Packet Filter Firewalls

Linus Bein Fahlander (*linusfa@kth.se*)

A public repository containing the files generated and used in this lab can be found on GitHub.

Task 1 - Building a Firewall

1.2 Network Permission

UFW configuration after completing this sub task

```
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere on eth1           ALLOW IN    10.0.20.0/24

Anywhere                   ALLOW OUT   10.0.20.0/24 on eth1
```

Q 1

`ufw` comes with an easy way of setting default policies, the ones needed for the desired configuration are:
`ufw default deny incoming`
`ufw default deny outgoing`
`ufw default deny routed`

To allow communication within the internal network I set these rules:
`ufw allow in on eth1 from 10.0.20.0/24`
`ufw allow out on eth1 from 10.0.20.0/24`

Q 2

The difference between `deny` and `reject` is that packets that are not allowed will be answered with a reject if you use `reject` where with `deny` the packet will simply be dropped.

The one you choose depends on what you want to achieve with the firewall.
Is the interface only internal facing maybe you want to reject so that developers will get a clearer message that the request was rejected and that the interface doesn't just not answer.
However it is a good idea in most cases to just drop the packet to not give the possible attacker any information and not to waste computing cycles creating the reject message.

1.3 Permitting a Service

UFW configuration after completing this sub task

```
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere on eth1           ALLOW IN    10.0.20.0/24
10.0.10.1 22/tcp on eth0   ALLOW IN    10.0.10.0/24

Anywhere                   ALLOW OUT   10.0.20.0/24 on eth1
```

Command used to add this rule:
`ufw allow in on eth0 to 10.0.10.1 port 22 from 10.0.10.0/24 proto tcp`

Q 3

The main advantage is that opening the port allows for remote maintenance outside of the internal network.
The disadvantages are that this opens up an attack vector to the firewall and also if no restriction is set to the user that login via ssh, then they can access the rest of the network freely by having nested ssh sessions.

1.4 Stateful Filtering

UFW configuration after completing this sub task

```
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere on eth1           ALLOW IN    10.0.20.0/24
10.0.10.1 22/tcp on eth0   ALLOW IN    10.0.10.0/24

Anywhere                   ALLOW OUT   10.0.20.0/24 on eth1

Anywhere on eth0           ALLOW FWD   10.0.20.0/24 on eth1
```

Command used to add this rule:
`ufw route allow in on eth1 out on eth0 from 10.0.20.0/24 to any`

Q 4

TCP and UDP can be verified using `nc` servers as you can run that server is either TCP or UDP mode. When I host a `nc` server on the `outside-host` I can successfully connect to it from the `inside-host`, both when the server is in TCP and UDP mode.

This is not the case if I instead host the server on the `inside-host` and try to connect from the `outside-host`.

1.5 Opening Ports

UFW configuration after completing this sub task

```
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere on eth1           ALLOW IN    10.0.20.0/24
10.0.10.1 22/tcp on eth0   ALLOW IN    10.0.10.0/24

Anywhere                   ALLOW OUT   10.0.20.0/24 on eth1

Anywhere on eth0           ALLOW FWD   10.0.20.0/24 on eth1
10.0.20.2 9000 on eth1     ALLOW FWD   10.0.10.2 on eth0
```

Command used to add this rule:
`ufw route allow in on eth0 out on eth1 from 10.0.10.2 to 10.0.20.2 port 9000`

Q 5

The configuration can be verified by hosting a `nc` server on the `inside-host` first on a random port, `1024` for example, and then on `9000`.
The `outside-host` can only connect to the server when it is hosted on port `9000`.

1.6 Blocking Ports

UFW configuration after completing this sub task

```
Status: active
Logging: on (low)
Default: deny (incoming), deny (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
Anywhere on eth1           ALLOW IN    10.0.20.0/24
10.0.10.1 22/tcp on eth0   ALLOW IN    10.0.10.0/24

Anywhere                   ALLOW OUT   10.0.20.0/24 on eth1

135 on eth0                DENY FWD    10.0.20.0/24 on eth1
Anywhere on eth0           ALLOW FWD   10.0.20.0/24 on eth1
10.0.20.2 9000 on eth1     ALLOW FWD   10.0.10.2 on eth0
```

Command used to add this rule:
`ufw route insert 1 deny in on eth1 out on eth0 from 10.0.20.0/24 to any port 135`

Q 6

The configuration can be verified by hosting a `nc` server on the `outside-host` first on a random port, `1024` for example, and then on `135`.
The `inside-host` can only connect to the server when it is not hosted on port `135`.