

1 ЧАСТЬ. Honeypot, Nmap

Настраиваем сеть. Меняем тип подключения на сетях мост на машине Server и Hacker.

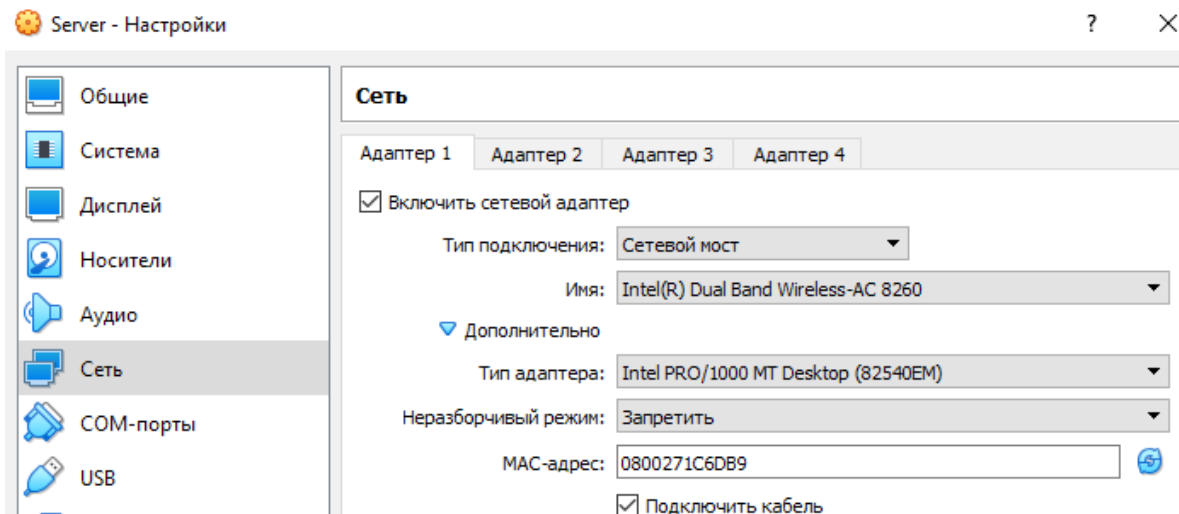


Рис. 1 Сервер

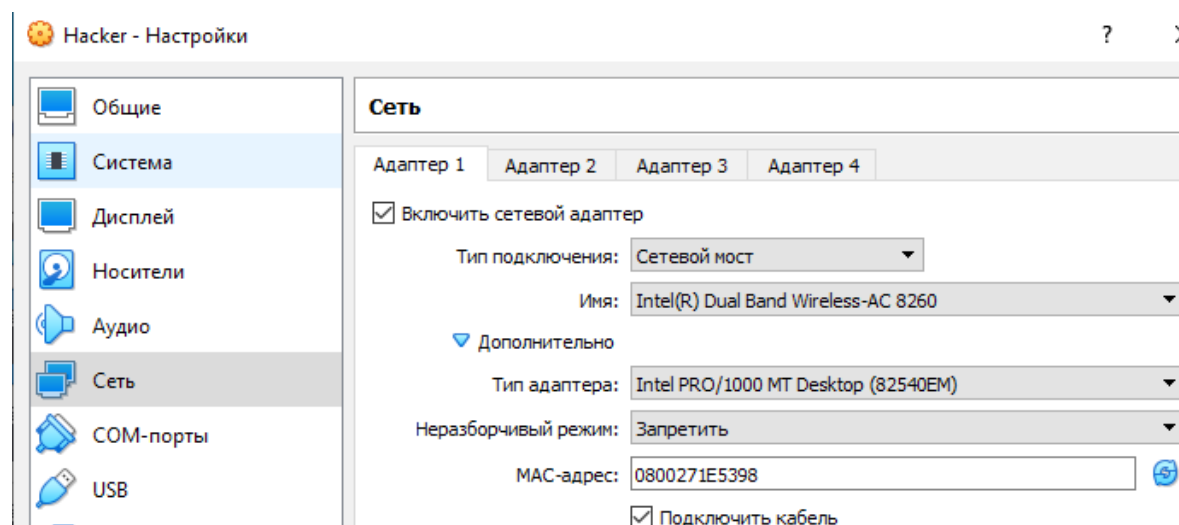


Рис. 2 Хакер

Далее устанавливаем на Hacker Nmap, а Server apache2.

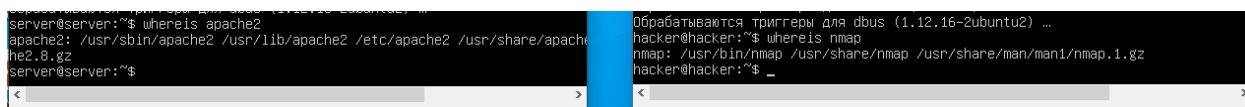
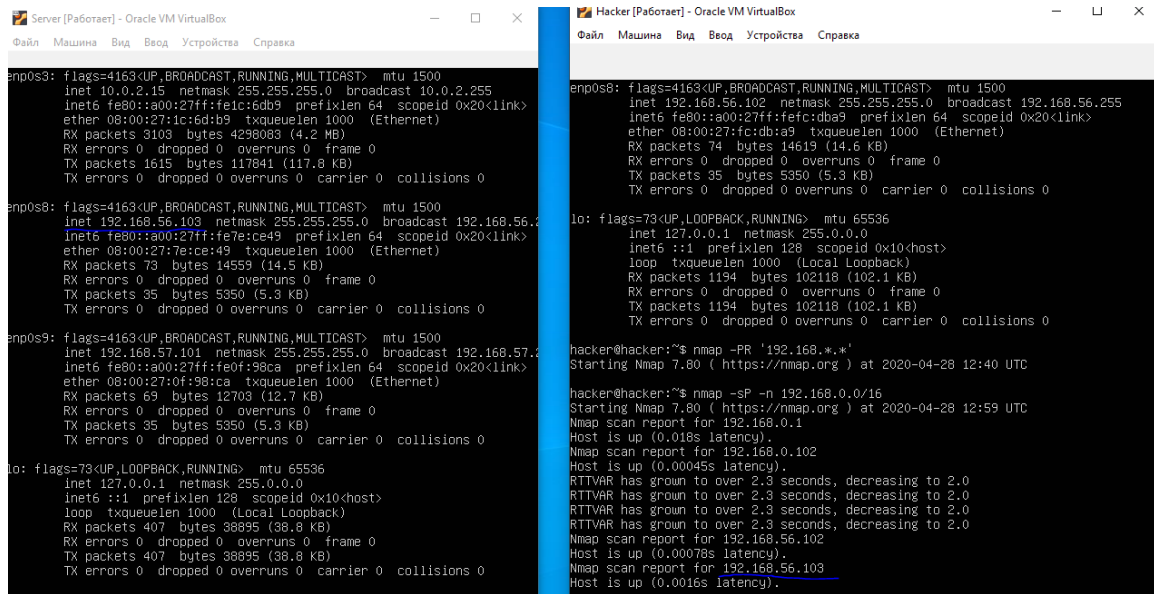


Рис. 3 Nmap и apache2

С машины Hacker через Nmap начинаем сканировать машину Server.



```
Server [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe1c:6db9  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:1c:6d:b9  txqueuelen 1000  (Ethernet)
    RX packets 3103  bytes 4298083 (4.2 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1615  bytes 117841 (117.8 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.103  netmask 255.255.255.0  broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe7e:ce49  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:7e:ce:49  txqueuelen 1000  (Ethernet)
    RX packets 73  bytes 14559 (14.5 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 35  bytes 5350 (5.3 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe0f:98ca  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:0f:98:ca  txqueuelen 1000  (Ethernet)
    RX packets 69  bytes 12703 (12.7 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 35  bytes 5350 (5.3 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 407  bytes 38895 (38.8 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 407  bytes 38895 (38.8 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

Hacker [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

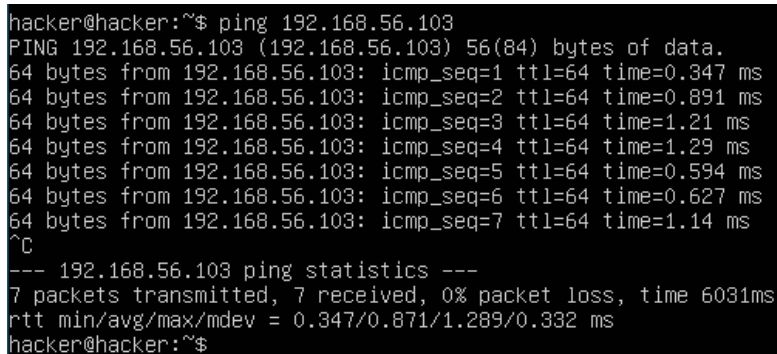
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe1c:6db9  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:1c:6d:b9  txqueuelen 1000  (Ethernet)
    RX packets 74  bytes 14619 (14.6 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 35  bytes 5350 (5.3 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 1194  bytes 102118 (102.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1194  bytes 102118 (102.1 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

hacker@hacker:~$ nmap -PR '192.168.*.*'
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 12:40 UTC

hacker@hacker:~$ nmap -sP -n 192.168.0.0/16
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 12:59 UTC
Nmap scan report for 192.168.0.1
Host is up (0.018s latency).
Nmap scan report for 192.168.0.102
Host is up (0.00046s latency).
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
Nmap scan report for 192.168.56.102
Host is up (0.00078s latency).
Nmap scan report for 192.168.56.103
Host is up (0.0016s latency).
```

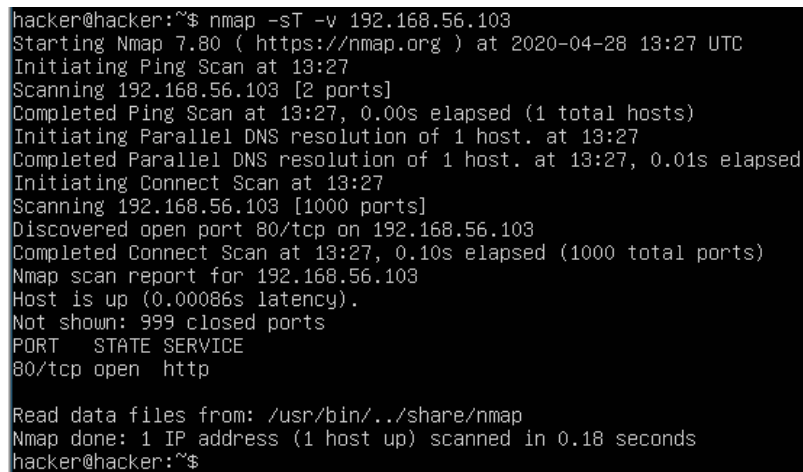
Рис. 4 Определение IP-адрес сервера



```
hacker@hacker:~$ ping 192.168.56.103
PING 192.168.56.103 (192.168.56.103) 56(84) bytes of data:
64 bytes from 192.168.56.103: icmp_seq=1 ttl=64 time=0.347 ms
64 bytes from 192.168.56.103: icmp_seq=2 ttl=64 time=0.891 ms
64 bytes from 192.168.56.103: icmp_seq=3 ttl=64 time=1.21 ms
64 bytes from 192.168.56.103: icmp_seq=4 ttl=64 time=1.29 ms
64 bytes from 192.168.56.103: icmp_seq=5 ttl=64 time=0.594 ms
64 bytes from 192.168.56.103: icmp_seq=6 ttl=64 time=0.627 ms
64 bytes from 192.168.56.103: icmp_seq=7 ttl=64 time=1.14 ms
^C
--- 192.168.56.103 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6031ms
rtt min/avg/max/mdev = 0.347/0.871/1.289/0.332 ms
hacker@hacker:~$
```

Рис. 5 ping

Через метод TCP Connec командой nmap -sT 192.168.56.103



```
hacker@hacker:~$ nmap -sT -v 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 13:27 UTC
Initiating Ping Scan at 13:27
Scanning 192.168.56.103 [2 ports]
Completed Ping Scan at 13:27, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:27
Completed Parallel DNS resolution of 1 host. at 13:27, 0.01s elapsed
Initiating Connect Scan at 13:27
Scanning 192.168.56.103 [1000 ports]
Discovered open port 80/tcp on 192.168.56.103
Completed Connect Scan at 13:27, 0.10s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00086s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
hacker@hacker:~$
```

Рис. 6 nmap -sT 192.168.56.103

Через метод TCP SYN командой nmap -sS 192.168.56.103

```
hacker@hacker:~$ sudo nmap -sS -v 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 13:29 UTC
Initiating ARP Ping Scan at 13:29
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 13:29, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:29
Completed Parallel DNS resolution of 1 host. at 13:29, 0.01s elapsed
Initiating SYN Stealth Scan at 13:29
Scanning 192.168.56.103 [1000 ports]
Discovered open port 80/tcp on 192.168.56.103
Completed SYN Stealth Scan at 13:29, 0.07s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00018s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.032KB)
hacker@hacker:~$
```

Рис. 7 nmap -sS 192.168.56.103

Далее через метод FIN командой nmap -sF 192.168.56.103

```
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:8E:18:63 (Cadmus Computer Systems)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.77 seconds
Raw packets sent: 1086 (47.768KB) | Rcvd: 1086 (43.432KB)
sit@ubuntusit:~$ sudo nmap -v -sF 192.168.1.213

Starting Nmap 6.40 ( http://nmap.org ) at 2015-10-11 06:17 EDT
Initiating ARP Ping Scan at 06:17
Scanning 192.168.1.213 [1 port]
Completed ARP Ping Scan at 06:17, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 06:17
Completed Parallel DNS resolution of 1 host. at 06:17, 0.00s elapsed
Initiating FIN Scan at 06:17
Scanning 192.168.1.213 [1000 ports]
Completed FIN Scan at 06:17, 6.80s elapsed (1000 total ports)
Nmap scan report for 192.168.1.213
Host is up (0.00046s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    openifiltered http
MAC Address: 08:00:27:8E:18:63 (Cadmus Computer Systems)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 7.07 seconds
Raw packets sent: 1161 (46.428KB) | Rcvd: 1158 (46.308KB)
sit@ubuntusit:~$
```

Рис. 8 nmap -sF 192.168.56.103

Через метод Xmas Tree командой nmap -sX 192.168.56.103

```
hacker@hacker:~$ sudo nmap -sX -v 192.168.56.103
[sudo] password for hacker:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 13:58 UTC
Initiating ARP Ping Scan at 13:58
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 13:58, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:58
Completed Parallel DNS resolution of 1 host. at 13:58, 0.01s elapsed
Initiating XMAS Scan at 13:58
Scanning 192.168.56.103 [1000 ports]
Completed XMAS Scan at 13:58, 1.21s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00044s latency).
Not shown: 999 closed ports
PORT      STATE      SERVICE
80/tcp    open|filtered http
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.34 seconds
           Raw packets sent: 1002 (40.068KB) | Rcvd: 1000 (39.988KB)
hacker@hacker:~$ _
```

Рис. 9 nmap -sX 192.168.56.103

Через метод NULL командой nmap -sN 192.168.56.103

```
hacker@hacker:~$ sudo nmap -sN -v 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 13:59 UTC
Initiating ARP Ping Scan at 13:59
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 13:59, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:59
Completed Parallel DNS resolution of 1 host. at 13:59, 0.01s elapsed
Initiating NULL Scan at 13:59
Scanning 192.168.56.103 [1000 ports]
Completed NULL Scan at 13:59, 1.21s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00024s latency).
Not shown: 999 closed ports
PORT      STATE      SERVICE
80/tcp    open|filtered http
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.34 seconds
           Raw packets sent: 1002 (40.068KB) | Rcvd: 1000 (39.988KB)
hacker@hacker:~$
```

Рис. 10 nmap -sN 192.168.56.103

Через метод сканирования IP командой nmap -sO 192.168.56.103

```
Initiating IPProto Scan at 14:00
Scanning 192.168.56.103 [256 ports]
Increasing send delay for 192.168.56.103 from 0 to 5 due to max_successful_tryno
Increasing send delay for 192.168.56.103 from 5 to 10 due to max_successful_tryno
Increasing send delay for 192.168.56.103 from 10 to 20 due to 11 out of 12 dropped
increase.
Discovered open port 1/ip on 192.168.56.103
Increasing send delay for 192.168.56.103 from 20 to 40 due to 11 out of 12 dropped
increase.
IPProto Scan Timing: About 47.04% done; ETC: 14:01 (0:00:35 remaining)
Increasing send delay for 192.168.56.103 from 40 to 80 due to 11 out of 13 dropped
increase.
Increasing send delay for 192.168.56.103 from 80 to 160 due to 11 out of 11 dropped
t increase.
Discovered open port 6/ip on 192.168.56.103
Increasing send delay for 192.168.56.103 from 160 to 320 due to max_successful_tr
Increasing send delay for 192.168.56.103 from 320 to 640 due to max_successful_tr
Increasing send delay for 192.168.56.103 from 640 to 1000 due to max_successful_t
IPProto Scan Timing: About 66.33% done; ETC: 14:02 (0:00:34 remaining)
Discovered open port 17/ip on 192.168.56.103
Completed IPProto Scan at 14:04, 258.04s elapsed (256 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00067s latency).
Not shown: 250 closed protocols
PROTOCOL STATE SERVICE
1 open icmp
2 open|filtered igmp
6 open tcp
17 open udp
103 open|filtered pim
136 open|filtered udplite
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 258.11 seconds
Raw packets sent: 1232 (25.076KB) | Rcvd: 256 (12.280KB)
hacker@hacker:~$
```

Рис. 11 nmap -sO 192.168.56.103

Через метод АСК-сканирования командой nmap -sA 192.168.56.103

```
hacker@hacker:~$ sudo nmap -sA -v 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 14:08 UTC
Initiating ARP Ping Scan at 14:08
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 14:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:08
Completed Parallel DNS resolution of 1 host. at 14:08, 0.01s elapsed
Initiating ACK Scan at 14:08
Scanning 192.168.56.103 [1000 ports]
Completed ACK Scan at 14:08, 0.06s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00025s latency).
All 1000 scanned ports on 192.168.56.103 are unfiltered
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
Raw packets sent: 1001 (40.028KB) | Rcvd: 1001 (40.028KB)
hacker@hacker:~$ _
```

Рис. 12 nmap -sA 192.168.56.103

Через метод TCP Window командой nmap -sW 192.168.56.103

```
hacker@hacker:~$ sudo nmap -v -sW 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 14:11 UTC
Initiating ARP Ping Scan at 14:11
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 14:11, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:11
Completed Parallel DNS resolution of 1 host. at 14:11, 0.01s elapsed
Initiating Window Scan at 14:11
Scanning 192.168.56.103 [1000 ports]
Completed Window Scan at 14:11, 0.07s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00017s latency).
All 1000 scanned ports on 192.168.56.103 are closed
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
Raw packets sent: 1001 (40.028KB) | Rcvd: 1001 (40.028KB)
hacker@hacker:~$
```

Рис. 13 nmap -sW 192.168.56.103

Через метод RPC-сканирование командой nmap -sR 192.168.56.103

```
hacker@hacker:~$ sudo nmap -v -sR 192.168.56.103
WARNING: -sR is now an alias for -sV and activates version detection as well as R
Warning: The -sR option is deprecated. Please use -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 14:13 UTC
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 14:13
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 14:13, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:13
Completed Parallel DNS resolution of 1 host. at 14:13, 0.01s elapsed
Initiating SYN Stealth Scan at 14:13
Scanning 192.168.56.103 [1000 ports]
Discovered open port 80/tcp on 192.168.56.103
Completed SYN Stealth Scan at 14:13, 0.07s elapsed (1000 total ports)
Initiating Service scan at 14:13
Scanning 1 service on 192.168.56.103
Completed Service scan at 14:13, 6.02s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.56.103.
Initiating NSE at 14:13
Completed NSE at 14:13, 0.02s elapsed
Initiating NSE at 14:13
Completed NSE at 14:13, 0.01s elapsed
Nmap scan report for 192.168.56.103
Host is up (0.00023s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp open  http  Apache httpd 2.4.41 ((Ubuntu))
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.
Nmap done: 1 IP address (1 host up) scanned in 6.97 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.032KB)
hacker@hacker:~$
```

Рис. 14 nmap -sR 192.168.56.103

Через метод сканирования ОС командой nmap -O 192.168.56.103

```

hacker@hacker:~$ sudo nmap -O 192.168.56.103
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-28 14:25 UTC
Nmap scan report for 192.168.56.103
Host is up (0.00059s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org)
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=4/28%OT=80%CT=1%CU=39929%PV=Y%DS=1%DC=D%G=Y%M=080027%T
OS:M=5EA83CCF%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=106%TI=Z%CI=Z%II-I
OS:%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS:5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS:=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS:%A=S+F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=0%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%
OS:S=A%A=Z%F=R%O=0%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)U1(
OS:R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 12.87 seconds
hacker@hacker:~$

```

Рис. 15 nmap -O 192.168.56.103

После сканирования устанавливаем на Server Honeypot - ресурс, представляющий собой приманку для злоумышленников.

```

-rw-r--r-- 1 server server
server@server:~$ ls
Honeyd
server@server:~$

```

Рис. 16 Ресурс

В файле /etc/honeypot/honeyd.conf изменяем его содержание для нашей работы.

```

#route entry 10.0.0.1
#route 10.0.0.1 link 10.2.0.0/24
#route 10.0.0.1 add net 10.3.0.0/16 10.3.0.1 latency 8ms bandwidth 10Mbps
#route 10.3.0.1 link 10.3.0.0/24
#route 10.3.0.1 add net 10.3.1.0/24 10.3.1.1 latency 7ms loss 0.5
#route 10.3.1.1 link 10.3.1.0/24

create default
set default default tcp action filtered
set default default udp action filtered
set default default icmp action filtered

# Example of a simple host template and its binding
create windows
set windows personality "Microsoft Windows XP"
set windows uptime 1728650
set windows maxfds 35
add windows tcp port 80 "scripts/web.sh"
add windows tcp port 22 "scripts/test.sh"
add windows tcp port 23 "scripts/router-telnet.pl"
add windows udp port 53 open
set windows ethernet "dell"
set windows default tcp action closed

#create router
#set router personality "Cisco 2514 router (IOS 12.1)"
#set router default tcp action closed
#add router tcp port 22 "scripts/test.sh"
#add router tcp port 23 "scripts/router-telnet.pl"

bind 192.168.56.120 windows
#bind 10.3.1.1 router
#bind 10.3.1.12 template

```

Рис. 17 Замена

Устанавливаем и запускаем farpd.

```

server@server:~$ sudo farpd -d
arpd[1450]: listening on enp0s3: arp and not ether src 08:00:27:1c:6d:b9
arpd[1450]: arpd_lookup: no entry for 192.168.0.101
arpd[1450]: arpd_send: who-has 192.168.0.101 tell 192.168.0.104
arpd[1450]: arpd_send: who-has 192.168.0.101 tell 192.168.0.104
arpd[1450]: arpd_rcv_cb: 192.168.0.101 at 08:00:27:1e:53:98
arpd[1450]: arpd_rcv_cb: 192.168.0.101 at 08:00:27:1e:53:98
arpd[1450]: arpd_lookup: no entry for 192.168.0.100
arpd[1450]: arpd_send: who-has 192.168.0.100 tell 192.168.0.104
arpd[1450]: arpd_send: who-has 192.168.0.100 tell 192.168.0.104
arpd[1450]: arpd_rcv_cb: 192.168.0.100 at e4:b3:18:75:97:5c
arpd[1450]: arpd_rcv_cb: 192.168.0.100 at e4:b3:18:75:97:5c
arpd[1450]: arpd_rcv_cb: 192.168.0.101 is allocated
arpd[1450]: arpd_rcv_cb: 192.168.0.100 is allocated
^Z

```

Рис. 18 farpd

Запускаем honeyd с файлом honeyd.conf, который мы переделали.

```

server@server:~$ sudo honeyd -d -f /etc/honeypod/config.sample
[sudo] password for server:
Honeyd V1.6d Copyright (c) 2002-2007 Niels Provos
honeyd[1478]: started with -d -f /etc/honeypod/config.sample
honeyd[1478]: fopen(/usr/share/honeyd/xprobe2.conf)
server@server:~$

```

Рис. 19 honeyd

Ответы на вопросы

Что такое статический и динамический IP-адреса? В чём разница?

IP-адрес называют статическим (постоянным, неизменяемым), если он назначается пользователем в настройках устройства, либо если назначается автоматически при подключении устройства к сети и не может быть присвоен другому устройству.

IP-адрес называют динамическим (непостоянным, изменяемым), если он назначается автоматически при подключении устройства к сети и используется в течение ограниченного промежутка времени, указанного в сервисе назначавшего IP-адрес (DHCP).

В чём заключается метод сканирование протоколов IP?

Метод заключается в том, что хосту передаются IP пакеты без заголовков для каждого протокола сканируемого хоста. Если получено сообщение, говорящее о недоступности протокола, то этот протокол не поддерживается хостом. В противном случае — поддерживается.

На какие пакеты большинство ОС должны ответить флагом RST?

На пакеты FIN, Xmas Tree и NULL-сканирования. Однако это не касается Windows-систем. Поэтому благодаря этим типам сканирования можно определить семейство операционных систем.

Назначение, цели, описание Honeypot.

Основная задача Honeypot — подвергнуться атаке или несанкционированному сканированию с целью изучения стратегии и методов сканирования и определения перечня средств, необходимых для предотвращения будущих атак.

Суть работы Honeypot заключается в создании ловушек — образов систем, которые извне воспринимаются как полноценные машины с установленными на них операционными системами, а, следовательно, поддающиеся сканированию.

Использование Honeypot имеет смысл, так как если на сервере установлена хорошая система защиты, долгое время можно не замечать постоянных попыток сканирования — Honeypot укажет на их наличие. Также Honeypot позволяет узнать информацию о методах и средствах, используемых злоумышленниками.

Какие цели может преследовать злоумышленник, взламывая сервера?

Целью могут быть ресурсы сервера, которые в своих целях будет использовать злоумышленник. Также целью может быть конфиденциальная информация.

Какое наказание предусмотрено в РФ за взлом?

Есть целый ряд статей УК РФ, которые предусматривают различные наказания. Среди них статьи 183, 272, 273, 274 УК РФ. Например, статья 272 УК РФ за неправомерный доступ к охраняемой законом компьютерной информации, если это деяние повлекло уничтожение, блокирование, модификацию либо копирование компьютерной информации предусматривает наказание с виде штрафа в размере до двухсот тысяч рублей или в размере заработной платы или иного дохода, осужденного за период до восемнадцати месяцев, либо исправительных работ на срок до одного года, либо ограничением свободы на срок до двух лет, либо принудительными работами на срок до двух лет, либо лишением свободы на тот же срок.

Как выявлять Honeypot?

Грамотно настроенный Honeypot практически невозможно распознать. Однако Honeypot всё же имеет ожидаемые характеристики или особенности поведения. Например, Honeypot имитирует работу web-сервера. Всякий раз, когда злоумышленник соединяется с Honeypot, Web-сервер отвечает, посылая общее сообщение об ошибках, используя стандартный HTML. Это - точный ответ, который ожидается от любого Web-сервера. Однако имеется орфографическая ошибка в одной из команд HTML, такой как проверка правописания длины слова - "legnht". Эта орфографическая ошибка является особенностью для данного Honeypot.

Или, если в результате сканирования nmap, отображается слишком много открытых портов. Современный сервер, оснащенный файерволом, никогда бы не допустил такого. Также может насторожить слишком устаревшая операционная система.

Что такое DHCP?

DHCP - сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

Для чего используется RPC-сканирование?

RPC-сканирование используется для определения программы, обслуживающей порт и её версии, и заключается в «затоплении» NULL-пакетами оболочки SunRPC открытых TCP или UDP портов хоста.

Перечислите основные методы сканирования Nmap.

Ping-сканирование, TCP Connect, TCP-SYN, сканирования FIN, Xmas Tree и NULL, сканирование протоколов IP, ACK-сканирование, TCP Window, RPC-сканирование, сканирование ОС.

2 ЧАСТЬ. Iptables, WEB APPLICATION FIREWALL

Смотрим список текущих правил iptables таблицы filter.

```
server@server:~$ sudo iptables -l
[sudo] password for server:
iptables v1.8.4 (legacy): unknown option "-l"
Try `iptables -h' or 'iptables --help' for more information.
server@server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

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

Рис. 1 Список текущих правил

Смотрим список, который отражает команды, необходимые для активации правил и политик.

```
server@server:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
server@server:~$
```

Рис. 2 Команды, необходимые для активации правил и политик

Теперь командой `sudo iptables -A INPUT -i lo -j ACCEPT` вносим локальный интерфейс. Потом блокируем весь исходящий трафик, кроме портов для SSH и веб-сервера. Для этого в цепочку ACCEPT добавляем два порта (порт SSH 22 и порт http 80), что разрешит трафик на эти порты.

```
server@server:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
server@server:~$ sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
server@server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere            tcp dpt:ssh
ACCEPT     tcp  --  anywhere              anywhere            tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

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

Рис. 3 sudo iptables

Удаляем ненужное правило.

```

server@server:~$ sudo iptables -D INPUT -p tcp -m tcp --dport 22 -j ACCEPT
server@server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  anywhere               anywhere
ACCEPT     tcp  --  anywhere               anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
server@server:~$ _

```

Рис. 4 Удаление

Добавляем ещё одно правило, которое позволит устанавливать исходящие соединения. И блокируем всё остальное и разрешаем все исходящие соединения.

```

server@server:~$ sudo iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
server@server:~$ sudo iptables -A INPUT -j ACCEPT
server@server:~$ sudo iptables -F INPUT
server@server:~$ sudo iptables -P INPUT DROP
server@server:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source                destination
ACCEPT     all  --  anywhere               anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere               anywhere
ACCEPT     tcp  --  anywhere               anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
server@server:~$ _

```

Рис. 5 Добавление и блокировка

Добавляем ещё несколько правил для блокировки наиболее распространенных атак. Для начала нужно заблокировать нулевые пакеты. Следующее правило отражает атаки syn-flood. Далее защищаем сервер от разведывательных пакетов XMAS.

```

server@server:~$ sudo iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
server@server:~$ sudo iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
server@server:~$ sudo iptables -L
Chain INPUT (policy DROP)
target    prot opt source                destination            state
ACCEPT    all  --  anywhere               anywhere               state RELATED,ESTABLISHED
ACCEPT    all  --  anywhere               anywhere
ACCEPT    tcp  --  anywhere               anywhere               tcp dpt:http
DROP      tcp  --  anywhere               anywhere               tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
DROP      tcp  --  anywhere               anywhere               tcp flags:!FIN,SYN,RST,ACK/SYN state N
NEW
DROP      tcp  --  anywhere               anywhere               tcp flags:FIN,SYN,RST,PSH,ACK,URG/FIN,
SYN,RST,PSH,ACK,URG

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
server@server:~$ _

```

Рис. 6 Дополнительное

С машины Hacker проводим XMAS сканирование.

```

hacker@hacker:~$ sudo nmap -sX -v 192.168.56.103
[sudo] password for hacker:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-29 17:20 UTC
Initiating ARP Ping Scan at 17:20
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 17:20, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:20
Completed Parallel DNS resolution of 1 host. at 17:20, 0.01s elapsed
Initiating XMAS Scan at 17:20
Scanning 192.168.56.103 [1000 ports]
Completed XMAS Scan at 17:21, 21.16s elapsed (1000 total ports)
Nmap scan report for 192.168.56.103
Host is up (0.00044s latency).
All 1000 scanned ports on 192.168.56.103 are open|filtered
MAC Address: 08:00:27:7E:CE:49 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 21.31 seconds
Raw packets sent: 2001 (80.028KB) | Rcvd: 1 (28B)
hacker@hacker:~$

```

Рис. 6 Сканирование

На машину Server устанавливаем mod_security.

```

server@server:~$ sudo apt-get install libapache2-mod-security2
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет libapache2-mod-security2 самой новой версии (2.9.3-1).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакет
новлено.
server@server:~$

```

Рис. 7 mod_security

Находим новый лог-файл.

```

server@server:/var/log/apache2$ ls -la
total 20
drwxr-x---  2 root adm    4096 апр 29 19:30 .
drwxrwxr-x 11 root syslog 4096 апр 29 17:27 ..
-rw-r----- 1 root adm    1011 апр 28 14:13 access.log
-rw-r----- 1 root adm    4846 апр 29 18:05 error.log
-rw-r--r--  1 root root      0 апр 29 19:30 modsec_audit.log
-rw-r----- 1 root adm      0 апр 28 12:08 other_vhosts_access.log
server@server:/var/log/apache2$ _

```

Рис. 8 Новый лог-файл

Переименовываем файл.

```

server@server:/etc/modsecurity$ ls -la
total 76
drwxr-xr-x  3 root root  4096 апр 29 17:35 .
drwxr-xr-x 100 root root  4096 апр 29 17:35 ..
drwxr-xr-x  2 root root  4096 апр 29 17:35 crs
-rw-r--r--  1 root root 8452 дек 10 2018 modsecurity.conf-recommended
-rw-r--r--  1 root root 53146 дек  4 2018 unicode.mapping
server@server:/etc/modsecurity$ sudo mv modsecurity.conf-recommended modsecurity.conf
server@server:/etc/modsecurity$ ls -la
total 76
drwxr-xr-x  3 root root  4096 апр 29 19:32 .
drwxr-xr-x 100 root root  4096 апр 29 17:35 ..
drwxr-xr-x  2 root root  4096 апр 29 17:35 crs
-rw-r--r--  1 root root 8452 дек 10 2018 modsecurity.conf
-rw-r--r--  1 root root 53146 дек  4 2018 unicode.mapping
server@server:/etc/modsecurity$ _

```

Рис. 9 Переименование

Редактируем файл modsecurity.conf.

```

# -- Rule engine initialization -----
#
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On

# -- Request body handling -----
#
# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess Off_

# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#

```

Рис. 10 Редактура

Теперь надо отредактировать файл mod-security.conf.

```

GNU nano 4.8                                mod-security.conf
<IfModule security2_module>

Include "/usr/share/modsecurity-crs/*.conf"
Include "/usr/share/modsecurity-crs/activated_rules/*.conf"

<IfModule>_

```

Рис. 11 mod-security.conf

Создаём символические ссылки в каталоге `activated_rules`, чтобы активировать правила.

```

server@server:/usr/share/modsecurity-crs/activated_rules$ sudo ln -s /usr/share/modsecurity-crs/base
rules/modsecurity_crs_30_http_policy.conf
server@server:/usr/share/modsecurity-crs/activated_rules$ sudo ln -s /usr/share/modsecurity-crs/base
rules/modsecurity_crs_49_generic_attacks.conf
server@server:/usr/share/modsecurity-crs/activated_rules$ ls
modsecurity_crs_30_http_policy.conf  modsecurity_crs_49_generic_attacks.conf
server@server:/usr/share/modsecurity-crs/activated_rules$ ls -l
total 8
lrwxrwxrwx 1 root root 73 апр 29 20:33 modsecurity_crs_30_http_policy.conf -> /usr/share/modsecurity
-crs/base_rules/modsecurity_crs_30_http_policy.conf
lrwxrwxrwx 1 root root 77 апр 29 20:34 modsecurity_crs_49_generic_attacks.conf -> /usr/share/modsecu
rity-crs/base_rules/modsecurity_crs_49_generic_attacks.conf
server@server:/usr/share/modsecurity-crs/activated_rules$

```

Рис. 12 Символические ссылки

Ответы на вопросы

Что такое межсетевой экран?

Межсетевой экран, сетевой экран, файервол, брандмауэр — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

Для чего используется межсетевой экран?

Межсетевой экран используется для защиты компьютерных сетей или отдельных узлов от несанкционированного доступа.

Принцип работы Netfilter.

Когда сетевые пакеты попадают в сетевой интерфейс, они после ряда проверок ядром проходят последовательность так называемых цепочек. Пакет обязательно проходит через цепочку `PREROUTING`, после чего определяется, кому он, собственно, был адресован. Если пакет не адресован локальной системе (в нашем случае серверу), он попадает в цепочка `FORWARD`, а иначе — в цепочку `INPUT`, после прохождения которой отдается локальным демонам или процессам. После этого при

необходимости формируется ответ, который направляется в цепочку OUTPUT. После цепочек OUTPUT или FORWARD пакет в очередной раз встречается с правилами маршрутизации и направляется в цепочку POSTROUTING. В результате прохождения пакетом цепочек фильтрации несколько раз, проверка его принадлежности определенным критериям осуществляется несколько раз.

Таблицы межсетевого экрана Netfilter. Для чего они используются?

raw - используется для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепочках PREROUTING и OUTPUT

mangle — содержит правила модификации IP-пакетов.

nat - предназначена для подмены адреса отправителя или получателя. Данную таблицу проходят только первые пакеты из потока - трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT, и POSTROUTING.

filter — основная таблица, используется по умолчанию если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD, и OUTPUT.

Что такое правила межсетевого экрана?

Правила определяют функционирование межсетевого экрана. Правило состоит из критерия, действия и счётчика. *Критерий* - это условие, под которое должны подпадать параметры пакета или текущее соединение, чтобы сработало действие. Это может быть IP-адрес источника, порт получателя, диапазон портов и т.д. *Действие* - операция, которую нужно проделать с пакетом или соединением в случае выполнения условий критерия. Например, отбросить, принять и т.д. *Счётчик* - сущность, которая считает сколько пакетов было подвержено действию правила и на основании этого, показывает их объём в байтах.

Как создавать правила для межсетевого экрана утилитой Iptables?

- `iptables -t nat -A PREROUTING -i eth0 -j DNAT --to-destination 192.168.57.102`

Данное правило определяет первоначальную обработку всех пакетов, приходящих на адаптер eth0:

- -t определяет подключаемую таблицу, в данном случае — nat — для подмены адреса отправителя или получателя
- -A — выбор цепочки
- -i — входящий интерфейс
- -j — действие с пакетами, удовлетворяющими условию — в данном случае DNAT — подмена адреса получателя
- -to-destination — выбор адреса, на который перенаправляются пакеты

Как сохранить правила для последующей автозагрузки?

Самый простой способ — загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. Во время инсталляции пакет уточнит, нужно ли сохранить текущие правила для дальнейшей автоматической загрузки, если текущие правила были протестированы и соответствуют всем требованиям, их можно сохранить.

Что такое Web Application Firewall?

WAF (Web Application Firewall) - это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.

Как настроить правила в WAF mod_security?

По умолчанию mod_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/. Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого надо определённым образом отредактировать файл mod-security.conf, а именно, внести следующие параметры: *Include "/usr/share/modsecurity-crs/*.conf"* и *Include "/usr/share/modsecurity-crs/activated_rules/*.conf"*.

Правила доступны в каталогах: /usr/share/modsecurity-crs/base_rules ;
/usr/share/modsecurity-crs/optional_rules ;
/usr/share/modsecurity-crs/experimental_rules. Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated_rules.

Можно добавить несколько своих правил, чтобы они вступили в исполнение нужно перезапустить Apache командой *sudo service apache2 reload*.