SOC Project (Shadow Sentry)

Prepared By:
Michelle Lai

CFC020823

S23

PROJECT OBJECTIVES

1 Deploy Elastic Cloud on DigitalOcean

- Step-by-step guide for deploying Elastic Cloud on DigitalOcean.
- Instructions for configuring the Elastic Stack components.
- Integration with Elastic Stack.
  - Configure Logstash to ingest logs from the sample infrastructure.
  - Integrate Elasticsearch for storing and indexing logs.

2 Choose Honeypot Solution(s)

- Choose a honeypot solution.
- Possible Honeypots: Cowrie, Honeyd, Glastopf, Kippo.
- Configure network settings.
  - Network Configuration: Ensure that the honeypot server is configured to listen on the desired network interfaces and ports. You may want to simulate common services such as SSH, Telnet, FTP, HTTP, or SMB.

PROJECT OBJECTIVES

3 Pentest Scripts

- Create attack scripts that can simulate at least three (3) different attack types using functions.
- Each attack should have a description to display once chosen.
- The system should display the IP addresses on the network.
- Display a list of all possible attacks with descriptions.
- The user can choose a specific attack or random from the list.
- If the user enters a different key, display a message and exit.
- For each attack, allow the user to choose a target or random from the found IPs.
- Everything other than the user input should be automated.
- Use functions.
- Possible tools: Nmap, Hydra, Masscan, Msfconsole, Hping3, Arpspoof etc.

PROJECT OBJECTIVES

4 Testing, Validation and Logging

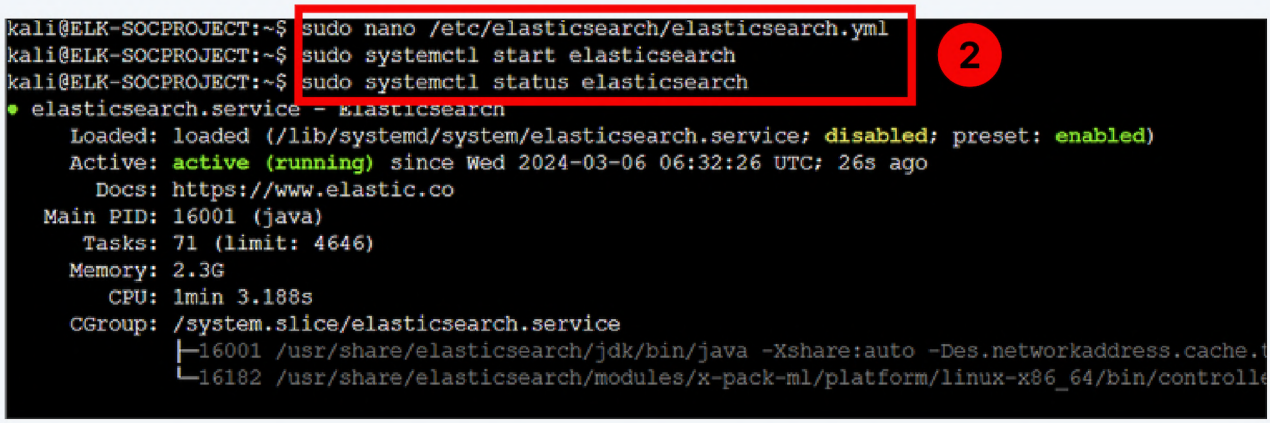
- Execute penetration testing scripts against the sample infrastructure.
- Report on the execution of pentest scripts and observed security events.
- Analysis of the effectiveness of the alerting and monitoring system.

METHODOLOGIES

1. Deploy Elastic Cloud on DigitalOcean

- **Installing and Configuring Elasticsearch**

The ELK (Elasticsearch, Logstash & Kibana) stack provides a framework to collect, store, and investigate network security data. It is used to solve problems such as log analytics, document search, security information and event management (SIEM), and observability .



Step 1:

- *sudo apt install elasticsearch* -> Install Elasticsearch.

Step 2:

- *sudo nano /etc/elasticsearch/elasticsearch.yml* -> Allows configuration options for your cluster, node, paths, memory, network, discovery, and gateway.
- *sudo systemctl start elasticsearch* -> Starts elasticsearch.
- *sudo systemctl status elasticsearch* -> Checks the status of the elasticsearch.

METHODOLOGIES

1. Deploy Elastic Cloud on DigitalOcean

- **Installing and Configuring Elasticsearch**

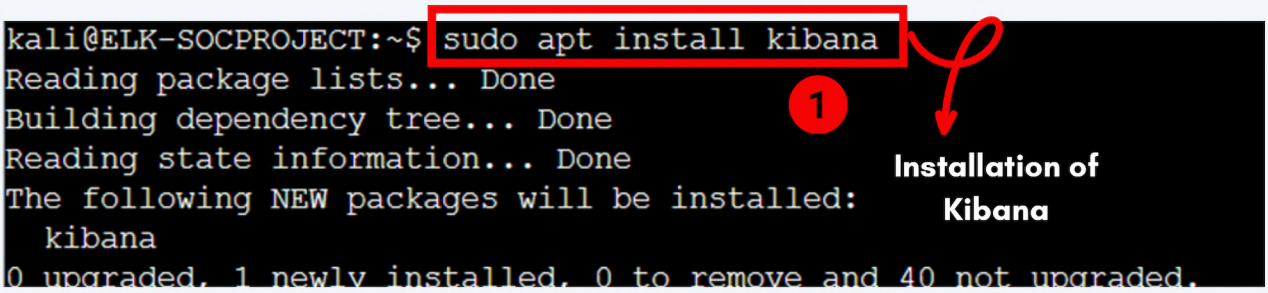
Step 3:

- *curl -X GET "localhost:9200"* -> Test if elasticsearch is running by sending an HTTP request.
- Based on the response, Elasticsearch is up and running.

METHODOLOGIES

1. Deploy Elastic Cloud on DigitalOcean

- **Installing and Configuring Kibana**



Step 1:

- *sudo apt install kibana* -> Install Kibana.

Step 2:

- *sudo systemctl start kibana* -> Starts kibana.
- *sudo systemctl status kibana* -> Checks the status of the kibana.

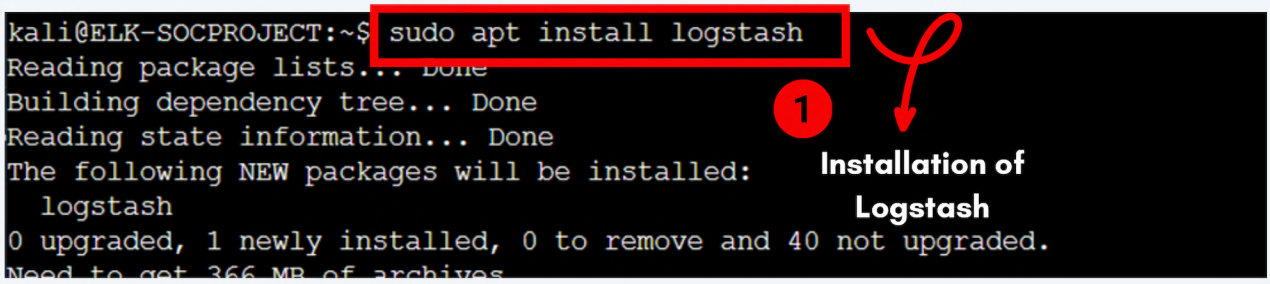
Step 3:

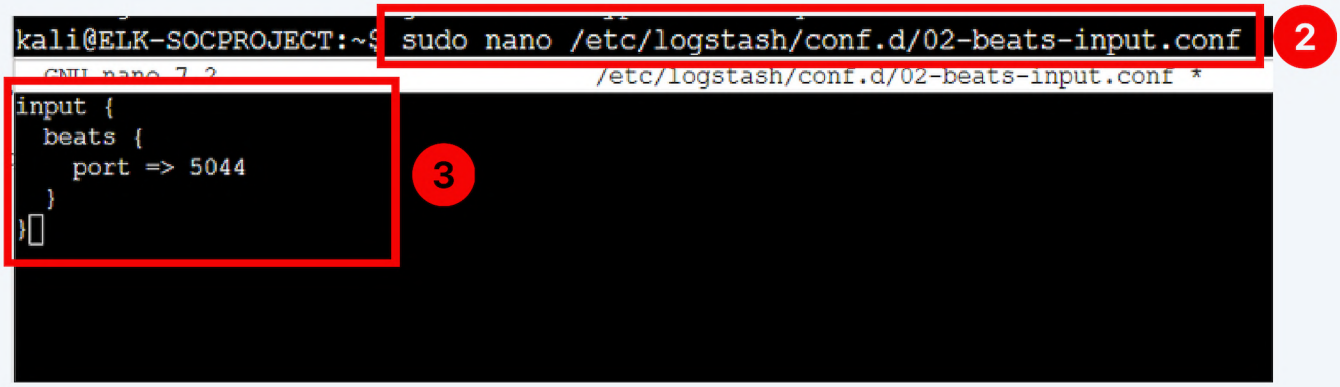
- *echo "kibanaadmin:`openssl passwd -apr1`" | sudo tee -a /etc/nginx/htpasswd.users* -> Create an administrative kibana user which is used to access the kibana web interface and store the passwords in htpasswd.users file.

METHODOLOGIES

1. Deploy Elastic Cloud on DigitalOcean

- **Installing and Configuring Logstash**





Step 1:

- sudo apt install logstash -> Install Logstash.

Step 2:

- *sudo nano /etc/logstash/conf.d/02-beats-input.conf* -> Set up a filebeat input.

Step 3:

- Specifies for the beats to listen on TCP port 5044.

METHODOLOGIES

1. Deploy Elastic Cloud on DigitalOcean

- **Installing and Configuring Logstash**

4

```
kali@ELK-SOCPROJECT:~$ sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

```
output {
  if [@metadata][pipeline] {
        elasticsearch {
        hosts => ["localhost:9200"]
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        pipeline => "%{[@metadata][pipeline]}"
        }
  } else {
        elasticsearch {
        hosts => ["localhost:9200"]
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        }
  }
}
```

5

```
kali@ELK-SOCPROJECT:~$ sudo systemctl start logstash
kali@ELK-SOCPROJECT:~$ sudo systemctl enable logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstas
.
```

6

Step 4:

- *sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf* -> Create a configuration file to store the Beats data in Elasticsearch.

Step 5:

- Configures Logstash to store the Beats data in Elasticsearch, which is running at localhost:9200, in an index named after the Beat used.

Step 6:

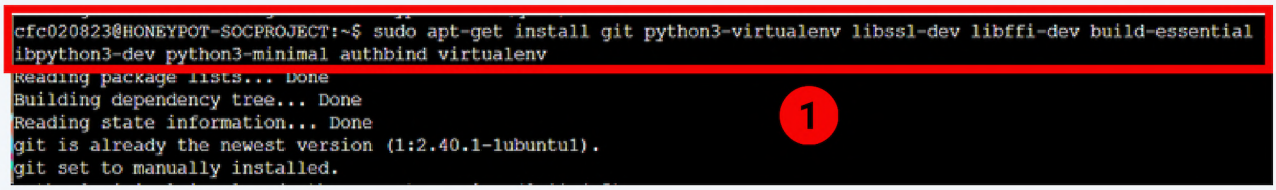
- *sudo systemctl start logstash* -> Starts logstash.
- *sudo systemctl enable logstash* -> Enable logstash to start up every time your server boots.

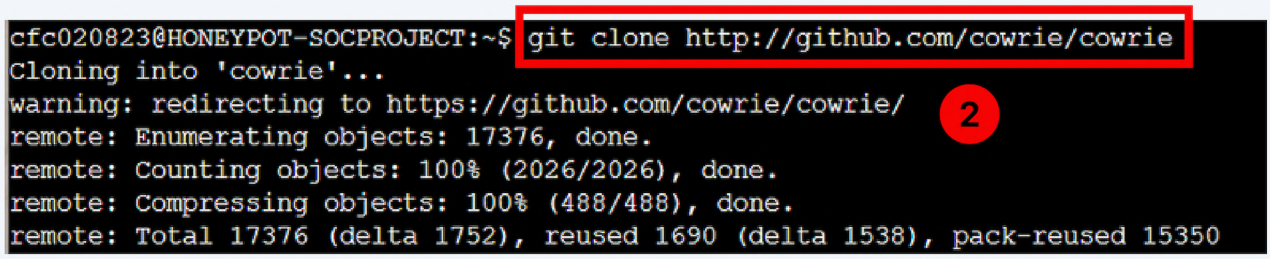
METHODOLOGIES

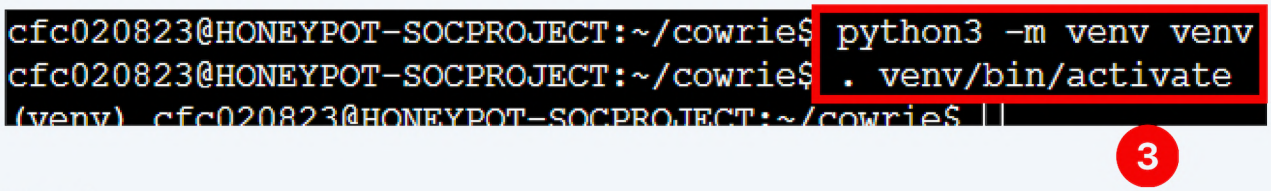
2. Honeypot Solution

- **Installing and Configuring Cowrie**

Cowrie is a SSH and Telnet honeypot which is designed to log brute force attacks and shell interaction with the attacker. It emulate a UNIX system in Python, or function as an SSH and telnet proxy to observe the attacker's behavior on the target system.







Step 1:

- *sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv* -> Install dependencies.

Step 2:

- *git clone http://github.com/cowrie/cowrie* -> Copy repository to local terminal.

Step 3:

- *python3 -m venv venv* -> Create virtual environment.
- *. venv/bin/activate* -> Activate the virtual environment.

METHODOLOGIES

2. Honeypot Solution

- **Installing and Configuring Cowrie**

```
cfc020823@HONEYPOT-SOCPROJECT:~/cowrie$ sudo touch /etc/authbind/byport/22
[sudo] password for cfc020823:
cfc020823@HONEYPOT-SOCPROJECT:~/cowrie$ sudo chown cowrie:cowrie /etc/authbind/byport/22
cfc020823@HONEYPOT-SOCPROJECT:~/cowrie$ sudo chmod 770 /etc/authbind/byport/22
(venv) cfc020823@HONEYPOT-SOCPROJECT:~/cowrie/bin$ nano cowrie
(venv) cfc020823@HONEYPOT-SOCPROJECT:~/cowrie/bin$ cd
(venv) cfc020823@HONEYPOT-SOCPROJECT:~$ sudo touch /etc/authbind/byport/23
(venv) cfc020823@HONEYPOT-SOCPROJECT:~$ sudo chown cowrie:cowrie /etc/authbind/byport/23
(venv) cfc020823@HONEYPOT-SOCPROJECT:~$ sudo chmod 770 /etc/authbind/byport/23
```

4

```
  GNU nano 7.2                                    cowrie.cfg
[telnet]
enabled = true
listen_endpoints = tcp:23:interface=0.0.0.0

[ssh]
listen_endpoints = tcp:22:interface=0.0.0.0
```

5

```
(venv) cfc020823@HONEYPOT-SOCPROJECT:~/cowrie$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using activated Python virtual environment "/home/cfc020823/cowrie/venv"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.
...
/home/cfc020823/cowrie/venv/lib/python3.11/site-packages/twisted/conch/ssh/transport
```

6

Step 4 & 5:

- Allow default SSH and Telnet listening port.

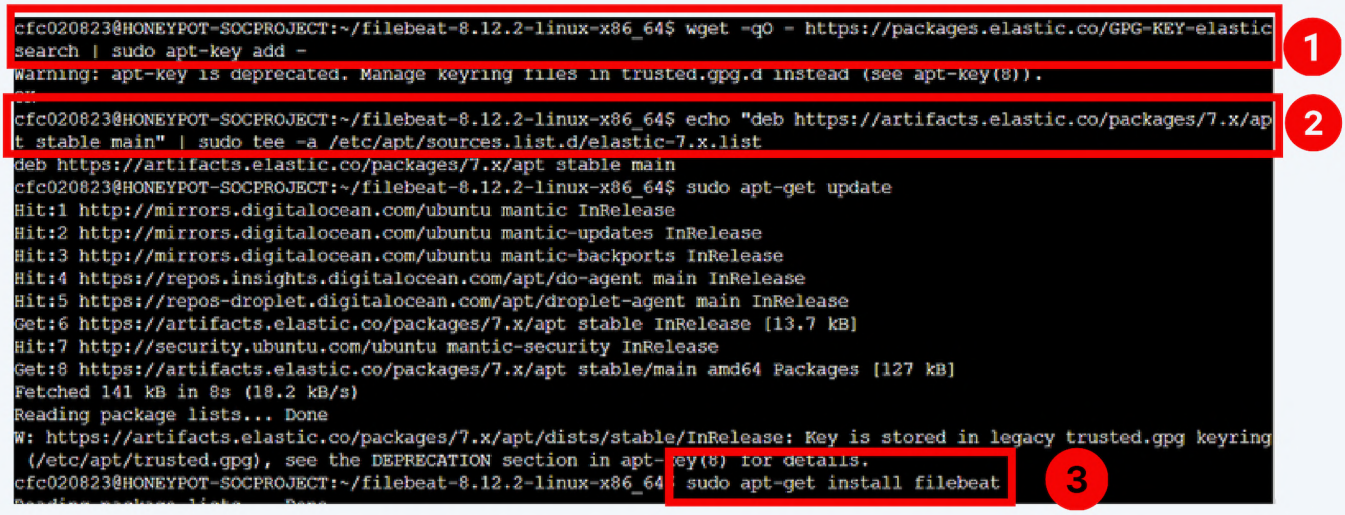
Step 6:

- *bin/cowrie start* -> Start cowrie.

METHODOLOGIES

2. Honeypot Solution

- **Installing and Configuring Filebeat**



Step 1:

- *wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -* -> Add the Elastic Stack key.

Step 2:

- *echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list* -> Add the Elastic Stack 7 Apt repository.

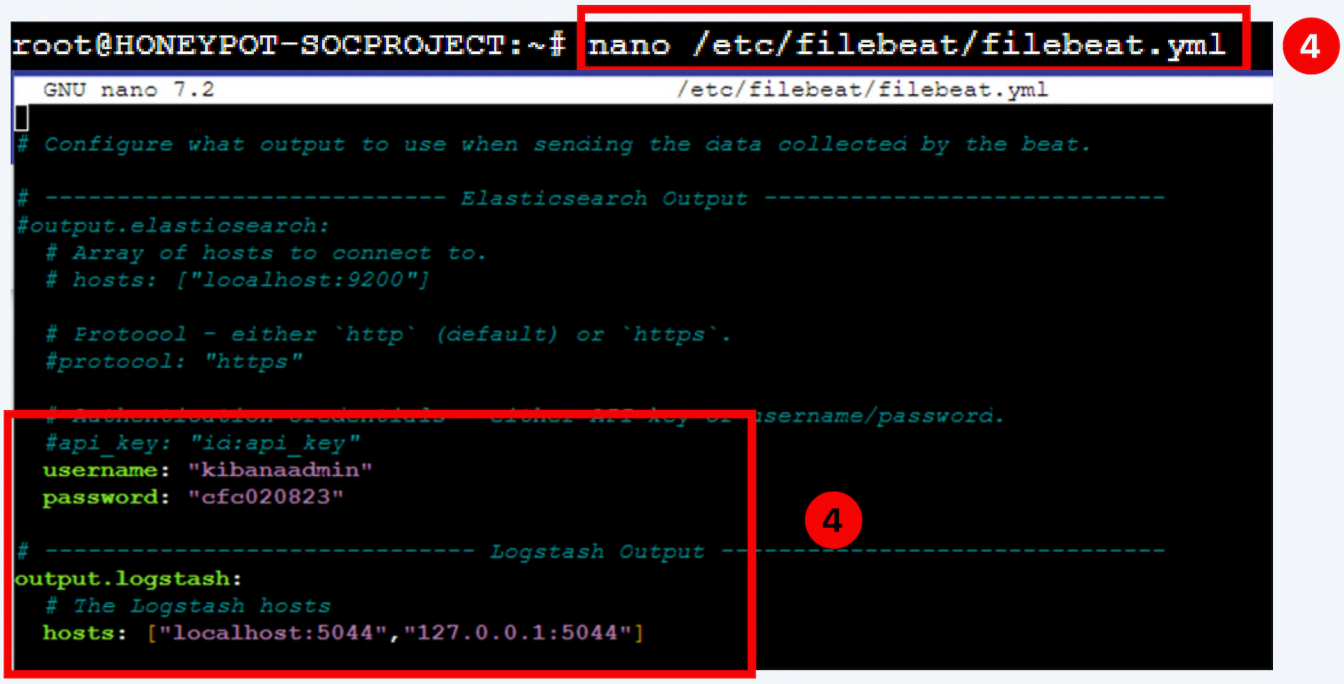
Step 3:

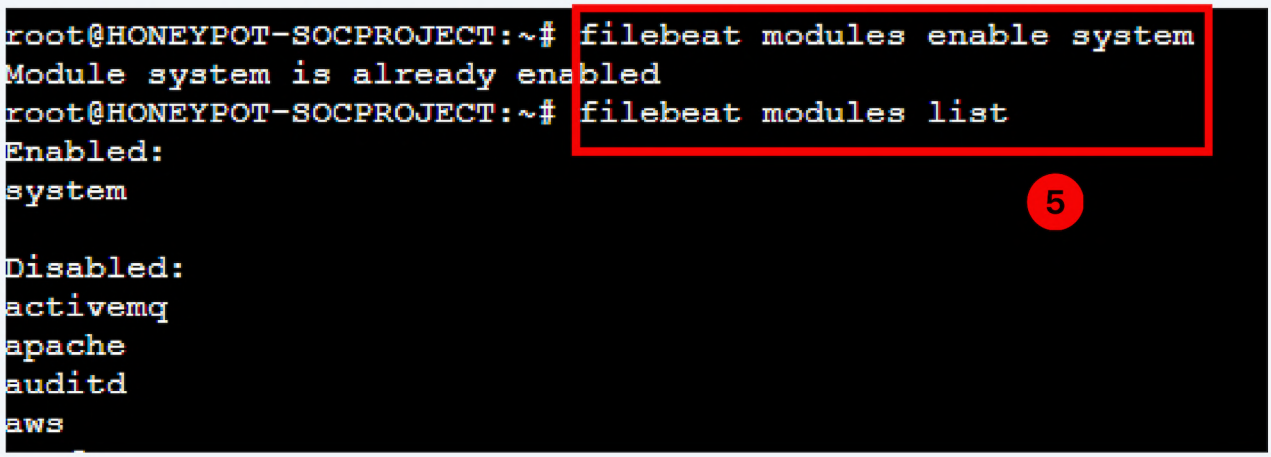
- *sudo apt-get install filebeat* -> Install filebeat.

METHODOLOGIES

2. Honeypot Solution

- **Installing and Configuring Filebeat**





Step 4:

- Configure Filebeat to send event data to Logstash.

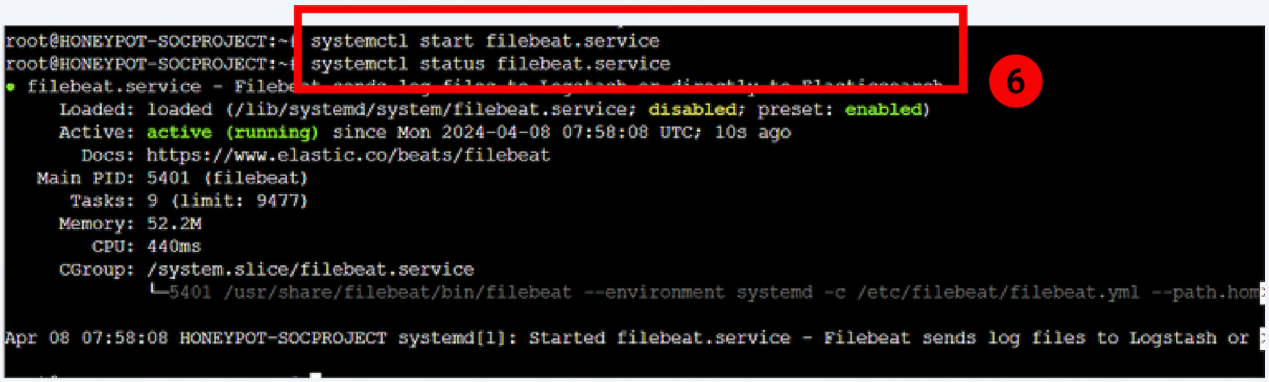
Step 5:

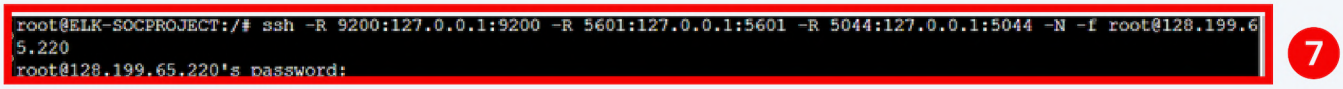
- Enable the system module to collects and parses logs created by the system logging service.

METHODOLOGIES

2. Honeypot Solution

- **Installing and Configuring Filebeat**





Step 6:

- Start and check the status of the filebeat service.

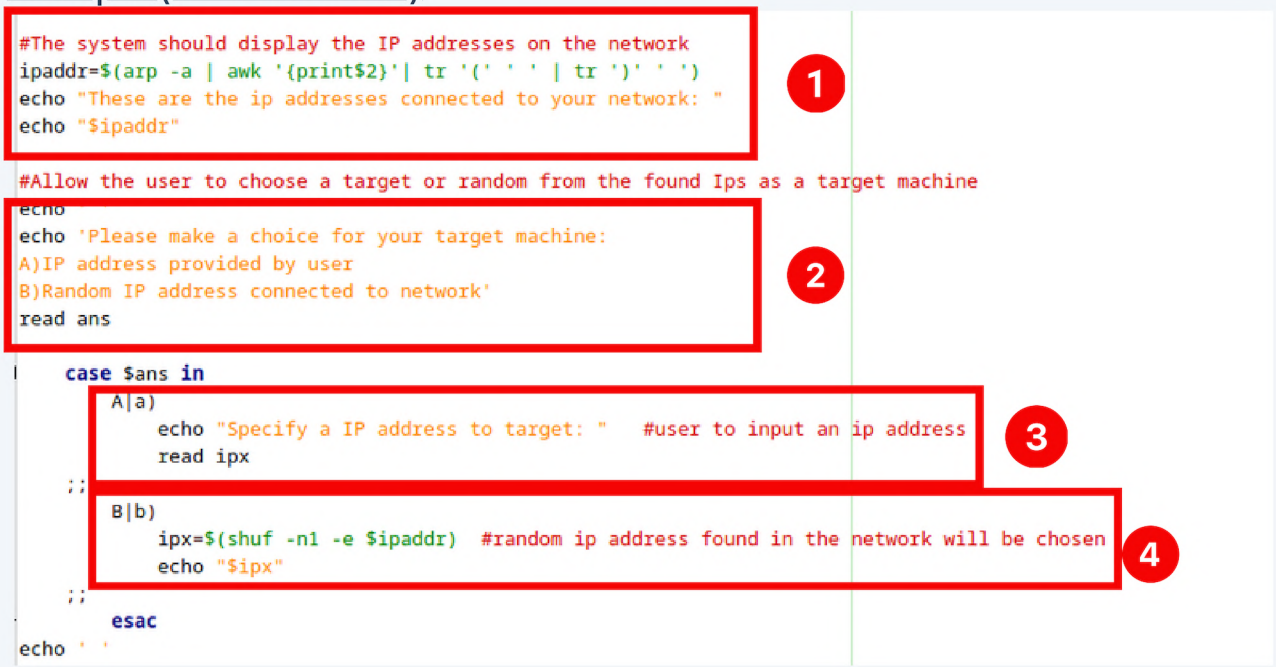
Step 7:

- Use a SSH connection to forward the necessary ports (5044,5061,9200) to the localhost address of your target machine.

METHODOLOGIES

3.Pentest Script (Attack)

Script (IP address)



```
#The system should display the IP addresses on the network
ipaddr=$(arp -a | awk '{print$2}'| tr '(' ' ' | tr ')' ' ')
echo "These are the ip addresses connected to your network: "
echo "$ipaddr"

#Allow the user to choose a target or random from the found Ips as a target machine
echo
echo 'Please make a choice for your target machine:
A)IP address provided by user
B)Random IP address connected to network'
read ans

    case $ans in
        A|a)
            echo "Specify a IP address to target: "    #user to input an ip address
            read ipx
    ;;
        B|b)
            ipx=$(shuf -n1 -e $ipaddr)  #random ip address found in the network will be chosen
            echo "$ipx"
    ;;
        esac
echo ' '
```

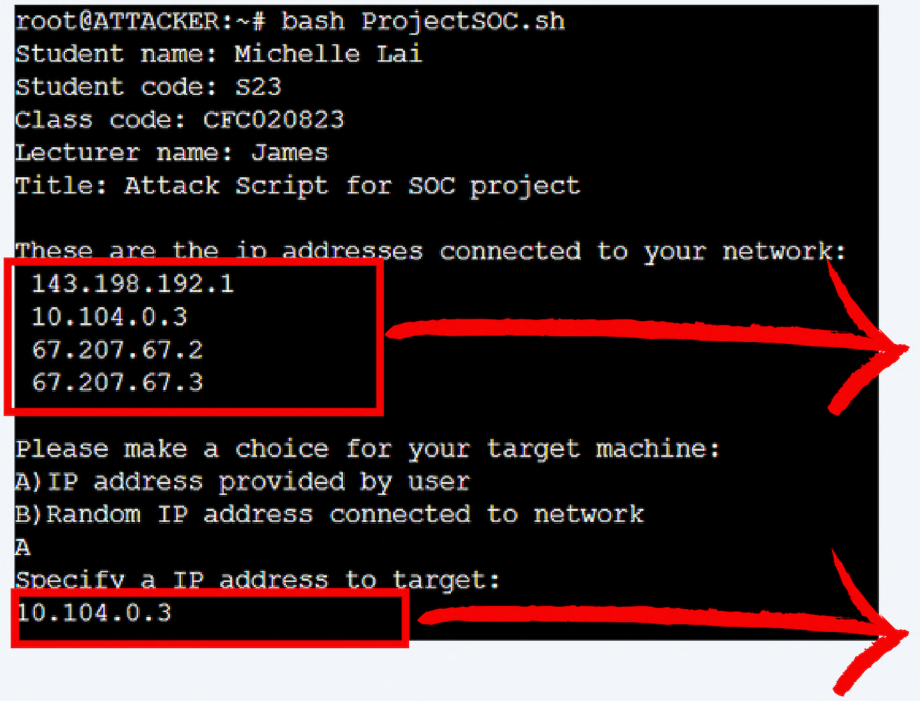
1. The ‘ipaddr’ variable calls for all IP addresses that is connected on the network.
2. The user is able to make a choice on what target they want to attack on.
3. Choice A allows user to provide a IP address they want to attack.
4. Choice B will pick a random IP address connected to the network and used it as a target IP address.

METHODOLOGIES

3.Pentest Script (Attack)

Result (IP address)

- Choice A

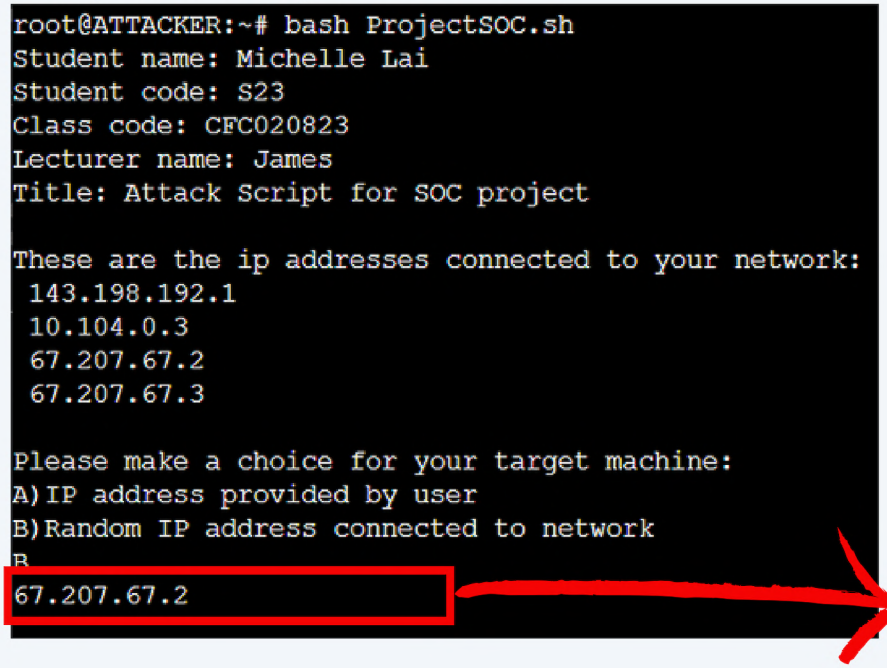


IP addresses connected on the network

IP address that user specifies.

Result (IP address)

- Choice B



The random IP address chosen will be shown to the user.

METHODOLOGIES

3.Pentest Script (Attack)

Script (Functions of attack continued)

```
#Each attack should have a description to display once chosen
#Display a list of all possible attacks with descriptions
echo "What does the user want to do?
A)Medusa - password cracking on the target's machine
B)Nmap - to find out which ports are open in the target's machine
C)Hping3 - send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies
D)Randomised Attack"
read attack

#Create attack scripts that can simulate at least three (3) different attack types using functions.
#The user can choose a specific attack or random from the list

    case $attack in
        A|a) #password cracking on the target machine.
        wkpass
    ;;
        B|b) #checking open ports.
        openports
    ;;
        C|c) #sends custom ICMP/UDP/TCP packets and display target replies.
        pack
    ;;
        D|d) #random attack from choice A,B or C chosen
        echo "Random attack chosen!"
        atk=( "wkpass" "openports" "pack" )
        $(shuf -n1 -e "${atk[@]}" )
    ;;
        *)  #exit when choice A,B,C or D not chosen
        echo "Wrong Input given!"
        exit
        ;;
    esac
```

A

B

C

D

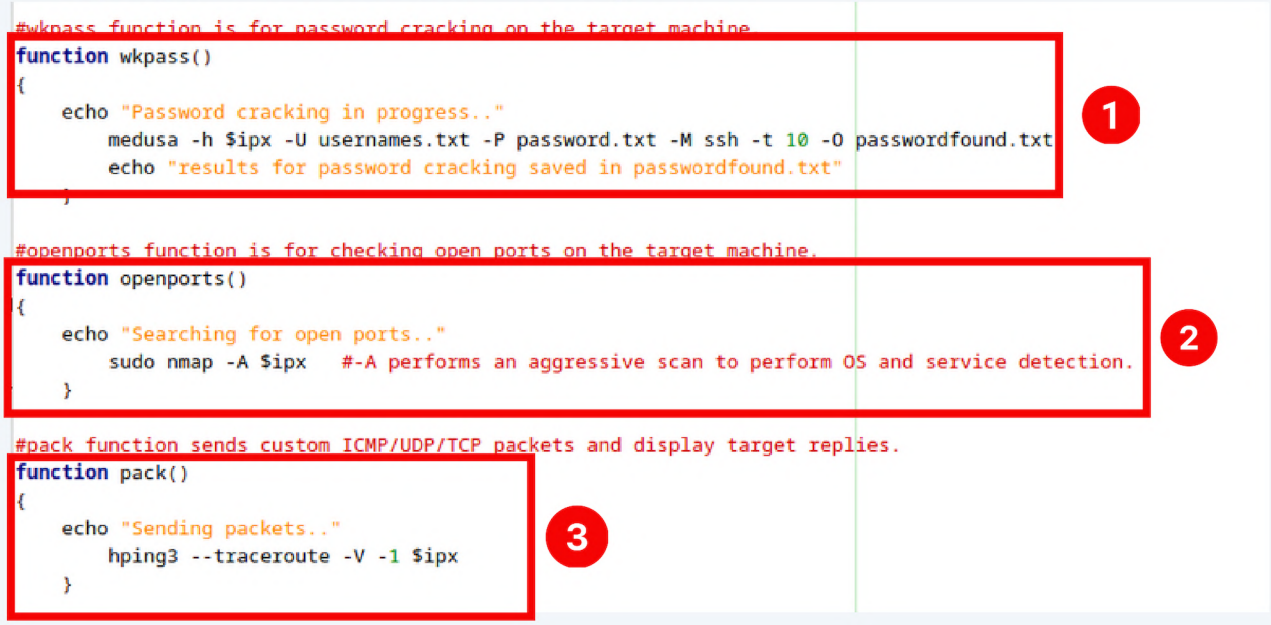
*

This part of the script shows the different attacks that the user can choose to use on the target IP address.
Choice A: Uses medusa to perform password cracking on the target.
Choice B: Uses nmap to check for open port on the target.
Choice C: Uses hping3 to send custom packets and display target replies like ping does with ICMP replies.
Choice D: Chooses random attack from choice A, B or C.
Choice *: * represents anything that is not A, B, C or D. As long as any alphabets or numbers or symbols is chosen, it will auto display a message and exit.
“wkpass”, “openports” and “pack” are functions which will be explained next.

METHODOLOGIES

3.Pentest Script (Attack)

Script (Functions of attack)



```
#wkpass function is for password cracking on the target machine.
function wkpass()
{
    echo "Password cracking in progress.."
        medusa -h $ipx -U usernames.txt -P password.txt -M ssh -t 10 -O passwordfound.txt
        echo "results for password cracking saved in passwordfound.txt"
    }

#openports function is for checking open ports on the target machine.
function openports()
{
    echo "Searching for open ports.."
        sudo nmap -A $ipx   #-A performs an aggressive scan to perform OS and service detection.
    }

#pack function sends custom ICMP/UDP/TCP packets and display target replies.
function pack()
{
    echo "Sending packets.."
        hping3 --traceroute -V -1 $ipx
    }
```


This part of the script shows the different functions of attack.

1. wkpass (medusa):
    - -h $ipx -> target ip address.
    - -U usernames.txt -> username list to test.
    - -P password.txt -> password list to test.
    - -M ssh -> service to execute.
    - -t 10 -> total number of logins to be tested concurrently.
    - -O passwordfound.txt -> save the log and results into a text file.
2. openports (nmap):
    - -A -> perform OS and service detection.
3. pack (hping3):
    - --traceroute -> increase ttl for each ICMP time to live 0 during transit received.
    - -V -> verbose mode.
    - -1 -> icmp mode.

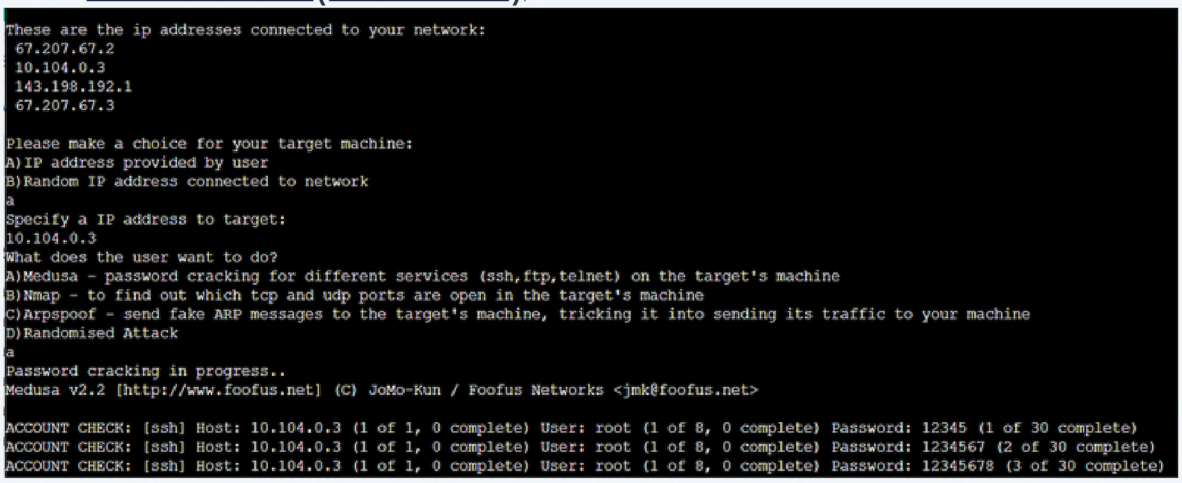
METHODOLOGIES

4. Testing, Validation and Logging

Medusa is a powerful and lightweight login brute-forcer used to brute-force credentials in as many protocols as possible which eventually lead to remote code execution.

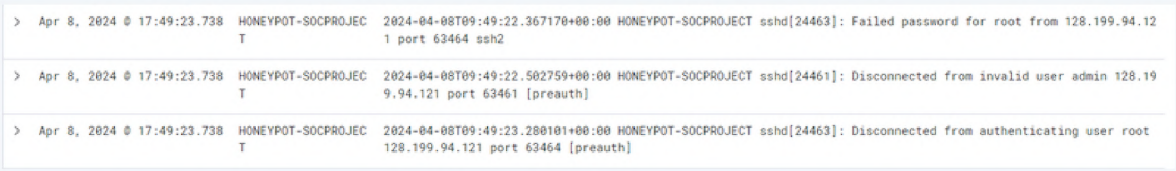
Result (Functions of attack)

- Choice A (Medusa)



Result (Kibana Dashboard)

- Choice A (Medusa)



Results shown in kibana dashboard via filebeats when medusa was used on target machine.

METHODOLOGIES

4. Testing, Validation and Logging

Nmap is a network scanning tool used for network exploration, host discovery, and security auditing. It was created to help map an entire network easily and find its open ports and services.

Result (Functions of attack)

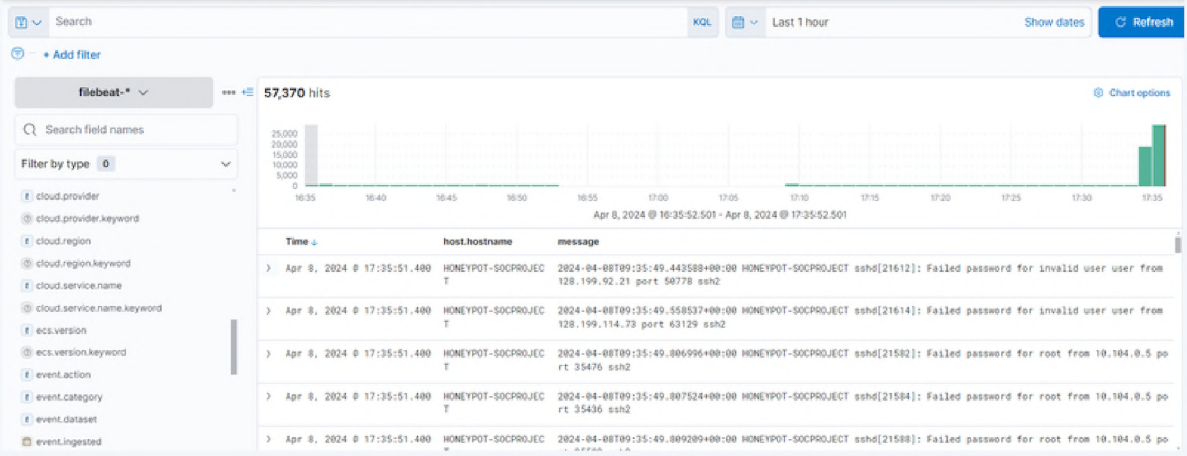
- Choice B (NMAP)

```
Please make a choice for your target machine:
A)IP address provided by user
B)Random IP address connected to network
a
Specify a IP address to target:
10.104.0.3
What does the user want to do?
A)Medusa - password cracking for different services (ssh,ftp,telnet) on the target's machine
B)Nmap - to find out which tcp and udp ports are open in the target's machine
C)Arpspoof - send fake ARP messages to the target's machine, tricking it into sending its traffic to your machine
D)Randomised Attack
b
Searching for open ports..
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-08 09:57 UTC
Nmap scan report for 10.104.0.3
Host is up (0.0019s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.3p1 Ubuntu lubuntu3.2 (Ubuntu Linux; protocol 2.0)
MAC Address: 82:01:7F:DC:58:C6 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.71 seconds
```

Result (Kibana Dashboard)

- Choice B (NMAP)



Results shown in kibana dashboard via filebeats when nmap was used on target machine.

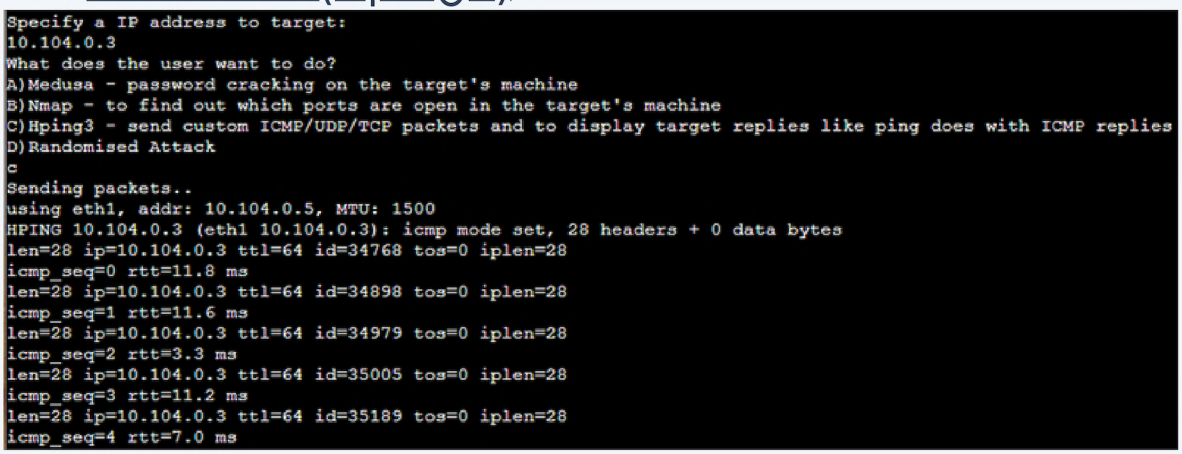
METHODOLOGIES

4. Testing, Validation and Logging

Hping3 is a command-line tool that analyze TCP/IP messages on a network. It is also used to assemble network packets, which can be beneficial to a penetration tester in performing device and service discovery and offensive actions, such as a Denial-of-Service (DoS) attack.

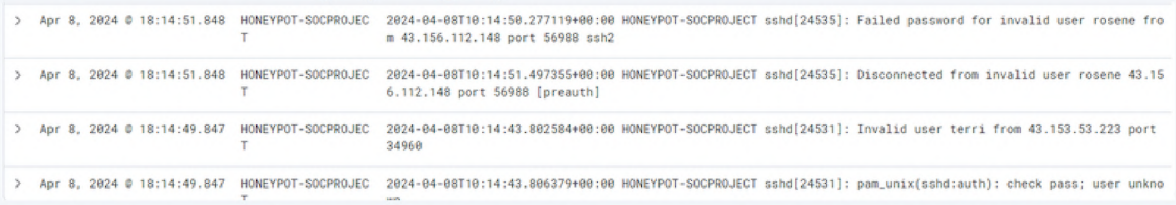
Result (Functions of attack)

- Choice C (Hping3)



Result (Kibana Dashboard)

- Choice C (Hping3)



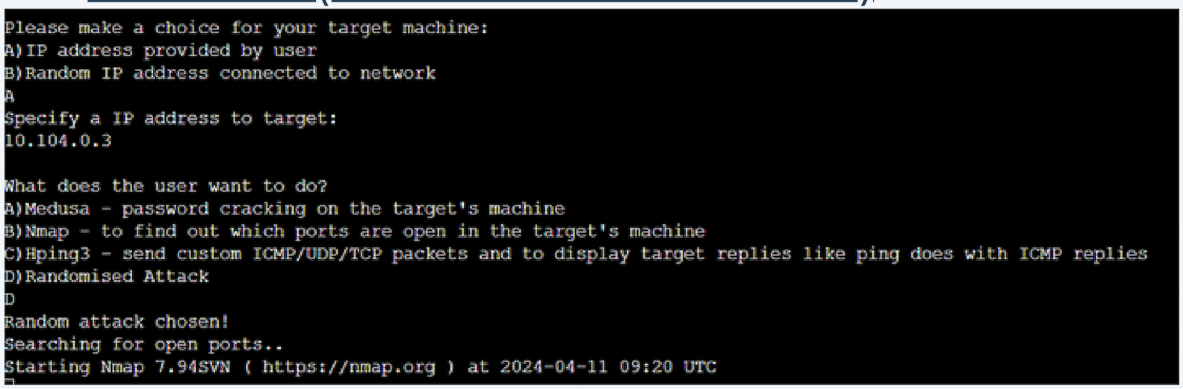
Results shown in kibana dashboard via filebeats when hping3 was used on target machine.

METHODOLOGIES

4. Testing, Validation and Logging

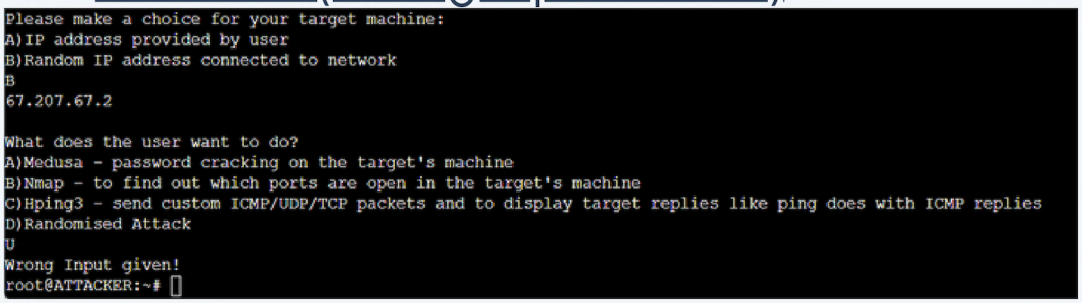
Result (Functions of attack)

- Choice D (Random Attack chosen)



Result (Functions of attack)

- Choice * (Wrong Input -> Exit)

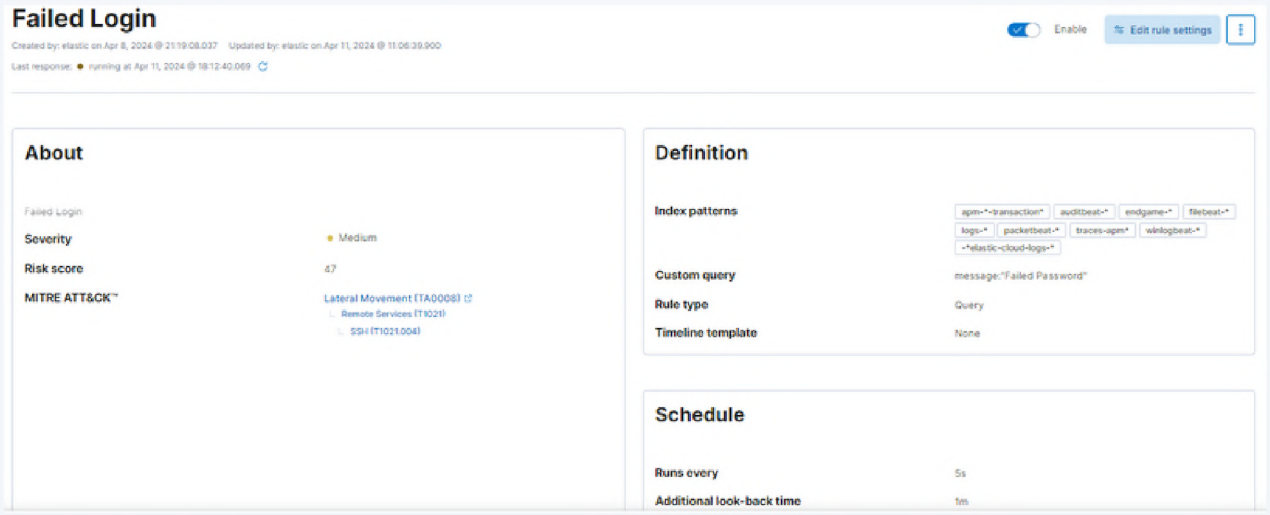


METHODOLOGIES

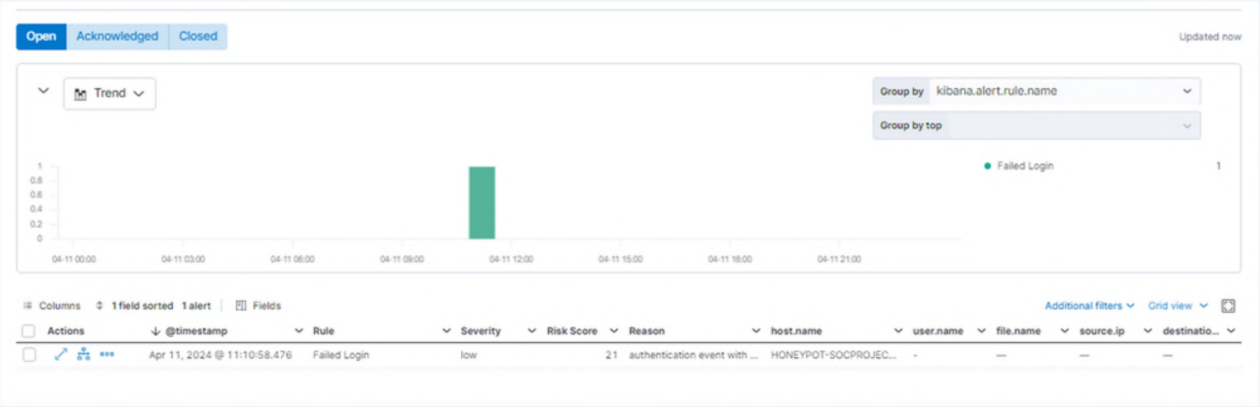
4. Testing, Validation and Logging

- **Alerting**

Rule



Rule Alert



When there are failed attempts trying to log into the target machine, alertf 'Failed Login' will be detected.

Recommendation:
An alert for the threshold count of failed logins can be included. This will filter out attempts that includes valid users inputting their wrong passwords.

CITATIONS

1. Glass, E., & Camisso, J. (2022, April 26). How to install Elasticsearch, Logstash, and Kibana (Elastic Stack) on ubuntu 22.04. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-22-04
2. Oosterhof, M. (2019). Installing cowrie in seven steps . Installing Cowrie in seven steps - cowrie 2.5.0 documentation. https://cowrie.readthedocs.io/en/latest/INSTALL.html
3. cyberandspace, P. (2019, May 10). Setting up a honeypot with AWS and collecting the data with an elastic stack (elk) server. cyber and space. https://cyberandspace.wordpress.com/2019/05/04/setting-up-a-honeypot-with-aws-and-collecting-the-data-with-an-elastic-stack-elk-server/
4. Jethva, H. (2022, September 8). How to install FileBeat on ubuntu. HowtoForge. https://www.howtoforge.com/how-to-install-filebeat-on-ubuntu/
5. BORGES, E. (2021, April 6). Securitytrails | top 16 nmap commands: Nmap port scan tutorial guide. Top 16 Nmap Commands to Scan Remote Hosts - Tutorial Guide. https://securitytrails.com/blog/nmap-commands
6. Tools, K. (2024, March 11). HPING3: Kali linux tools. Kali Linux. https://www.kali.org/tools/hping3/
7. Sanfilippo, S. (2017). HPING3(8) - linux man page. hping3(8) - Linux man page. https://linux.die.net/man/8/hping3