ECE - 567

PROJECT REPORT

Part 2

APRIL 9, 2020

SUBMITTED TO:

Professor Issa Traore

SUBMITTED BY:

SANNATH REDDY VEMULA - V00949217

SOMAYEH ROSHANDEL - V00942553

ANJALI - V00037453

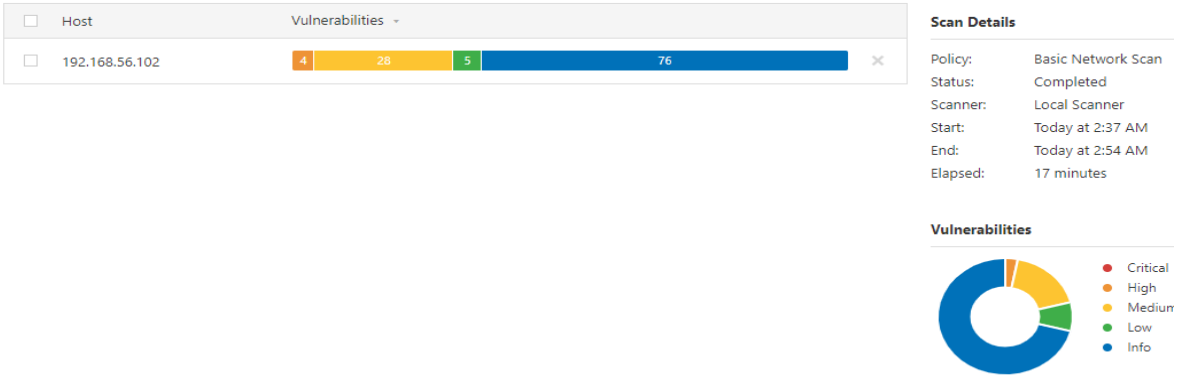
Defense Strategies

In the second part of the project, we will use the attack intelligence obtained in part 1 to implement adequate defense strategy to prevent or detect similar attacks in the future. As part of the protection mechanisms, we will setup snort IDS on the machine NeptuneR and IP Tables on the machine Neptune N.

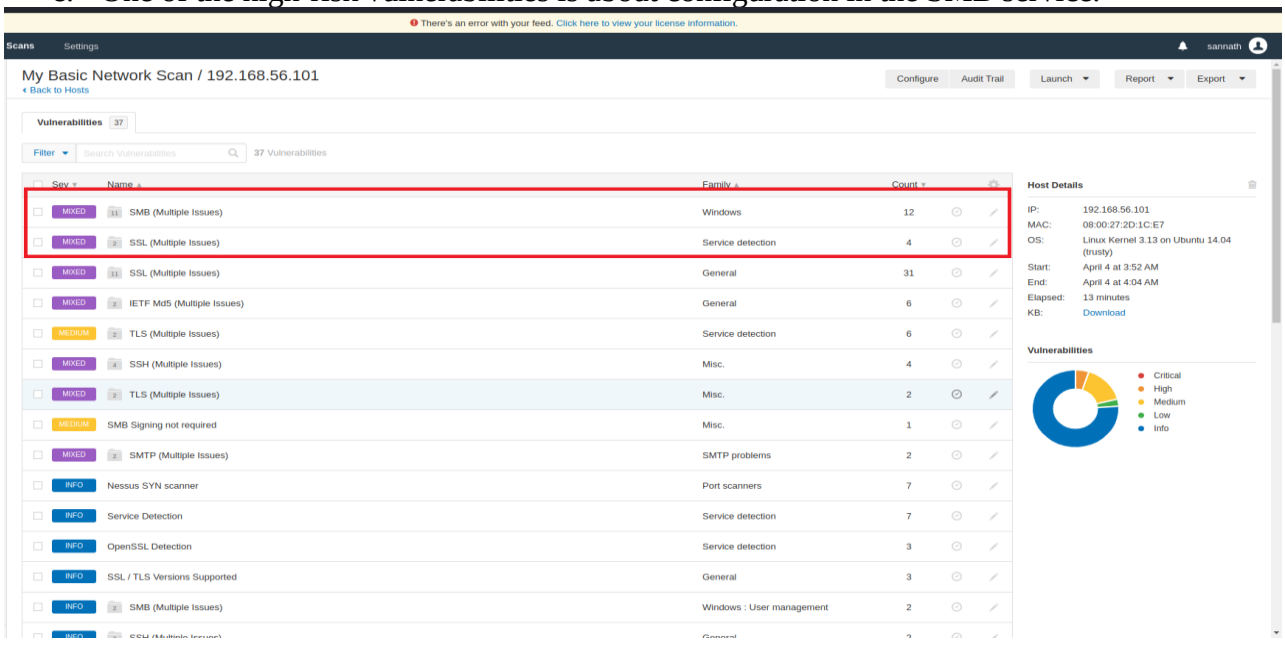
Phase 1: Intrusion Detection

Identify and select one medium or high-risk vulnerability (other than password cracking or unsupported OS version) against NeptuneN, for which you can identify an exploit code and execute successfully the exploit using Metasploit.

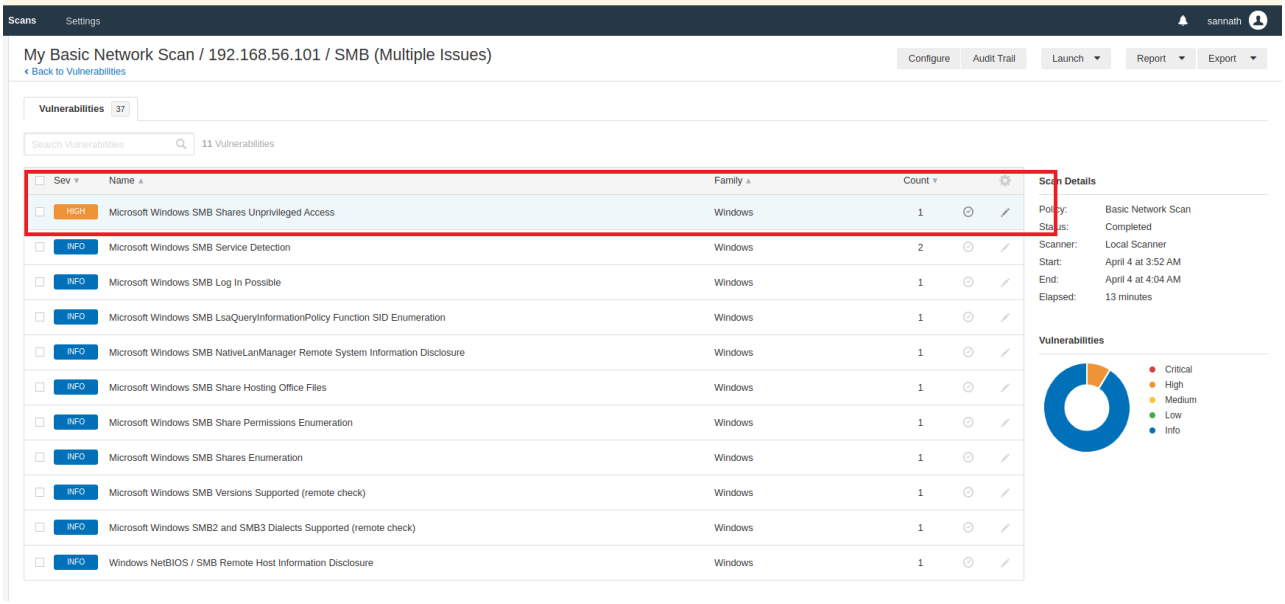
a. Turned on Neptune N machine.
b. Searched for the IP addresses of N machine (192.168.56.102) using Kali.
c. The Zenmap scan revealed for Naptune N seven opened ports with the services they were using.
d. Nessus results after scanning the IP address (192.168.56.102) of machine Neptune N. NeptuneN with 4 high risk vulnerabilities and 26 medium vulnerabilities.

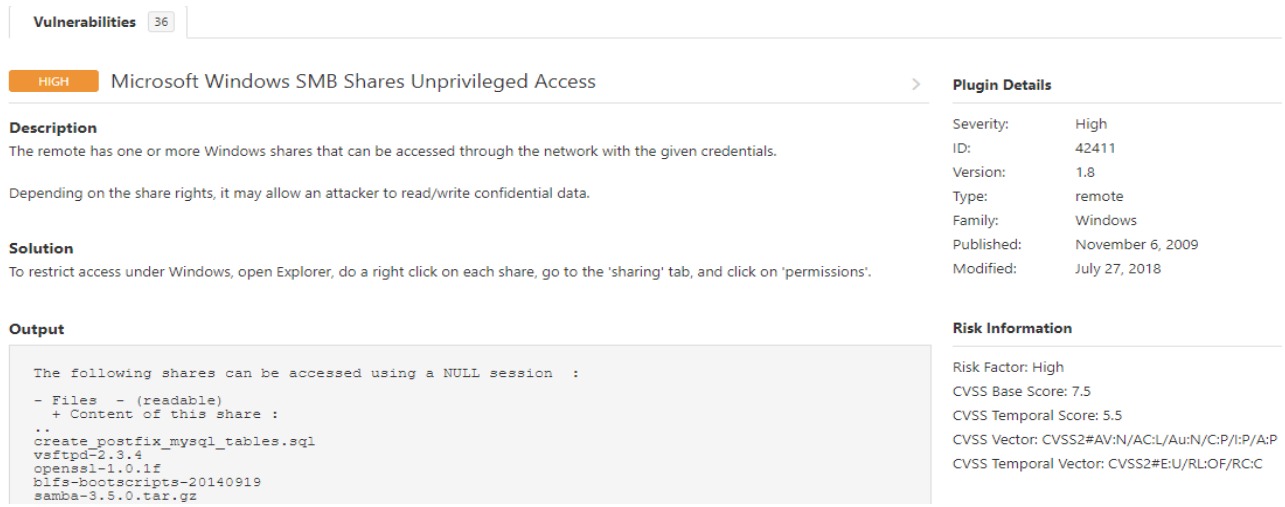


e. One of the high-risk vulnerabilities is about configuration in the SMB service.

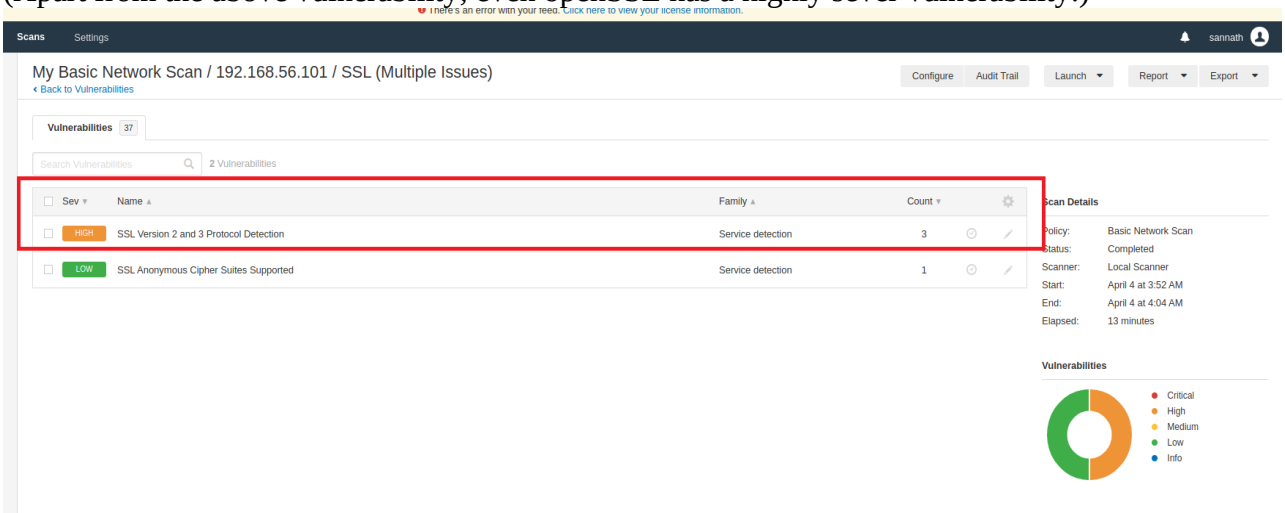


f. It is possible to access SMB shares without acquiring any credentials:





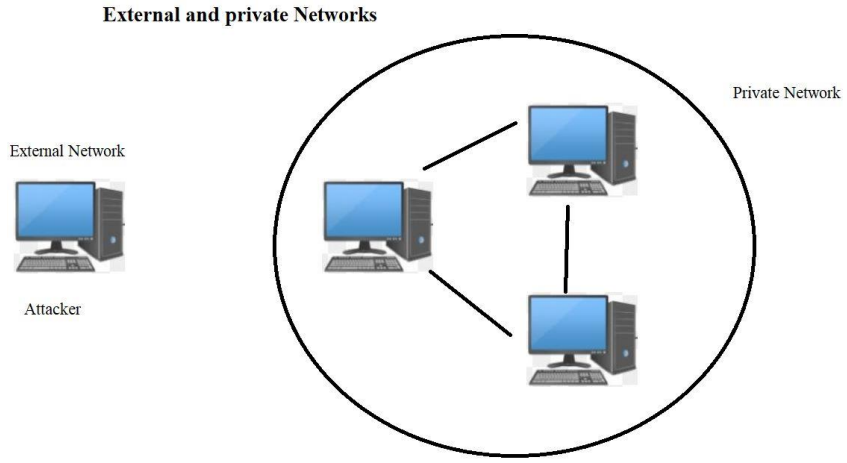
(Apart from the above vulnerability, even openSSL has a highly sever vulnerability.)



1. Explain briefly the generic attack scenario associated with the selected vulnerability (2 paragraphs maximum); a graphical sketch (in addition to the explanations) is required. Do not copy and paste paragraphs and figures from the Internet. You can search for such information online; but make sure that the explanation is reworded using your own terms, and the graph is your own (1.5%).

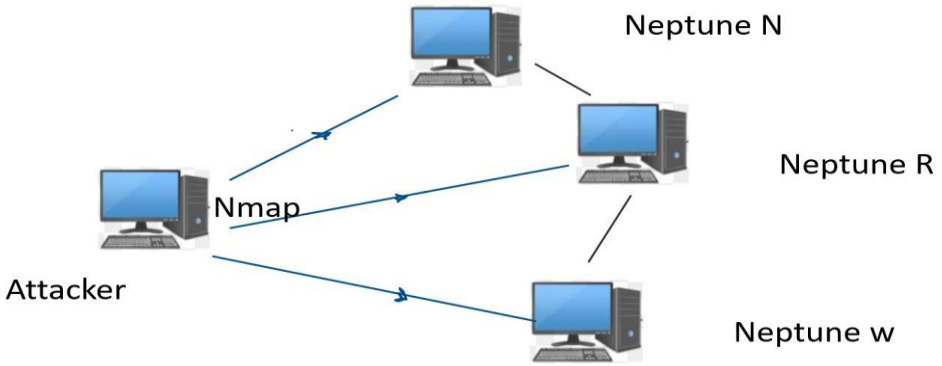
SMB: Using the Server Message Block protocol, an application (or the user of an application within the network) can access files or other resources from the other machines in the same network. This allows applications to read, create, and update files on the remote server. It can also communicate with any server program that is set up to receive an SMB client request. SMB functions as a request-response or client-server protocol. Once the connection is established, the client computer or program can then open, read/write, and access files like the file system on a local computer.

smb_enumuser vulnerability – we can see the general scenario of the attack in the diagram below.
The attacker intrudes into the organization's (private) network and scans each of the machines in the network. The attacker looks for open services and selects the machine which has SMB as open service.



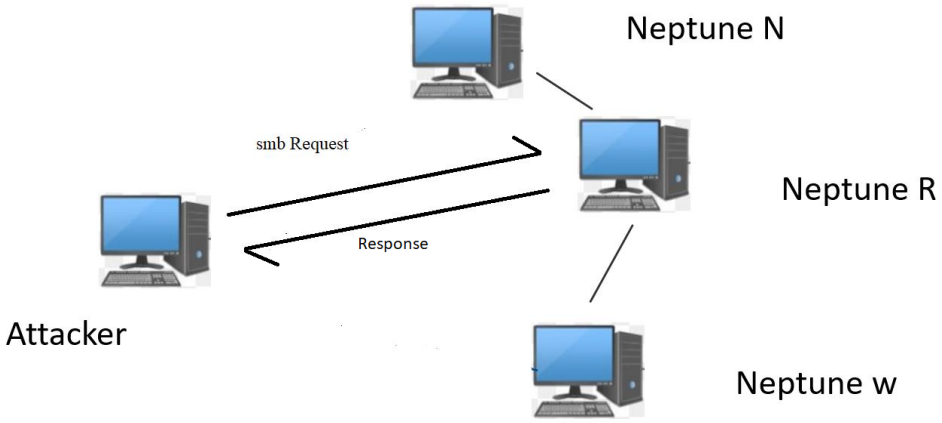
Attacker tries to execute the null sessions (User=NULL, password=NULL) using one of the tools like metasploit, enum4linux or smbclient. On successful execution, the tool should be capable of fetching data like machines list, group and users list, password protection policy and share lists (including comments in share-list). Sends request by sending user as null and password as null to by-pass the authentication conducted by SMB protocol. With unprivileged access through null session the attacker can fetch the above-mentioned information.
When the attack is initiated, the attack machine starts the request for smb ports. The target then responds to the request through one of the smb ports along with the authentication string. The authentication string is then processed by the attack machine to complete the negotiation which is sent in the next packet to the target. The target now receives the authentication results as response which decides the if to accept the connection and allow access or deny it. With the vulnerability of null session, the attacker by-passes this authentication with empty strings for username and password.

Scanning open services of every machines in the network:



(Reads the open services and select the machine which have ports 139 or 445 open_

Creating Null Session with the Machine



2. Execute the attack against machine NeptuneN by exploiting the selected vulnerability using Metasploit. Capture sample attack packets using wireshark (you can start wireshark, just before typing the "run" or "exploit" command in Metasploit; and stop it after the attack has completed).

We identified and selected high-risk vulnerability in SMB protocol for Neptune N By reviewing the Nessus results for which we identified an exploit in Metasploit. Below are the exploits of smb which have been executed successfully. Exploits from metasploit related to SMB on Ports 139, 445.

```
msf5 > search scanner/smb

Matching Modules
================

   #   Name                                      Disclosure Date  Rank    Check  Description
   -   ----                                      ---------------  ----    -----  -----------
   0   auxiliary/scanner/smb/impacket/dcomexec   2018-03-19       normal  No     DCOM Exec
   1   auxiliary/scanner/smb/impacket/secretsdump                 normal  No     DCOM Exec
   2   auxiliary/scanner/smb/impacket/wmiexec    2018-03-19       normal  No     WMI Exec
   3   auxiliary/scanner/smb/pipe_auditor                         normal  No     SMB Session Pipe Auditor
   4   auxiliary/scanner/smb/pipe_dcerpc_auditor                  normal  No     SMB Session Pipe DCERPC Auditor
   5   auxiliary/scanner/smb/psexec_loggedin_users                normal  No     Microsoft Windows Authenticated Logged In Users Enumeration
   6   auxiliary/scanner/smb/smb1                                 normal  No     SMBv1 Protocol Detection
   7   auxiliary/scanner/smb/smb2                                 normal  No     SMB 2.0 Protocol Detection
   8   auxiliary/scanner/smb/smb_enum_gpp                         normal  No     SMB Group Policy Preference Saved Passwords Enumeration
   9   auxiliary/scanner/smb/smb_enumshares                       normal  No     SMB Share Enumeration
   10  auxiliary/scanner/smb/smb_enumusers                        normal  No     SMB User Enumeration (SAM EnumUsers)
   11  auxiliary/scanner/smb/smb_enumusers_domain                 normal  No     SMB Domain User Enumeration
   12  auxiliary/scanner/smb/smb_login                            normal  No     SMB Login Check Scanner
   13  auxiliary/scanner/smb/smb_lookupsid                        normal  No     SMB SID User Enumeration (LookupSid)
   14  auxiliary/scanner/smb/smb_ms17_010                         normal  No     MS17-010 SMB RCE Detection
   15  auxiliary/scanner/smb/smb_uninit_cred                      normal  Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   16  auxiliary/scanner/smb/smb_version                          normal  No     SMB Version Detection


msf5 >
```

a. auxiliary/scanner/smb/smb_1 - SMB 2.0 Protocol Detection Detect systems that support the SMB 1.0 protocol

```
File  Actions  Edit  View  Help
msf5 >
msf5 >
msf5 >
msf5 > use auxiliary/scanner/smb/smb1
msf5 auxiliary(scanner/smb/smb1) >
msf5 auxiliary(scanner/smb/smb1) > show options

Module options (auxiliary/scanner/smb/smb1):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    445              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/smb/smb1) >
msf5 auxiliary(scanner/smb/smb1) >
msf5 auxiliary(scanner/smb/smb1) > set rhosts 192.168.56.103
rhosts => 192.168.56.103
msf5 auxiliary(scanner/smb/smb1) >
msf5 auxiliary(scanner/smb/smb1) > run

[+] 192.168.56.103:445    - 192.168.56.103 supports SMBv1 dialect.
[*] 192.168.56.103:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb1) >
```

b. auxiliary/scanner/smb/smb_2 - SMB 2.0 Protocol Detection Detect systems that support the SMB 2.0 protocol

```
msf5 > use auxiliary/scanner/smb/smb2
msf5 auxiliary(scanner/smb/smb2) >
msf5 auxiliary(scanner/smb/smb2) > show options

Module options (auxiliary/scanner/smb/smb2):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    445              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/smb/smb2) > set rhosts 192.168.56.103
rhosts => 192.168.56.103
msf5 auxiliary(scanner/smb/smb2) >
msf5 auxiliary(scanner/smb/smb2) > run

[+] 192.168.56.103:445    - 192.168.56.103 supports SMB 2 [dialect 255.2] and has been online for 3675191 hours
[*] 192.168.56.103:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb2) >
```

c. auxiliary/scanner/smb/smb_enumshares - determines what shares are provided by the SMB service and which ones are readable/writable. It also collects additional information such as share types, directories, files, timestamps, etc.

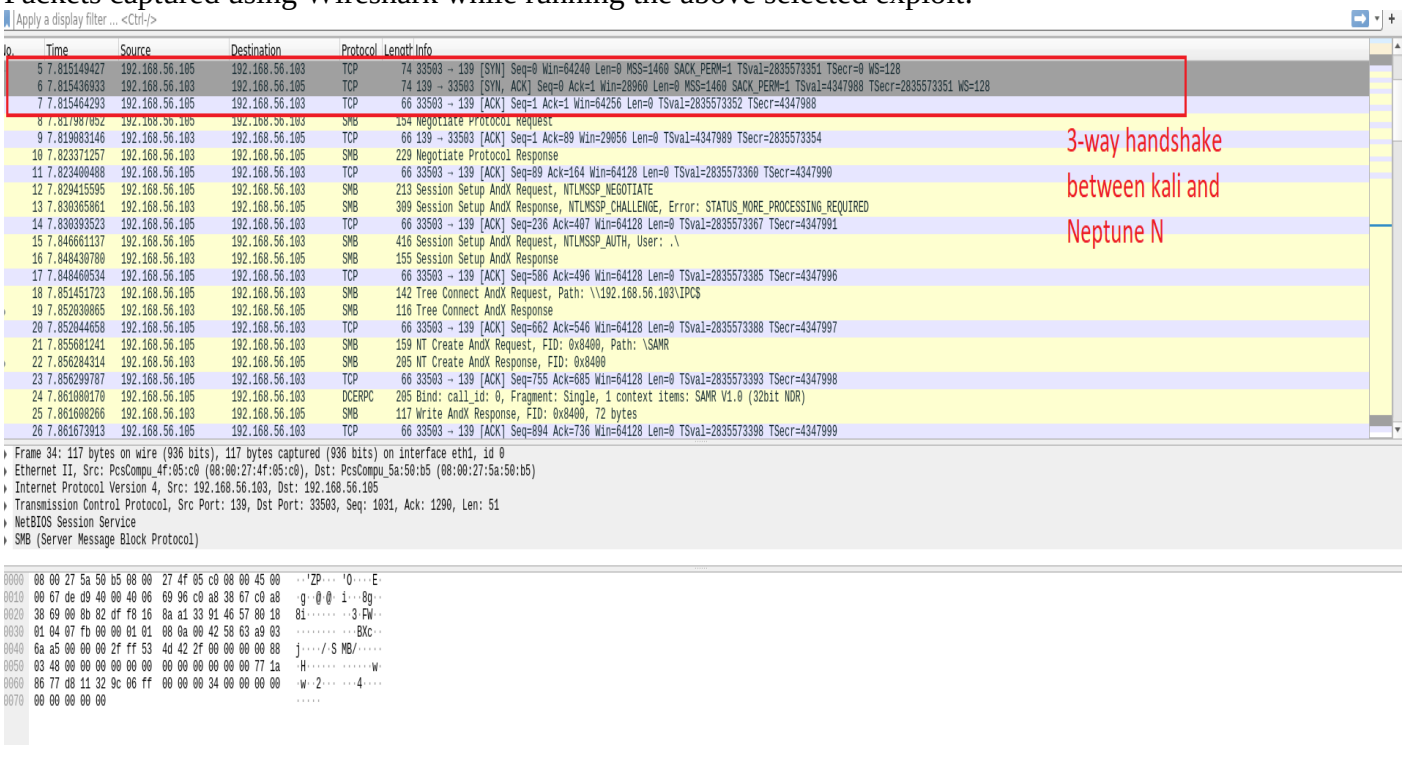
```
msf5 >
msf5 > use auxiliary/scanner/smb/smb_enumshares
msf5 auxiliary(scanner/smb/smb_enumshares) >
msf5 auxiliary(scanner/smb/smb_enumshares) > set rhosts 192.168.56.103
rhosts => 192.168.56.103
msf5 auxiliary(scanner/smb/smb_enumshares) >
msf5 auxiliary(scanner/smb/smb_enumshares) > show options

Module options (auxiliary/scanner/smb/smb_enumshares):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   LogSpider        3                no        0 = disabled, 1 = CSV, 2 = table (txt), 3 = one liner (txt) (Accepted: 0, 1, 2, 3)
   MaxDepth         999              yes       Max number of subdirectories to spider
   RHOSTS           192.168.56.103   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   SMBDomain        .                no        The Windows domain to use for authentication
   SMBPass                           no        The password for the specified username
   SMBUser                           no        The username to authenticate as
   ShowFiles        false            yes       Show detailed information when spidering
   SpiderProfiles   true             no        Spider only user profiles when share = C$
   SpiderShares     false            no        Spider shares recursively
   THREADS          1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/smb/smb_enumshares) > set showfiles true
showfiles => true
msf5 auxiliary(scanner/smb/smb_enumshares) >
msf5 auxiliary(scanner/smb/smb_enumshares) > run

[+] 192.168.56.103:139    - print$ - (DISK) Printer Drivers
[+] 192.168.56.103:139    - Files - (DISK)
[+] 192.168.56.103:139    - IPC$ - (IPC) IPC Service (neptuneN server (Samba
[+] 192.168.56.103:139    - Ubuntu))
[*] 192.168.56.103:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_enumshares) >
```

d. auxiliary/scanner/smb/smb_enumusers

```
msf5 >
msf5 > use auxiliary/scanner/smb/smb_enumusers
msf5 auxiliary(scanner/smb/smb_enumusers) >
msf5 auxiliary(scanner/smb/smb_enumusers) > set rhosts 192.168.56.103
rhosts => 192.168.56.103
msf5 auxiliary(scanner/smb/smb_enumusers) >
msf5 auxiliary(scanner/smb/smb_enumusers) >
msf5 auxiliary(scanner/smb/smb_enumusers) > show options

Module options (auxiliary/scanner/smb/smb_enumusers):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS     192.168.56.103   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   SMBDomain  .                no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads (max one per host)

msf5 auxiliary(scanner/smb/smb_enumusers) > run

[+] 192.168.56.103:139    - NEPTUNEN [ nobody, jpesci ] ( LockoutTries=0 PasswordMin=5 )
[*] 192.168.56.103:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_enumusers) >
```

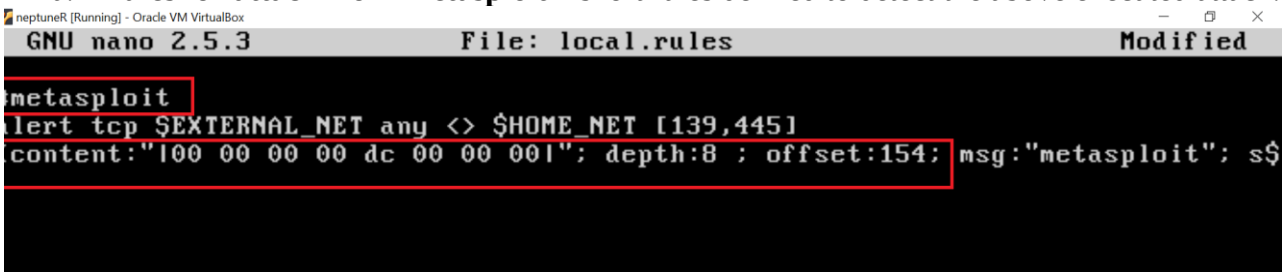
- The above exploits reveal certain information from the target machine. Smb_enumusers fetches valuable information like some of the password policies and users in target machine (Neptune N).

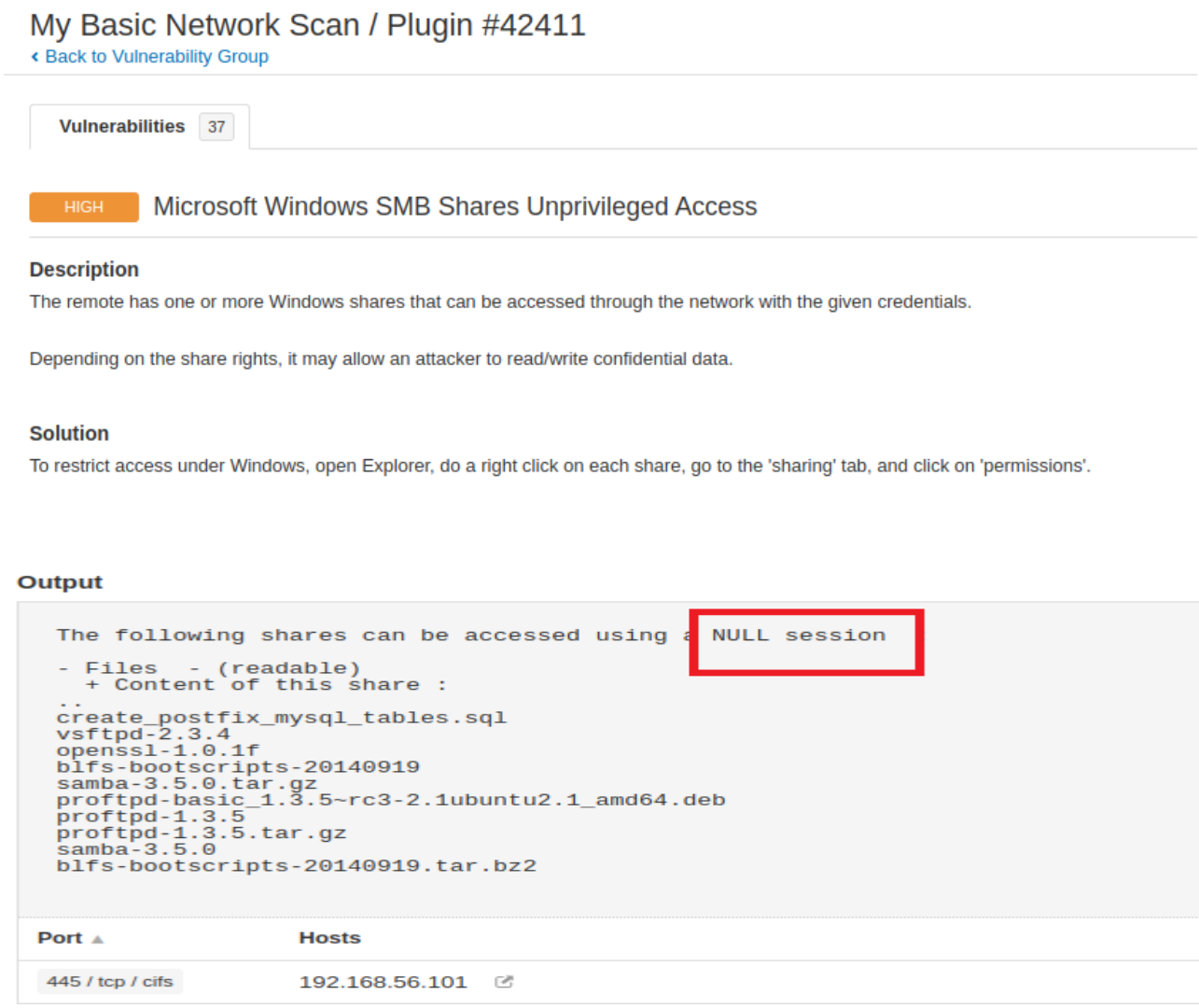
Packets captured using Wireshark while running the above selected exploit:



3. Analyze the sample attack packets and define new Snort rules (as many as you think are necessary) to detect the attack, and add the rules to the snort rule set. Justify the rationale for the rules. Make sure your Snort rules do not over-fit the attack scenarios.

a. **Rules for attack from Metasploit –** snort rules defined to detect the above executed attack.





Above shown vulnerability says that a null session (user = NULL, password = NULL) can get unprivileged access.

Explaining keywords in the rule:

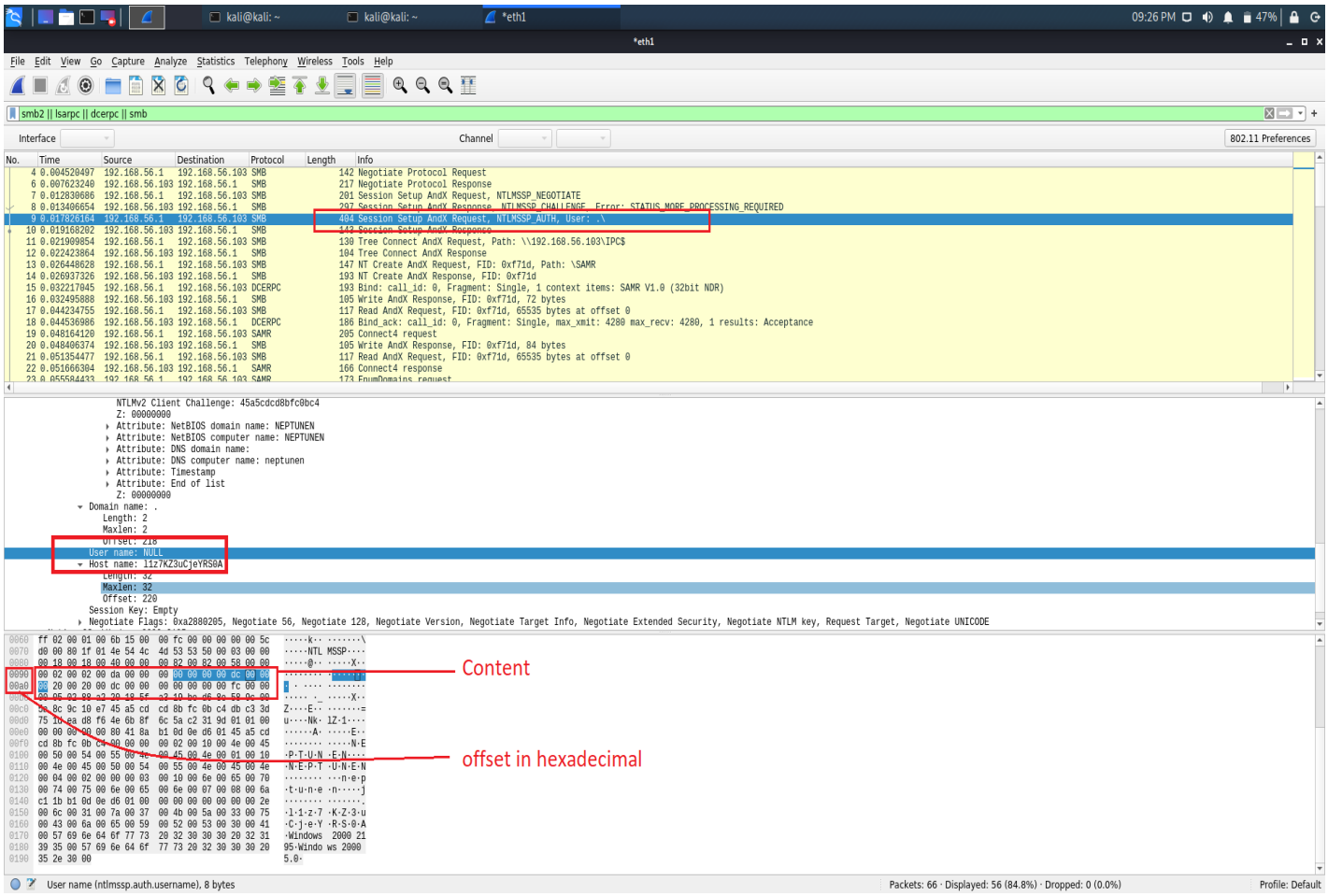
Alert- to create an alert record in output CSV file TCP protocol

$EXTERNAL_ANY – the source ip is any of t he ip addresses listed under this variable in snort.conf

any – source port number

→,←, < > - defines direction/flow of packet
The default port number is 139 and as the port number can be changed to 445 to execute the attack, both the port numbers are mentioned to track attacks from metasploit.

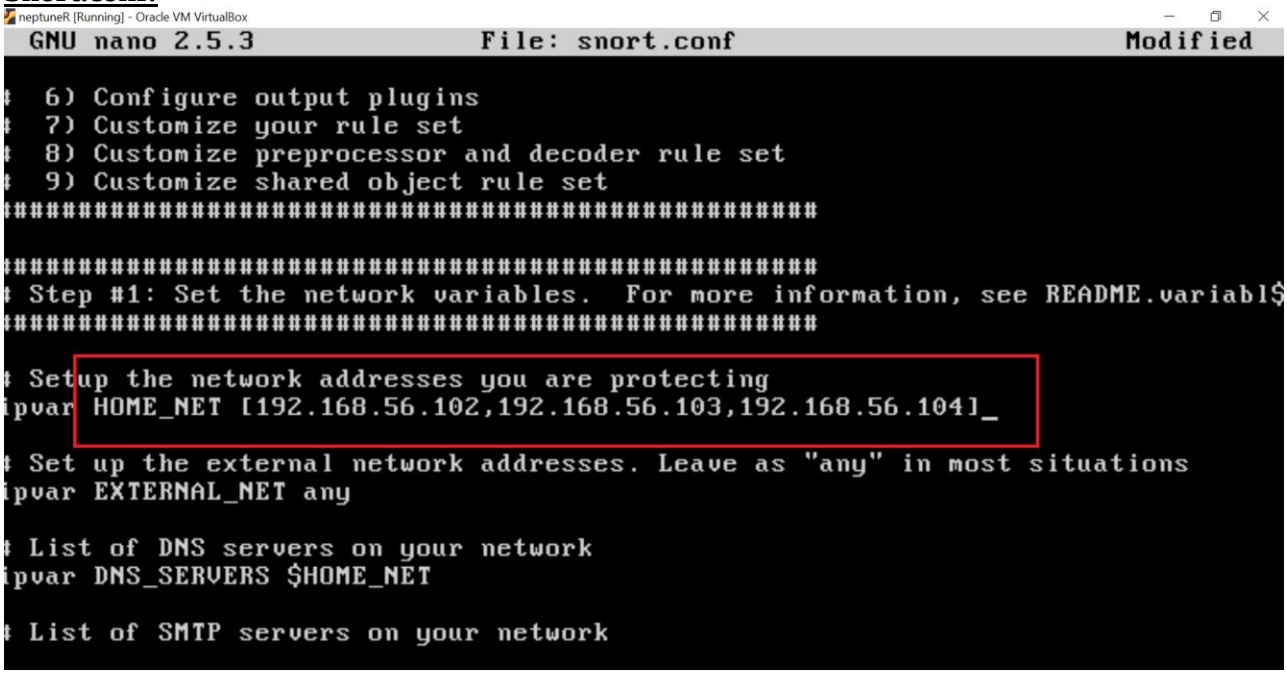
Offset has been used to make sure that the snort captures the packet only with the USER: NULL but nothing else. (As there is a possibility of having a NULL value assigned to other attributes in the packet).

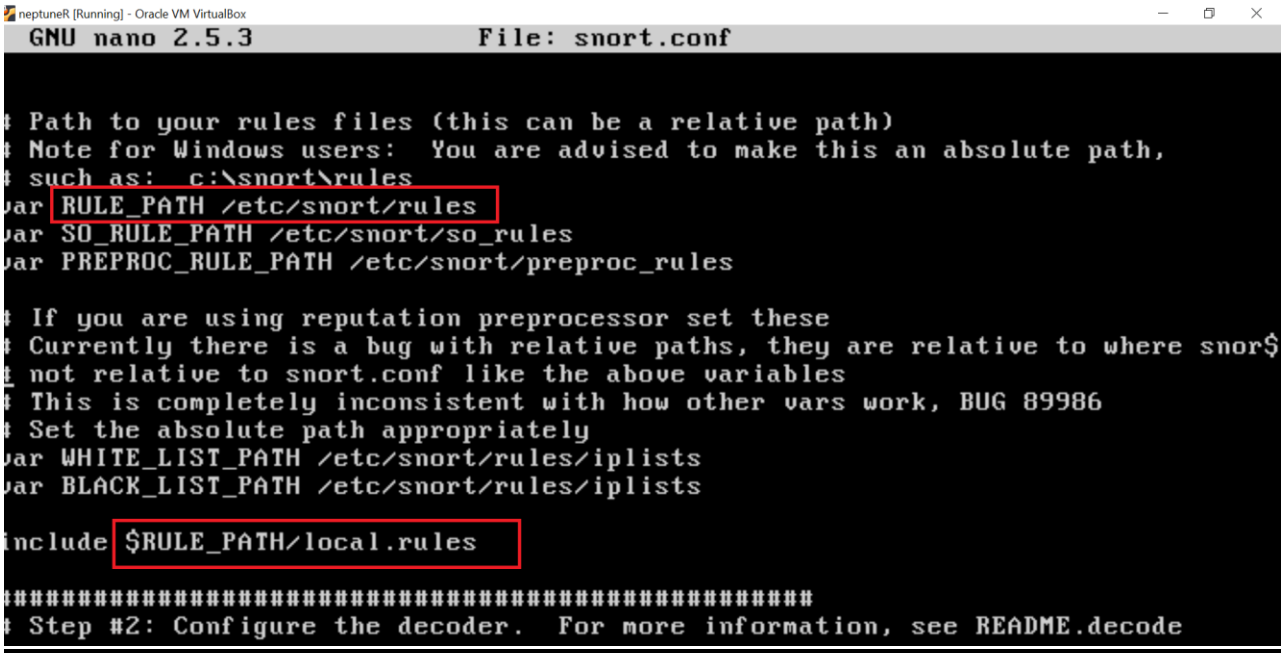


4. Configure Snort (on the Neptune machine) and run it in intrusion detection mode. Execute the relevant exploit for the vulnerability using your attack machine (i.e. Kali).

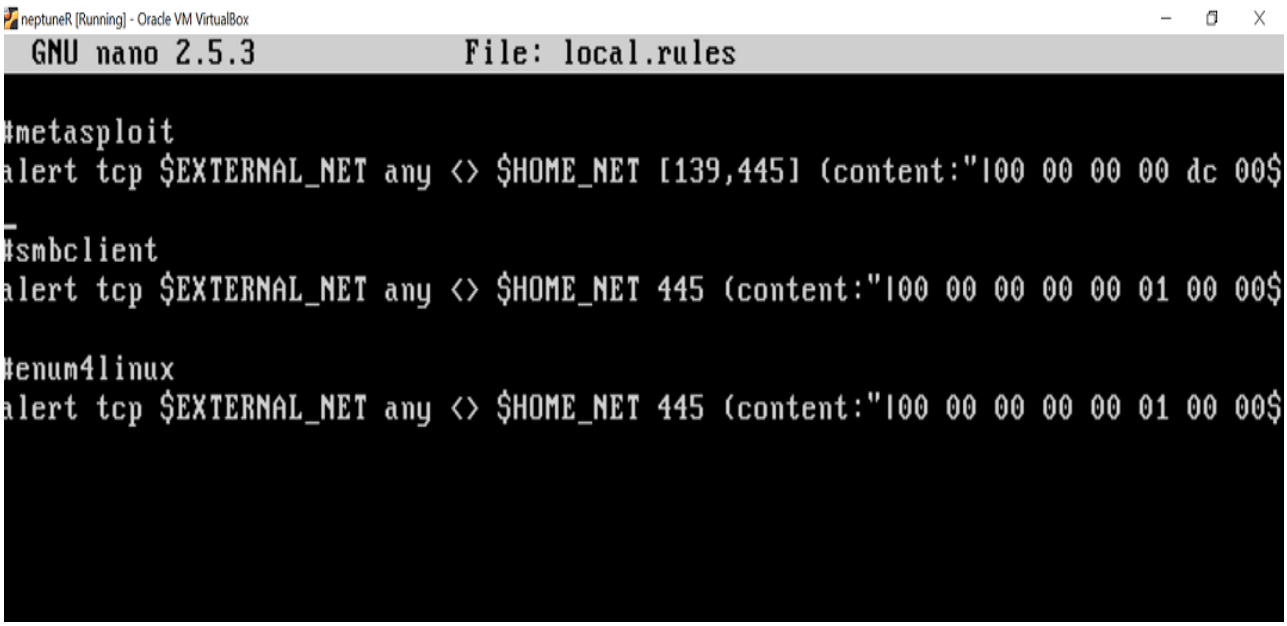
Including the machines belonging to the neptune's network through HOME_NET variable in configuration file.

Snort.conf:



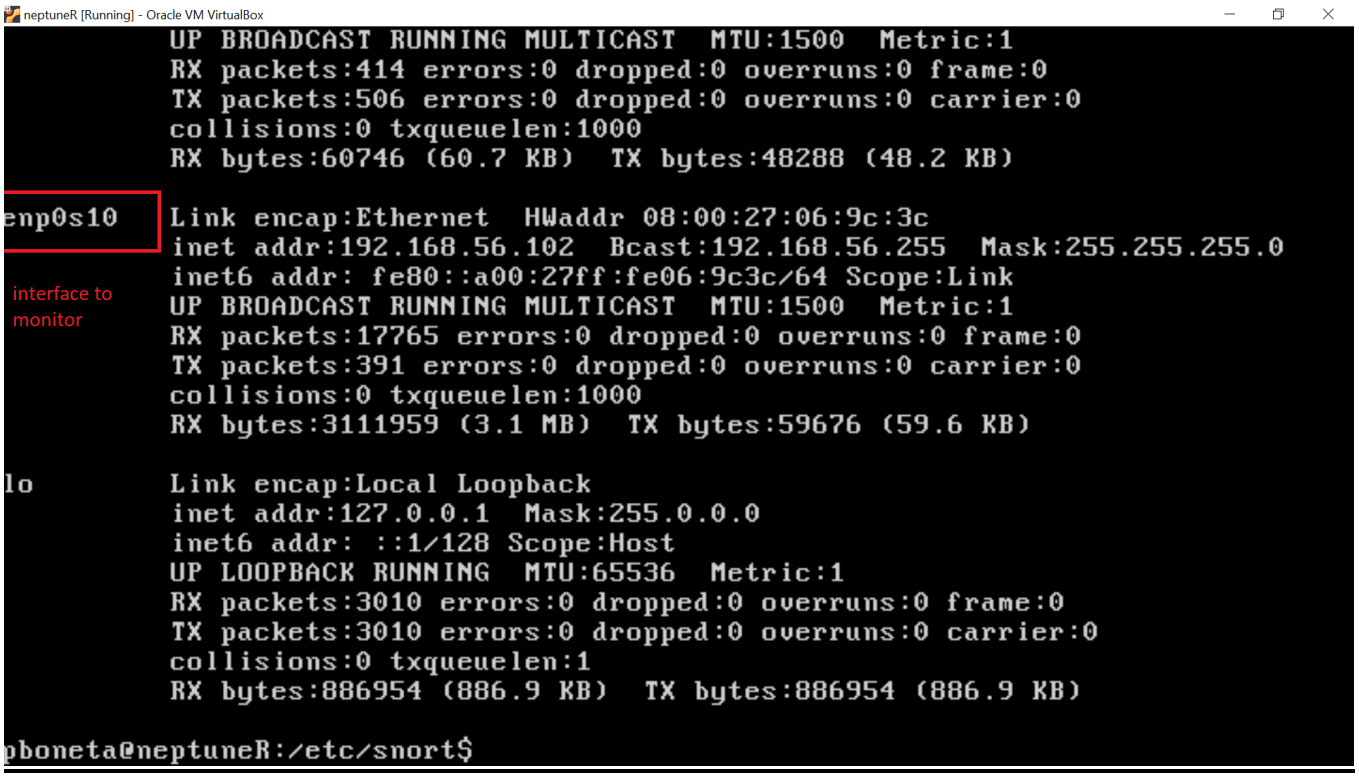


(HOME_NET and path for local rules and output CSV file.)

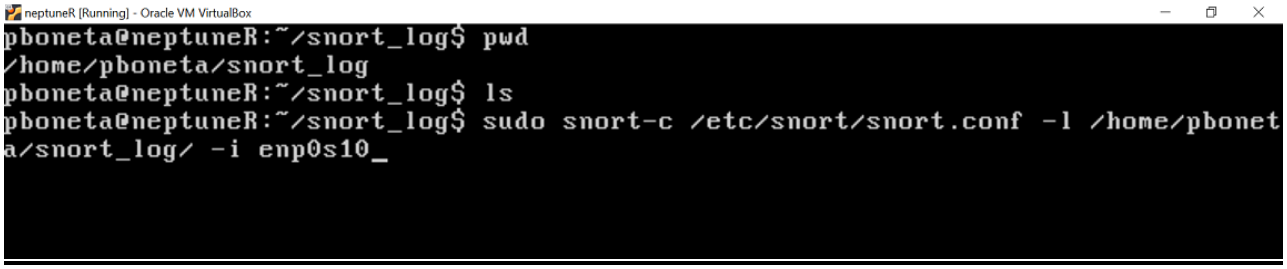


(Rules to detect all the variations of the attack)

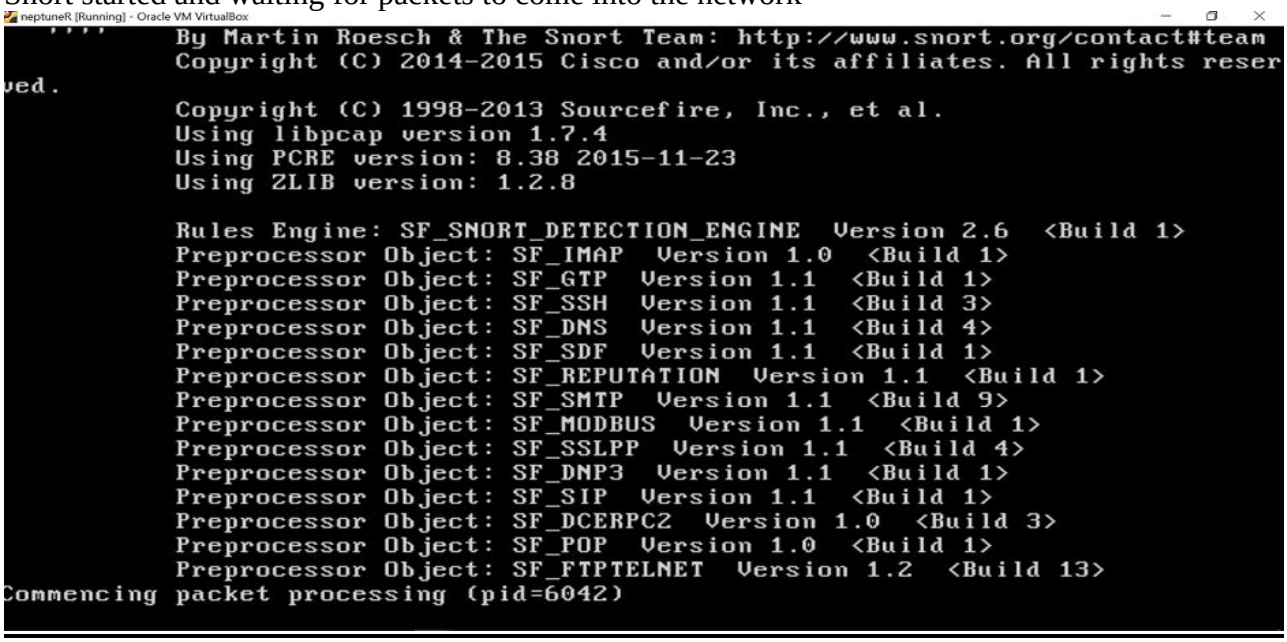
Rule for Metasploit looks for the packet which has username: NULL. The same is conveyed in the rule as hexa-decimal format using content rule option. Rules for smbclient/enum4linux have the same content as the packets of these tools are structured the similar way and have the same content value.



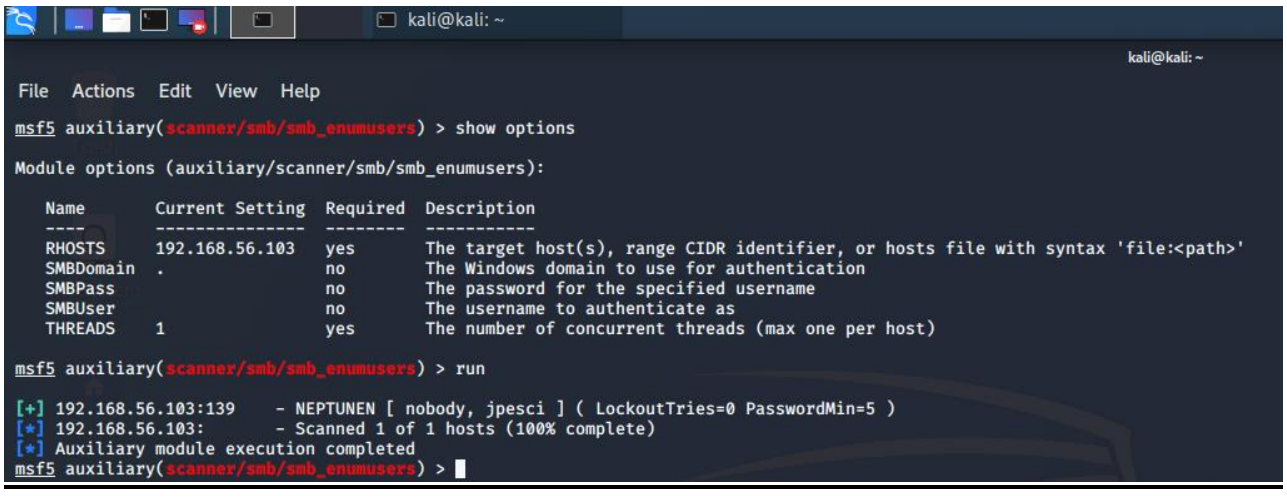
Command to start snort:



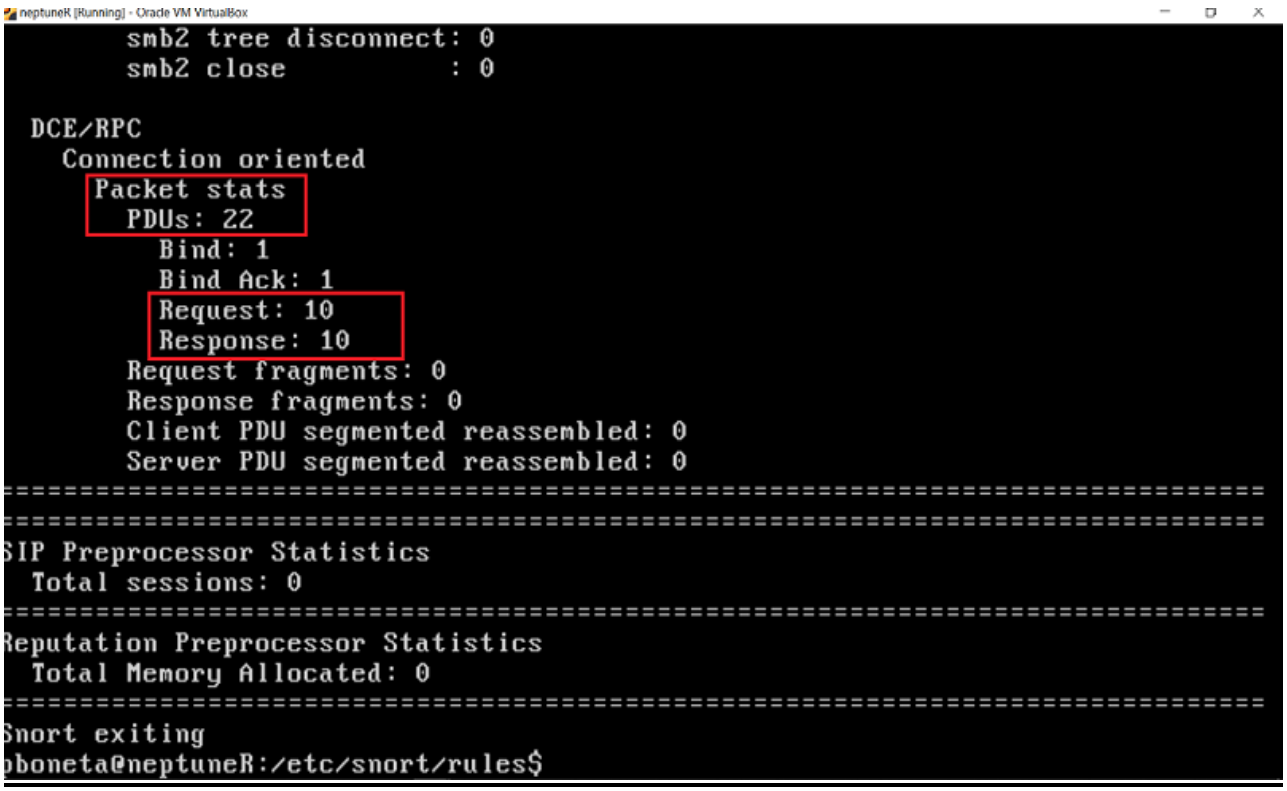
Snort started and waiting for packets to come into the network



Metasploit - Running exploit

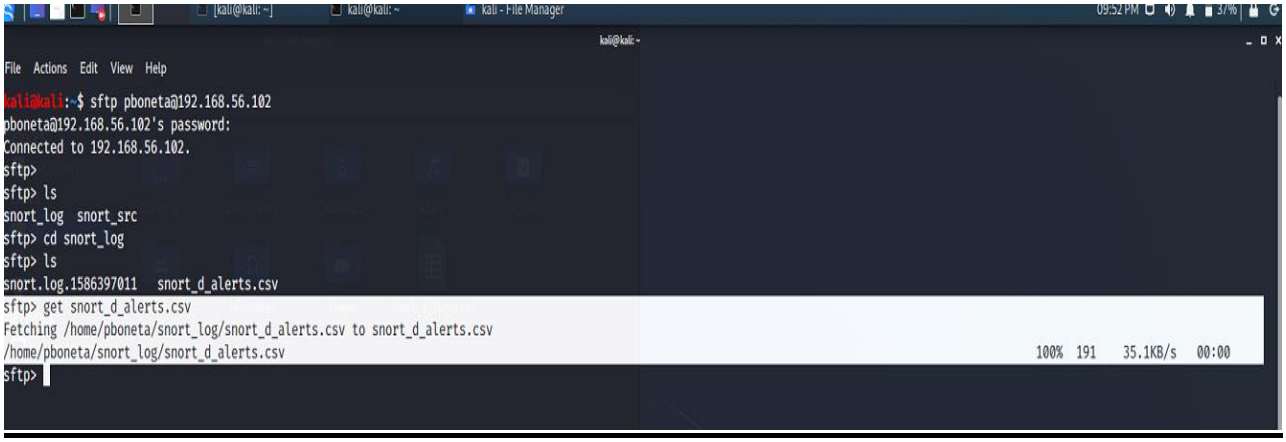


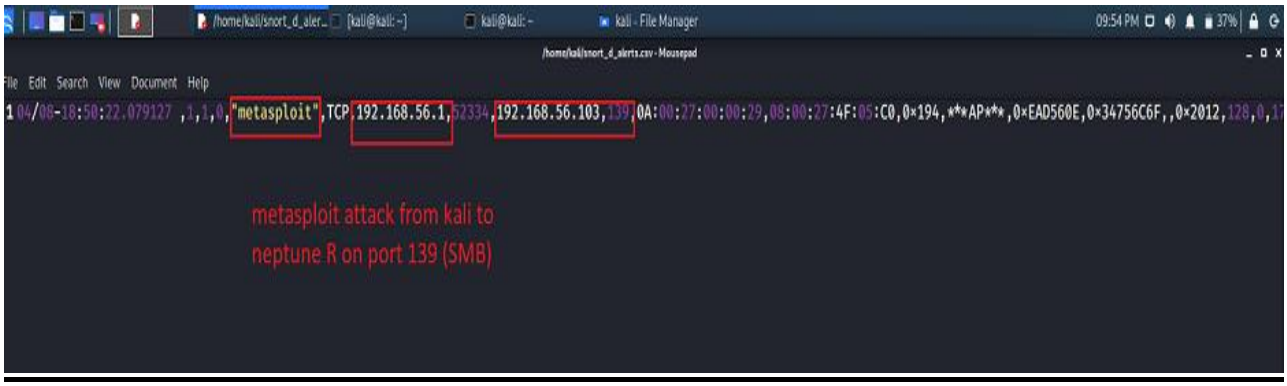
Stopping snort



(Stats of the packets encountered by snort)

5. Analyze the Snort alerts log generated after the attack, and discuss the results in terms of false positives and false negatives (in principle the snort configuration must successfully alerts on all suspicious packets, while not raising alerts on legitimate traffic) . Note: It matters that the exploit be relevant (you cannot pick one at random), and you must complete all the proper steps in Metasploit (initializing, launching, and completion). Provide screenshots documenting the different steps.

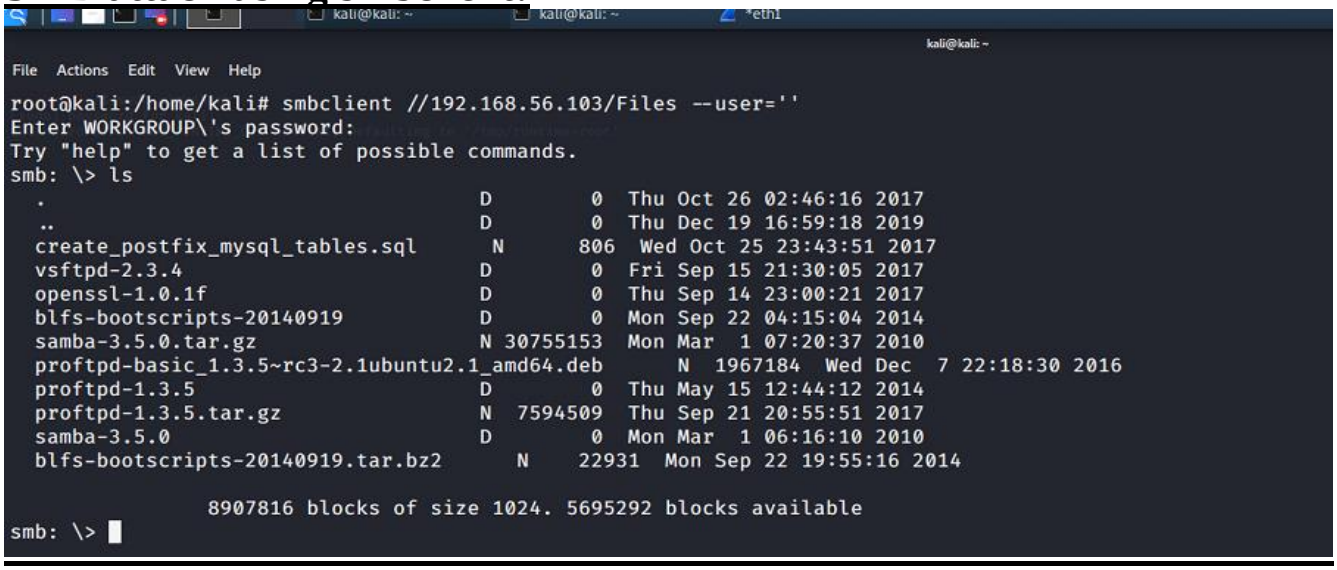


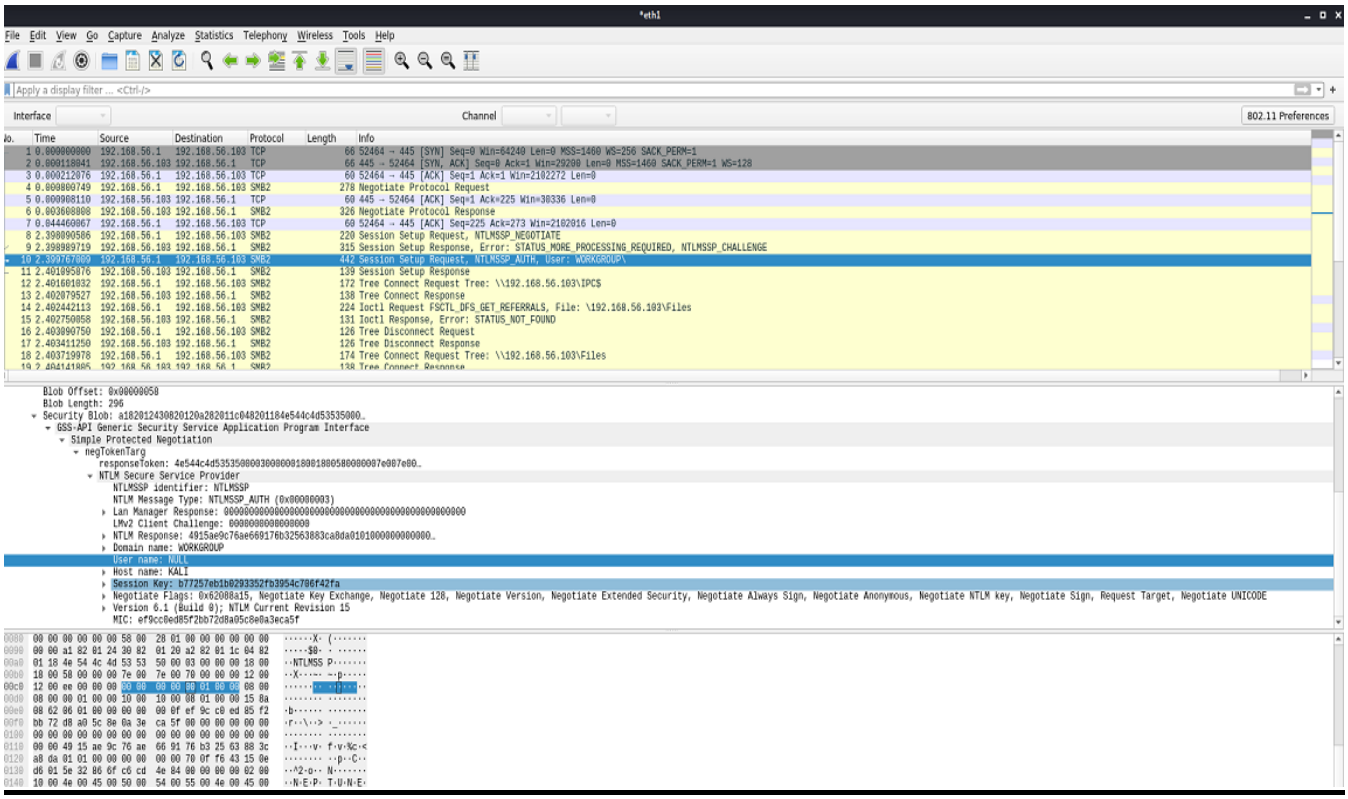


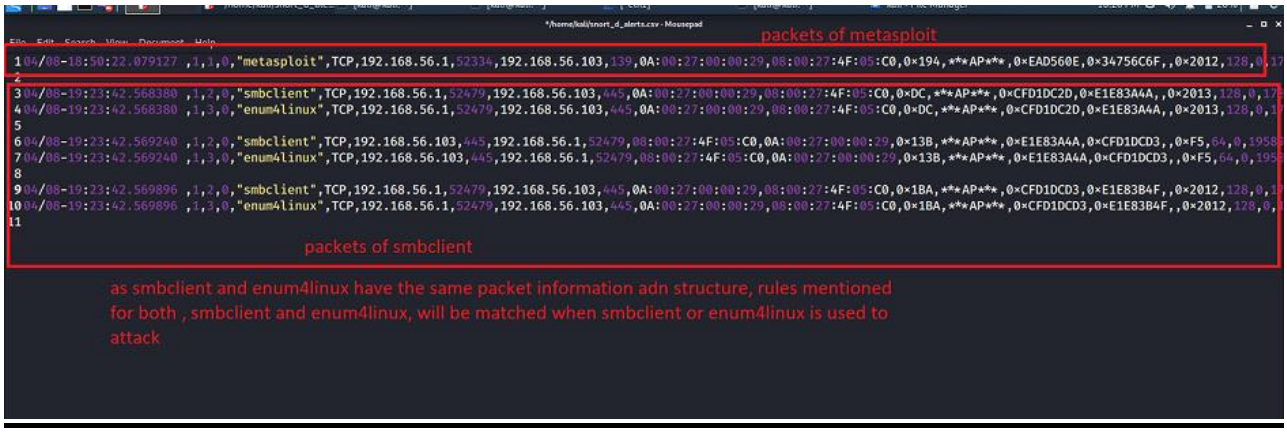
Packet that matched in local rules

For all the three approaches of SMB attack, the packets are being captured by Snort. There are no false negatives for the snort rules as no malicious packet have gone undetected.
We can check for false positives, when a legit user within a Neptune network from one of the machines is trying to access the other machines in network and if such a packet is detected as malicious packet then it can be considered as a false negative.
This can happen if the content rule option is checking for the existing users and and a new user is not updated in the content rule option of snort rule.

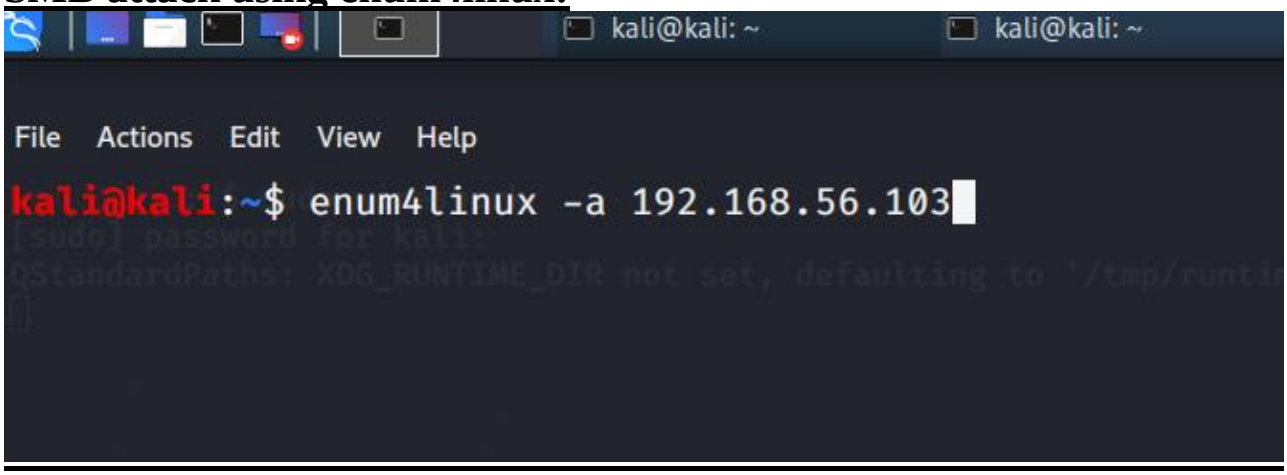
SMB attack using smbclient:

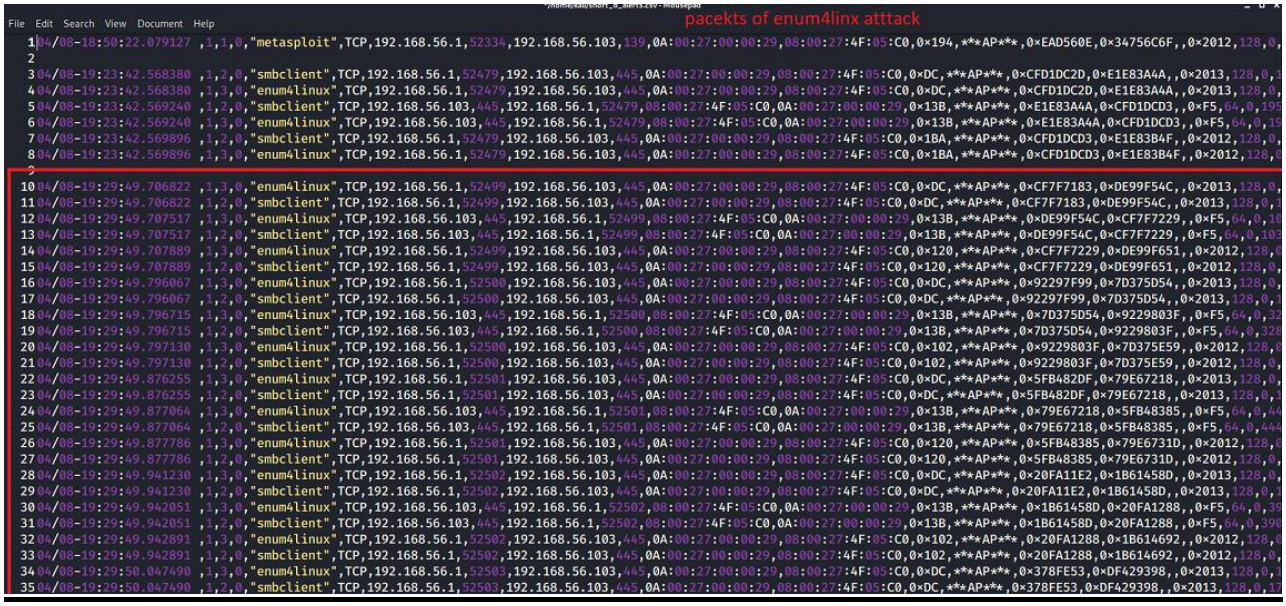






SMB attack using enum4linux:



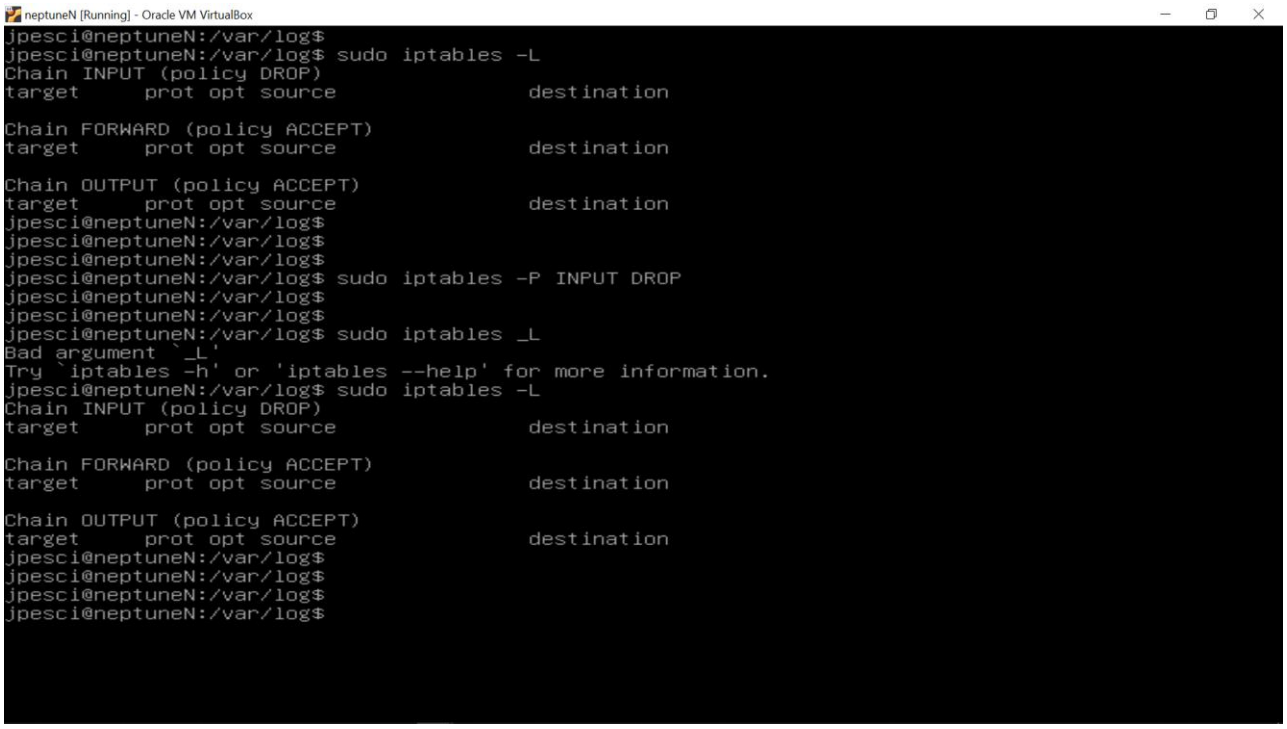


Since the flag -a in the command raises multiple requests more than 2000 packets are captured in this approach.

Phase 2: Intrusion Prevention

To protect against the above attacks, we would like to reinforce the IDS protection using IP Tables firewall. The protection scope (in this phase) will be the Neptune N machine, i.e., the IP Tables rules will be deployed on Neptune N. Since this part of the project focuses on protection, it is assumed that you'll have direct access to the internal network. This means you can update the firewall rules on the machine directly. The default root credentials for Neptune N will be given after the deadline for Part 1. Note that snort will run only in non-inline mode. That is, it does detection only, and does not actively prevent anything. We use IP Tables for that.

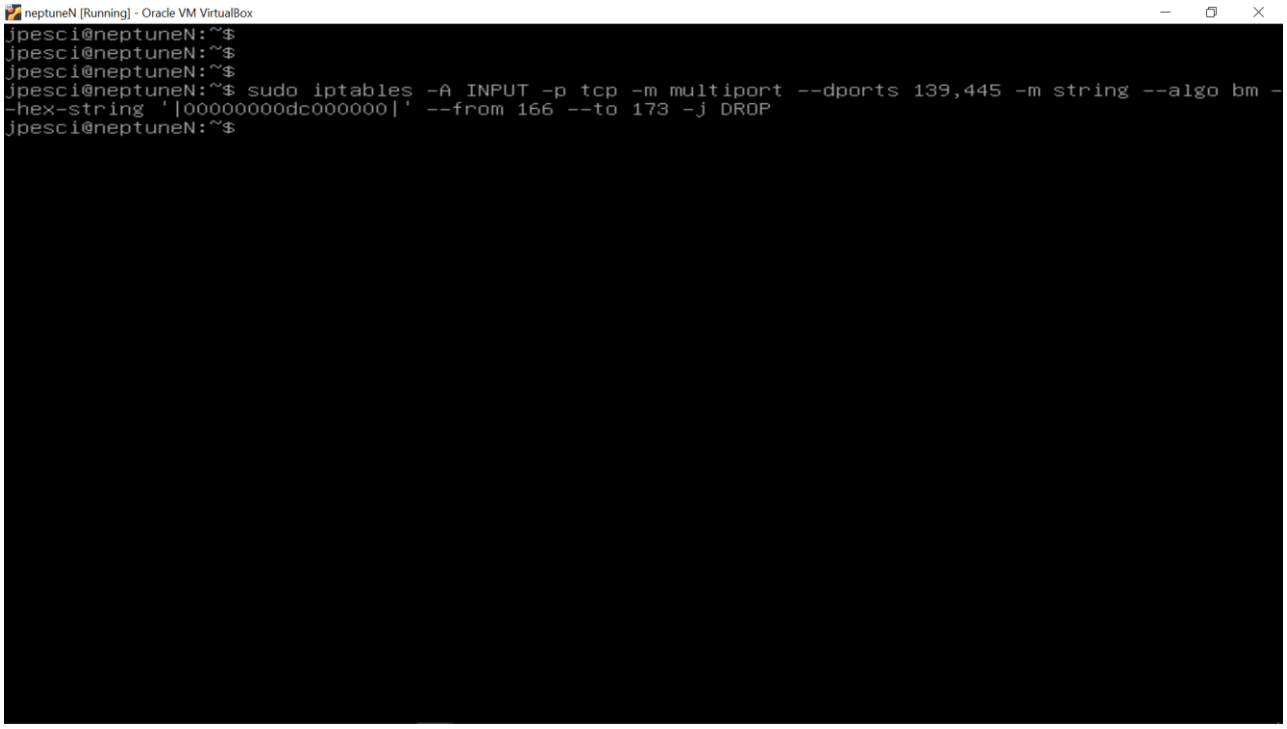
1. **Define the IP Tables rules and provide rationale for each of the rules. You should minimize false negatives and false positives so that a legitimate client is allowed access, but a client that attempts the selected attacks is blocked.**

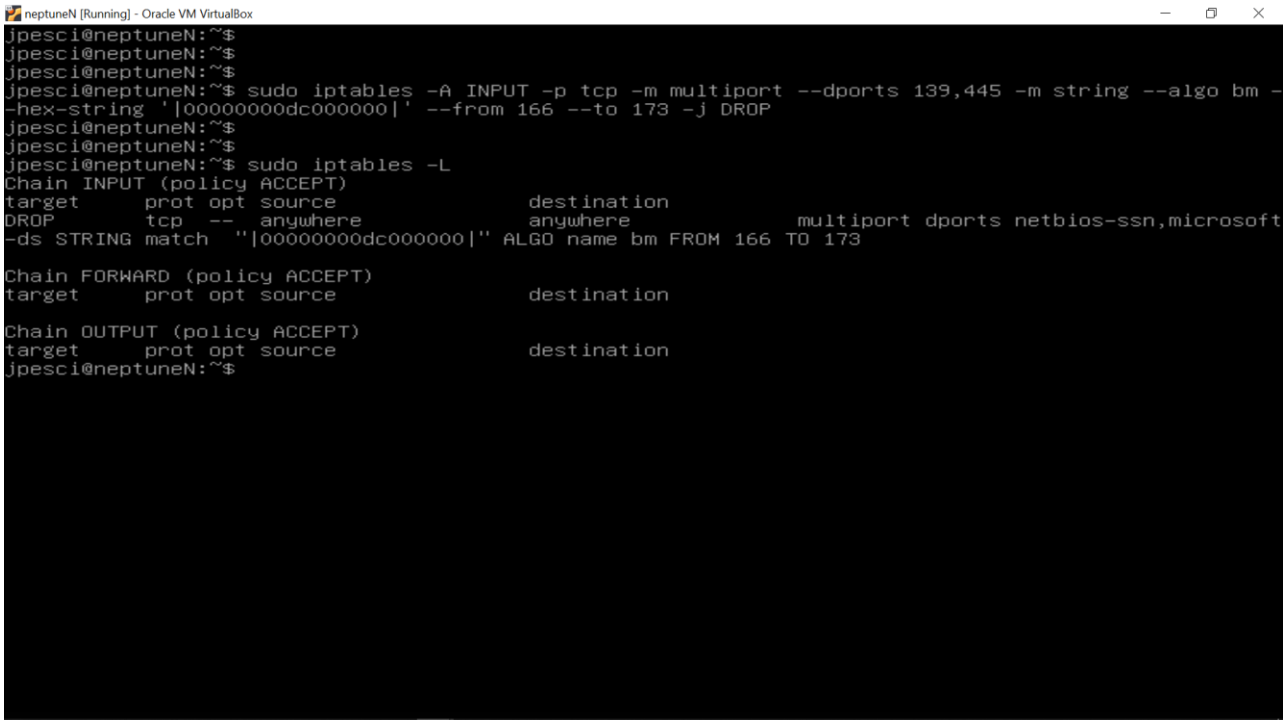


Default iptables : sudo iptables -L

Above screenshot shows changing of policy of INPUT chain from ACCEPT to DROP.

Below is the rule added for the Metasploit attack :





As the request packets from the attacker has to be blocked, we add this rule to INPUT chain and change default policy from ACCEPT to DROP.

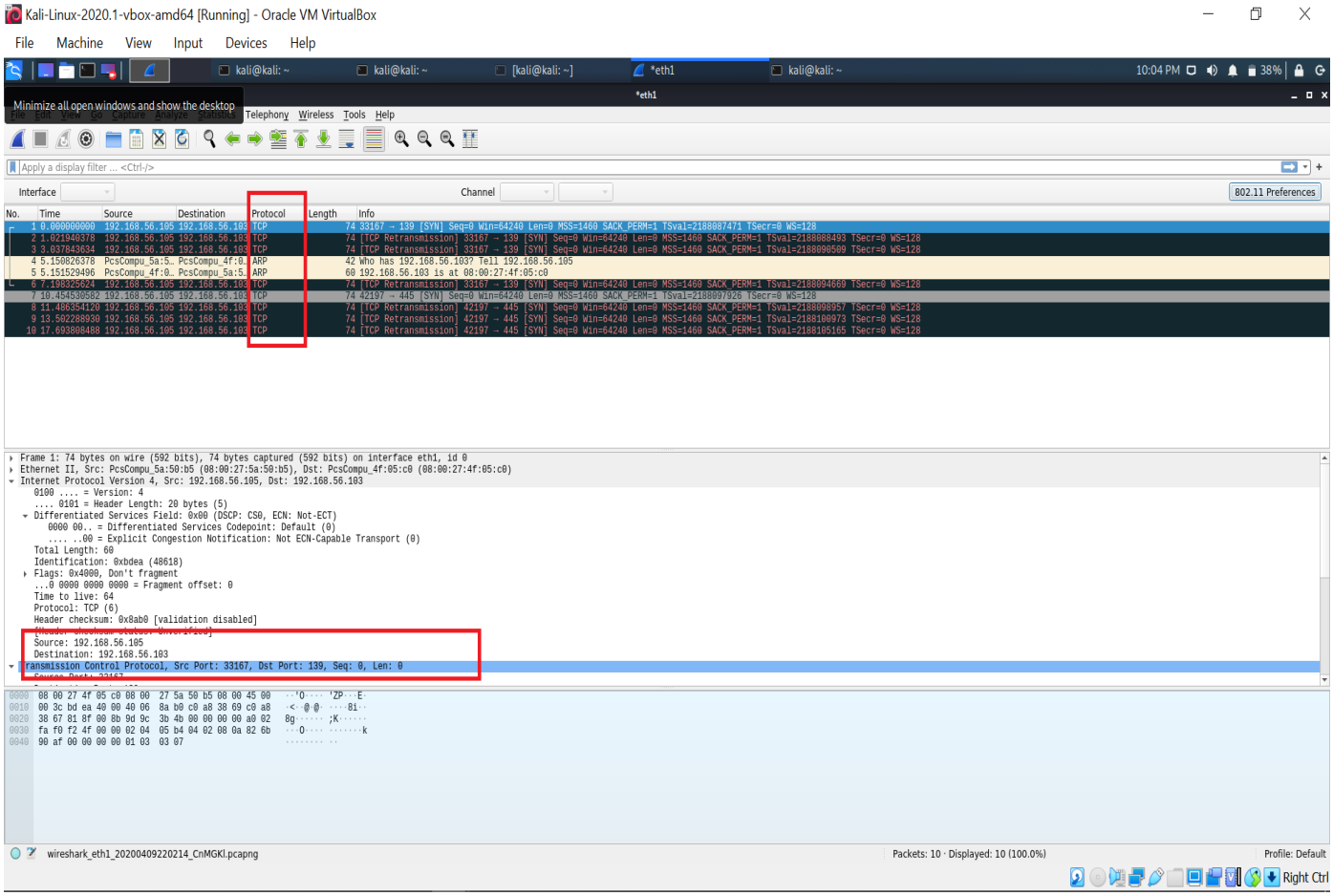
In the above rule, if TCP packets coming into Neptune ‘s network with following information:

- Port number 135 and 445
- Match the hex-string of (username) NULL
- If the string which is matched starts from 166 byte to 173 bytes

Then such packets are considered as malicious packets and they are blocked(dropped).

The above rule is for the attack from Metasploit but the same attack from other tools (, enum4linux) can be blocked in the same way by changing the --hex- string to be matched and by mentioning the start position(from offset) of the matched string (Username: NULL)

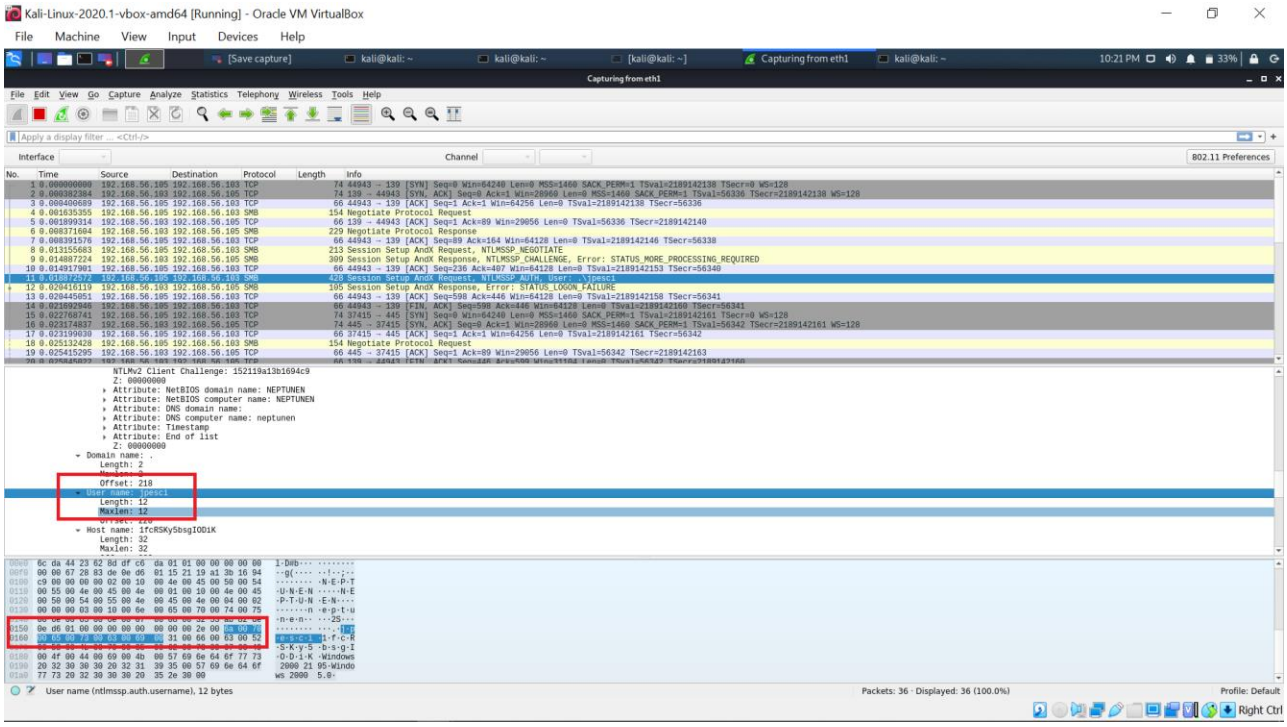
This figure shows Wireshark for null session:



And for legit client since the username field will be having a value other than null, the user would not be blocked.

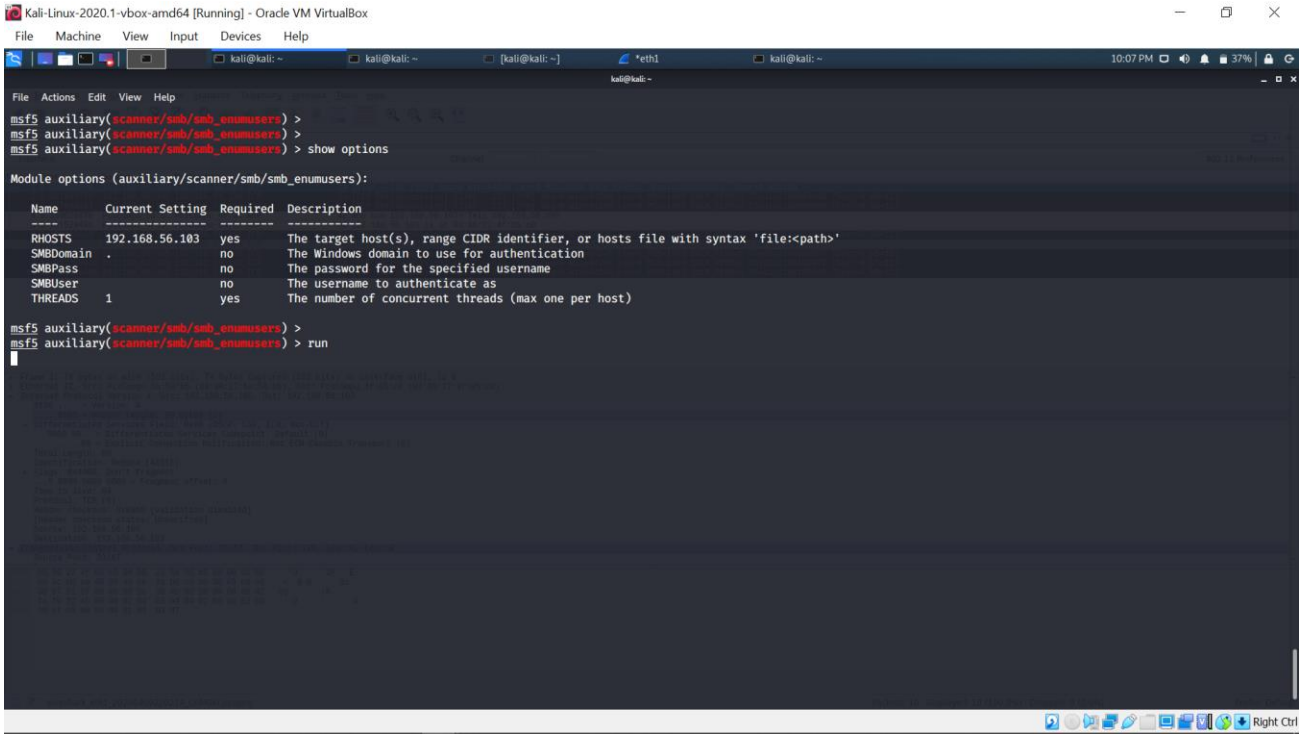
Wireshark packets for jpesci:

By setting the SMBuser and SMBpass, we tried to simulate the genuine packet and below is the screenshot of those packets.

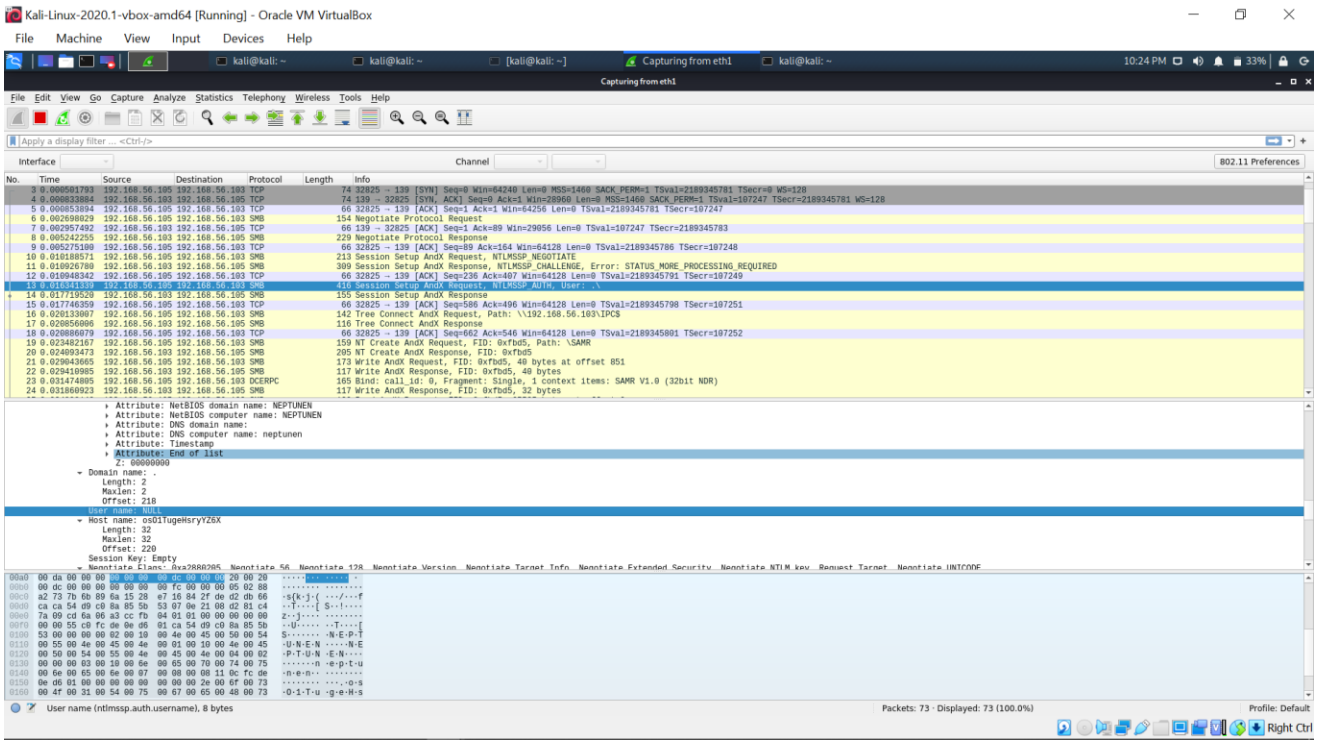


2. Test the firewall rules by executing the attacks (in metasploit, when relevant exploit exists); provide screenshots documenting the results.

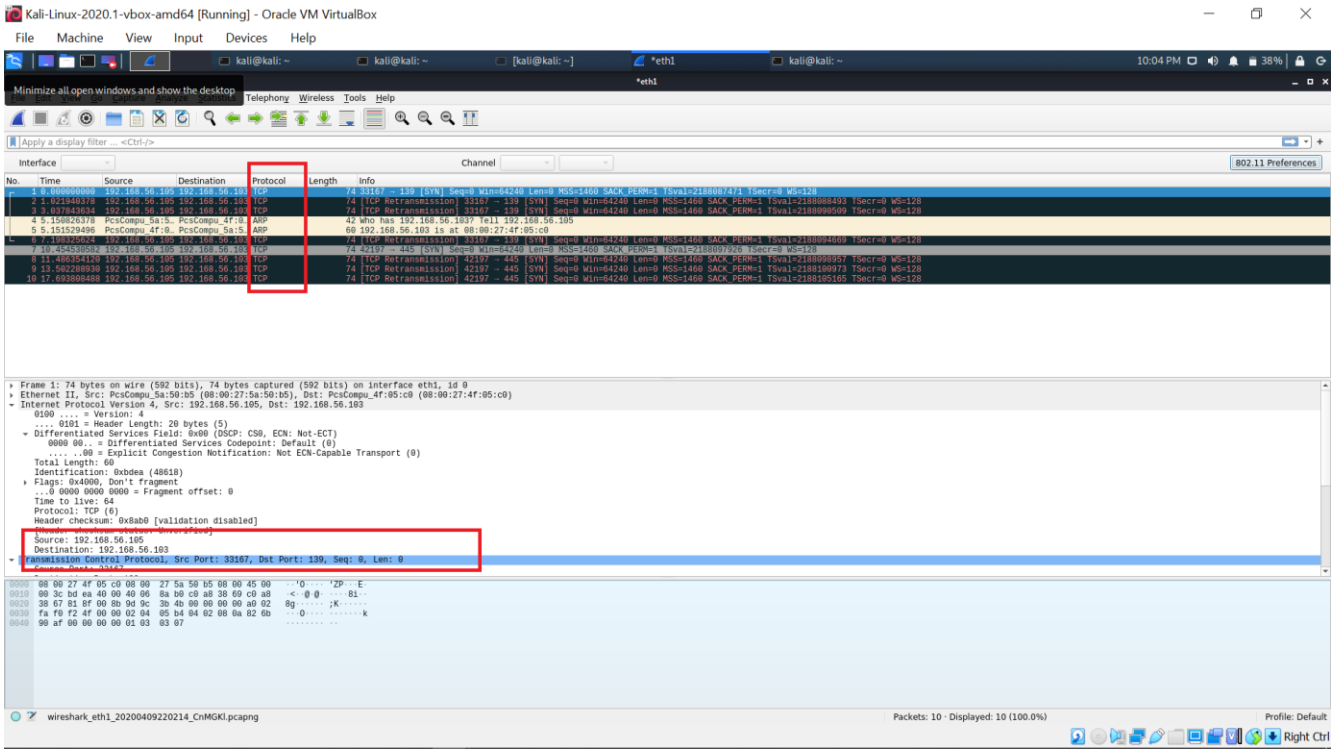
Running the exploit:



Packet of the attack are captured using Wireshark:
Figure bellow show the packets of the attack before adding the rule to IPtable (packet without Rule)

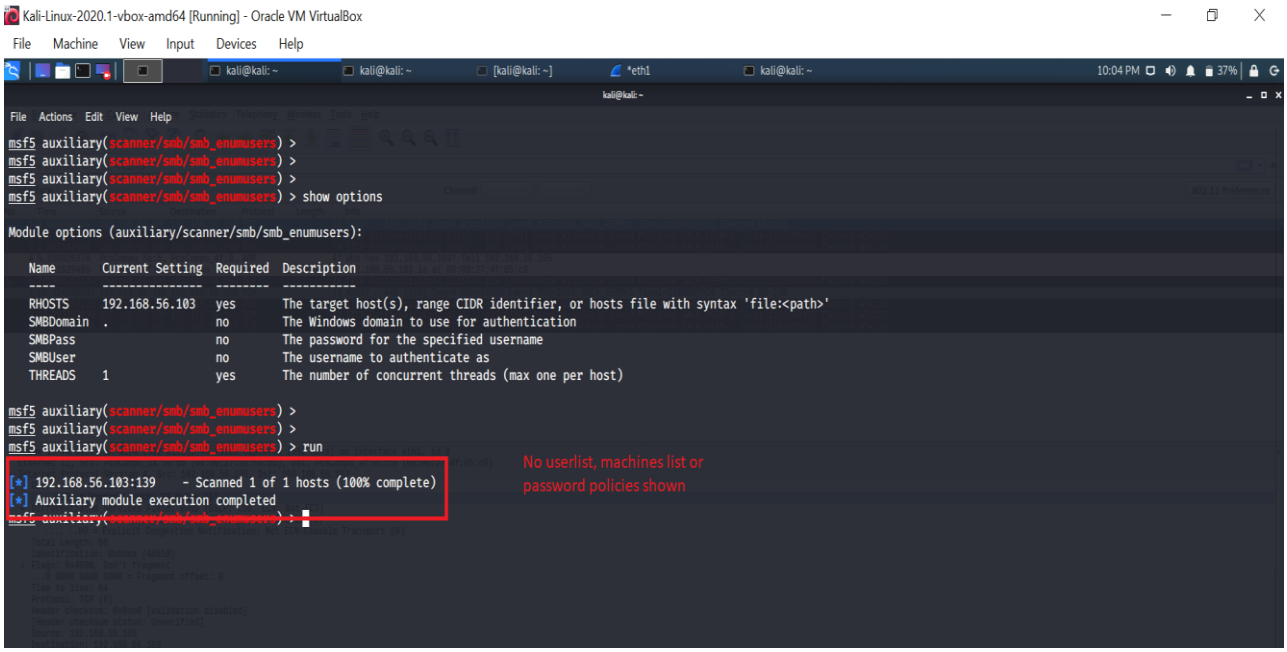


Packets captured after adding rule to IPtable



In the above screenshots, there are no SMB packets after adding the rule. Which says that attack was not successful.

Metasploit output:



3.By reviewing the scan results (obtained in project – Part 1), suggest and describe any additional defense strategy to protect the target systems.

By reviewing the Zenmap scans from project 1 and at the beginning of project 2, we could see that most of the critical services on the network were exposed and seemed to be using the default configurations. In addition, mailing services (SMTP, IMAP, and POP3) were using weak ciphers. FTP and SSH are possibly vulnerable to brute force attack, since there may not be any back-off mechanism configured with these services.

Using firewall, it is possible to protect many information inside the private network invisible to the external world. But for example, when somebody from inside organization need to have access to resources from outside, we need virtual private network (VPN). VPN lets user from outside of a private network to take a detour around the firewall and make it possible to have access to internal network. This happen through using combination of software and security measures. VPN works based on encryption of all the traffic entering and leaving by making encrypted tunnel between a user and remote server, so all date traffics are going to be routed through this tunnel. So, our data stay secure and out IP address and identity and location have mask. At the end the security is improved, and VPN improves security by offering confidentiality and integrity.

a. The first step is to have a strong firewall system. No doubt, antivirus is the first thing in mind when it comes to the cyber security. Enable a firewall or endpoint protection to protect these ports from attackers. Most solutions include a blacklist to prevent connections from known attackers IP addresses.

b. Keeping the system updated with all the recent versions or patches is extremely important to block the attacker from getting access to the system.

c. Creating strong snort rules as we did up in this report will help to prevent the organisation from intended attacker.

d. Install a VPN to encrypt and protect network traffic.

e. Implement VLANs to isolate internal network traffic.

f. Use MAC address filtering to keep unknown systems from accessing the network. This tactic requires significant management to keep the list maintained.

g. In addition to the network specific protections above, we can implement a data centric security plan to protect the most important resource – the data that lives on our SMB file shares.