Lab 2: IPTables

Group A

Fabrizio Demaria, <demaria@kth.se> (900402-T417)
Paolo Forte, <forte@kth.se> (910907-T291)
Samia Khalid, <samiak@kth.se> (910323-T248)

Contributions: Fabrizio Demaria worked with his personal computer to complete the various steps of the lab; Paolo Forte and Samia Khalid were focused on the theoretical aspects and questions. However, we worked together in solving all tasks. Also the report has been written as a group work.

1. Setup

- Output of ping when verifying connectivity

```
root@iptables:~# route add default gw 10.0.0.1
root@iptables:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        *               255.255.0.0     U     0      0        0 eth0
default         10.0.0.1        0.0.0.0         UG    0      0        0 eth0
root@iptables:~# ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_req=1 ttl=63 time=1.91 ms
64 bytes from 192.168.0.2: icmp_req=2 ttl=63 time=1.80 ms
64 bytes from 192.168.0.2: icmp_req=3 ttl=63 time=1.61 ms
^C
--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.613/1.779/1.917/0.125 ms
```

Ping from external host to internal host, through the firewall

2. Nmap enumeration

- How does nmap detect active hosts using:
  - ***Link Layer*** : ARP Requests (IPv4) and Neighbour Discovery (IPv6);
  - ***Network Layer*** : ICMP Echo packets;
  - ***Transport Layer*** : TCP and UDP connections.

- What are the advantages and disadvantages of each type of scanning?
  - ***Link Layer*** : this scan is related to the local subnet since it involves OSI-Layer 2. This is the default discovery method for targets on the local ethernet since it is fast and effective;
  - ***Network Layer*** : this scan can be used to discover hosts in the target network but ICMP packets can be discarded by the host or firewalls;
  - ***Transport Layer*** : this kind of probe can give more detailed information about the applications running on the remote hosts. Port scanning can be performed with this method. Also in this case TCP ping can be blocked, by using stateful firewalls. [1][2]

- Which parameters did you use to locate the server?
  nmap -sP 10.0.0.0/16

- What is the address of the server?
  The IP address of the server is: 10.0.32.42

- How long did it take?
  1335.86 seconds

- How many addresses did you scan?
  65536 IP addresses

```
root@iptables:~# nmap -sP 10.0.0.0/16

Starting Nmap 5.21 ( http://nmap.org ) at 2014-12-16 11:27 CET
Nmap scan report for 10.0.0.1
Host is up (0.0012s latency).
MAC Address: 00:16:3E:3E:00:03 (Xensource)
Nmap scan report for 10.0.0.2
Host is up.
Nmap scan report for 10.0.32.42
Host is up (0.0011s latency).
MAC Address: 00:16:3E:3E:00:10 (Xensource)
Stats: 0:11:10 elapsed; 32771 hosts completed (3 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 6.49% done; ETC: 11:40 (0:01:12 remaining)
Stats: 0:12:51 elapsed; 36867 hosts completed (3 up), 4096 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 26.97% done; ETC: 11:41 (0:01:02 remaining)
Nmap done: 65536 IP addresses (3 hosts up) scanned in 1335.86 seconds
```

3. Nmap scanning

- What command did you use for TCP discovery
  nmap -sT 10.0.32.42
- What command did you use for UDP discovery
  nmap -sU 10.0.32.42
- UDP discovery is much slower than TCP discovery. Why?
  UDP takes longer because it uses timeouts to wait for possible answers. Remote hosts might not send any response when open or filtered port is tested. In case of a closed port, a “ICMP port unreachable” message is sent back but this is not guaranteed, while in the case of a TCP port RST packets are sent back in response to a SYN or connect scan. [3]
- List all open TCP services
  22/tcp open ssh
- List all open UDP services
  All scanned ports were closed.
- What is the difference between Open, Filtered, Unfiltered and Closed ports?
  - ***OPEN*** : an application is accepting TCP and UDP packets. These ports show services available for use on the network;
  - ***FILTERED*** : a firewall is blocking the access to the port and it is not possible to determine whether is open.
  - ***UNFILTERED*** : the port is reachable by nmap and no firewall is blocking the access. This state doesn’t give information wether the port is open or not.
  - ***CLOSED*** : the port is accessible but no running application is listening to it. [4]

4. Nmap service identification

- What operating system does nmap detect?
  nmap guesses Linux 2.6.X | 2.4.X with probability of 96%

- How are the services identified?
  nmap -sV 10.0.32.42

- Are these sane guesses?
  Yes. The OS guess is compliant with the version of OpenSSH identified by the services' scan (*OpenSSH 5.5p1 Debian 4ubuntu4*).

- What other methods can be used to check the operating system and service implementations of an unknown server?
  - Looking at the open ports (e.g. *netbios* is usually open on Windows Servers);
  - Send HTTP GET requests and look for the info on the Server;
  - By means of "fuzzing" in order to obtain a HTTP 500 Internal Server Error that might include such details;
  - Inspect the initial Time To Live (TTL) and the TCP window size of the first packet in a TCP session, since those values are typically known for each operating system;
  - DHCPREQUEST or DHCPDISCOVER packets' options can be inspected to identify the remote OS/Device. [5][6]

5. Basic IPTables

- Explain the order in which the rules are evaluated

  The rules are evaluated following the rules' line numbers, in ascending order.

- Show your IPTables rules where you drop ICMP Echo packets

```
root@iptables:/# iptables -A FORWARD -p icmp --icmp-type echo-request -j DROP
root@iptables:/# iptables -vL
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 6 packets, 504 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       icmp --  any    any     anywhere             anywhere             icmp echo-request

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

- List of ping logs showing everything works correctly

```
root@iptables-A1:~# ping 10.0.0.2 -c 1
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=63 time=2.19 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.193/2.193/2.193/0.000 ms
```

Ping from internal host to external host

```
root@iptables:~# ping 192.168.0.2 -c 1
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Port Unreachable

--- 192.168.0.2 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

Ping from external host to internal host

```
root@iptables:/# ping 192.168.0.2 -c 1
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_req=1 ttl=64 time=1.05 ms

--- 192.168.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.057/1.057/1.057/0.000 ms
root@iptables:/# ping 10.0.0.2 -c 1
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_req=1 ttl=64 time=1.17 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.177/1.177/1.177/0.000 ms
```

Ping from firewall to both internal and external hosts

- Can you ping from the external host to the internal interface on the firewall?

  Yes.

- Why/Why not?

  Because the interfaces of the firewall are related to the INPUT/OUTPUT chains, while we just modified the FORWARD rules that handles the traffic passing through the firewall.

- Can this have any security implications?

  Without implementing correctly the INPUT chain, firewall's interfaces can be exposed to remote attacks that could compromise its functionalities.

6. Logging, blocking and rejecting

- What is the difference between rejecting and dropping blocked traffic?

  If a packet is rejected an error message is sent back as a response to the source host. “Dropping” simply discards the packet without sending any response.

- What are the advantages of blocking and what are the advantages of dropping?

  The main advantage of using “dropping” is that no reachability information is given to the source host. This method requires timeouts slowing down possible attacks. “Rejection” is computationally heavier but avoids eventual retrials by the sender. [7]

- A sample from the system log showing what you have logged

  The log policy was set by adopting the following command:

```
root@iptables:/# iptables -I LOGREJECT 1 -m limit --limit 5/min -j LOG --log-prefix "Ping dropped by A.2: "
```

- List your complete set of rules at this point

```
root@iptables:/# iptables -vL
Chain INPUT (policy ACCEPT 4 packets, 336 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 62 packets, 5208 bytes)
 pkts bytes target     prot opt in     out     source               destination
   38  3192 LOGREJECT  icmp --  any    any     10.0.0.0/16          192.168.0.0/24      icmp echo-request

Chain OUTPUT (policy ACCEPT 45 packets, 4839 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain LOGGING (0 references)
 pkts bytes target     prot opt in     out     source               destination

Chain LOGREJECT (1 references)
 pkts bytes target     prot opt in     out     source               destination
   23  1932 LOG        all  --  any    any     anywhere             anywhere            limit: avg 5/min bu
    9   756 DROP       all  --  any    any     anywhere             anywhere
```

7. Building a firewall

- What kind of security advantage does a setup with a SSH terminal server offer?

  SSH provides a secure remote shell. It encrypts all traffic preventing attacks such as eavesdropping. [8]

- What kind of security disadvantage does a setup with a SSH terminal server introduce?

  SSH is not a complete security solution and problems can arise in case of misconfigurations in port forwarding. A client accessing the Intranet can expose it by port forwarding. [8]

- List the rules you used to setup the firewall as a terminal server for SSH

  iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT

  iptables -I OUTPUT 1 -p tcp --sport 22 -j ACCEPT

```
root@iptables-A2:~# iptables -vL
Chain INPUT (policy ACCEPT 68 packets, 6647 bytes)
 pkts bytes target     prot opt in     out     source               destination
  146 14006 ACCEPT     tcp  --  any    any     10.0.0.0/16          10.0.0.1            tcp dpt:ssh
  122 14957 ACCEPT     all  --  any    any     192.168.0.0/24       anywhere
    9   540 DROP       all  --  any    any     anywhere             anywhere

Chain FORWARD (policy ACCEPT 64 packets, 5376 bytes)
 pkts bytes target     prot opt in     out     source               destination
    3   180 DROP       all  --  any    any     anywhere             anywhere

Chain OUTPUT (policy ACCEPT 47 packets, 5007 bytes)
 pkts bytes target     prot opt in     out     source               destination
  365 53811 ACCEPT     tcp  --  any    any     anywhere             10.0.0.0/16         tcp spt:ssh
   14  1232 ACCEPT     icmp --  any    any     anywhere             anywhere
  116 14337 ACCEPT     all  --  any    any     anywhere             192.168.0.0/24
   81  6804 DROP       all  --  any    any     anywhere             anywhere
```

- How did you verify that the firewall works as intended?

  After having set the complete set of rules (that can be found in the next page), we tested them with SSH, FTP and PING.
  The external host could only connect to the internal host by first establishing a SSH connection to the firewall.
  We established FTP connections and checked the rules' counters after uploading and downloading files to verify the rules' matching.

8. Your rule set

- List your final set of firewall rules

```
root@iptables-A2:~# iptables -vL
Chain INPUT (policy ACCEPT 68 packets, 6647 bytes)
 pkts bytes target     prot opt in     out     source               destination
  146 14006 ACCEPT     tcp  --  any    any     10.0.0.0/16          10.0.0.1            tcp dpt:ssh
  129 15417 ACCEPT     all  --  any    any     192.168.0.0/24       anywhere
   18   900 DROP       all  --  any    any     anywhere             anywhere

Chain FORWARD (policy ACCEPT 64 packets, 5376 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0            udp  --  any    any     192.168.0.0/24       anywhere            multiport dports netbios-ns,netbios-dgm
    0     0            tcp  --  any    any     192.168.0.0/24       anywhere            multiport dports netbios-ssn,microsoft-ds
   64  3663 ACCEPT     tcp  --  any    any     10.0.0.0/16          192.168.0.2         tcp dpt:ftp state NEW,RELATED,ESTABLISHED
   19  1042 ACCEPT     tcp  --  any    any     10.0.0.0/16          192.168.0.2         tcp dpt:ftp-data state RELATED,ESTABLISHED
   43  3832 ACCEPT     tcp  --  any    any     192.168.0.2          10.0.0.0/16         tcp spt:ftp state RELATED,ESTABLISHED
    5   274 ACCEPT     tcp  --  any    any     192.168.0.2          10.0.0.0/16         tcp spt:ftp-data state RELATED,ESTABLISHED
   58  5375 ACCEPT     tcp  --  any    any     192.168.0.0/24       10.0.0.0/16         state NEW,RELATED,ESTABLISHED
   30  4463 ACCEPT     tcp  --  any    any     10.0.0.0/16          192.168.0.0/24      state RELATED,ESTABLISHED
   19  1192 DROP       all  --  any    any     anywhere             anywhere

Chain OUTPUT (policy ACCEPT 47 packets, 5007 bytes)
 pkts bytes target     prot opt in     out     source               destination
  375 54331 ACCEPT     tcp  --  any    any     anywhere             10.0.0.0/16         tcp spt:ssh
  133 15717 ACCEPT     all  --  any    any     anywhere             192.168.0.0/24
   86  7336 DROP       all  --  any    any     anywhere             anywhere

Chain LOGGING (0 references)
 pkts bytes target     prot opt in     out     source               destination

Chain LOGREJECT (0 references)
 pkts bytes target     prot opt in     out     source               destination
```

9. References

[1] *Host Discovery, Chapter 15. Nmap Reference Guide*. Last accessed December 17, 2014, http://nmap.org/book/man-host-discovery.html

[2] Rajesh Deodhar, (Last modified December 1, 2010). *Advanced Nmap: Scanning Techniques Continued.* Last accessed December 17, 2014, http://www.opensourceforu.com/2010/12/advanced-nmap-scanning-techniques-continued

[3] *Port Scanning Techniques, Chapter 15. Nmap Reference Guide*. Last accessed December 17, 2014, http://nmap.org/book/man-port-scanning-techniques.html

[4] *Port Scanning Basics, Chapter 15. Nmap Reference Guide*. Last accessed December 17, 2014, http://nmap.org/book/man-port-scanning-basics.html

[5] *Chatter on the Wire: How excessive network traffic gives away too much!*. Last accessed December 17, 2014, http://chatteronthewire.org/

[6] Erik Hjelmvik, (Last modified November 5, 2011). *Passive OS Fingerprinting.* Last accessed December 17, 2014, http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting

[7] Peter Benie. *Drop versus Reject.* Last accessed December 17, 2014, http://www.chiark.greenend.org.uk/~peterb/network/drop-vs-reject

[8] Sean Boran. *All About SSH - Part I/II.* Last accessed December 17, 2014, http://www.boran.com/security/sp/ssh-part1.html#Disadvantages