COSC 3364 – Principles of Cybersecurity
Lab 10

Provide screenshots where * is indicated.

Ethan Conner

Firewalls

To create firewall rules on a system, you can use the **iptables** command. This command allows you to create rules that provide one or more of the following functions:

- Creates rules that:
  - Filter (block or allow) network packets
  - Forward packets to another system
  - Perform network address translation (NAT)
  - Mangle (modify) network packets

Keep in mind that when you use the iptables command to create firewall rules, the rules take effect immediately. This can be an issue if you are remotely logged in to a system and create a rule that ends up immediately blocking access to your own session.

Important terms:

- *Filtering point*: Point where filtering rules are applied to packets and packets are dealt with appropriately
- *Table*: A list of rules
- *Chain*: A set of rules that determine what action to take on a specific packet at a specific filtering point
- *Target*: An action that takes place once a matching rule is found

Target Types:

- When one of these types of targets is executed, additional rules are ignored:
  - ACCEPT: Allow the packet to continue to the next step
  - DROP: Discard the packet
  - REJECT: Send a response to the origin of the packet informing it of the rejection, and discard the packet
- When this type of target is executed, additional rules are still evaluated
  - LOG: Create a log entry

Typically DROP is considered a more secure method than REJECT because hackers will use REJECT responses as a means to probe a system or network. Even a negative response provides the hacker with useful information. For example, a REJECT could indicate that the destination machine might be worth hacking into (why secure an unimportant system), or it could indicate that some ports are blocked but others are allowed.

Default Chain Policy:

- Each chain has a default chain policy
- If you have not edited a chain, it should be set to ACCEPT
  - If a packet does not match any DROP or REJECT rules in the chain, it will continue to the next step
- On a high-security system you might want to change the default to DROP
  - Only packets that match an ACCEPT rule are allowed to move to the next step

Using iptables to Filter Incoming Packets:

- To see the current firewall rules:
  - -t filter means you are working with the filter table
  - -L specifies the INPUT chain
- To delete an individual rule use the -D option
  - Example: `iptables -D INPUT 1`
  - This deletes the first rule in the INPUT chain
  - You do not have to specify -t filter because filter is the default table
- To delete all rules in a chain use the -F option
  - Example: `iptables -F INPUT`
- To block all packets from a specific host, use the -s option
  - Example: `iptables -A INPUT -s 192.168.10.100 -j DROP`
  - -s specifies the source
  - -A places the new rule at the end of the chain
  - -j jumps to the specified target

Filtering by Protocol:

- It is common to filter packets by protocol

- Could be a protocol like ICMP, TCP, or UDP
- Could be a protocol associated with a specific port, such as telnet, which uses port 22

- Example: to block ICMP:
  - `iptables -A INPUT -p icmp -j DROP`
- See /etc/protocols for a list of protocols that can be used with the -p option

Filtering by Port:

- To block a specific port, use the -m option and either --sport or --dport
  - For incoming packets, use --dport
  - Example: iptables `-A INPUT -m tcp -p tcp --dport 23 -j DROP`
  - You can also specify a range of ports, such as --dport 1:1024

Multiple Criteria:

- You can combine criteria to create a more complex rule
- For the rule to match, all the criteria must match
- Example: To match both a protocol and source IP address:

```
iptables -A INPUT -p icmp -s 192.168.125.125 -j DROP
```

Saving the Rules:

- Unless saved, all changes made using iptables are lost upon reboot
- Save the rules into a file using the iptables-save command
- Normally the output of this command is sent to the screen but you can redirect it to a file: `iptables-save > /etc/iptables/rules.txt`
- Where to save the rules and how they are loaded automatically depends on the distro
- Some distros have front-end utilities that configure firewall rules and also save them
  - firewalld on Red Hat Enterprise Linux
  - UFW on Ubuntu
- You can create a shell script that restores rules from the saved file and then execute the script during the boot process
  - Example: `iptables-restore < /etc/iptables/rules.txt`

Using iptables to Filter Outgoing Packets:

- To block access to external sites, create a firewall rule on the OUTPUT-filter chain
  - Example: `iptables -A OUTPUT -m tcp -p tcp -d 10.10.10.10 --dport 80 -j DROP`
- You could use REJECT instead of DROP to be more user-friendly
- You could choose to allow the access but create a log entry
  - Example: `iptables -A OUTPUT -m tcp -p tcp -d 10.10.10.10 --dport 80 -j LOG`

Implementing NAT:

- Forms of NAT
  - DNAT: Destination NAT, used when you want to place servers behind a firewall and still provide access from an external network
  - SNAT: Source NAT, used when you have an internal network with statically assigned private IP addresses
  - MASQUERADE: Used when you have an internal network with dynamically assigned private IP addresses (e.g. DHCP) Using MASQUERADE, you can funnel access to the Internet via a single machine that has a live IP address (an address that is routable on the Internet).
    - A single command handles all the internal systems
    - Example: `iptables -t nat -A POSTROUTING -j MASQUERADE`

1. Display the current firewall rules in the filter table with line numbers*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
```

2. Display the current firewall rules for incoming traffic.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
labuser1@ML-RefVm-535928:~/Desktop$
```

3. Display the current firewall rules for outgoing traffic.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L OUTPUT --line-numbers
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
labuser1@ML-RefVm-535928:~/Desktop$
```

4. Develop a filter table with the following rules:*
    a. Accept incoming traffic on TCP for ports 1-1023
    b. Accept incoming traffic on UDP for ports 1-1023
    c. Log outgoing traffic on IP addresses 192.168.1.0/24 for port 23
    d. Drop incoming traffic on IP address 192.168.10.100 for port 55555
    e. Reject incoming TCP traffic on port 54321

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1                tcp  --  anywhere             anywhere             tcp dpts:tcpmux:1023
2                udp  --  anywhere             anywhere             udp dpts:1:1023
3    DROP        tcp  --  192.168.10.100       anywhere             tcp dpt:55555
4    REJECT      tcp  --  anywhere             anywhere             tcp dpt:54321 reject-with
icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    LOG        tcp  --  anywhere             192.168.1.0/24       tcp dpt:telnet LOG level w
arning
labuser1@ML-RefVm-535928:~/Desktop$
```

5. Delete rule b from the filter table.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -D INPUT 2
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1                tcp  --  anywhere             anywhere             tcp dpts:tcpmux:1023
2    DROP        tcp  --  192.168.10.100       anywhere             tcp dpt:55555
3    REJECT      tcp  --  anywhere             anywhere             tcp dpt:54321 reject-with
icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    LOG        tcp  --  anywhere             192.168.1.0/24       tcp dpt:telnet LOG level w
arning
labuser1@ML-RefVm-535928:~/Desktop$
```

6. Save the current firewall rules in the filter table.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables-save > /home/labuser1/Documents/ruleslab10.
txt
labuser1@ML-RefVm-535928:~/Desktop$
```

7. Flush all incoming traffic rules.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -F INPUT
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    LOG        tcp  --  anywhere             192.168.1.0/24       tcp dpt:telnet LOG level w
arning
labuser1@ML-RefVm-535928:~/Desktop$
```

8. Flush all outgoing traffic rules.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -F OUTPUT
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
labuser1@ML-RefVm-535928:~/Desktop$
```

9. Restore the saved firewall rules in the filter table.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables-restore < /home/labuser1/Documents/ruleslab
10.txt
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1                tcp  --  anywhere             anywhere             tcp dpts:tcpmux:1023
2    DROP       tcp  --  192.168.10.100       anywhere             tcp dpt:55555
3    REJECT     tcp  --  anywhere             anywhere             tcp dpt:54321 reject-with
icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    LOG        tcp  --  anywhere             192.168.1.0/24       tcp dpt:telnet LOG level w
arning
labuser1@ML-RefVm-535928:~/Desktop$
```

10. Funnel access to the internet for output device eth0.*

```
labuser1@ML-RefVm-535928:~/Desktop$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```