Security Topics

Firewall exercises using iptables

Contents

1 Introduction

In this exercise we will see examples of how to set up packet filtering on a host running Ubuntu Linux using the iptables firewall.

2 Notes

- Commands preceded with “$” imply that you should execute the command as a general user - not as root.
- Commands preceded with “#” imply that you should be working as root.
- Commands with more specific command lines (e.g. “RTR-GW>” or “mysql>”) imply that you are executing commands on remote equipment, or within another program.

3 Goals

- Install iptables
- Understand basic iptables commands
- Build packet filtering rules to restrict access to certain applications depending on the target audience
- Learn how to permanently save the iptables rules

4 Installation

You can check if iptables is installed in your system by doing

```
$ sudo iptables -V
```

If it is not installed:

```
$ sudo apt-get install iptables
```

5 Filtering policy:

5.1 Host must be able to access all of its own applications (localhost)

5.2 Host must allow SMTP connections from anywhere

5.3 Host must allow SSH access only from the campus network

5.4 Host must allow access to to the web server only from the local network

5.5 Host must allow access to SNMP only from the local network

5.6 All incoming TCP/UDP traffic must be blocked, except for established

connections ## ICMP must be rate-limited to 3 packets per second

For that, we are going to create a text file, which will help us build the ruleset more easily. Make sure to replace "X" with your group number when necessary

```
$ vi /home/sysadm/iptables.sh

# Flush any existing rules
iptables -F

# Permit any incoming packet on the loopback interface
iptables -A INPUT -i lo -j ACCEPT

# SMTP must be open so that we can accept mail from the world
iptables -A INPUT -p tcp --dport 25 -j ACCEPT

# SSH restricted to the campus network
iptables -A INPUT -s 10.10.0.0/16 -p tcp --dport 22 -j ACCEPT

# HTTP and HTTPS restricted to the local network only
iptables -A INPUT -s 10.10.X.0/24 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -s 10.10.X.0/24 -p tcp --dport 443 -j ACCEPT

# SNMP restricted to the local network only
iptables -A INPUT -s 10.10.X.0/24 -p udp --dport 161 -j ACCEPT
```

```
# Rate-limit ICMP traffic to 3 packets per second
iptables -A INPUT -p icmp -m recent --set
iptables -A INPUT -p icmp -m recent --update --seconds 1 --hitcount 3 -j DROP

# Then, permit all traffic initiated from this machine to come back
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# And finally, block all incoming TCP traffic
iptables -A INPUT -s 0/0 -p tcp --tcp-flags SYN,RST,ACK SYN -j REJECT

# and all UDP traffic
iptables -A INPUT -s 0/0 -p udp -j REJECT
```

Now, let's apply those rules

```
$ sudo sh /home/sysadm/iptables.sh
```

And verify that the rules are there:

```
$ sudo iptables -L
```

You should see the rules you have created. If you'd rather see numeric output, do the following

```
$ sudo itpables -L -n
```

Now, let's test to make sure that the rules are working:

Check that you can connect to services on localhost:

(if you have not installed Apache, do so now)

```
$ telnet localhost 80
```

You should see something like this:

```
# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

To exit, type 'Ctrl-]', and then 'quit'

Now, ask the members of your group to check connectivity against your web server:

```
sysadm@ext2:~$ telnet ext1 80
```

They should be able to connect.

Now, ask someone from ANOTHER group, to test:

```
sysadm@ext5:~$ telnet ext1 80
```

(If they are able to connect, then you did something wrong. Go back to your file, fix the rules, and run the sh command again).

Now, test the ICMP rate limiting. Ask one of your classmates to do the following against your extX:

```
sysadm@ext2:~$ sudo ping -f ext1
```

What is that "-f"? It stands for "flood", which means that ext2 will try to send as many ICMP echo request packets as possible. Ask your classmate to run that for about 5 seconds, and then stop with 'Ctrl-C'. Then, ask them to check the statistics. There should be a high "packet loss" value, and the number of packets received should not be greater than 3 per second (15 packets total if they ran it for 5 secs)

If all the tests look good, then you should save those rules in order to have Linux re-apply them when it reboots:

```
$ sudo iptables-save > /etc/iptables.rules
```

And now, tell Ubuntu to restore those rules at boot time:

```
$ sudo vi /etc/network/if-pre-up.d/iptablesload

#!/bin/sh
iptables-restore < /etc/iptables.rules
exit 0
```

Make it executable:

```
sudo chmod +x /etc/network/if-pre-up.d/iptablesload
```

Don't forget to use "iptables-save" each time you modify your rules. Otherwise, you will lose your changes next time you reboot.