λWireless Lab
λ802.1x Authentication

Network Startup Resource Center
www.nsrc.org

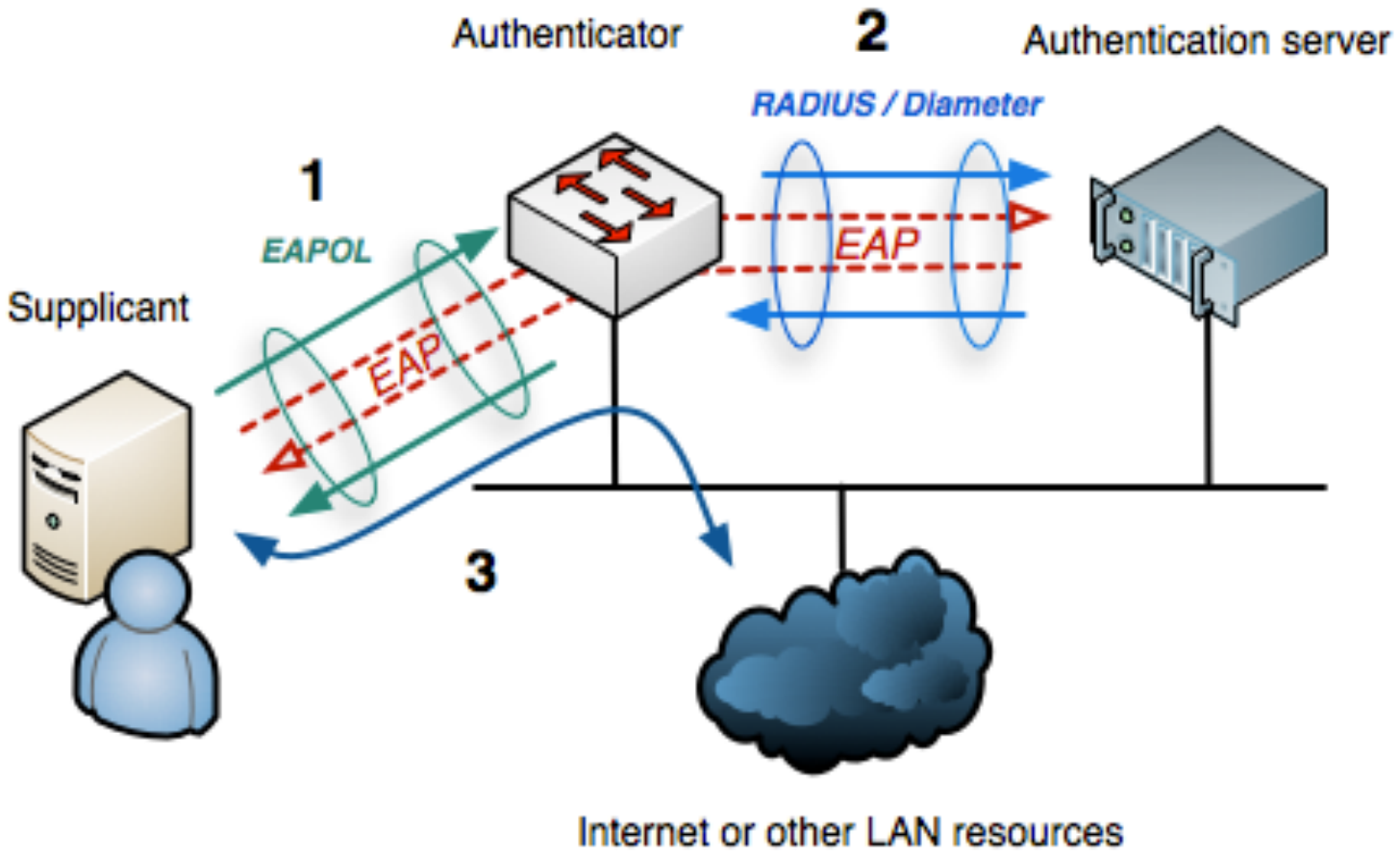
Last edit: Patrick Okui, Nov 2015

Recap

- Recall that 802.1x has multiple components to it:
  - The authentication server is what has the usernames and passwords or knows how to find them.
  - The authenticator is the network device the client is trying to connect to. Can be an Ethernet switch or in our case WiFI access point. The authenticator will only grant network access (even to DHCP) if the authentication server accepts the credentials given.
  - The supplicant is the software on the client that tries to sign in. Handled by the operating system

The following sections cover the various components you need to have configured to get this to work.

802.1x/EAP – How does it work



Source: Wikipedia

Authentication server: radius

FreeRadius installation

- Your instructors will have assigned one or more virtual machines to your group.
- You will work together to install radius on the server(s).
- If you have more than one machine per group you can run the same installation steps on each server – most authenticators (dealt with in the next session) can try multiple authentication servers one after the other incase one is down.

Your instructors may have also installed a Radius server that you can all use and skip to the Authenticator section.

Write down the ip address(es) of the authentication server(s) you are going to use.

FreeRadius

- Very popular high performance open source radius server.
- We shall install version 3 (version 2 as of this writing is deemed legacy).
- It has very active development and can authenticate against many sources including SQL databases, Active directory, etc.
- Lots of options available:
  - Some access points with controllers have an inbuilt radius server.
  - Windows server has an optional (free) install of Network Policy Server (NPS) or
  - Internet Authentication Service (IAS) for Microsoft only environments.

Basic installation

As of Nov 2015 Ubuntu still ships with FreeRadius 2.x which is a legacy release.
We will install version 3 from an *unofficial* repository.
Note that version 3 configuration is not fully compatible with version 2.
If you have version 2 you should recreate your configuration
Type the following on the server you'll use for authentication

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:freeradius/stable-3.0
$ sudo apt-get update
$ sudo apt-get install build-essential freeradius
```

Radius clients

- The FreeRadius server refers to any device that asks it to authenticate a user as a “client”.
- For our purposes the clients are the authenticators.
- i.e the access points that are running 802.1X.
- Each authenticator needs an entry in /etc/freeradius/clients.conf that looks as follows.

```
client grpX-wifi1 {
    secret                          =       secrets321
        ipaddr                              =       10.10.0.1X
    nastype             =       other
}
```

Concerning heartbleed

- Since most radius auth (especially as required by 802.1X) is wrapped in SSL,
- the radius project updated its distribution to refuse to start if linked against a
- version of SSL that has significant vulnerabilities. In this case the heartbleed
- (the version we have just installed has the fix).
- To get FreeRadius to start on our systems,
- we need to edit the file /etc/freeradius/radiusd.conf. Under the section that starts

```
security {
```

- Add the line

```
allow_vulnerable_openssl = 'CVE-2014-0160'
```

Authentication sources

Radius can authenticate from a number of sources where the actual username/password credentials are stored.

- Active Directory is popular amongst Microsoft Windows environments. This is a specialised form of LDAP.
- SQL databases can hold your user authentication and be updated by web based frontends.
- Users can be statically defined in the freeradius configuration.
- While this may be useful for testing we strongly recommend you use another method for production as this does not scale.

auth: Active Directory

Your instructors have installed an active directory server for you to authenticate against. They will avail the following details which you will need to write down to use this server:

- The workgroup
- The realm
- The DNS name of the server
- The admin account + password you will use to join the domain
- A username/password you will use for web access in the next section

auth: Active Directory

- Now we install samba to allow the linux authentication server talk to the active directory
- and open the configuration file in a text editor

```
$ sudo apt-get install samba winbind krb5-user
$ sudo vi /etc/samba/smb.conf
```

- Insert the correct workgroup in the line that says
- workgroup = WORKGROUP
- Bellow that line add the following lines substituting as necessary:

```
security = ads
winbind use default domain = no
password server = KDC1.WS.NSRC.ORG
  //your AD-server
realm = WS.NSRC.ORG          //your realm
```

auth: Active Directory

- Next, edit /etc/krb5.conf in the [libdefaults] section just under the line that says default_realm = WS.NSRC.ORG add lines as follows (watch for case sensitivity)

```
dns_lookup_realm = false
dns_lookup_kdc = false
```

- Next, look for the [realms] section, and
- add the following just bellow at the top of the
- section (replace with the correct REALM given
- by the instructor). The realm name must
- be uppercase.s

```
[realms]

WS.NSRC.ORG = {
    kdc = kdc1.ws.nsrc.org
}
```

- Next, ensure the file /etc/nsswitch.conf has winbind added to the following lines.

```
passwd:         compat winbind
group:          compat winbind
shadow:         compat winbind
protocols:      db files winbind
services:       db files winbind
netgroup:       nis winbind
```

- Reboot the machine at this point with `sudo reboot`

- Once the machine is back up try and join the
- domain using the administrator username and password availed to you.

```
$ net join -U Administrator
```

- Next is to attempt to authenticate with the wireless username/password that has been given to you.

```
$ ntlm_auth --request-nt-key
--domain=WSNSRCORG --username=<your username>
```

- If this was successful, then NTLM will report it thus:

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

- We also need freeradius to have read access to the winbind privileged directory

auth: Active Directory

- Next, edit the mods-available/mschap file
- Add a new line just under the mschap { line that looks like this:

```
with_ntdomain_hack = yes
```

- Find the line that starts with:

```
#       ntlm_auth = "/path/to/ntlm_auth
```

- And modify it so that it begins like:

```
        ntlm_auth = "/usr/bin/ntlm_auth
```

auth: Active Directory

- Edit the file /etc/freeradius/mods-available/eap
- Find the line that reads

```
default_eap_type = md5
```

- Change it to

```
default_eap_type = peap
```

- Find the section that begins

```
tls-config tls-common {
```

- In there remove the comment on the line

```
#       random_file = /dev/urandom
```

Authenticator

Wifi AP configuration

- Next we need to configure our access points to do 802.1X authentication against the radius we have built or the instructors have built.
- How to do this depends on the units available. In most cases you either use a web interface to the unit or a controller software. The authentication for the relevant SSID needs to be set to WPA2/Enterprise.
- Ensure you have the following details to be filled into the authenticator:
  - IP address of the radius server. DNS names work in many cases but introduce possible fragility to the auth process.
  - Shared secret on the radius server as configured in the clients.conf file.

Supplicant

End user device configuration

- The end user device needs to join the wifi domain. Any operating system after WindowsXP service pack 3 has a supplicant capable of 802.1X. Only new additions (which we haven't used) require newer operating systems.
- The user needs the wifi username/password that has been entered into the authenticator back-end in the previous section.
- Since we have used self signed certificates, the devices will popup an error saying they don't trust the certificate. In this case it is fine.
- For larger deployments, the opensource CAT tool developed by eduroam (can be extended to provision other SSIDs) is available. See https://cat.eduroam.org