Advanced Registry Operations Curriculum

Contents

1 Introduction

1.1 Goals

- Learn how to create an initial firewall on Ubuntu/Debian

2 1. Creating an initial firewall

Let's create the first firewall to understand how this is done. We'll create a very simple rule. Let's block access to ping to your box. It's a good example, but we don't want to do this in real life. Ping is too valuable to block.

To do this do the following:

```
$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -i lo -j DROP
```

See if the new rule that is now in place is working:

```
$ ping localhost
```

This should now fail. Press ctrl-c to exit from the ping.

If you wanted to make this rule be permanent you would do:

```
$ sudo iptables-save > /etc/iptables.rules
$ sudo editor /etc/network/interfaces
```

In this file you will see something like:

```
# The primary network interface
auto eth0
iface eth0 inet static
        address 67.218.55.101
        netmask 255.255.255.192
        network 67.218.55.64
        broadcast 67.218.55.127
        gateway 67.218.55.65
        # dns-* options are implemented by the resolvconf package, if installed
        dns-nameservers 67.218.55.67
        dns-search pacnog.bluesky.as
```

At the end of this, on a separate line just after "dns-search..." you should a line that looks like:

```
       pre-up iptables-restore < /etc/iptables.rules
```

Then save and exit from the file (":wq" in vi).

Now each time your machine boots the iptables rule will be applied.

3 2. Removing the initial iptables ping blocking rule

To remove the rule is simple. There are two ways to do this. You can do:

```
$ sudo iptables -D INPUT -p icmp --icmp-type echo-request -i lo -j DROP
```

Now try pinging your local machine:

```
$ ping localhost
```

It should be working again. But, you saved the old rule to /etc/iptables.rules. This means that if you were to reboot or restart your network interface the ping blocking rule would come back. You can do:

```
$ sudo iptables -F
```

to flush all rules, or you can leave things as they are. In either case, run:

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

and you will have a file with no iptables in it that gets loaded next time you reboot.

4 3. Creating an initial, restrictive iptables rule-set

To test this you may wish to do the following:

```
$ su -          [enter in the root password]
# cd
# editor firewall.sh
```

In this file add the following:

```
#!/bin/bash

iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP

iptables -A INPUT -i lo -j ACCEPT

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

iptables -A INPUT -j REJECT
iptables -A FORWARD -j REJECT
```

Now save and exit from the file (":wq" in vi).

Make the file executable:

```
# chmod 755 firewall.sh
```

execute the firewall rules

```
# ./firewall.sh
```

Do some testing. Can you to the services on your box from another machine (ssh, web, ping, anything else?).

If you have problems try to figure out what is blocking the service and add a rule in to iptables to let the packets through.

There are endless possible iptables rules you can add - including dynamic rules to deal with potential DDoS attacks, port scanning on the ports you do open, allowing access from certain addresses or ranges only, etc., etc.

Here are some good web pages with more in-depth iptables rulesets:

- https://help.ubuntu.com/community/IptablesHowTo
- http://www.shanghaiwebhosting.com/ssh-hosting/typical-iptables-firewall-rules-for-a-server-that-hosts-websites
- http://forcespike.altervista.org/articles/setting-firewall-with-iptables.php
- http://blogs.techrepublic.com.com/10things/?p=539
- http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:*Ch14*:*Linux*Firewalls_Using_i
- http://wiki.vpslink.com/HOWTO:*Building*IPTables_rules
- http://www.pizon.org/articles/building-a-linux-firewall-with-iptables.html

You can view your current iptables ruleset by typing:

```
# iptables -L
```

To make the current firewall rules permanent remember you must do:

```
# iptables-save > /etc/iptables.rules
```

Below is a more in-depth description of each rule in our iptables ruleset:

```
# Flush the current iptables ruleset in memory
iptables -F
```

```
# drop all packets on the INPUT chain in the Filter table
iptables -P INPUT DROP

# drop all packets on the FORWARD chain on the Filter tables
iptables -P FORWARD DROP

# accept all packets on our local loopback interface
iptables -A INPUT -i lo -j ACCEPT

# allow us to connect out from our box
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# allow incoming tcp connection on port 22 (ssh)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# allow incomding tcp connections on port 80 (http)
iptables -A INPUT -p tcp --dport 80 -j ACCEPT

# allow incoming tcp connections on port 443 (https)
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# allow incoming udp connections on port 53 (dns)
iptables -A INPUT -p udp --dport 53 -j ACCEPT

# allow incoming tcp connections on port 53 (dns)
iptables -A INPUT -p tcp --dport 53 -j ACCEPT

# allow icmp requests of type 8 (ECHO or ping)
iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

# reject anything else on these Chains that gets to here. Do this explicitly even though it
iptables -A INPUT -j REJECT
iptables -A FORWARD -j REJECT
```

You are now running your server with a firewall that allows you to get out, but which only allows access to your currently running services.