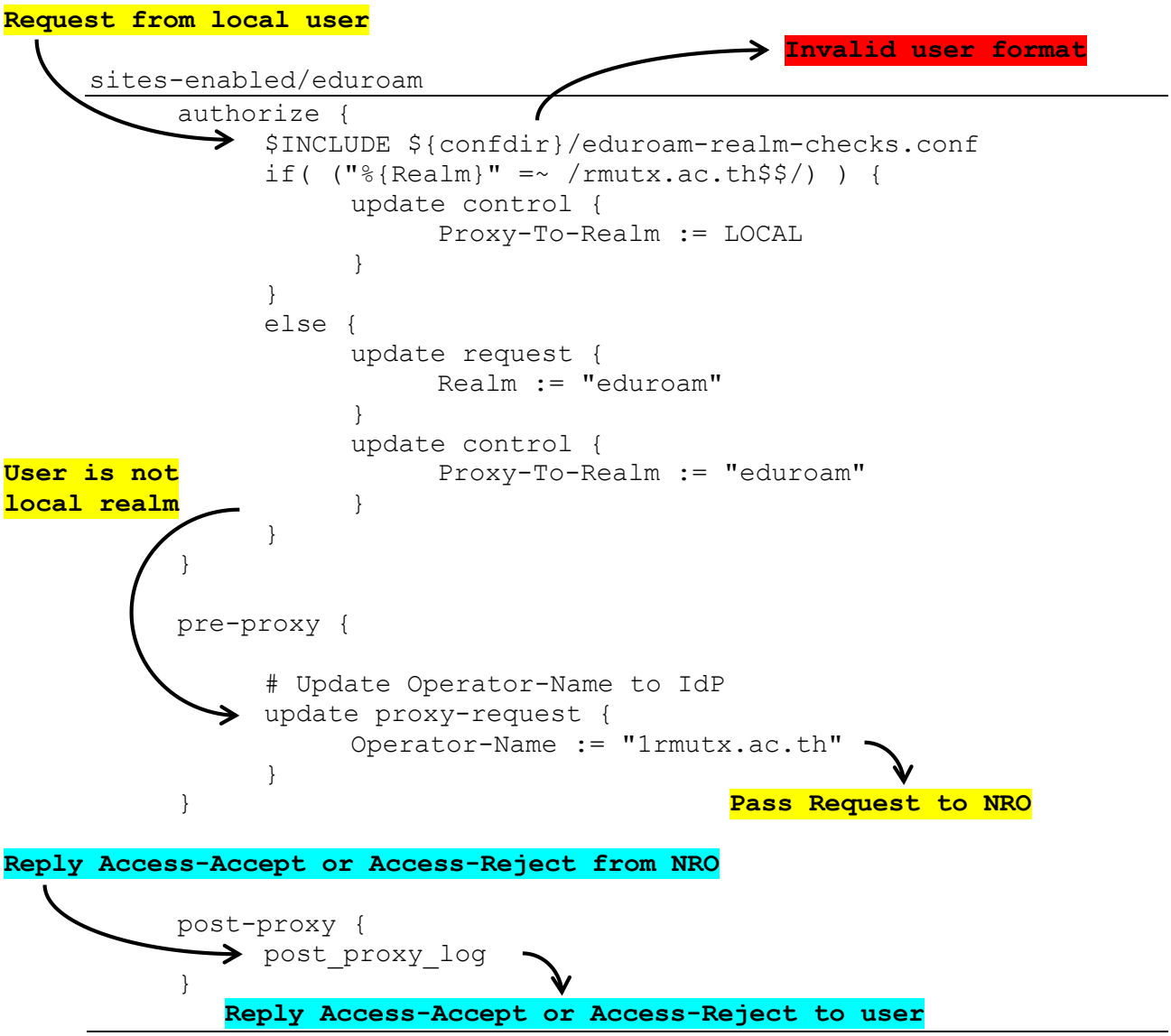



User information from authorize
authenticate {
eap-eduroam
Auth-Type PAP {
pap
}
Auth-Type MS-CHAP { # for LDAP
mschap
}
Auth-Type MS-CHAP { # for Active Directory
mschap-eduroam
}
}
Challenge and verify password
Fail of challenged or verified password
Pass Access-Reject state to post-auth
Success of challenged and verified password
Pass Access-Accept state to post-auth


User not found: Access-Reject
Fail of challenged or verified password: Access-Reject
Access-Accept state from authenticate
post-auth {
update reply {
User-Name := "%{request:User-Name}"
}
update outer.reply {
User-Name = "%{request:User-Name}"
}
}
Pass Access-Accept state to outer post-auth


Invalid user format: Access-Reject
Success: Access-Accept
sites-enabled/eduroam
post-auth {
Post-Auth-Type REJECT {
attr_filter.access_reject
}
}
Reply Access-Accept or Access-Reject to local user or NRO

7. ลำดับการทำงานของคุณสมบัติกรณีทำหน้าที่เป็น SP (Access-Request)



อ้างอิง

- https://www.eduroam.us/node/89
- http://confluence.diamond.ac.uk/display/PAAUTH/Using+Active+Directory+as+authentication+source
- https://wiki.samba.org/index.php/Setup_a_Samba_AD_Member_Server
- http://freeradius.org/radiusd/man/unlang.html