IDS Module 9 Lab

Audrey Long

07/25/2020

Of the many responses available in the event of an incident, one might choose to tighten firewall rules/settings, block access to a domain or certain IP address/range, or perhaps implement white or blacklisting. In this lab we'll look at a popular host-based Linux firewall called IPTABLES and see how it can be used as an effective response to a cyber-attack.

1. First, start with a brief introduction to what IPTABLES are and briefly explain why they might be useful in the face of a network-based attack in terms of detecting and responding to, as well as eliminating connections from known threats.

In general iptables are a command line interface tool which enables users to manage linux firewall rules. The firewall is a device which inspects network traffic and makes decisions based on the inbound and outbound traffic on linux boxes. More specifically iptables administers network based configurations for the kernel to perform actions to inspect, modify, or drop network packets based on written rules.The sniffing of the packets can eliminate threats by dropping malicious packets coming into or leaving the network.

2. What do the ACCEPT, REJECT, DROP and RETURN targets do in an IPTABLES rule, and which command-line option should be used to specify it?

- ACCEPT – Allows a packet.(Accept the incoming/outgoing connection)
  - ex: `sudo iptables -A INPUT -i lo -j ACCEPT`
- DROP – Drops a packet. (Drop the connections)
  - ex: `sudo iptables -A INPUT -j DROP`
- LOG – Logs a packet to Syslog.(Log the connection status for network monitoring, TCP Built/teardown)
  - ex: `sudo iptables -N LOGGING`
- REJECT – Drops a packet sends an appropriate response packet (TCP Reset or an ICMP Port Unreachable message).
  - ex: `sudo iptables -A INPUT -s 15.15.15.51 -j REJECT`
- RETURN- Continues processing a packet within the calling chain.
  - ex: `sudo iptables -A INPUT -p tcp -m tcp --dport 100 -j RETURN`

3. List and describe the 3 default chains provided by IPTABLES.

- Input Chain – Incoming connections which are traversed from Prerouting. Example: External IP trying to establish an SSH connection on your system
- .Output Chain- Packets that are passed or outgoing connections from your system. Example: If you're trying to visit cybersecuritynews.com, user traffic is verified in chain rule to allow or deny the connection.

- Forward Chain – Forwarding connections to specific networks or ports.Example: Port Forwarding.

4. Breakdown each part of the following rules and tell what each segment does:

a. iptables -t filter -A INPUT -s 221.98.146.40 -j DROP

- **Iptables** - calls the iptables program
- **-t, --table** *table*

  This option specifies the packet matching table which the command should operate on. If the kernel is configured with automatic module loading, an attempt will be made to load the appropriate module for that table if it is not already there.

- **filter**:

  This is the default table (if no -t option is passed). It contains the built-in chains **INPUT** (for packets destined to local sockets), **FORWARD** (for packets being routed through the box), and **OUTPUT** (for locally-generated packets).

- **-A, --append** *chain rule-specification*

  Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.

  In this case we are appending the INPUT

  -s, --source [!] *address*[/*mask*]

  Source specification. *Address* can be either a network name, a hostname (please note that specifying any name to be resolved with a remote query such as DNS is a really bad idea), a network IP address (with /mask), or a plain IP address. The *mask* can be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask. Thus, a mask of *24* is equivalent to *255.255.255.0*. A "!" argument before the address

specification inverts the sense of the address. The flag **--src** is an alias for this option.

In this command the source specified is **221.98.146.40.**

- **-j DROP**

-j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule (and -g is not used), then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

In this case we are telling the targ to use the DROP rule.

- these parameters stand for jump which specifies the target of the rule and what action should be performed if the packet is a match

i. What would happen here if we omitted the '-t filter option?

If we omitted the -t filter option we would essentially be using the default table for the build in chains INPUT for packets destined to local sockets.

The following commands will not repeat the defined commands above such as "iptables -A INPUT"

b. iptables -A INPUT **-p tcp --dport ssh -j DROP**

- **-p, --protocol** [!] *protocol*

The protocol of the rule or of the packet to check. The specified protocol can be one of *tcp*, *udp*, *icmp*, or *all*, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted.\

In this command we are specifying the protocol to be tcp.

- **--destination-port**,**--dport** [**!**] *port*[**:***port*]

In this command we are specifying the destination port to be ssh

- **-j DROP**

  -j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule (and -g is not used), then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

  In this case we are telling the target to use the DROP rule to drop the connection.

c. iptables -A INPUT **-p udp --dport ftp -j REJECT**

- **-p, --protocol** [!] *protocol*

The protocol of the rule or of the packet to check. The specified protocol can be one of *tcp*, *udp*, *icmp*, or *all*, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted.\

In this command we are specifying the protocol to be udp.

- **--destination-port**,**--dport** [**!**] *port*[**:***port*]

In this command we are specifying the destination port to be ftp

- **-j REJECT**

  -j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule (and -g is not used), then matching the rule will

have no effect on the packet's fate, but the counters on the rule will be incremented.

In this case we are telling the target to use the REJECT rule to refuse connections.

d. iptables -A INPUT **-p tcp --dport 443 -j ACCEPT**

- **-p, --protocol** [!] *protocol*

The protocol of the rule or of the packet to check. The specified protocol can be one of *tcp*, *udp*, *icmp*, or *all*, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from /etc/protocols is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted.\

In this command we are specifying the protocol to be tcp.

- **--destination-port**,**--dport** [**!**] *port*[**:***port*]

In this command we are specifying the destination port to be 443

- **-j ACCEPT**

-j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule (and -g is not used), then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

In this case we are telling the target to use the ACCEPT rule to accept connections.

e. iptables -A INPUT **-m iprange --src-range 192.168.40.25-192.168.40.35 -j DROP**

- **-m or --match options**, followed by the matching module name; after these, various extra command line options become available, depending on the specific module.
- **ip range**
- This matches on a given arbitrary range of IPv4 addresses
- **[!]***--src-range* **ip-ip**
- Match source IP in the specified range.
- **[!]***--dst-range* **ip-ip**
- Match destination IP in the specified range.
-

In this context we are trying to match the ip source address range from **192.168.40.25-192.168.40.35**

- **-j DROP**

  -j, --jump *target* This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special builtin targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule (and -g is not used), then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.

  In this case we are telling the target to use the DROP rule to drop the connection within the given src address range.

5. Please provide the IPTABLES commands to accomplish each of the following:

a. Drop all incoming TCP traffic destined for port 80

```
sudo iptables -A INPUT -p tcp --dport 80 -j DROP
```

b. Allow incoming traffic coming from the 172.16.40.0/24 subnet on ports 23 and 443

```
sudo iptables -A INPUT -p tcp -s 172.16.40.0/24 -m multiport --dports 80,223 -j ACCEPT
```

c. Allow ping (ICMP echo-request) from MAC source address 00:0D:EC:82:03:09 and allow the (echo-reply) to be returned internally

```
sudo iptables -A INPUT -p icpm --icpm-type echo-request, echo-reply --mac-source 00:0D:EC:82:03:09 -j ACCEPT
```

d. Only accept connections to TCP port 443 if IP is between 192.166.1.102 and 192.166.1.206

```
sudo iptables -A INPUT -p tcp --dport 443 -m iprange --src-range 192.166.1.102-192.166.1.203 -j ACCEPT
```

e. Allow DNS queries (which use UDP) outbound and allow DNS responses inbound on interface eth0

```
sudo iptables -A OUTPUT -p udp -j ACCEPT

sudo iptables -A INPUT -p udp -i eth0 -j ACCEPT
```

Please submit your lab activity as either a pdf or word document through the link on Blackboard. Let me know if you have any questions.

References

https://cybersecuritynews.com/linux-firewall-iptables/
https://www.cyberciti.biz/tips/linux-iptables-9-allow-icmp-ping.html