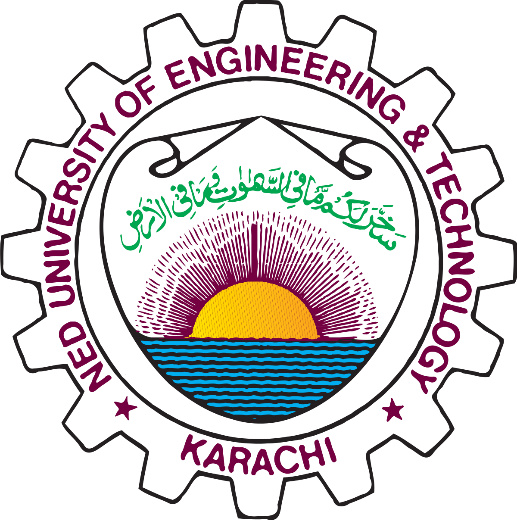
**CT-541 – NETWORK SECURITY**MS-IS 004 2019/20– Evening Fall 2019

**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](#_Toc45976774)

[STEPS OF ICMP REDIRECT ATTACK: 4](#_Toc45976775)

[ICMP REDIRECT ATTACK RESULT / OUTPUT: 7](#_Toc45976776)

[ICMP REDIRECT ATTACK MITIGATION / SECURITY DEFENCE MECHANISM: 8](#_Toc45976777)

[TCP SYN FLOOD WHEN SYN COOKIE SET TO 1: 10](#_Toc45976778)

[STEPS FOR TCP SYN FLOOD: 12](#_Toc45976779)

[TCP SYN FLOOD ATTACK RESULT / OUTPUT WHEN SYN COOKIE SET TO 0: 13](#_Toc45976780)

[TCP SYN FLOOD SECURITY MECHANISMS: 14](#_Toc45976781)

[TCP RST ATTACK ON TELNET AND SSH CONNECTIONS: 15](#_Toc45976782)

[TCP RESET ON TELNET 15](#_Toc45976783)

[TCP RESET ON SSH: 17](#_Toc45976784)

[PROTECT AGAINST TCP RESET AND SYN FLOOD ATTACKS: 18](#_Toc45976785)

[Protecting against a TCP reset attack using the RST bit 18](#_Toc45976786)

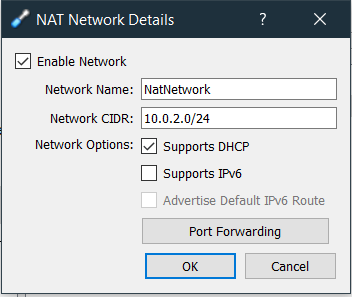
[Best Practice - Protect Against TCP SYN Flooding Attacks with TCP Accept Policies 18](#_Toc45976787)

[Acknowledgements: 19](#_Toc45976788)

**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:

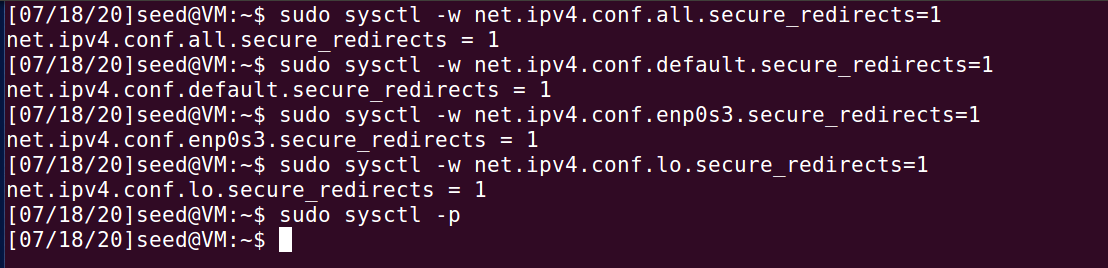
# STEPS OF ICMP REDIRECT ATTACK:

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

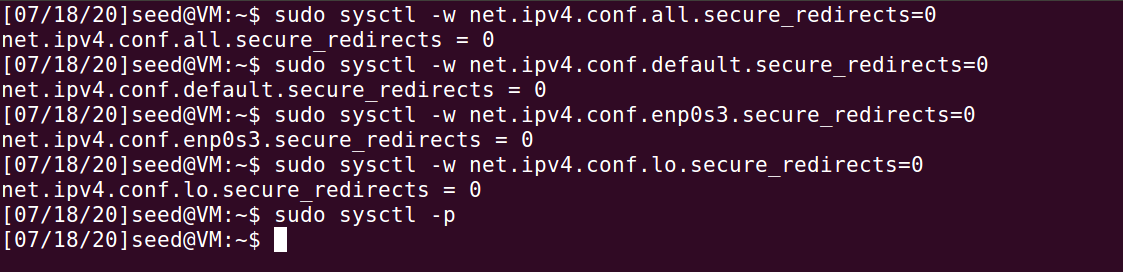
net.ipv4.conf.all.secure\_redirects = 1

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



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

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



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:

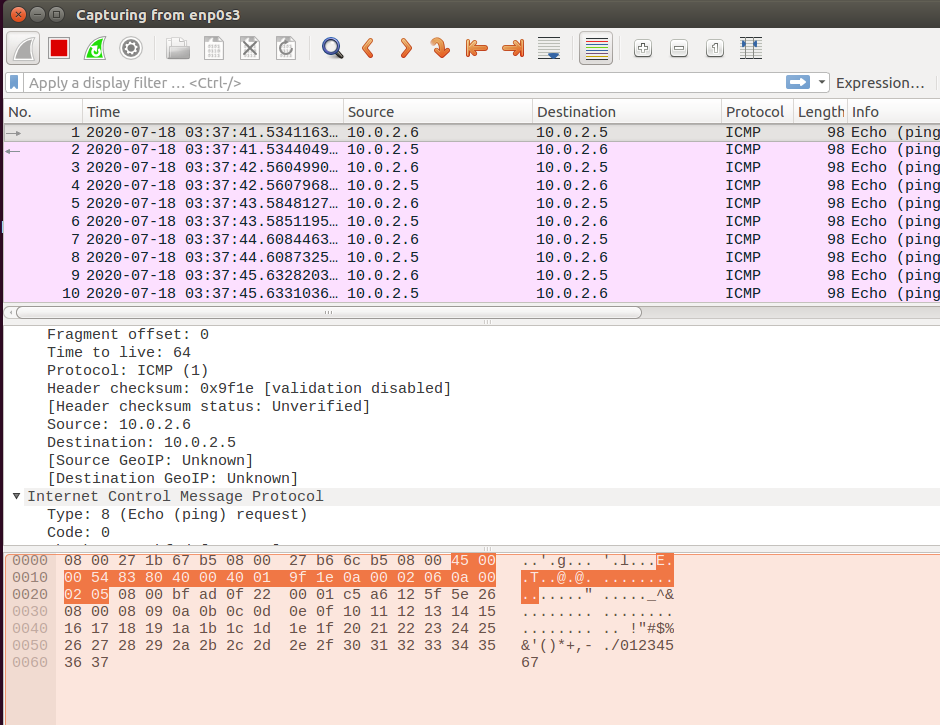
Sudo netwox 86 --device "enp0s8" --filter "src host VICTIMIP" --gw VICTIM\_REDIRECTED\_GATEWAY\_ATTACKERS\_IP --code 1 --src-ip ACTUAL\_GATEWAY\_OF\_VICTIM

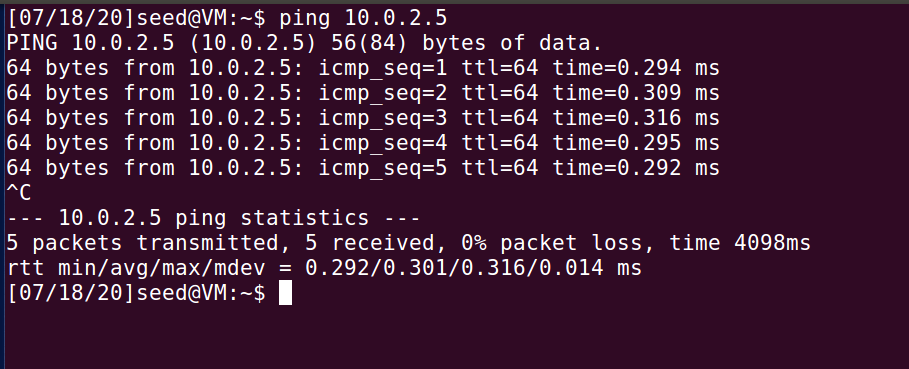
Sudo netwox 86 --device "enp0s8" --filter "src host 192.168.56.102" --gw 192.168.56.101 --code 1 --src-ip 192.168.56.1

**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**

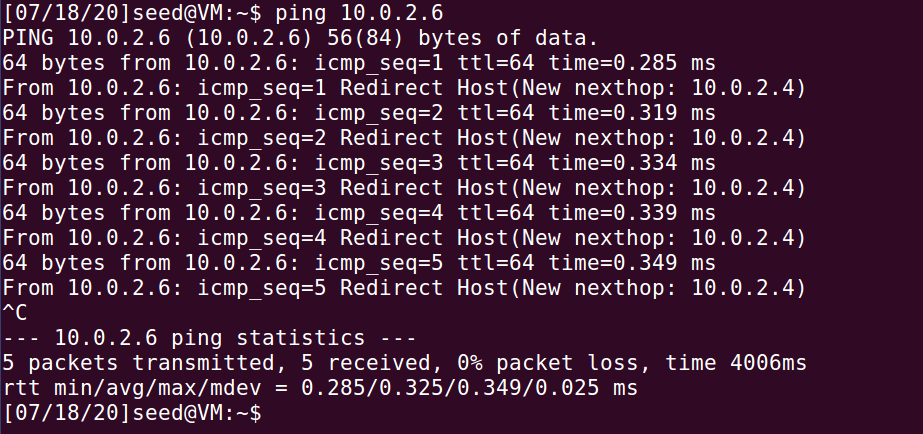
**Before Attack:**Wireshark And Ping Before the Attack, Observing from VM3(10.0.2.6) Source and Victim is VM2(10.0.2.5) Destination:



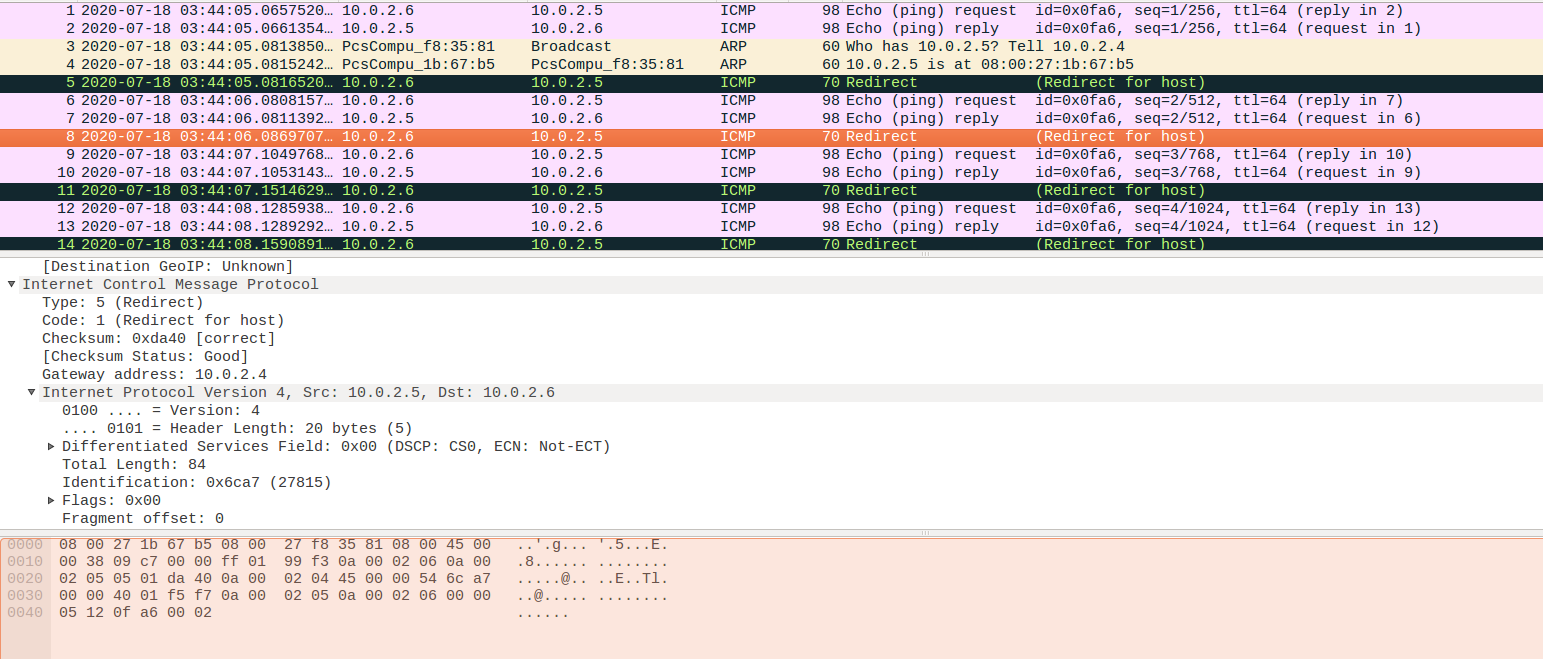


# ICMP REDIRECT ATTACK RESULT / OUTPUT:

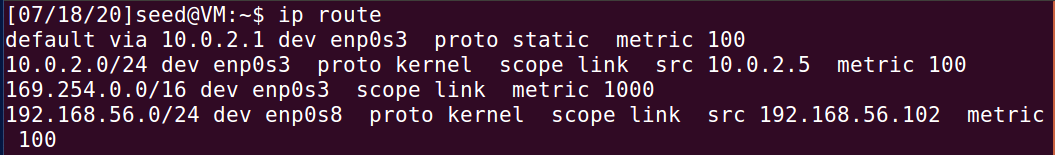
After Attack, **We Ping VM3 from VM2, Now Note The Redirect Host(New Next Hop)**:



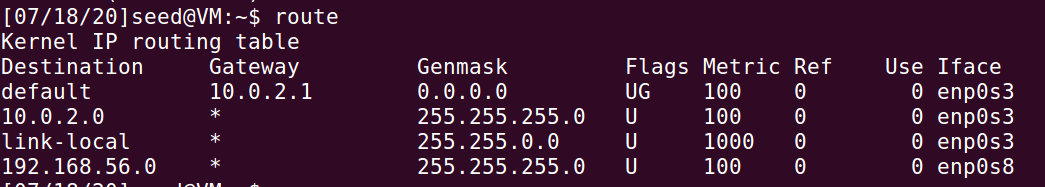
Also from VM3, we can see the ICMP Redirect and changed Gateway:



# ICMP REDIRECT ATTACK MITIGATION / SECURITY DEFENCE MECHANISM:



Route



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 SYS 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.

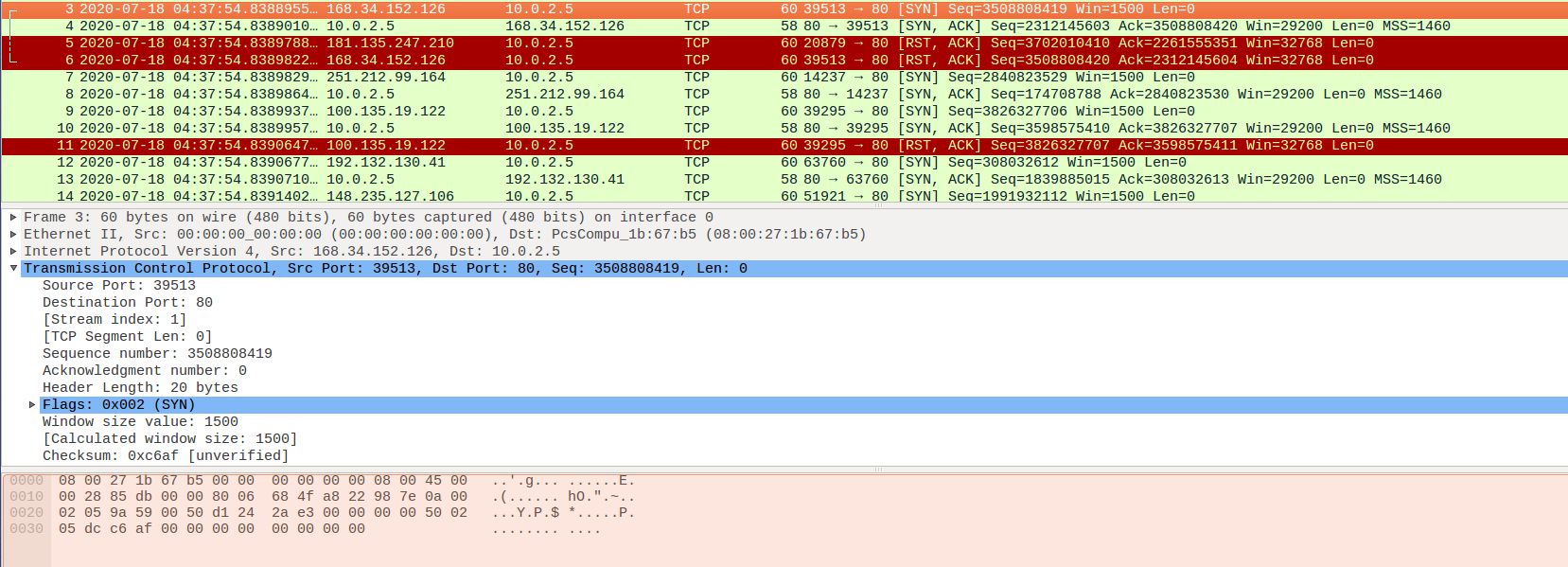
If the value is anything less, reboot the server.

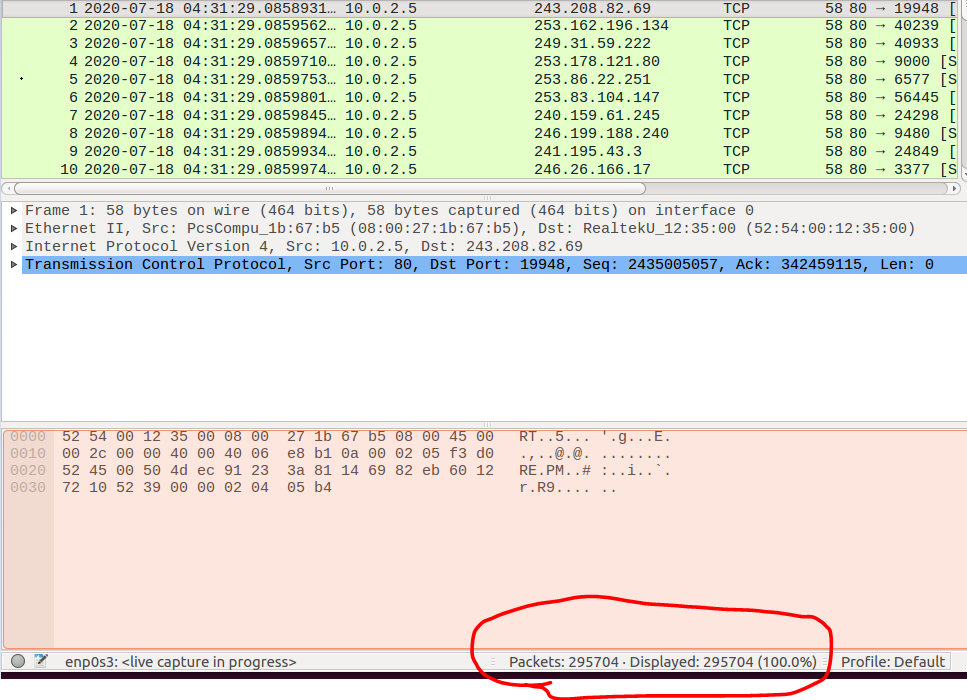
At this point, your Linux server should be better protected against SYN attacks and IP address spoofing.

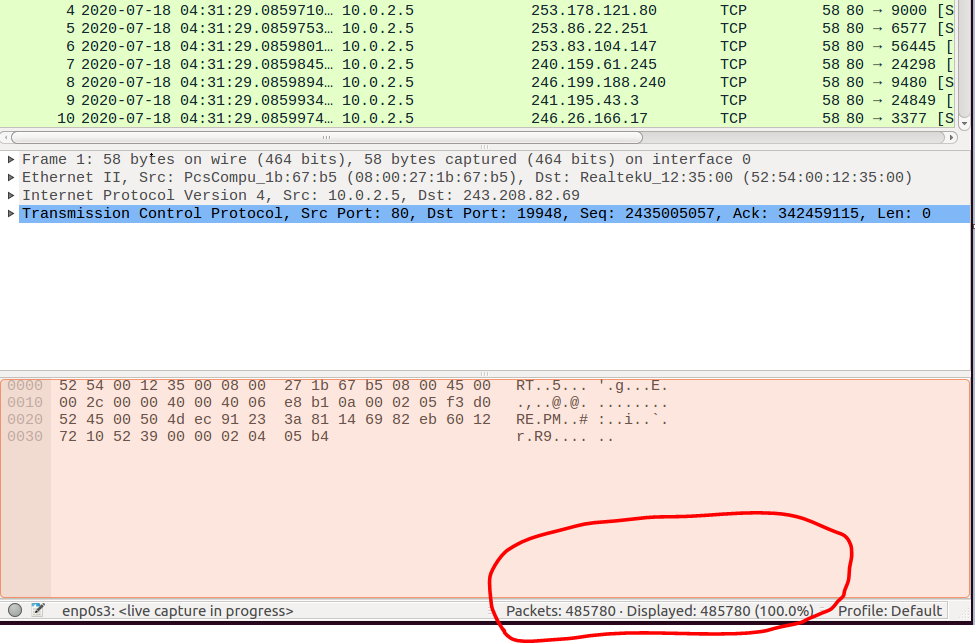
# TCP SYN FLOOD WHEN SYN COOKIE SET TO 1:

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.**





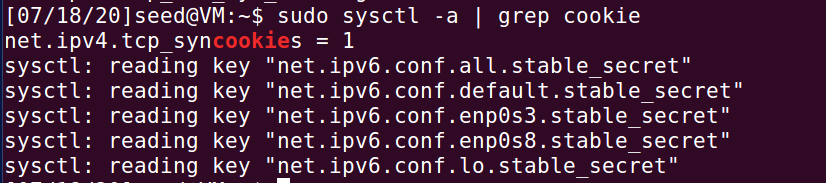


With SYN COOKIE mechanism off -> they remain half-open until the size of queue described.

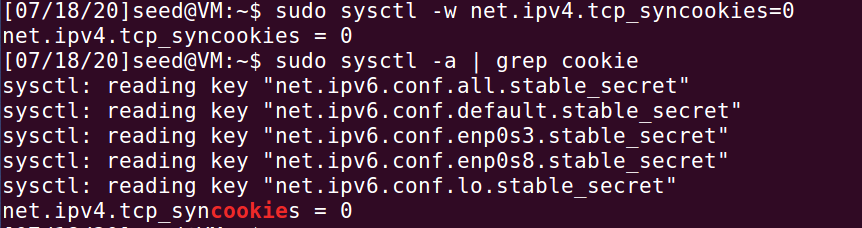
# STEPS FOR TCP SYN FLOOD:

**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:**



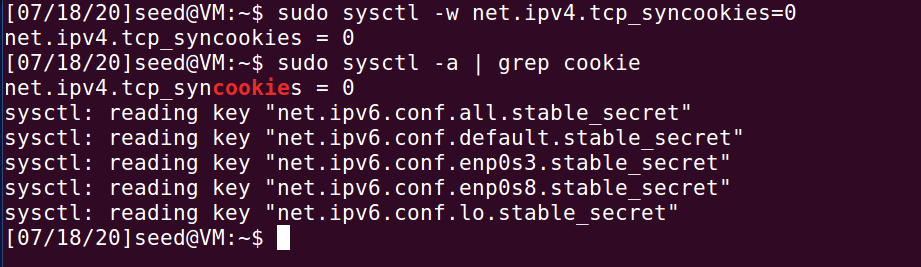
Victim, VM2: **SET SYN COOKIE TO 0**



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

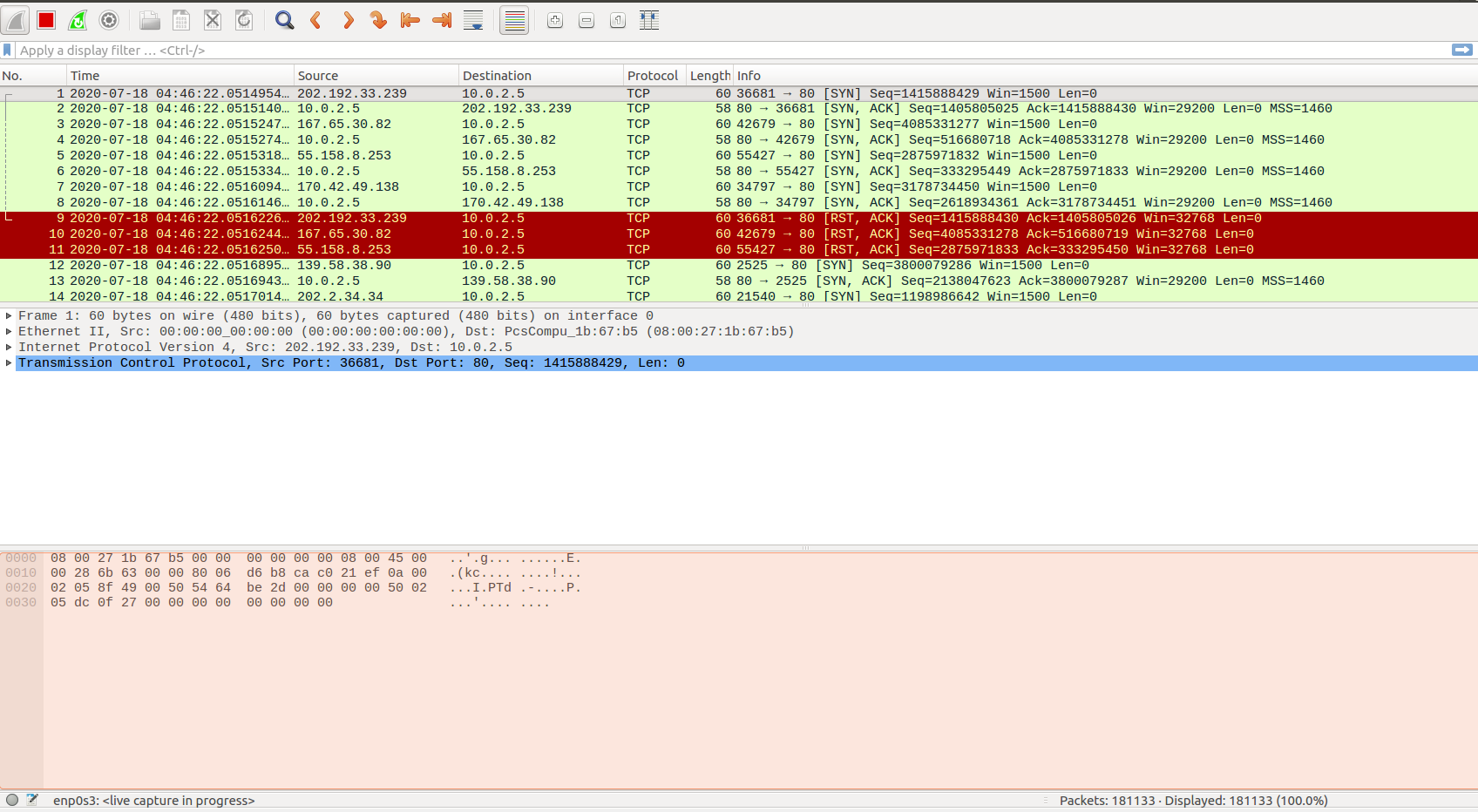
sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 53 (port 53 is the port used by DNS)

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

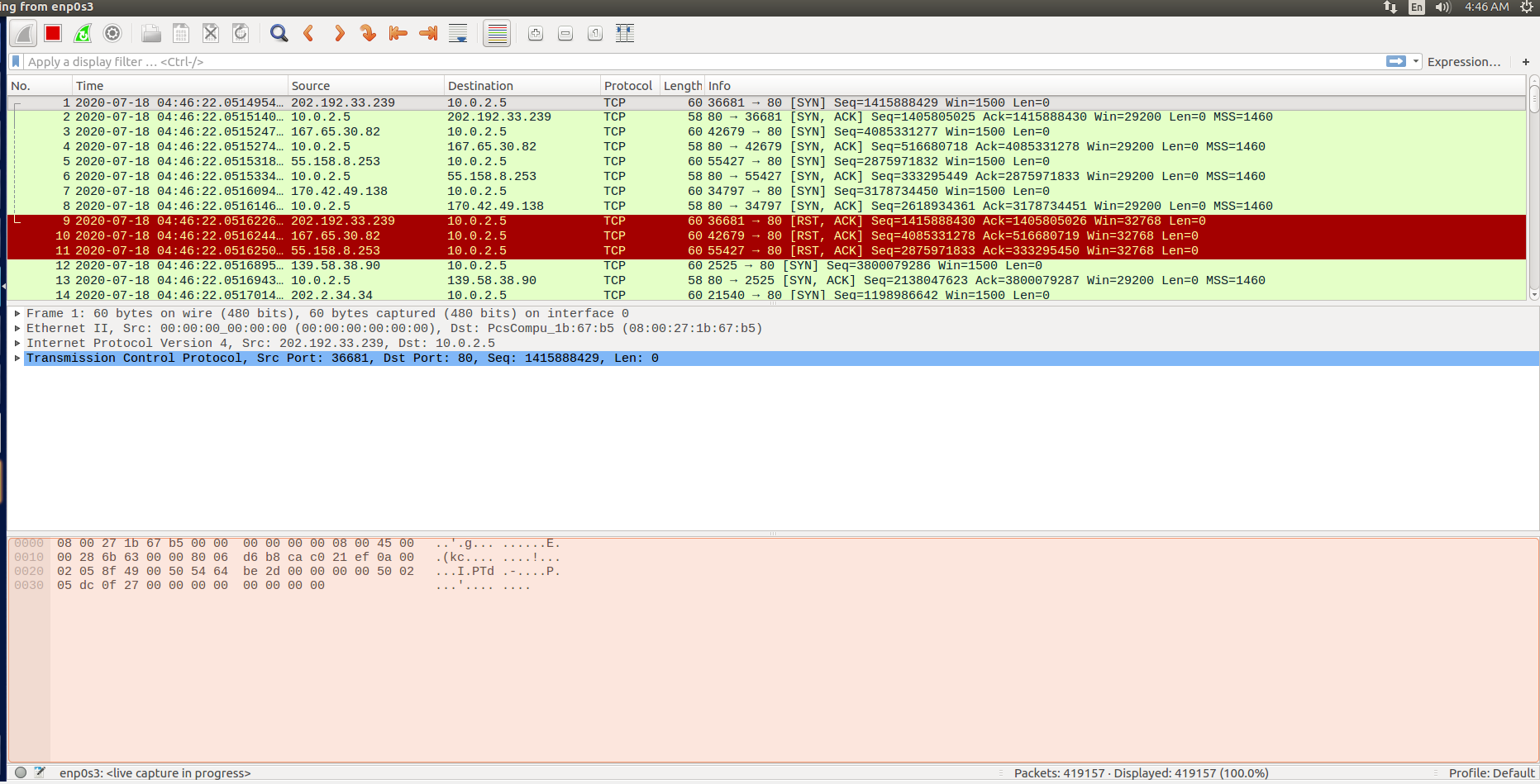


sudo netwox 76 --dst-ip 10.0.2.5 --dst-port 80 (port 80 is the port at which server listens for incoming connections)

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



Packet count : 181133



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:

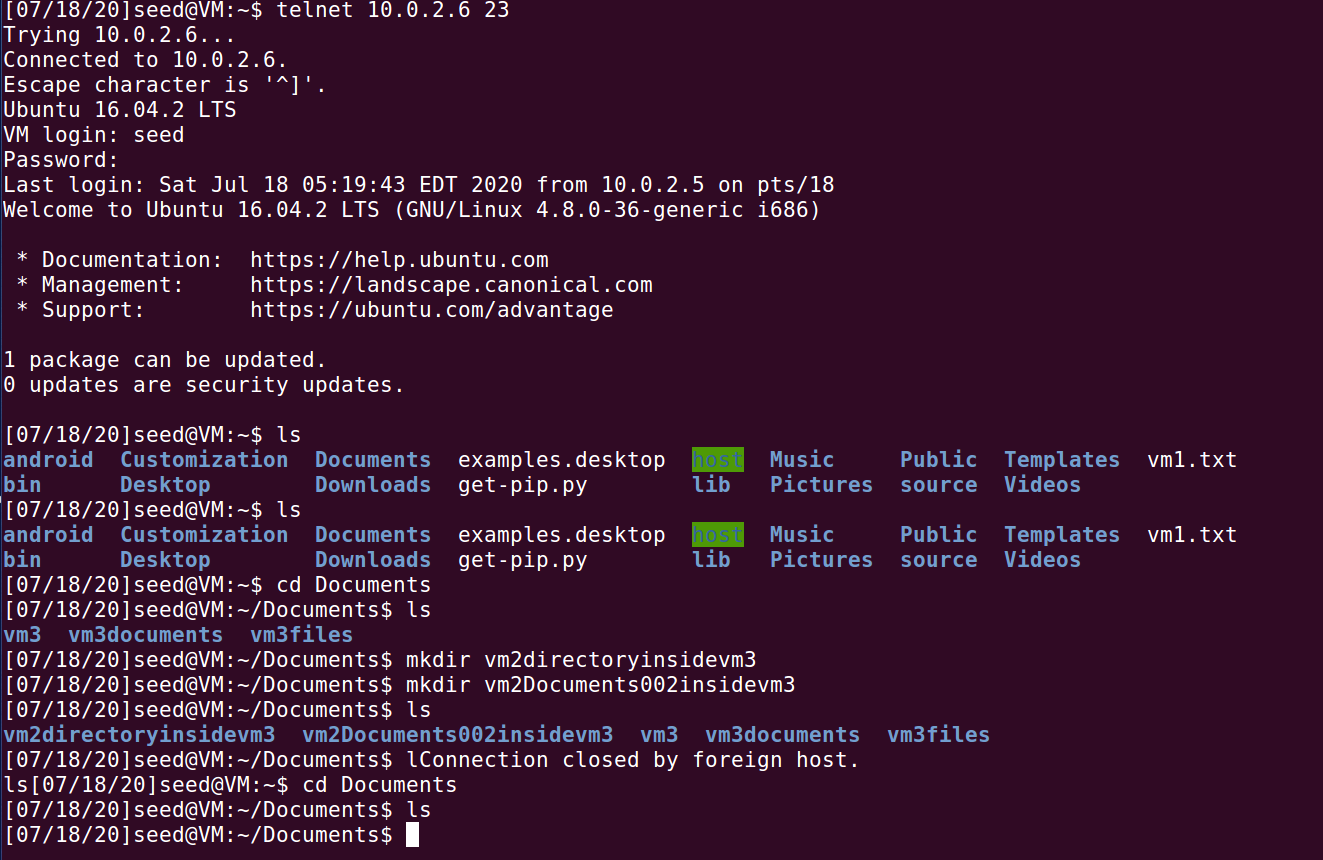
**Attacker Virtual Machine 1: 10.0.2.4**

**Victim Virtual Machine 2 [HOST A]: 10.0.2.5**

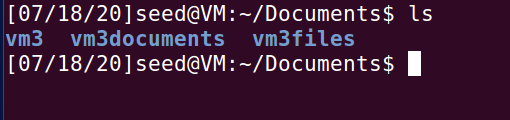
**Victim Virtual Machine 3 [HOST B]: 10.0.2.6**

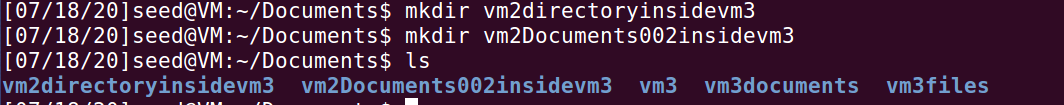
# TCP RESET ON TELNET

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



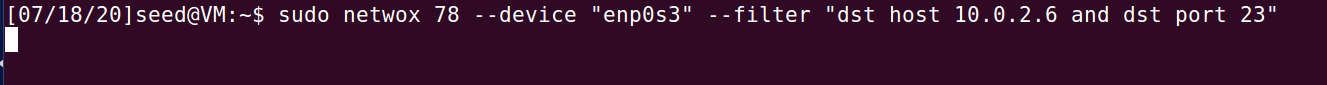
**Do some activity through the telnet connection from vm2 to vm3:**

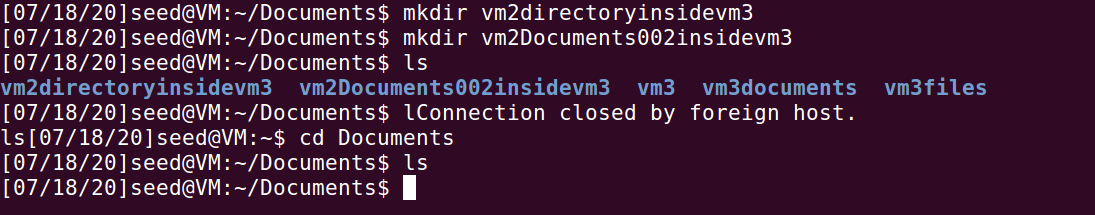




Now keep it at this stage , and now from VM1 Attacker virtual machine with IP = 10.0.2.4 we run the TCP RST Attack:

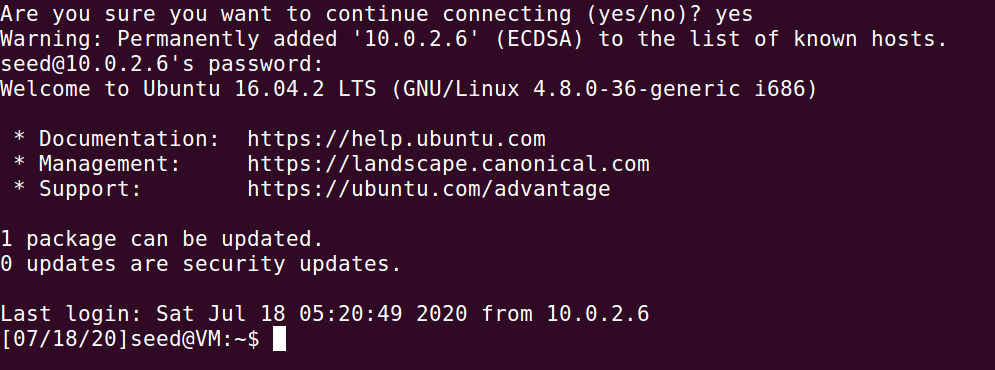
netwox 78 --device "enp0s3" --filter "dst host 10.0.2.6 and dst port 23"





# TCP RESET ON SSH:



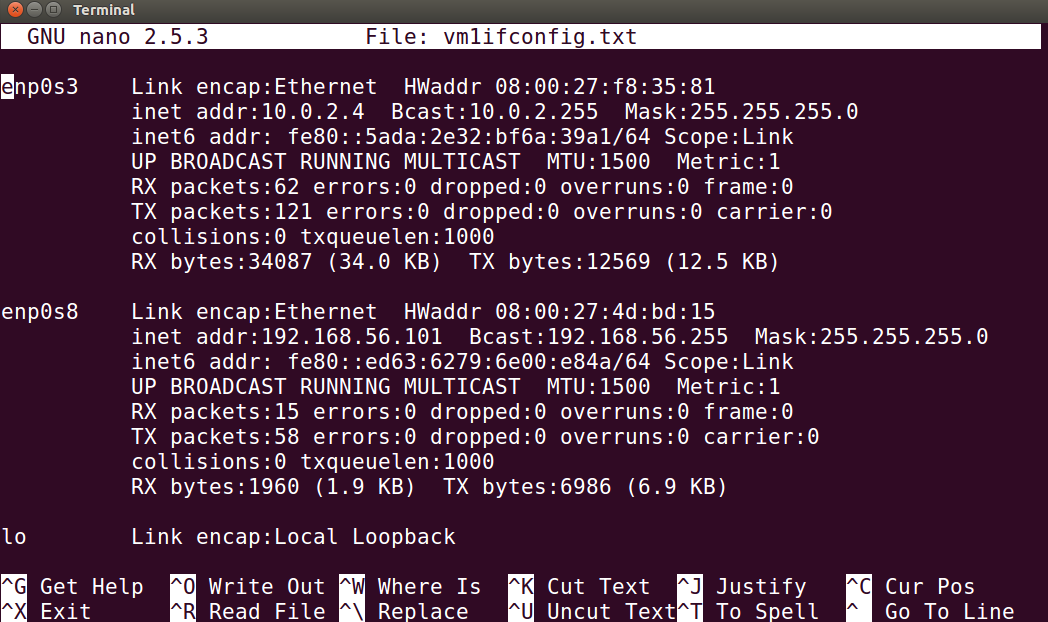


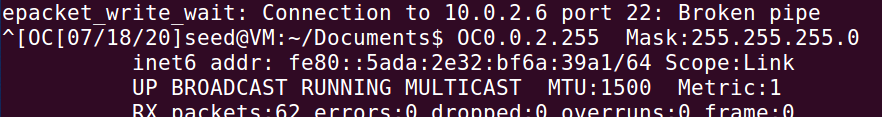
Sudo netwox 78 --device "enp0s3" --filter "dst host 10.0.2.6 and dst port 22"

SSH PORT 22

TELNET PORT 23

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







# 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/junose15.1/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>