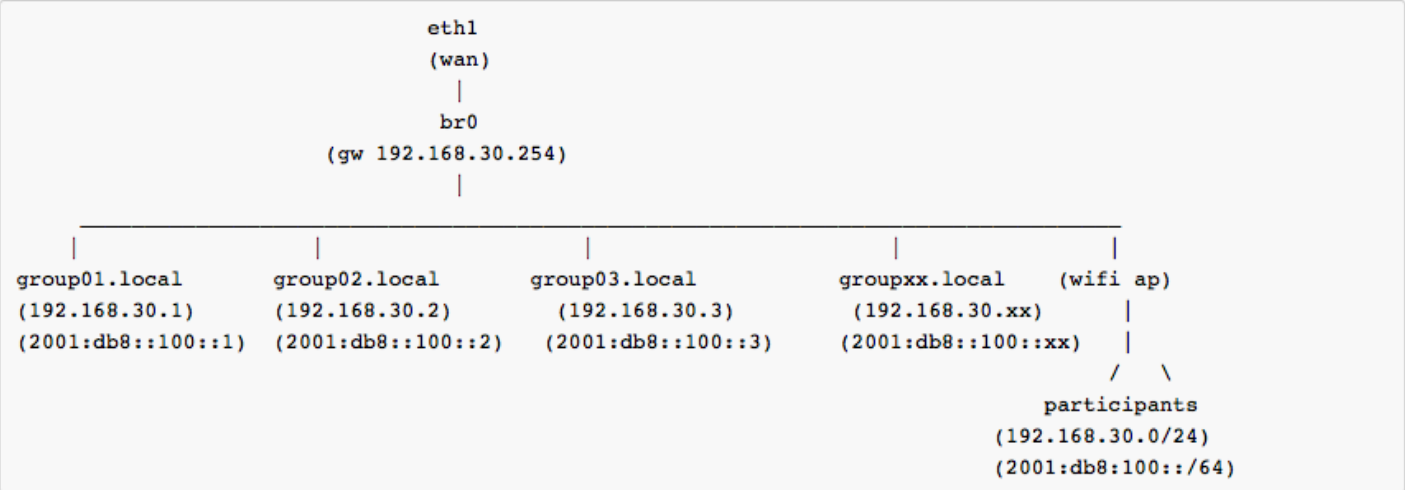
Module: Public Key based SSH

Objective: As part of this hands-on module, you will be configuring certificate based autehntication for Secure SHell (SSH) access.

Topology

The following will be the topology used for this lab. Note that the IP addresses are examples only. When working on the lab, use the actual IP addresses as indicated by the instructors. For the purpose of this guide, the IP address of 192.168.30.X or 2001:db8:100::X will refer to your Virtual Machine (VM).



Lab Notes

- Confirm interface name:
  - On the VM, check the IP configuration to see the interface Name

    ```
    ifconfig
    OR
    ip route show | grep " src " | cut -d " " -f 3,12
    ```

  - In this guide the interface name is `eth0`. Depending on the version of Ubuntu the interface name may be `enp0s3` or something different. Where `eth0` is used in this guide replace it with your interface name.

- Virtual Machine details
    - Ubuntu 18.04 LTS/LXC
    - Hostname = groupXX.apnictraining.net
    - Domain name = apnictraining.net
    - IPv4 Address = 192.168.30.xx
    - IPv6 Address = 2001:db8:100::xx
    - xx = group ID as allocated by the instructor
    - Username `apnic` and password `training`

Required Software

- Linux:
    - ssh utility
    - ssh-keygen

Lab setup

Login to the server (SSH from the jumphost to the container using the `username` and `password` given above) where `X` is the VM number.

1. MacOS - Open a terminal window, using `⌘ Cmd` + `spacebar` (this opens spotlight search) then type in `terminal`
2. Linux - Open a terminal window, using `Ctrl+Alt+t`
3. Windows - Open putty or command prompt

Log into the jumphost (refer to google document to confirm jumphost) using:

```
ssh apnic@202.XXX.XX.XX
```

Exercise 1 - Password Based Authentication

Confirm username and password based authentication by using SSH

1. Type in the following:

    ```
    ssh apnic@192.168.30.X
    ```

2. When asked for a password, type in `training` and press enter.

Exercise 2 - Public Key Authentication

To generate OpenSSH compatible keys for Linux

Linux Client

1. Change into the `.ssh` folder. If the SSH folder does not yet exist, create it manually:

```
mkdir .ssh
chmod 0700 .ssh
cd .ssh
```

2. Type in the following:

```
ssh-keygen -t rsa -b 4096 -C your_email@example.com
```

-t = Specifies the type of key to be created.

-b = Specifies the number of bits in the key (longer is more secure).

-C = Changes the comment for a keyfile

```
apnic@ubuntu:~$ ssh-keygen -t rsa -b 4096 -C your_email@example.com
Generating public/private rsa key pair.
Enter file in which to save the key (/home/apnic/.ssh/id_rsa): 20190920_example
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in 20190920_example.
Your public key has been saved in 20190920_example.pub.
The key fingerprint is:
SHA256:730v7hk88sF0aELFEA0j7l4kjzbl4dfxCdFh8GK4YA0 your_email@example.com
The key's randomart image is:
+---[RSA 4096]----+
|          E. =O+o.|
|          .o..o*. |
|          ooo=+ o |
|         ...Xo.oo+|
|        S =.* =.+|
|          + o B . |
|           o . *  |
|          . . oo= |
|           . .+=o.|
+----[SHA256]-----+
apnic@ubuntu:~$
```

3. The above command will create two files in the current directory. In the image the keys were saved as `20190920_example`

   - Private key (`20190920_example`) - DO NOT share
   - Public key (`20190920_example.pub`) - this can be shared

   NOTE: Replace `20190920_example` with the name that your file was saved as.

4. View the public key by typing the following:

```
cat 20190920_example.pub !# Linux
cat ~/.ssh/20190920_example.pub !# MacOS
```

5. Select the output and copy to clipboard

Save The Public Key On The Server

Now, you need to paste the copied public key in the file ~/.ssh/authorized_keys on the server. For this you will need to copy onto a server that you are not currently logged into, for example group30 server

1. Log in to the other server using ssh with username `apnic`

```
ssh apnic@192.168.30.30
```

2. If the SSH folder does not yet exist, create it manually:

```
mkdir .ssh
chmod 0700 .ssh
touch .ssh/authorized_keys
chmod 0644 .ssh/authorized_keys
```

3. Edit the ~/.ssh/authorized_keys file:

```
sudo vi .ssh/authorized_keys
```

4. Paste the SSH public key into the file. Tap the `i` key on your keyboard & right-click your mouse to paste.

5. To save, tap the following keys on your keyboard (in this order): `Esc` , `:wq` press Enter.

6. Type `exit` to log out of the server.

NOTE: On linux and MacOS you can use some alternative ways to copy the public key to the server.

```
ssh-copy-id apnic@192.168.30.XX
```

or

```
cat ~/.ssh/20190920_example.pub | ssh apnic@192.168.30.XX "mkdir -p ~/.ssh && cat
\
>> ~/.ssh/authorized_keys”
```

Create a Profile to Save Your Server’s Settings

Linux Client

1. Log in to the server using the Identity File

   ```
   ssh -i ~/.ssh/20190920_example apnic@192.168.30.XX
   ```

2. To specify which Identity File to use to login to a server, create a config file

   ```
   sudo vi ~/.ssh/config
   ```

3. Tap the `i` key on your keyboard, and type the following into the file

   ```
   Host 192.168.30.XX
     IdentityFile ~/.ssh/Lab1_rsa
   ```

4. To save, tap the following keys on your keyboard (in this order): `Esc` , `:wq` press Enter.

5. Type in the following:

   ```
   ssh apnic@192.168.30.XX
   ```

Now log in and it should not prompted for a password. However, if a passphrase was set on the public key, it will ask for you to enter the passphrase at that time (and every time you log in, in the future).

References:

https://wiki.apnictraining.net/*media/netsec20190923-mo/1.3b*ssh_lab.pdf

https://www.ssh.com/ssh/