LAB: Filtering (UFW)

Lab Environment

The workshop WiFi:

- SSID: `workshop`
- PASS: `iij/2497`

Hosts - Virtual machines (Ubuntu 18.04LTS/LXC)

- Hostname: `nsXX.workshop`
- IPv6: `fd00:2497:1::X`
- IPv4: `10.0.0.X`

Where `X` and `XX` is yoru group ID. For group 1, hostname is `ns01.workshop`, IPv6 address is `fd00:2497:1::1`, and IPv4 is `10.0.0.1`.

Exercise 1: Check UFW status

UFW (Uncomplicated Firewall) is filtering management package bundled with recent Linux distributions.

Check the current status of UFW using the `ufw` command.

```
$ sudo ufw status
```

By default, UFW is inactivated.

Exercise 2: Activate UFW

The default policy of UFW is deny all the incoming connections. Before activating UFW, we need to setup a ssh filtering rule, otherwise, we will lose ssh access from outside just after activating UFW.

```
$ sudo ufw allow ssh
```

And activate UFW.

```
$ sudo ufw enable
```

Check the status.

```
$ sudo ufw status
```

`ufw allow ssh` is equal to `ufw allow 22/tcp`. You can specify the rules in more detailed manner whenever needed.

Exercise 3: Check the filter

Now the only ssh port is accessible. Check what happens.

- HTTP
- DNS

Check which ports are open using `nmap`.

Exercise 4: Open ports

Now the only ssh port is available on your virtual server. You cannot access your web server anymore.

Add a new rule to allow HTTP and DNS access.

```
$ sudo ufw allow http
$ sudo ufw allow domain
```

Exercise 5: Close ports

You have two different filter rules to close a port.

```
$ sudo ufw deny http
```

```
$ sudo ufw reject http
```

What are the difference?

Exercise 6: Delete rules

If you don't need some rules anymore, you can delete rules. First, check the index of each rule.

```
$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 53                         ALLOW IN    Anywhere
[ 4] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 5] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 6] 53 (v6)                    ALLOW IN    Anywhere (v6)
```

Remove a rule by specifing the rule index.

```
$ ufw delete 2
```

You need to check the index each time whenever you remove multiple rules, since the index number will change after deletion.

Exercise 7: Change default

By default, ufw deny all the incoming connections, but allow all the outgoing connections. In some cases, you may want to limit only specific services.

Let's try to configure to deny all the outgoing connections, and allow some selected servies only on yoru virtual server.

> Note: Check the man page to change the default behavior.

Exercise 8: Fine grained rulesets

If you want to access to your web server from a specific client, you can specify the rule as below, for example.

```
$ sudo ufw allow proto tcp from 10.0.0.2 to any port 80
```

Advanced Exercise: Deny ICMP Echo Request

ICMP/ICMPv6 rules are not handled by ufw. If you want to modify the filtering behavior of ICMP/ICMPv6, you need to change the rule files applied before the ufw rules are applied. The files are located under the `/etc/ufw/` directory. The ICMP/ICMPv6 rules are defined in `before.rules` and `before6.rules`.

To deny all the incoming ICMP Echo Request, change the line in `before.rules` as below.

```
root@ns49:/etc/ufw# diff -u before.rules-dist before.rules
--- before.rules-dist   2017-08-17 16:34:49.000000000 +0000
+++ before.rules    2019-02-21 03:43:53.835588294 +0000
@@ -35,7 +35,7 @@
 -A ufw-before-input -p icmp --icmp-type source-quench -j ACCEPT
 -A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
 -A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
--A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
+-A ufw-before-input -p icmp --icmp-type echo-request -j DROP

 # ok icmp code for FORWARD
 -A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
```

For ICMPv6, change the `before6.rules` file.

```
--- before6.rules-dist  2017-08-17 16:34:49.000000000 +0000
+++ before6.rules   2019-02-21 03:43:39.507619245 +0000
@@ -41,7 +41,7 @@
 -A ufw6-before-input -p icmpv6 --icmpv6-type time-exceeded -j ACCEPT
 # codes 0-2
 -A ufw6-before-input -p icmpv6 --icmpv6-type parameter-problem -j ACCEPT
--A ufw6-before-input -p icmpv6 --icmpv6-type echo-request -j ACCEPT
+-A ufw6-before-input -p icmpv6 --icmpv6-type echo-request -j DROP
 -A ufw6-before-input -p icmpv6 --icmpv6-type echo-reply -j ACCEPT
 -A ufw6-before-input -p icmpv6 --icmpv6-type router-solicitation -m hl --hl-eq 25
5 -j ACCEPT
 -A ufw6-before-input -p icmpv6 --icmpv6-type router-advertisement -m hl --hl-eq 2
55 -j ACCEPT
```

Please beware that ICMP/ICMPv6 messages are important control messages that may affect protocol behavior of other protocols or applications (e.g. Path MTU discovery). So please think twice when you change the behavior of ICMP responses. (Echo Request/Echo Reply may be fine. Actually, some administrators configure not to respond ICMP Echo Request messages)