Lab 1.1 - Single Packet Authorization Setup

Table of contents

1 Introduction

This lab contains the setup for a brief tutorial for implementing Single Packet Authorization (SPA) with Firewall Knock Operator (or fwknop for short, pronounced eff - double u - nop).

In this activity, it is recommended that you partner up, or have multiple machines available for use. You will need to designate one machine as the "client" in the following steps and

another machine will as the "server". It is recommened that you step through it twice where one machine is the client and the other machine is the server and then reverse roles for the second run through.

Note that this lab only includes the setup. Lab 1.2 includes the execution of SPA.

2 Overview of Commands and Relevant Bash Tips

The following commands will be used in this lab.

2.1 fwknop

The `fwknop` command is used to execute the Firewall Knock Operator software. fwknop is open source software that allows us to easily implement Single Packet Authorization. In short, fwknop will be used on the client to send a SPA packet. On the server, it will be used to verify the packet, add a firewall rule to allow access to the port we want access to for a limited time, then remove the firewall rule after the time expires.

2.2 iptables

The `iptables` command is used to manage the rules that determine what to do with network packets, that is, it is for managing firewall rules.

Common flags used with `iptables` are as follows:

```
-L      lists all the rules int he current tables
-v      displays additional details about each rule (verbose)
-n      displays numeric values where applicable
        ex: for the http port, show 80 instead of http
-A      appends a rule to a specified chain
-p      used to specify a protocol such as tcp for tcp traffice like HTTP and SSH
-j      used to specify an action to take if a packet matches (j for jump)
```

As part of this lab, you will eventually setup a firewall rule that blocks all traffic in order to work with fwknop.

2.3 ssh

The `ssh` command allows secure access to a remote machine.

2.4 scp

The `scp` command stands for `secure copy`. You can use this command to copy information from one machine to another using SSH. The general format is as follows:

```
scp [options] source destination
```

Common options are as follows:

```
-r     recursively copy subdirectories and files.
-P     specifies the port. The default port is 22.
```

2.5 Bash Tips

For this lab, you will be setting various variables in bash. Recall that in bash, when setting a variable, we do not put spaces on either side of the `=` sign.

Example:

```
x=10    # correct
y = 12  # incorrect
```

3 Wifi Setup

In order for this lab to work as intended, you may need to modify the settings in your virtualization software and in the network options within your virtual machine. Complete the following to ensure that everything is in working order.

1. If on campus, use `ip addr` to check if your ip address is in the form 138.47.X.X. If so, then you can skip this section. If not, complete the steps that follow.
2. In the virtual machine settings for your virtualization software, ensure that the network is set to use a bridged connection and that the proper connection is selected (most likely Wifi). To do this with VirtualBox, select the machine, open the settings, navigate to `Network` and be sure the adapter is "Attached to" the Bridged Adapter.
3. Within the virtual machine, open the system configuration and navigate to the network settings. Find the settings for IPv4 and make sure DHCP is set to Automatic.
4. At this point you may need to disconnect from the internet and reconnect, or you may need to restart your machine altogether.
5. Once complete, check that your ip address is in the form 138.47.X.X if on campus.

4 Client Installations

4.1 fwknop-client

1. On the client, search for fwknop in package directory:

```
sudo apt search fwknop
```

2. In the output, you should see some of the following (results may vary):

- fwknop-apparmor-profile
- fwknop-client
- fwknop-server

3. Install fwknop-client.

```
sudo apt install fwknop-client
```

4.2 nmap

On the client, if you do not already have nmap installed, install it with the following command.

```
sudo apt install nmap
```

5 Server Installations

5.1 fwknop-server

1. On the server, search for fwknop in package directory:

```
sudo apt search fwknop
```

2. In the output, you should see some of the following (results may vary):

- fwknop-apparmor-profile
- fwknop-client
- fwknop-server

3. Install fwknop-server.

```
sudo apt install fwknop-server
```

5.2 openssh-server

On the server, confirm you have the openssh server running on port 22 with the ss command (socket statistics). Using the flags `-a` and `-t` will filter the information down to just the TCP ports. Using the flag `-n` will show you the port number.

```
ss -atn # -n will show the port number (22 for ssh)
ss -at  # no -n will show the label `ssh` in place of the value 22.
```

You should see an output that contains at a minimum the following (results may vary):

```
State       Local Address:Port
LISTEN                *:22
```

or

```
State       Local Address:Port
LISTEN                *:ssh
```

If you don't already have openssh server installed, you will need to install it with the following command.

```
sudo apt install openssh-server
```

After installing it, confirm it is running with the `ss` command shown previously.

6 Client Setup

1. On the client, get your ip address with the command `ip addr`, `ip address`, or `ifconfig` and note it elsewhere (a text file or a piece of paper) as your client's ip address.

   ```
   # run one of the following to get your ip address
   ip addr
   ip address
   ifconfig
   ```

2. On the server, get the ip address and note it somewhere (a text file or a piece of paper).
3. On the client, set some variables to represent the `SERVER_IP` and `CLIENT_IP`. Replace the x's in the command shown below with the appropriate values. Recall that bash does not accept spaces before and after the `=` symbol.

```
CLIENT_IP=138.47.x.x
SERVER_IP=138.47.x.x
```

4. On the client, set another variable to represent the server's ssh port.

```
SSH_PORT=22
```

5. On the client, generate a key for the ssh port using the client's ip and the server's ip. Note that this should all be typed on one single line.

```
# write the following on 1 single line
fwknop -A tcp/$SSH_PORT -a $CLIENT_IP -D $SERVER_IP
--key-gen --use-hmac --save-rc-stanza
```

   You should receive an output similar to the following:

```
[+] Wrote Rijndael and HMAC keys to rc file: /home/josh/.fwknoprc
```

6. On the client, To confirm the keys were generated, view the contents of the `.fwknoprc` file

```
cat ~/.fwknoprc
```

   At a minimum, you should see something similar to the following two lines inside the file:

```
KEY_BASE64          dGhlIGJpZ2dlc3Qgam9rZSBpbiBjb21wdXRlciBzY2llbmNlIGlz
HMAC_KEY_BASE64     eW91ciBwcm9ncmFtcw==
```

7 Server Setup

1. On the client, copy the keys you generated to the server using the `scp` command (secure copy). Note that we are copying the keys to a file called `access.conf` at `/home/username` on the server. Additionally, be sure to replace the word `username` with the name of the user of the server.

```
scp -P $SSH_PORT ~/.fwknoprc username$SERVER_IP:~/copiedfile.conf
```

   When running the above command, you will be prompted for the password of the server.

2. On the server, open the file at /etc/fwknop/access.conf and find the section that has the following information:

```
SOURCE              ANY
KEY_BASE64          __CHANGE_ME__
HMAC_KEY_BASE64     __CHANGE_ME__
```

i. Remove `ANY` and replace it with the ip address for the client machine.

ii. Replace both instances of `__CHANGE_ME__` by copying and pasting the keys from the ~/access.conf file you moved onto the server.

iii. Add a line after SOURCE that is labeled OPEN_PORTS with a value of tcp/22.

iv. Confirm the file looks similar to the following. The values for SOURCE, KEY_BASE64, and HMAC_KEY_BASE64 will be different that those shown.

```
SOURCE              138.47.163.65
OPEN_PORTS          tcp/22
KEY_BASE64          dGhlIGJpZ2dlc3Qgam9rZSBpbiBjb21wdXRlciBzY2llbmNlIGlz
HMAC_KEY_BASE64     eW91ciBwcm9ncmFtcw==
```

v. Save and close the file.

3. On the server, remove the file you put at the home directory ~/access.conf.

```
rm ~/copiedfile.conf
```

4. Complete the following on the server to enable fwknop

i. Open the file `/etc/default/fwknop-server` with nano or vim.

ii. Change `START_DAEMON="no"` to `START_DAEMON="yes"`.

iii. Save and close the file.

iv. Open the file `/etc/fwknopd/fwknopd.conf` with nano or vim.

v. Find the line with PCAP_INTF and uncomment it. By default it shows eth0. It can stay as that for linux machines.

vi. Save and close the file.

vii. Restart the fwknop server.

```
sudo service fwknop-server restart
```

5. On the server, check the status of the fwknop-server to be sure it is running. This will also show some logs at the bottom. The status should show “active”.

```
sudo service fwknop-server status
```