

Overview of 802.1x and EduRoam Setup

802.1x is an IEEE standard which addresses the issue of how to provide network access to only authorized users. This is especially important in the case of WLAN since users do not need to be physically connected to the switch in order to get network connectivity, making it much harder to secure your LAN. The basic idea behind 802.1x is that all switches/access points that perform 802.1x authentication will only allow 802.1x traffic when users first connect to them. Only once they have been authenticated, and authorized will their normal traffic be allowed to pass through. In the case of wired Ethernet, the port through which an authenticated user connected will be enabled, while for wireless, the access point will negotiate a unique key with the authenticated user wireless card. The unique key is used to encrypted traffic between the user and the access point. Two common standards for the key are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). The important thing to note is that unlike static WEP or WPA keys which are not secure in large enterprise LAN environments, the key negotiated during 802.1x authentication is dynamic, unique for each user and also changes on a regular basis, making it immune to sniffing attacks.

Authorization in 802.1x is made possible through the use of Extensible Authentication Protocol (EAP) which allows client requests to be forwarded to the the authentication server. The decision of whether or not to grant access is sent back. Originally developed for authenticating dial-up access, EAP is now extended to be used for Ethernet (EAPOL ♦ EAP over LAN) and for wireless (EAPoW ♦ EAP over Wireless).

In 802.1x terminology, the client is referred to as the supplicant, the wireless access point or switch is the authenticator. The other major component is the authentication server which is typically a Radius server. A typical sequence of operations for 802.1x is shown in the figure below (taken from ref. [1]). Note that once the client is authenticated, the EAPoW 4-Way Handshake is the part which negotiates a dynamic key between the access point and the client station.

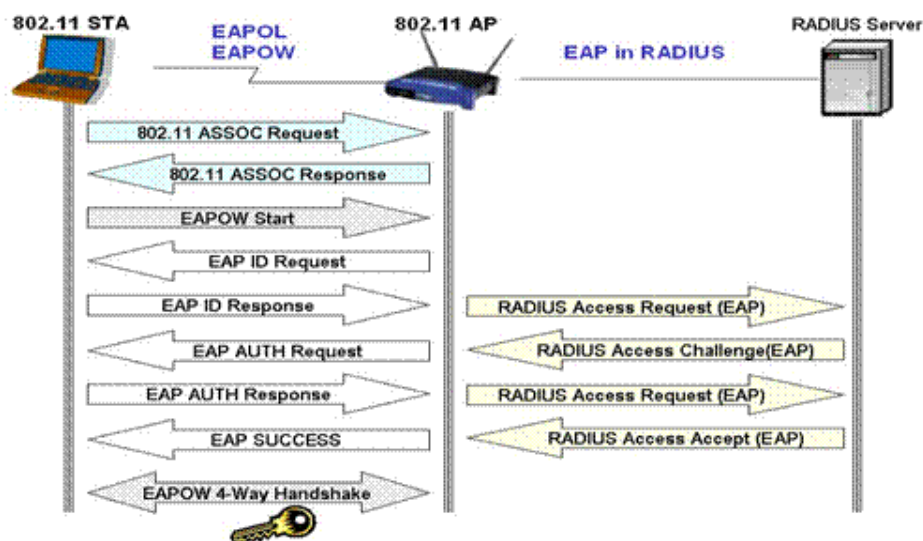


Figure 1 EAP Authentication

EAP

The advantage of using EAP is that it allows many different methods of authentication to be used. It simply wraps the authentication message inside EAP envelop to be forwarded between the supplicant, authenticator and the authentication server. Currently the following EAP methods are the most common:

EAP-MD5

A client sends a hashed password to the authentication server which checks it against the hashed password password stored in its database. This method is good for wired Ethernet but is no good for wireless since the hashed password can be intercept by the man-in-the-middle. For eg. a rouge station can masquerade as an access point and trick users into giving them the hashed passwords. EAP-MD5 also does not allow for the negotiation of dynamic keys.

- *LEAP.*

Cisco proprietary protocol which aims to improve on EAP-MD5 by requiring mutual authentication between the client and server (ie. client as well as server authentication) and allows for negotiation of encryption keys at the end of authentication phase. However, it is still susceptible to sniffing and dictionary attacks against hash passwords. Only suitable for all Cisco equipment network.

- *EAP-TLS.*

Requires both client and server to authenticate with each other via PKI which can be in the form of X.509 certificates or smart cards. The exchange is done inside a TLS tunnel which makes it resistant to man-in-the-middle attacks. The drawback is that it requires heavy PKI infrastructure to be in place.

- *EAP-TTLS and PEAP.*

Both of these remove the burden of client certificates since only the server needs to have a certificate. For client authentication, instead of certificate based, both of these allows an extensible set of authentication such as MD5, CHAP or MS-CHAP v2. At the moment, both of these are only Internet Draft stage. The ways they both work can roughly be described as implementing the EAP inside of TLS, inside of EAP, inside of RADIUS. PEAP is currently only supported by Microsoft and CISCO. EAP-TTLS are supported by many however, it requires new software to be installed on the client system.

Radius Authentication Server

The Remote Authentication Dial In User Service (RADIUS) protocol ([RFC 2865](#)) was designed to perform centralized authorization, authentication and accounting (AAA) for traditional dialup access using PPP and SLIP. Network Access Servers (NAS) no longer need to maintain a list of users and their passwords, but instead can delegate the authentication and authorization tasks to a central Authentication Server. This allows the number of NAS to grow and be distributed in many different places within a large organization without worries about their manageability. Additionally, Radius servers can be distributed to handle different ♦realms♦. For example, radius server 1 can handle ♦example1.org♦ realm while radius server 2 can handle ♦example2.org♦ realm. A user from realm ♦example2.org♦ but attempting to authenticate at radius server 1 can still be successful since radius server 1 simply acts as a proxy and forwards the authentication request to radius server 2 and relays back the success or failure status of the authentication request to the user.

With the introduction of 802.1x, Radius was extended to support EAP (RFC 3579). The sequence of operations for a client authenticating with a Radius server is illustrated below. Note the extra operations with a Radius server which allow, once the user is authenticated, the added ability to perform accounting. The first phase, Connect, using EAP was shown in more details in Figure 1.

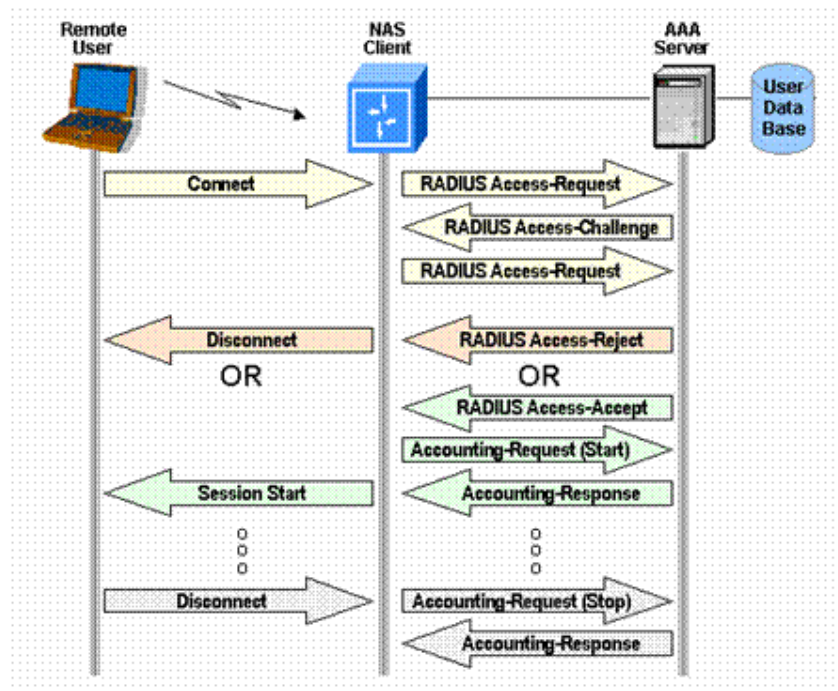


Figure 1 Radius Authentication and Accounting

All Radius messages are carried in UDP datagrams. The structure of a Radius message consists of the Message Type (e.g. Radius Access-Request), sequence number (to keep track of the authentication conversation since UDP is being used), the length, an Authenticator and a series of Attribute-Value pairs. The use of the Authenticator is to ensure the integrity of the communication between the NAS and the Radius server. It is also used to obscure user password from the NAS; only the AAA server needs to know about the user password for authentication (or at least that the hashes of the user password match on both sides). The Attribute-Value pairs are used to encapsulate EAP information within Radius messages.

Eduroam

EduRoam is an initiative to enable wireless network access for members from participating institutions. The basic idea is that if a user from one institution goes to another institution, he can login using his home institution credentials to obtain wireless access. More information about the EduRoam project can be obtained from <http://www.eduroam.org>. You will need to contact the maintainers of the project to get details on how to join the EduRoam network. For the setup below, we will simply assume that you are setting up for the domain `example1.org` and that you want people from `example2.org` to be able to login and obtain network access and vice versa. This is similar to the setup for the EduRoam project.

Behind the scene, Radius servers are used as the authentication servers because of their distributed nature; the radius server simply proxies the authentication request to the user home institution radius server. In the following section we outline the steps for setting up a Radius server to enable wireless access using EAP-TTLS with MD5 (PAP) as the inner authentication method. In addition, the Radius server uses an LDAP store to check whether the particular user is allowed to access the network as well as password verification. The radius server product we use is FreeRadius which has the advantage that it is free while supporting a wide variety of authentication methods.

Setting up FreeRadius

These instructions assume you have an LDAP store already at <ldap://ldap.example.edu.au> and each users are found under the Distinguished Name `ou=People,dc=example,dc=edu,dc=au`. Each user needs to implement the Radius Profile schema found inside the FreeRadius package (see below) under the `doc` directory. The schema file is called `RADIUS-LDAPv3.schema`. For each user that you want to allow network access, they need to have the

attribute `dialupAccess` set to `yes`. The Radius server uses this field to check whether a particular user is allowed network access, even before any password authentication is done. Values in *italics* for the configurations below need to be substituted with your particular settings. If you are using Debian unstable distribution, due a bug in the code, you need to roll back the package `libltdl3` to version 1.4.2-4 (ie. the version that is being used in the Debian stable distribution).

You can install the `freeradius` package from a recent Linux distribution or download and install the latest stable FreeRadius software from <http://www.freeradius.org>. If you choose the build the software yourself the instructions are:

1. Untar the software and change to the directory containing the software.
2. `$ make clean`
3. `$./configure --sysconfdir=/etc`
4. `$ make`
5. `# make install`

All the configuration files are found in the `/etc/raddb` directory.

To configure a wireless access point with address 1.2.3.4 add this to `clients.conf`:

```
client 1.2.3.4 {
    secret = radius_secret
    shortname = short_name
}
```

The `radius_secret` matches the one set in the wireless access point. It should be random and long.

The `short_name`

appears in log messages referring to the access point. The first few components of the DNS name of the access point is often used.

The EAP details are set in `eap.conf`:

```
eap {
    default_eap_type = tls
    timer_expire = 60
    ignore_unknown_eap_types = no
    tls {
        private_key_file = path to a private key for your Radius server
        certificate_file = path to the corresponding certificate file for the Radius server
        CA_file = path to the file containing the certificate of the Certificate Authority
        dh_file = ${raddbdir}/certs/dh
        random_file = /dev/urandom
        fragment_size = 1024
        include_length = yes
        copy_request_to_tunnel = no
        use_tunneled_reply = no
    }
    ttls {
        default_eap_type = md5
        copy_request_to_tunnel = no
        use_tunneled_reply = no
    }
}
```

Note that even though `ttls` is used, you still need to correctly set up the TLS section since TTLS relies on the TLS settings to provide the tunnel. Note that although EAP-MD5 is not recommended for use in wireless (since the password is not protected), EAP-MD5 inside an a TLS tunnel is quite secure. Other available options are CHAP and and MSCHAP v2. However both of these cannot be used in conjunction with LDAP unless user passwords in LDAP are stored in clear text (not recommended!).

In the `proxy.conf` file define the DNS domains for your site.

```
realm example.edu.au {
```

```

    type = radius
    authhost = LOCAL
    accthost = LOCAL
}

```

Perhaps you are not going to use FreeRADIUS's user database or FreeRADIUS's LDAP gateway, but another RADIUS server (say 2.3.4.5):

```

realm example.edu.au {
    type = radius
    authhost = 2.3.4.5:1812
    accthost = 2.3.4.5:1813
    secret = radius_secret
    nostrip
}

```

Note that the default port for RADIUS authentication is 1812 and the default port for RADIUS accounting is 1813.

When using LDAP edit `radius.conf`:

```

modules {
    ...
    ldap {
        server = ldap.example.edu.au
        identity = "cn=Administrator,dc=example,dc=edu,dc=au"
        password = bind password for "identity"
        basedn = ou=People,dc=example,dc=edu,dc=au
        filter = "(uid=%{Stripped-User-Name}:-%{User-Name})"
        # If your LDAP server supports TLS you should use it
        start_tls = yes
        tls_cacertfile = path to the certificate of the CA of your LDAP server certificate
        tls_randfile = /dev/urandom
        access_attr = "dialupAccess"
        dictionary_mapping = ${raddbdir}/ldap.attrmap
        ldap_connections_number = 5
        password_attribute = userPassword
    }
    ...
    $INCLUDE ${confdir}/eap.conf
}
...
authorize {
    preprocess
    auth_log
    eap
    files
    ldap
}
...
authenticate {
    Auth-Type PAP {
        pap
    }
    Auth-Type LDAP {
        ldap
    }
    eap
}
...

```

Now configure the access point to use WPA (TKIP) encryption.

And configure a client and test.

References

- [1] B. Aboba, P. Calhoun
RFC3579: RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)
<ftp://ftp.isi.edu/in-notes/rfc3579.txt>

[2] Lisa Phifer

Tutorials: 802.1x Port Access Control for WLANS

<http://www.wi-fiplanet.com/tutorials/article.php/3073201>

[3] Lisa Phifer

Using Radius for WLAN Authentication Part 1,2,3

<http://www.wi-fiplanet.com/tutorials/article.php/3289231>