Introduction

In this lab, we were going to exploit the Remote Executable Code vulnerability of the Samba version, then we leveraged it to get the root shell. Therefore, we utilized a Wireshark Command line, “tcpdump” to capture the packet before sending it to Suricata for detecting purposes.

Objective

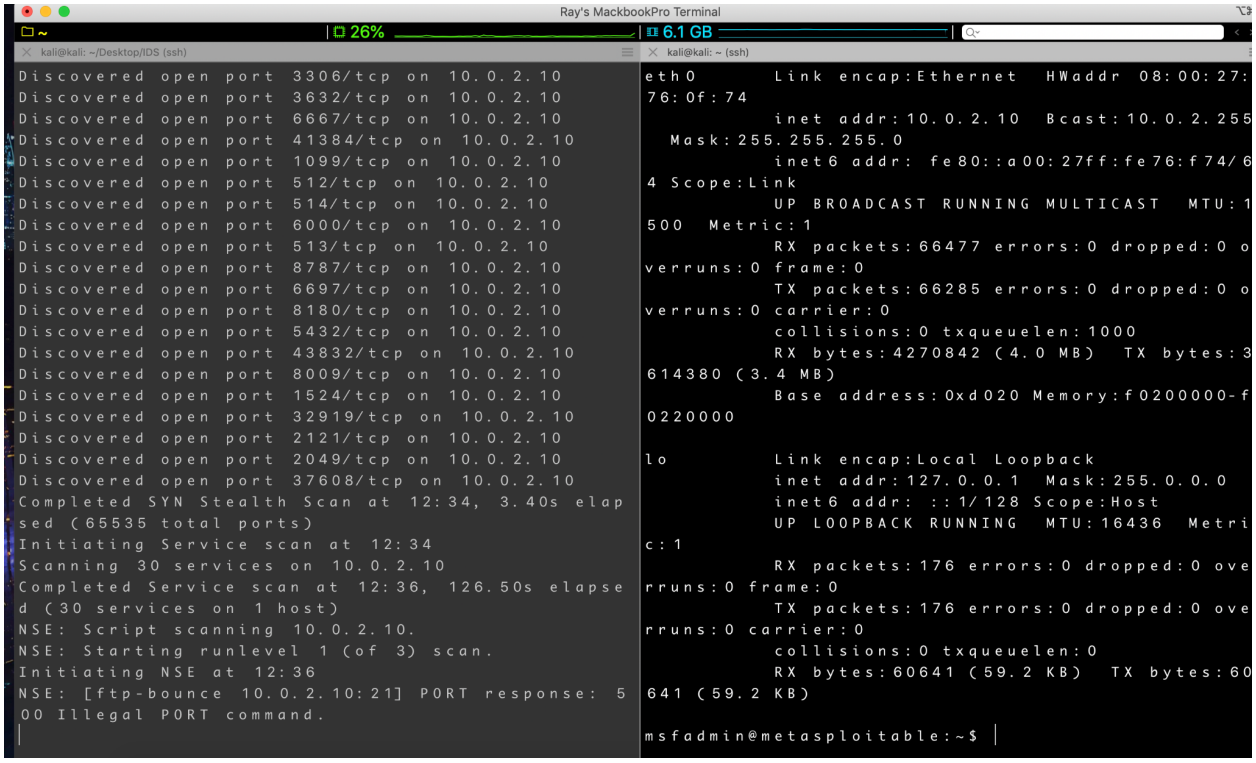
Task 1: Exploit the vulnerability in Samba

Kali IP Address: 10.0.2.8

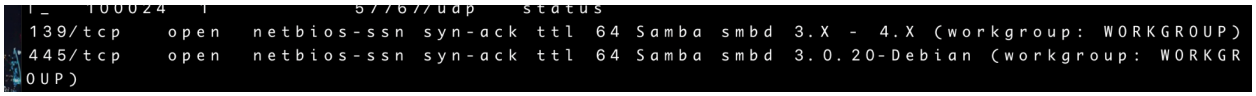
Metasploit IP Address: 10.0.2.10

Firstly, we scanned the port in the Metasploit server via “**sudo nmap -sC -sV -p- 10.0.2.10**”. Specifically, **-sC** means Nmap scan through the default script, **-sV** means Nmap scan the service version and **-p-** means Nmap scan all the ports from 1 - 65535.

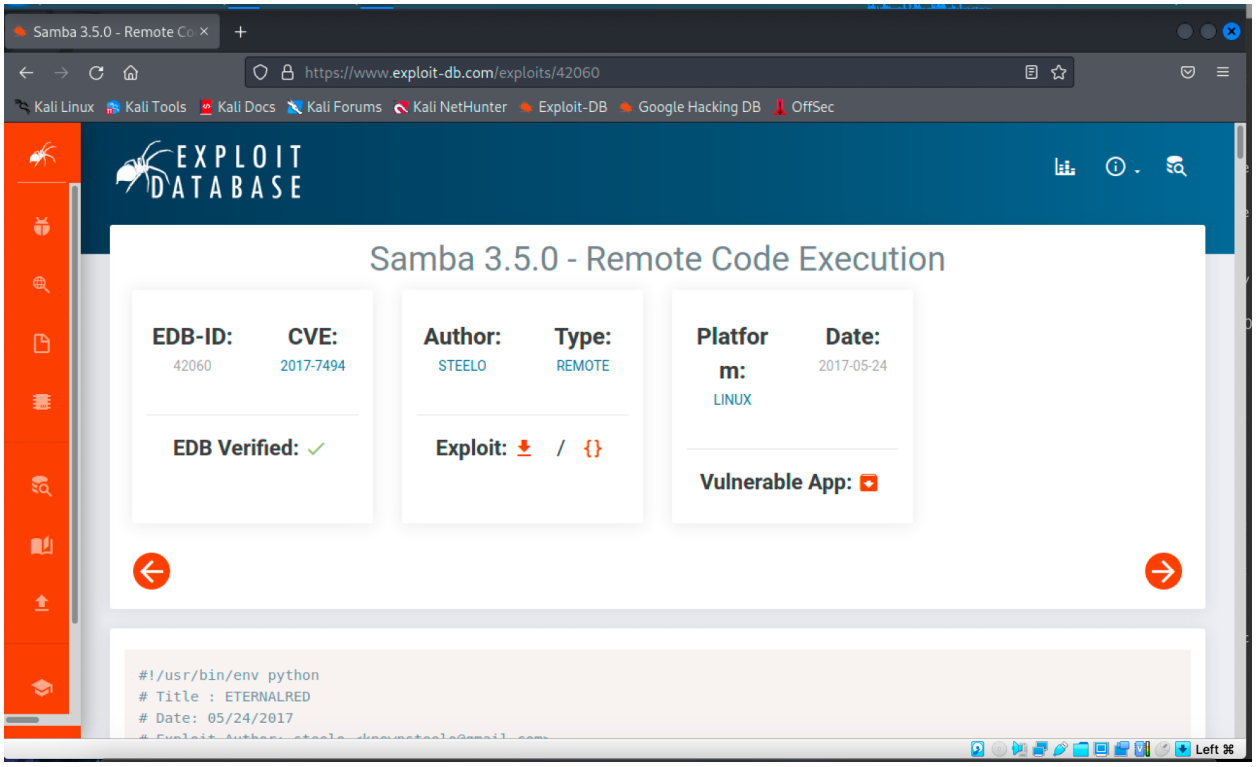
In terms of the below screenshots, the output of the command indicates that the samba ports 139 and 445 opened, and the version of those were **3.X - 4.X** and **3.0.20** respectively.



The differences between Samba port 139 and port 445 are that Samba port 139 is an older version, and it runs on NetBIOS for communicating with other machines on a Local Area Network (LAN). In contrast, port 445 is the latest version of samba, which uses for communication over the TCP Transport Layer. [1]



If we used ExploitDB to search the Samba version 3.x - 4.x, we knew it had an RCE vulnerability [2] [3] "**CVE-2007-2447**". Then, this vulnerability can be leveraged through the Metasploit exploit framework.



Starting Attack:

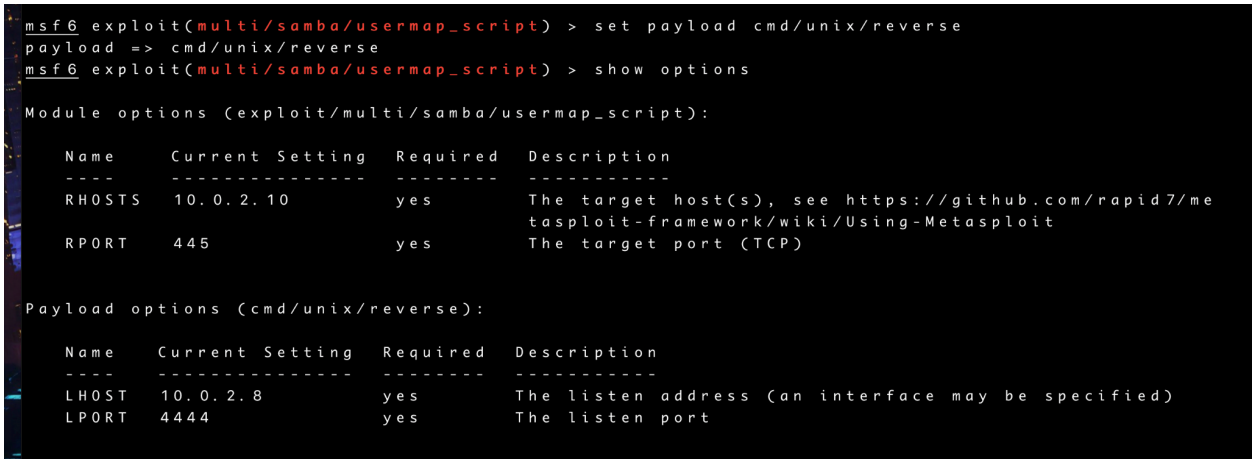
Firstly, we searched the samba through the Metasploit console and used the module of "**exploit/multi/samba/usermap_script**" for exploiting.

```
msf6 > search samba

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

   #    Name                                                  Disclosure Date   Rank       Check  Description
   -    ----                                                  ---------------   ----       -----  -----------
   0    exploit/unix/webapp/citrix_access_gateway_exec        2010-12-21        excellent  Yes    Citrix Access Gateway Command Execution
   1    exploit/windows/license/calicclnt_getconfig           2005-03-02        average    No     Computer Associates License Client GETCONFIG Overflow
   2    exploit/unix/misc/distcc_exec                         2002-02-01        excellent  Yes    DistCC Daemon Command Execution
   3    exploit/windows/smb/group_policy_startup              2015-01-26        manual     No     Group Policy Script Execution From Shared Resource
   4    post/linux/gather/enum_configs                                          normal     No     Linux Gather Configurations
   5    auxiliary/scanner/rsync/modules_list                                    normal     No     List Rsync Modules
   6    exploit/windows/fileformat/ms14_060_sandworm          2014-10-14        excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   7    exploit/unix/http/quest_kace_systems_management_rce   2018-05-31        excellent  Yes    Quest KACE Systems Management Command Injection
   8    exploit/multi/samba/usermap_script                    2007-05-14        excellent  No     Samba "username map script" Command Execution
   9    exploit/multi/samba/nttrans                           2003-04-07        average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10   exploit/linux/samba/setinfopolicy_heap                2012-04-10        normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
```

Setting the payload of the basic reverse shell "**cmd/unix/reverse**," the Remote Hosts (RHOSTS) is the Metaploitable Server "**10.0.2.10**" and the Remote Port (RPORT) is the Samba version either **139** or **445**.

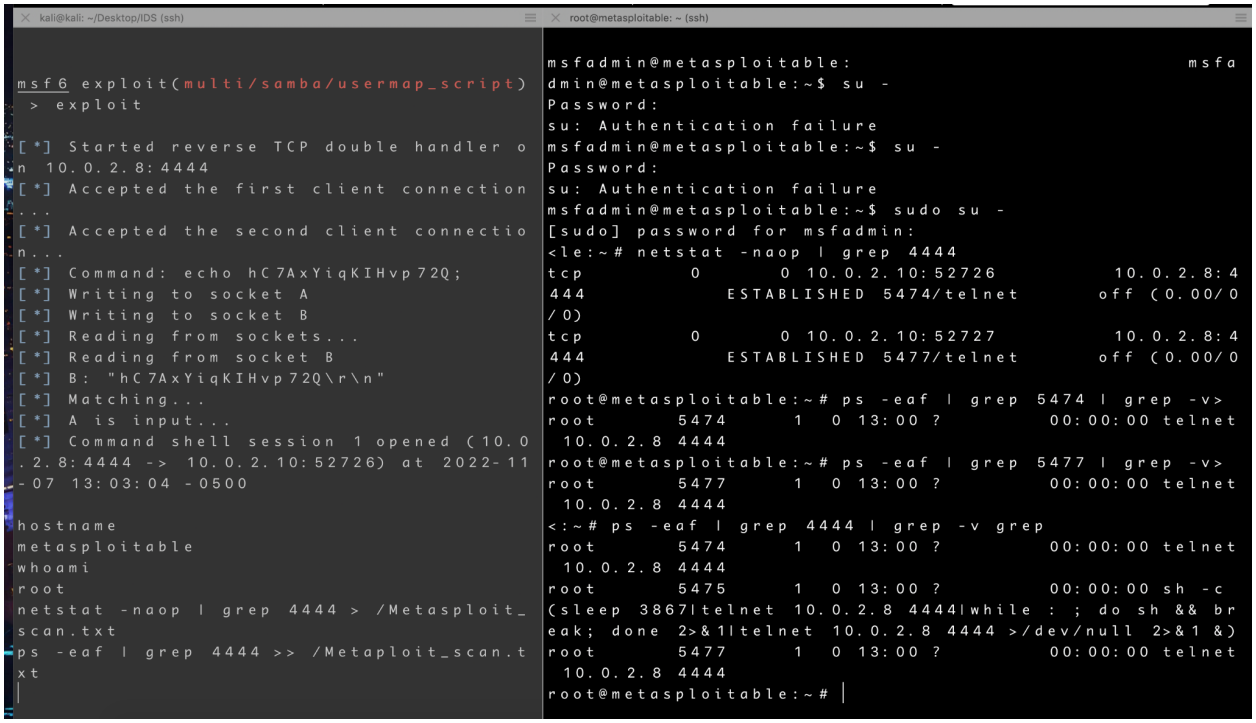


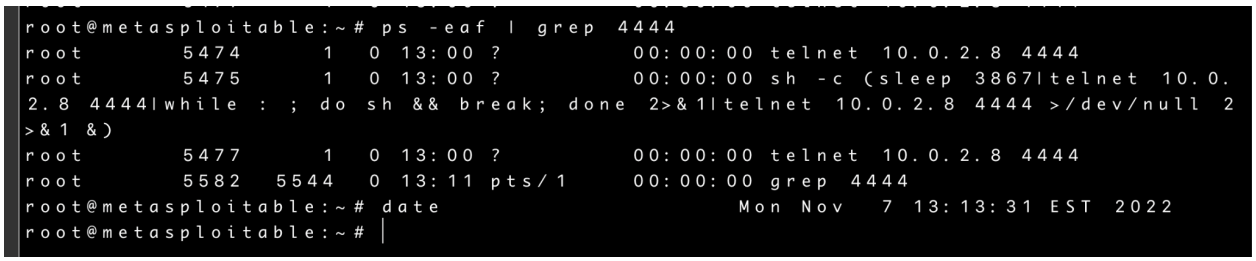
Expected Result:

Once the configuration was correct, we can start to exploit. The result of that shows that we got the root shell.

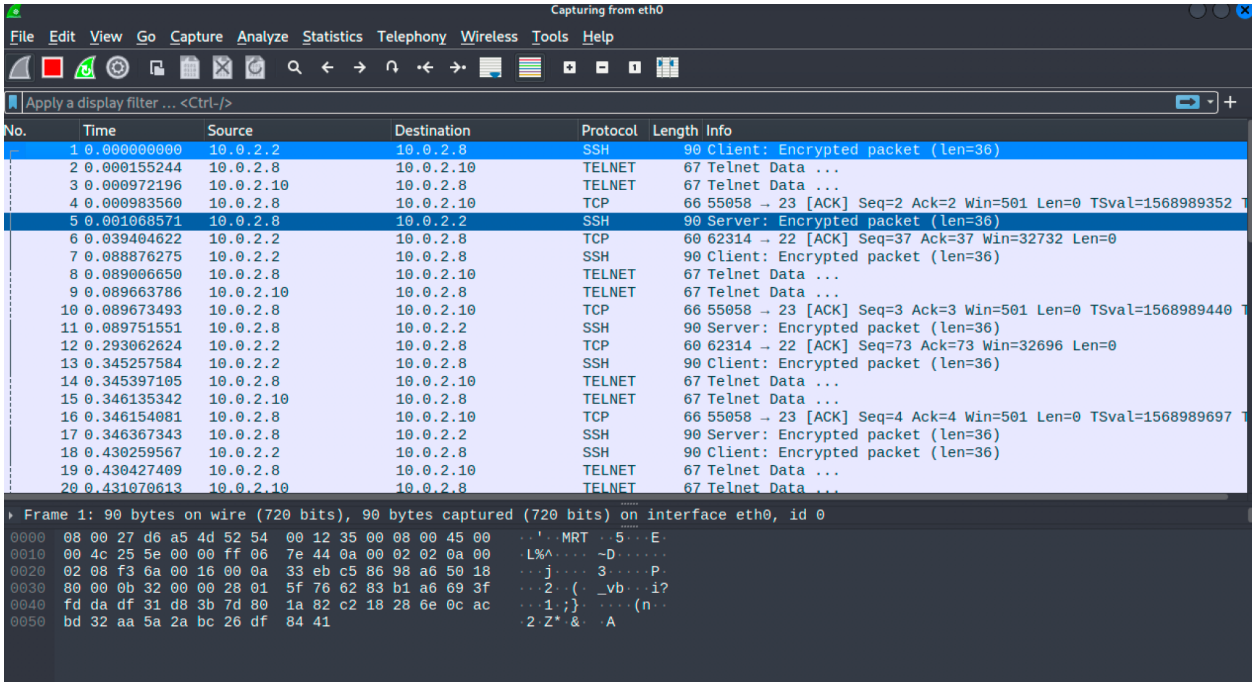
Analyzing Part:

Based on the two screenshots below, we can see that port 4444 connects to Metaploitable Server.



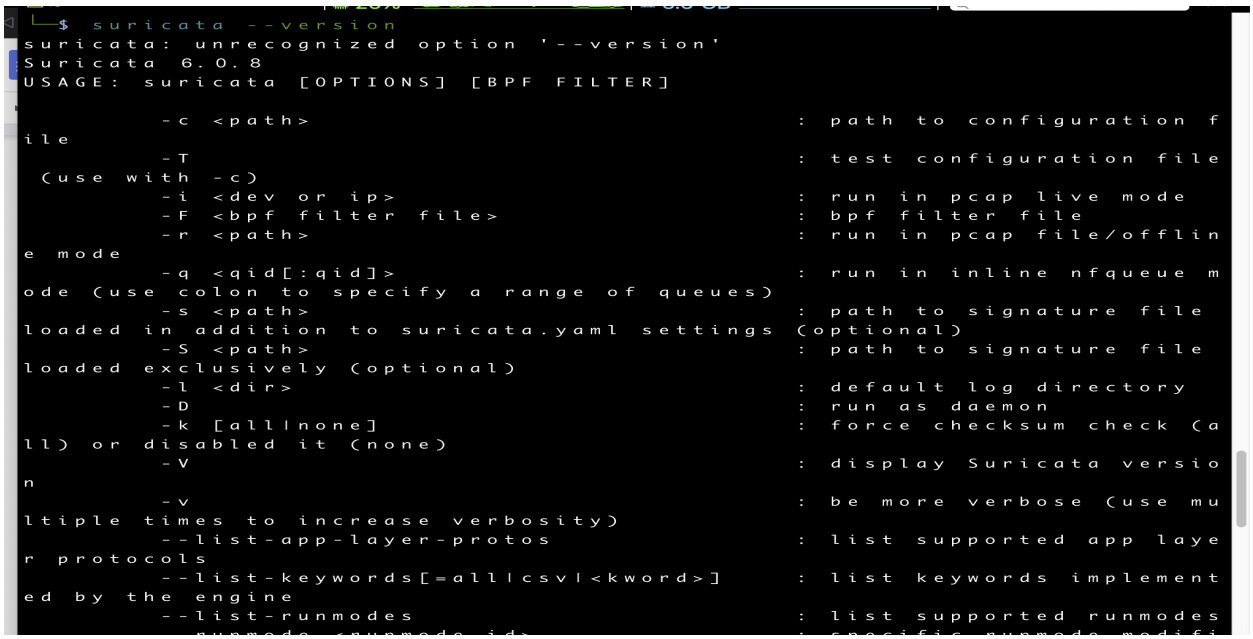


As to the below screenshot, we captured the packet via Wireshark on the Attack machine, which indicates that the communication was using the “telnet” application protocol.

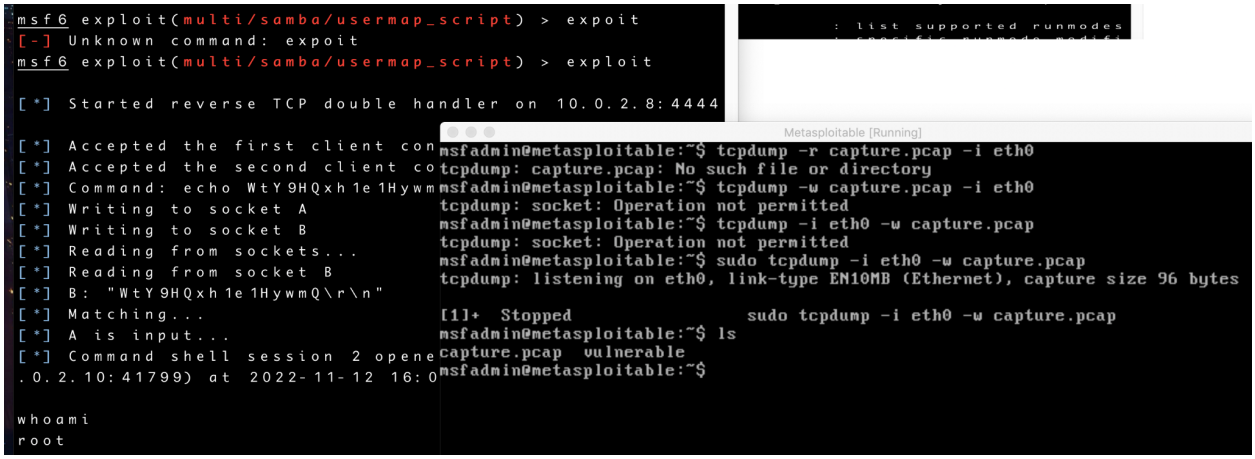


Task 2: Detect the attack

In Task 2, we need to set up the Suricata based on IDS/IPS on the Attack machine [4] [5].

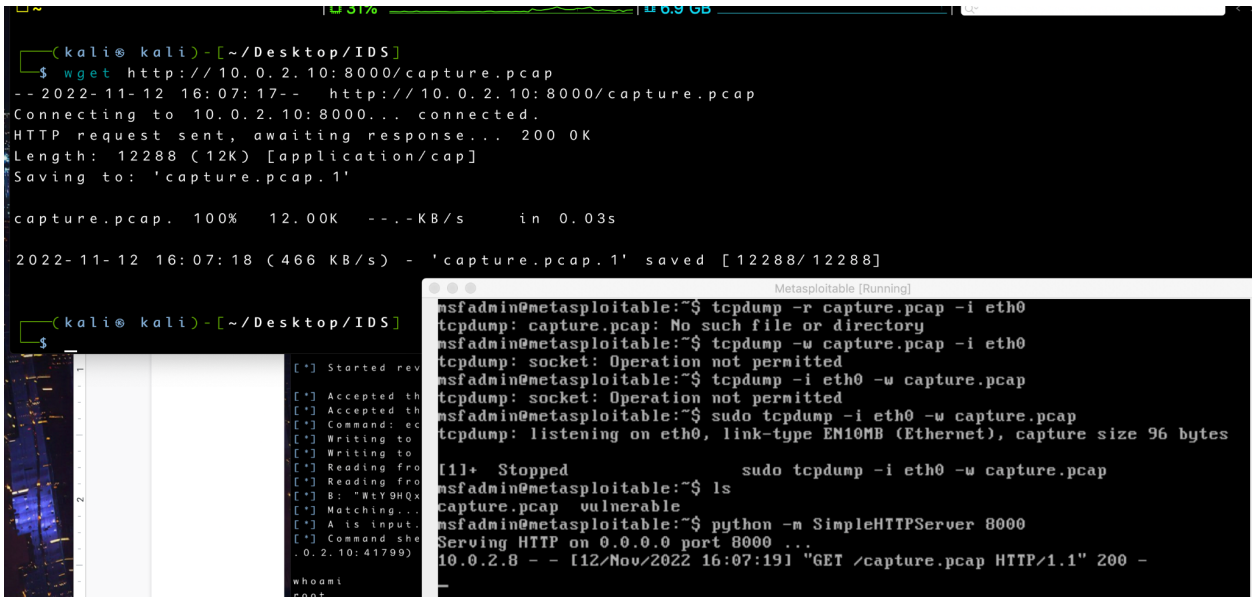


Then, we need to do the exploiting again in Task1 to capture the packet.



Then, we used "**tcpdump**" to capture the packet from the Attack machine, which was "**tcpdump -i eth0 -w capture.pcap**".

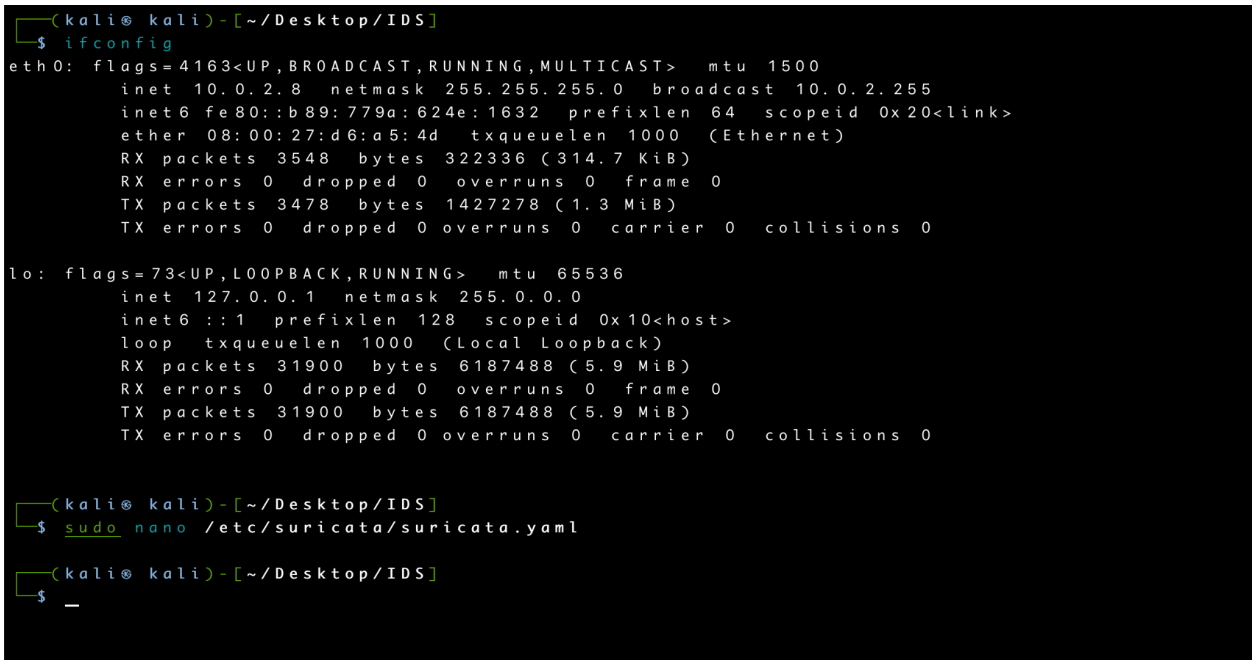
After that, we set up the Python server with port 8000 "**python -m SimpleHTTPServer 8000**" on the Metasploitalbe server so that we can download the "**capture.pcap"** file from the server.



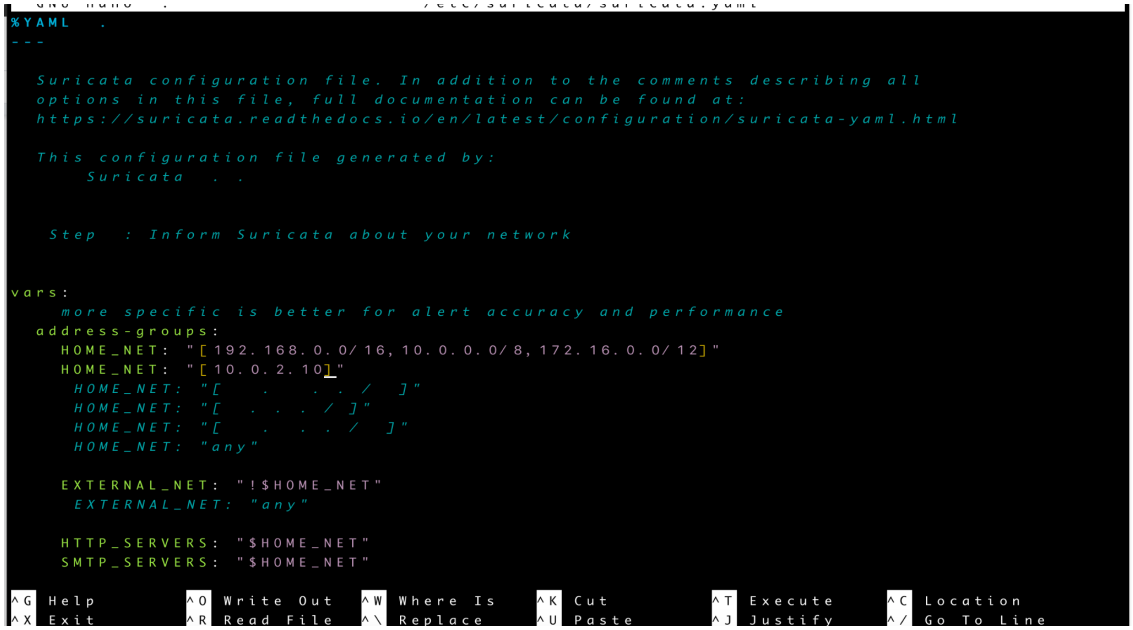
In the following screenshots, we are going to write some Suricata rules, then send the captured file to the Suricata system for detection.

Suricata Detecting:

As the below screenshot, we know that our Attack machine IP is “10.0.2.8” and the Target machine IP is “10.0.2.10”, we added to the Suricata YAML configuration file with **HOME_NET “[10.0.2.10]” [5] [6].**



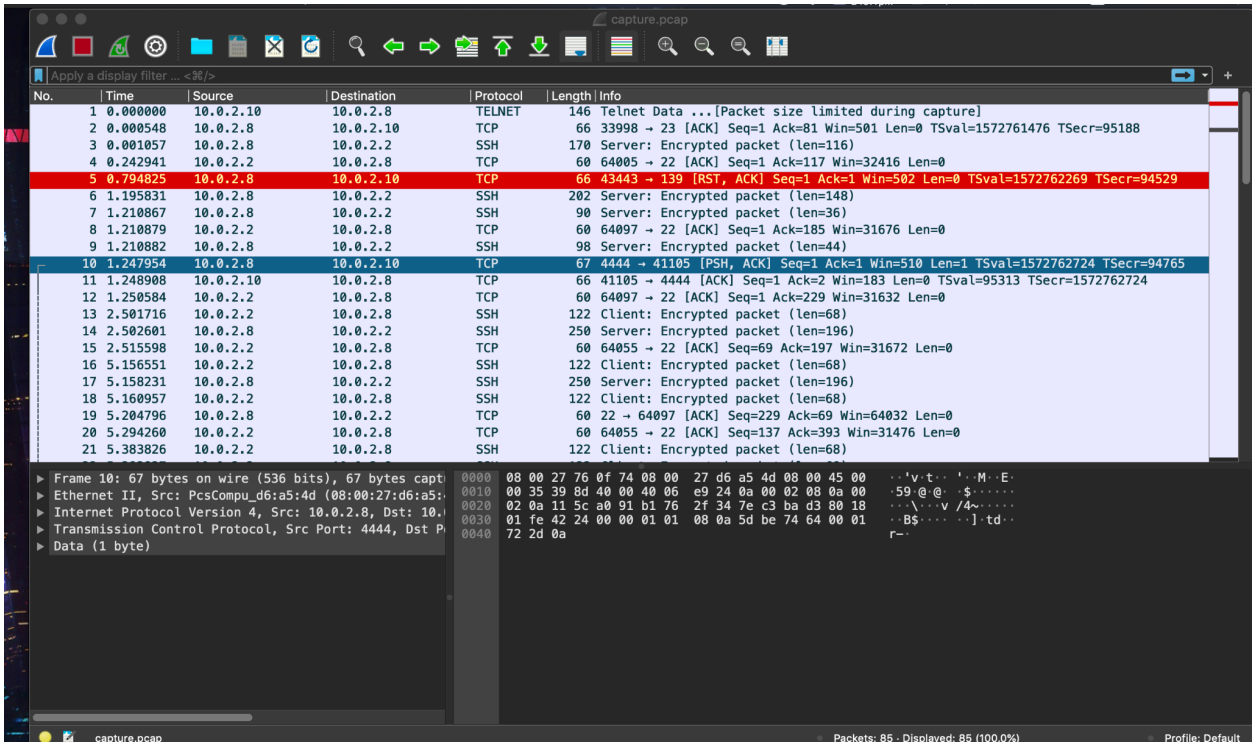
HOME_NET: “[10.0.2.10]”
EXTERNAL_NET: “!$HOME_NET” indicates that the Suricata will detect our of home IP address.



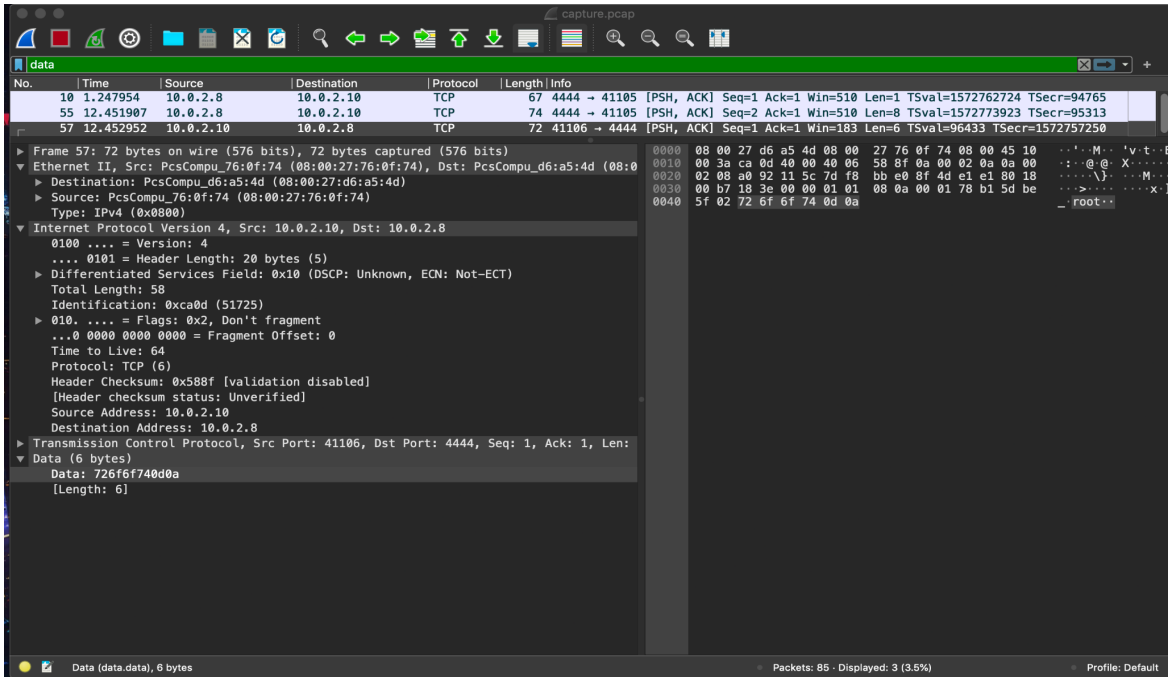
As for the **IDS-Task2.rules** configuration file, we added the following rules for detecting [7] before adding this rule of the file to **Suricata.yaml** configuration file.

Using Suricata to read the PCAP file with configured rules to see whether it is the attack [8] [9].

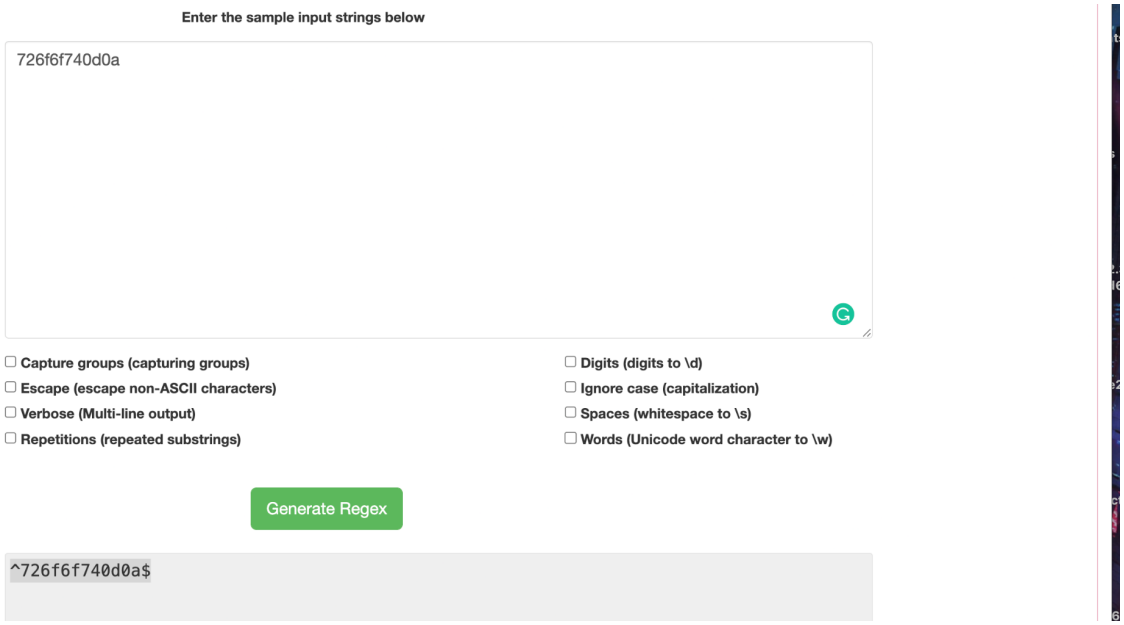
Based on the "**capture.pcap**" file, we need to generalize the packet via regular expression and write the specific rule for detecting communication.



According to the below screenshot, we found the packet that indicates the attack got the root shell, and the data is **726f6f740d0a**



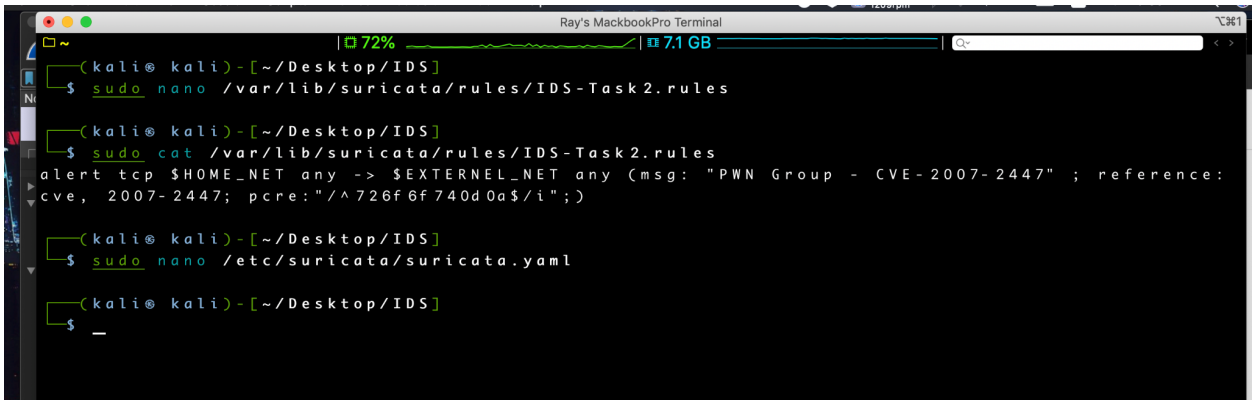
If we use a regular expression online generator, the data **726f6f740d0a** will be **^726f6f740d0a$** [10].



Based on the information we got, the detecting rule is as follows [11]:

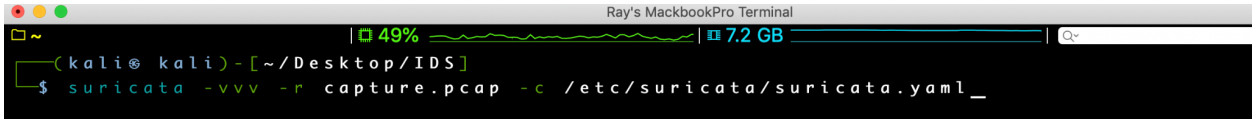
alert tcp $HOME_NET any -> $EXTERNEL_NET any (msg: "PWN Group - CVE-2007-2447"; reference:cve, 2007-2447; pcre:"/^766f6f740d0a$/i";)

This means that the communication is from target machine 10.0.2.10 to attack machine "10.0.2.8" if the Suricata detects the attack, the message will show the CVE-2007-2447.

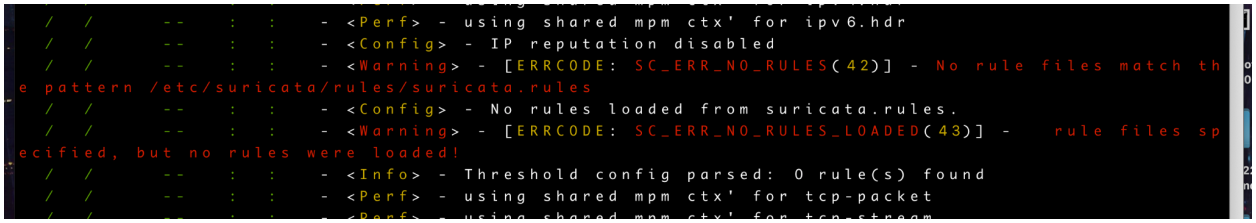


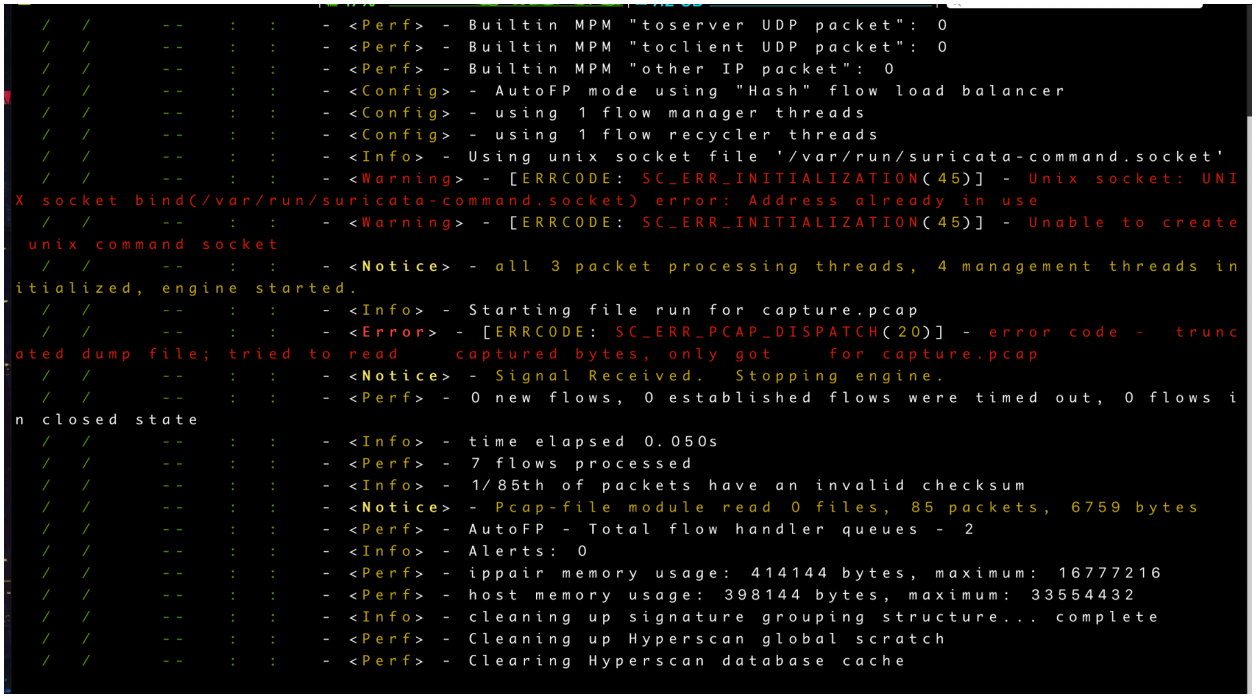
Start the Suricata service to read the capture.pcap file with Suricata Yaml configuration files.

“**suricata -vvv -r capture.pcap -c /etc/suricata/suricata.yaml”**



The result of Suricata detection is as below. The configured rules did not load into the Suricata Yaml files, and Suricata detected no alerts.





The expected result is that the Suricata is supposed to detect the CVE-2007-2447 attack, but I still need to troubleshoot the Suricata versions and their dependencies [12].

Conclusion

To summarize what we have done with this LAb, it was quite easy to perform in Task1 by exploiting Samba REC vulnerabilities and leveraging it to get the root shell while we were struggling with troubleshooting the Suricata version in our virtual machines, so we did lots of research. Therefore, the Suricata rule was straightforward if we captured the packet between the attack and target machines and know how to add regular expressions in the Suricata rule.

Reference

1. M. Buckbee, “What is an SMB port + ports 445 and 139 explained,” *Varonis*, 28-Jun-2022. [Online]. Available: https://www.varonis.com/blog/smb-port. [Accessed: 12-Nov-2022].

2. “CVE-2007-2447,” *CVE*. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2447. [Accessed: 12-Nov-2022].

3. Offensive-Security, “Offensive-security/exploitdb: The legacy exploit database repository - new repo located at https://gitlab.com/exploit-database/exploitdb,” *GitHub*. [Online]. Available: https://github.com/offensive-security/exploitdb. [Accessed: 12-Nov-2022].

4. “2. quickstart guide¶,” *2. Quickstart guide - Suricata 7.0.0-beta1 documentation*. [Online]. Available: https://suricata.readthedocs.io/en/latest/quickstart.html#running-suricata. [Accessed: 12-Nov-2022].

5. “How to configure Suricata as an intrusion prevention system (IPS) on ubuntu 20.04,” *DigitalOcean*, 09-Dec-2021. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-configure-suricat

a-as-an-intrusion-prevention-system-ips-on-ubuntu-20-04. [Accessed: 12-Nov-2022].

6. *Installing and configuring SURICATA rules*. [Online]. Available: https://nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-suricata-rules.htm#:~:text=Suricata%20rules%20are%20the%20de facto,threat%20intelligence%20against%20network%20traffic.&text=This%20rule%20consists%20of%20a,%2D%3E%20%24EXTERNAL_NET%20any%22). [Accessed: 13-Nov-2022].

7. *Suricata Rules - Suricata 6.0.0 documentation*. [Online]. Available: https://suricata.readthedocs.io/en/suricata-6.0.0/rules/. [Accessed: 13-Nov-2022].

8. MikeSchemMikeSchem 8811313 silver badges2727 bronze badges and JufajardiniJufajardini 3155 bronze badges, "How to run suricata on PCAP mode and get results in fast.log," *Stack Overflow*, 01-Jun-1967. [Online]. Available: https://stackoverflow.com/questions/61132410/how-to-run-suricata-on-pcap-mode-and-get-results-in-fast-log. [Accessed: 13-Nov-2022].

9. "Suricata & Iptables cheatsheet," *HackTricks*. [Online]. Available: https://book.hacktricks.xyz/generic-methodologies-and-resources/basic-fore

nsic-methodology/pcap-inspection/suricata-and-iptables-cheatsheet. [Accessed: 14-Nov-2022].

10. “Online regex generator tool: JavaInUse,” *Tool*. [Online]. Available: https://www.javainuse.com/rexgenerator. [Accessed: 14-Nov-2022].

11. “Suricata,” *Pcre (Perl Compatible Regular Expressions) - Suricata - Open Information Security Foundation*. [Online]. Available: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Pcre_(Perl_Compatible_Regular_Expressions). [Accessed: 14-Nov-2022].

12. https://www.reddit.com/r/OPNsenseFirewall/comments/i46tda/suricata_is_not_loading_rules_error/