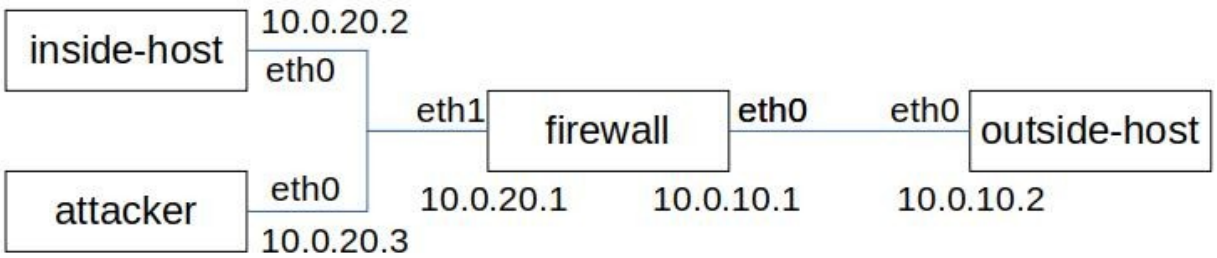
Packet Filter Firewall

The topology of the hosts present in these attacks is as follows:



Task 1 - Building a firewall

Question 1 - What rules do you need in order to set up the default policy, and to allow communication with the firewall and the internal network?

1,1. Default policy

It is good practice to take a conservative approach, and state that everything that is not explicitly allowed, is forbidden: Therefore, the firewall is set to drop all packets.

This was done with the following commands:

- `ufw default deny outgoing`
- `ufw default deny ingoing`
- `ufw default deny routed`

1.2 Network Permissions

Now it is time to start allowing some carefully chosen traffic through the firewall. To allow communication within the network the following rules were applied.

- `ufw allow in on eth1 from 10.0.20.0/24`
- `ufw allow out on eth1 from 10.0.20.0/24`

The ufw status verbose now looks like this:

```
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
```

Question 2 - with ufw you can choose between “deny” and “reject” when you want to disable traffic. What is the difference between deny and reject? What are the advantages/disadvantages? Make an experiment where you set up the firewall to reject instead of deny, and explain what you see.

The difference between the rules ”reject” and ”deny” is that when reject is used the destination will receive an error message of why the packets were not delivered while deny will simply just drop the packets without notifying the sender. What to choose depends on the nature of the implementation. If packets from the inside needs to be controlled it may be preferred to reject since it will give the user information about why the packets was not delivered. However, it may not be good to reject incoming packets since it will give a potential attacker more information to work with and also waste some computation cycles in order to send the messages.

1.3 Permitting a Service

SSH (Secure shell) is a common service for remote management of firewalls. Here, rules are created which allow a host from the external network to connect to the firewall with SSH. SSH runs on port 22 and uses only TCP. This was done with the following command:

- `ufw allow in on eth0 to 10.0.10.1 port 22 from 10.0.10.0/24 proto tcp`

The ufw status verbose now looks like this:

```
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
```

Question 3: What are the advantages and disadvantages with opening up for ssh login from the outside?

The advantages of having a SSH port open is that it enables for example maintenance of the firewall and inside network from the outside network which can be useful. The disadvantage is that this creates opportunities for a potential attacker to infiltrate the network with nested SSH given the right credentials.

1.4 Stateful Filtering

In most cases we want to allow hosts on the internal network to connect to the external network, e. g., the Internet. However, we do not want hosts on the Internet to be able to connect to hosts on the internal network. This can be done by creating a simple rule that forwards the traffic from the firewall with the following command.

- `ufw route allow in on eth1 out on eth0 from 10.0.20.0/24 to any`

The ufw status verbose now looks like this:

```
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
```

Question 4: How can you verify that traffic from the outside going in is prevented, while traffic from the inside going out is allowed? Consider TCP, UDP as well as ICMP traffic. What tests do you perform to validate your configuration?

This can be verified by hosting a netcat server on the outside host and then try to connect to it from the inside host, both in TCP and UDP mode. To verify that incoming connections are blocked a netcat server is started on the inside host while trying to connect to it from the outside host, this is as expected not allowed by the firewall.

1,5 Opening ports

Sometimes, you want computers on the outside of the network to have access to a specific service on the inside. Therefore, rules are added to the firewall so that the outside host can reach a service at port 9000 on the inside host, for UDP as well as TCP. The following command was used:

- `ufw route allow in on eth0 out on eth1 to any port 9000`

The ufw status verbose now looks like this:

```
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
9000 on eth1               ALLOW FWD   Anywhere on eth0
```

Question 5: How can you verify that traffic to the service on the inside host at port 9000 is allowed, but no other services?

As similar to the last verification a netcat server is started on the inside host, the outside host should only be able to connect to the inside host on port 9000.

1,6 Blocking Ports

Sometimes you do not want your internal users to be able to connect to the Internet on a specific port at all. One commonly blocked port is 135 which is used for Windows file sharing. Make sure your firewall blocks all traffic on port 135 (both TCP and UDP) from the computers on the internal network. This is done with the following command:

- `ufw route insert 1 deny in on eth1 out on eth0 to any port 135`

This will make sure that the command is on top and has the highest priority. The ufw status verbose now looks like this:

```
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    Anywhere on eth1

Anywhere on eth0           ALLOW FWD   10.0.20.0/24 on eth1

9000 on eth1               ALLOW FWD   Anywhere on eth0
```

Question 6: How can you verify that the inside host cannot reach port 135 on the outside, but no other outgoing traffic is blocked?

This verification process is similar to the previous ones. A netcat server is started on the outside host on port 135. The inside host shall only be able to connect to it if the port number is not 135.

The firewall now has the following capabilities:

- Connections on port 135 from the inside hosts are blocked.
- Connections on the SSH port are allowed to the firewall host from the outside.
- Connections on the port 9000 are allowed to the internal hosts from the outside.
- Connections directly to and from the firewall are allowed from the internal networks.
- Connections from the inside are allowed out.
- Connections from the outside are blocked.

Task 2 - Defending Against SSH Brute-force Attacks

Question 7: How does ufw support protection against denial of service attacks?

By using the `limit` command the firewall will deny connections from an IP if 6 or more logins are attempted in the last 30 seconds. This can be done with either of the two following commands:

- ufw limit to any port 22
- ufw limit ssh

This creates the following configuration:

```
Status: active

Logging: on (low)

Default: deny (incoming), deny (outgoing), deny (routed)

New profiles: skip

To                         Action      From
--                         ------      ----

22                         LIMIT IN    Anywhere

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    Anywhere on eth1

Anywhere on eth0           ALLOW FWD   10.0.20.0/24 on eth1

9000 on eth1               ALLOW FWD   Anywhere on eth0
```

Note that the rule is inserted at the top of the list in order to work as intended. Otherwise, the two other rules will match with the packet first and thereby allow an infinite number of logins.

Question 8: How can you verify that the protection works?

Blocking of SSH brute-force attacks is verified by making multiple connections to the firewall from the outside host. This is done by typing `ssh ubuntu@10.0.10.1` followed by ctrl c to be able to make six connections within 30 seconds. After six connection attempts the following message is provided:

```
ssh: connect to host 10.0.10.1 port 22: Connection refused
```

This verifies that the protection works as intended.

Task 3 – Ping and the Internet Control Message Protocol (ICMP)

The ping tool is usually used to test the reachability of a host that implements the Internet protocol (IP). It operates by sending packets using the Internet Control Message Protocol (ICMP) of the type 'echo request' to the host whose reachability is to be tested, and processing the response from that host (if any).
Ping can also be used in attacks, such as ping flooding, ping of death, and smurf attacks. Your overall task here is to design firewall rules that will allow inside-host to ping outside- host, but at the same time prevent an attacker from doing an attack where large amounts of ICMP echo requests or ICMP echo replies reach the inside-host.

This is done by adding rules IP packet filter with the help of Iptables which is a user-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall. With the following rule, incoming ICMP request are dropped by the firewall.

```
iptables –I INPUT 1 –j DROP –p icmp ––icmp–type echo–request
```

However, the packets are still being forwarded by the the firewall into the inside network. This can be prohibited by the following command:

```
iptables –I FORWARD 1 –j DROP –p icmp ––icmp–type echo–request
```

Now, the outside host is not able to ping either of the firewall or the inside network.