Shiba Insider

Contents

Scenario

Can you uncover the insider?

Pre-requisites

- Load kali
- Run sudo apt update && sudo apt -y upgrade > reboot
- Change network to host only instead of NAT – to restrict network so malware inside the pcap is contained within the VM
- Extract inner zip and pcap from the challenge zip
    - Load Wireshark and open the pcap file

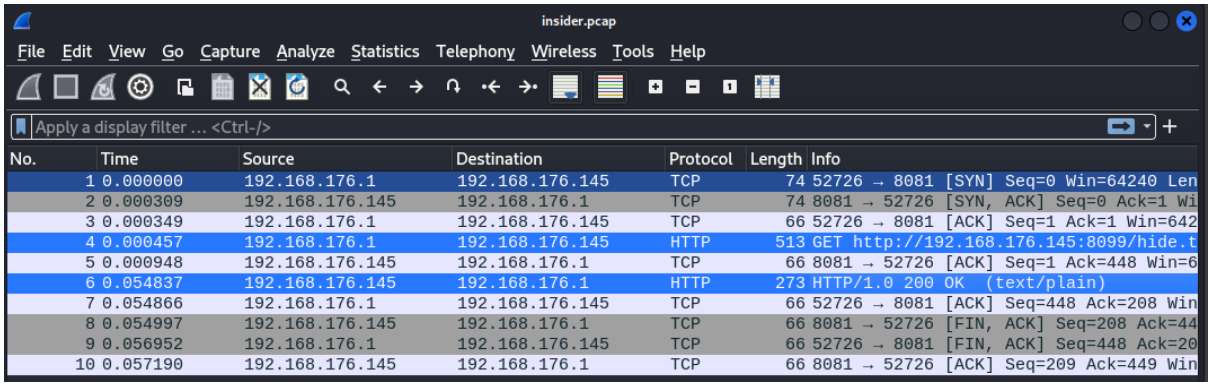
Initial thoughts from scenario

- Inner zip is password protected, so likely have to find the password in the pcap
    - Given I need credentials, maybe it's a pcap using a cleartext protocol such as FTP/HTTP? Could also be found in payload of TCP packet?

Challenge Questions

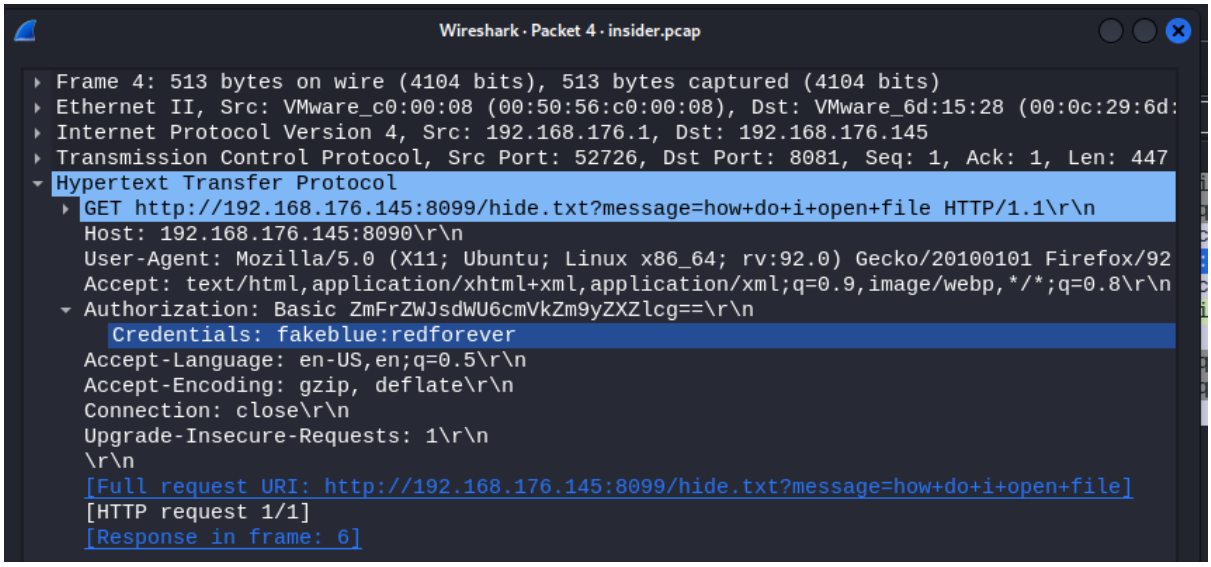
What is the response message obtained from the PCAP file?

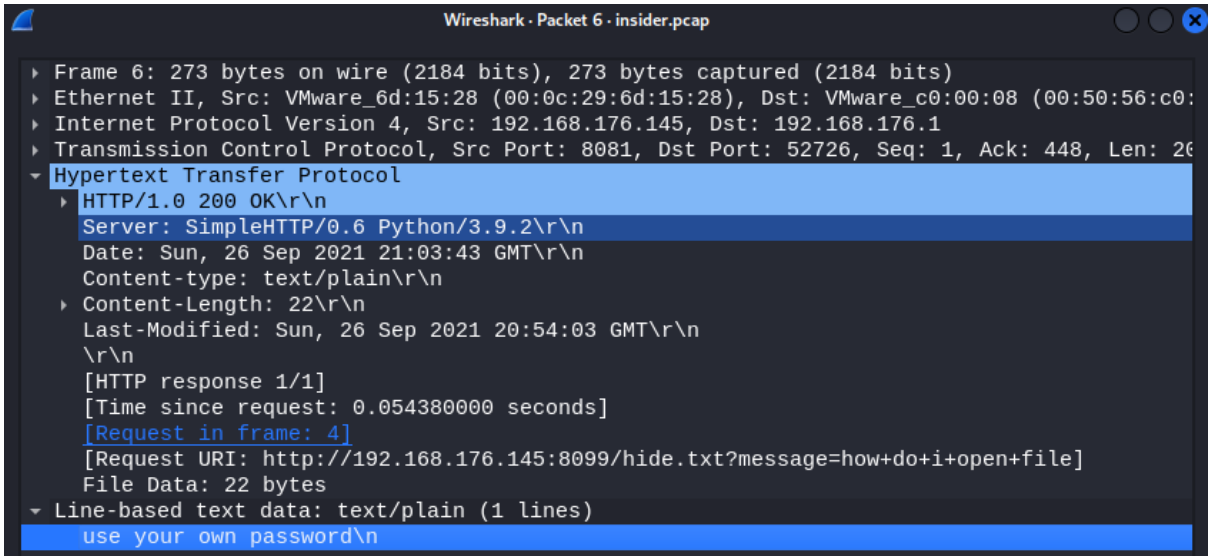
- Answer: use your own password



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.176.1	192.168.176.145	TCP	74	52726 → 8081 [SYN] Seq=0 Win=64240 Len
2	0.000309	192.168.176.145	192.168.176.1	TCP	74	8081 → 52726 [SYN, ACK] Seq=0 Ack=1 Wi
3	0.000349	192.168.176.1	192.168.176.145	TCP	66	52726 → 8081 [ACK] Seq=1 Ack=1 Win=642
4	0.000457	192.168.176.1	192.168.176.145	HTTP	513	GET http://192.168.176.145:8099/hide.t
5	0.000948	192.168.176.145	192.168.176.1	TCP	66	8081 → 52726 [ACK] Seq=1 Ack=448 Win=6
6	0.054837	192.168.176.145	192.168.176.1	HTTP	273	HTTP/1.0 200 OK (text/plain)
7	0.054866	192.168.176.1	192.168.176.145	TCP	66	52726 → 8081 [ACK] Seq=448 Ack=208 Win
8	0.054997	192.168.176.145	192.168.176.1	TCP	66	8081 → 52726 [FIN, ACK] Seq=208 Ack=44
9	0.056952	192.168.176.1	192.168.176.145	TCP	66	52726 → 8081 [FIN, ACK] Seq=448 Ack=20
10	0.057190	192.168.176.145	192.168.176.1	TCP	66	8081 → 52726 [ACK] Seq=209 Ack=449 Win

- As theorised in the "Initial Thoughts" section above, the pcap contains HTTP/TCP traffic.
- Frames 4 and 6 immediately stand out due to HTTP traffic being a cleartext protocol, reviewing them each in detail reveals:





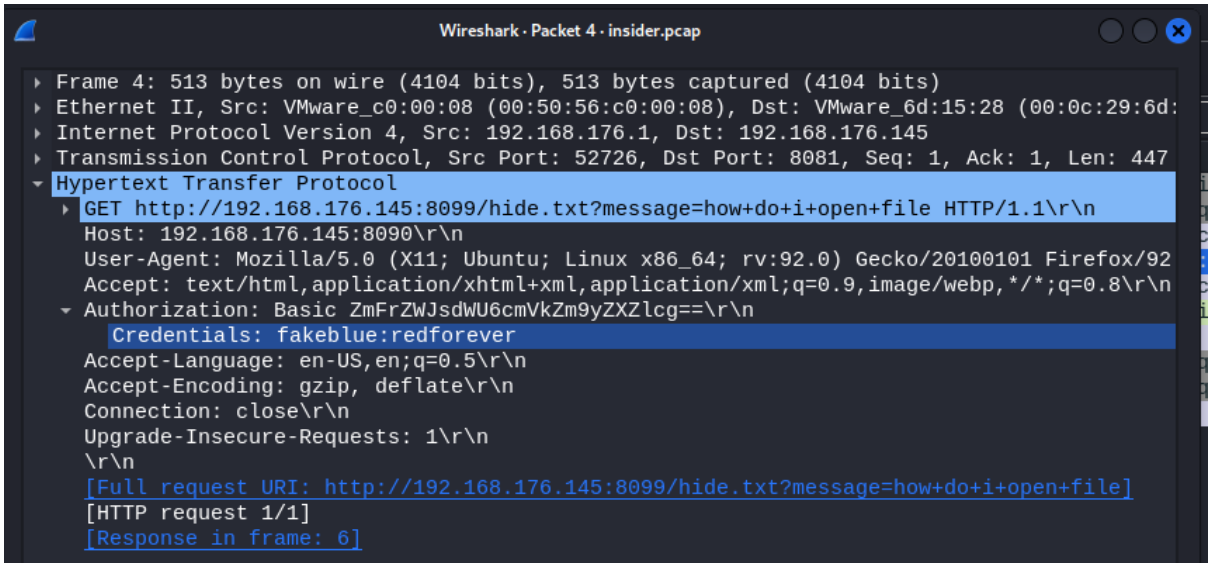
- Frame 4:
  - Src_ip = 192.168.176.1, src_port = 8099
  - Dst-IP = 192.168.176.145, dst_port = 8081
  - Requesting a file called hide.txt passing it a “message” parameter with value “how do I open file”
  - Authorisation credentials are fakeblue:redforever
    - In the format user:pass
- Frame 6:
  - Src_ip = 192.168.176.145, src_port = 8081
  - Dst_ip = 192.168.176.1, dst_port = 52576
  - Server responding is SimpleHTTP v0.6 using Python3.9.2
  - Text/data returned is “use your own password”

After analysing the communications, we can see that the server response contains the answer which is “use your own password”.

What is the password of the ZIP file?

- Answer: redforever

Due to our analysis for the previous question, we've already analysed the conversation that contains the answer. As seen in frame 4:

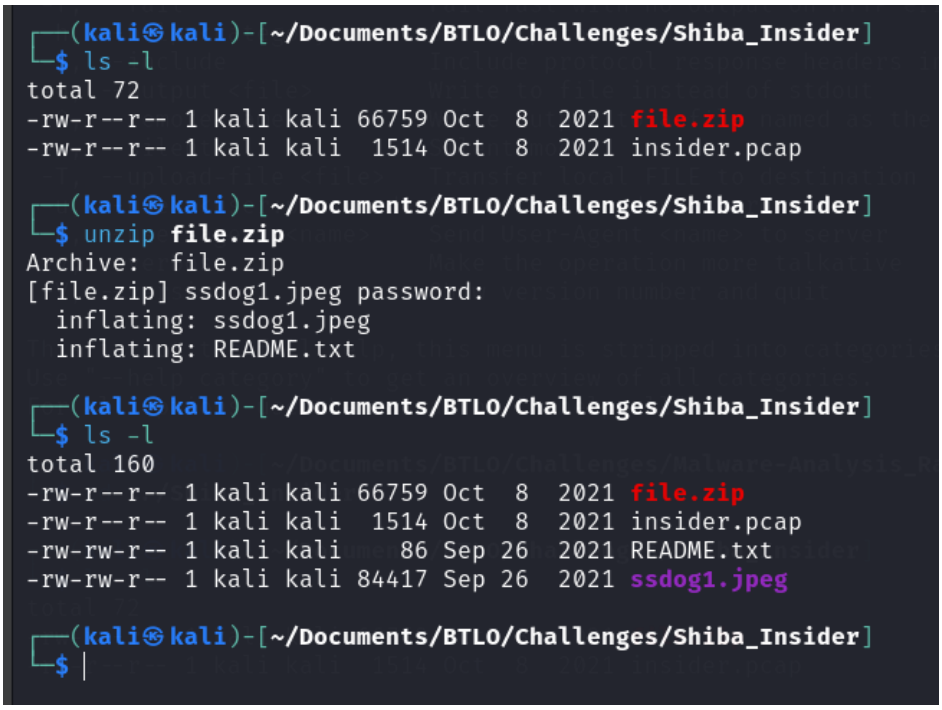


Authorisation credentials submitted are fakeblue:redforever, in the format user:pass

Will more passwords be required?

- Answer: No

Using the password "redforever", we can unzip "file.zip":



By using cat on the README.txt, we find out that no more passwords are required:

```
┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$ cat README.txt
Shiba Dog has everything you need and decided that no more passwords will be needed

┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$
```

What is the name of a widely-used tool that can be used to obtain file information?

- Answer: Exiftool

No analysis needed, when analysing files or metadata exiftool is often a go to.

What is the name & value of the interesting information obtained from the image file metadata?

- Answer: Technique:Steganography

Running Exiftool on the JPG reveals the following:

```
┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$ exiftool ssdog1.jpeg
ExifTool Version Number         : 12.67
File Name                       : ssdog1.jpeg
Directory                       : .
File Size                       : 84 kB
File Modification Date/Time     : 2021:09:26 21:07:52+01:00
File Access Date/Time           : 2021:09:26 21:07:57+01:00
File Inode Change Date/Time     : 2023:12:06 14:29:05+00:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
XMP Toolkit                     : Image::ExifTool 11.88
Technique                       : Steganography
Technique Command               : steghide
Image Width                     : 1080
Image Height                    : 1018
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 1080×1018
Megapixels                      : 1.1
```

After reviewing, there are two line that standout which are “Technique” and “Technique Command”. Steganography involves hiding data within digital files such as images, audio, video, or text documents. It typically utilises small, imperceptible changes to the least significant bits (LSBs) of the cover media to embed the secret data.

Therefore, as this is suspicious, we can confirm the answer.

Based on the answer from the previous question, what tool needs to be used to retrieve the information hidden in the file?

- Answer: Steghide

No analysis needed, Steghide is a popular tool to analyse files that utilise Steganography techniques.

Enter the ID retrieved.

- Answer: 0726ba878ea47de571777a

Install steghide if not already installed:

```
┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$ sudo apt install steghide
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
```

```
┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$ steghide extract -sf ssdog1.jpeg
Enter passphrase:
wrote extracted data to "idInsider.txt".

┌──(kali㉿kali)-[~/Documents/BTLO/Challenges/Shiba_Insider]
└─$ cat idInsider.txt
0726ba878ea47de571777a
```

Command above is as follows:

- Steghdie – the name of the tool
- Extract – tells steghide we want to extract information
- -sf – tells steghide we want to extract information from a file
- Ssdog1.jpeg – the name of the file to extract information from

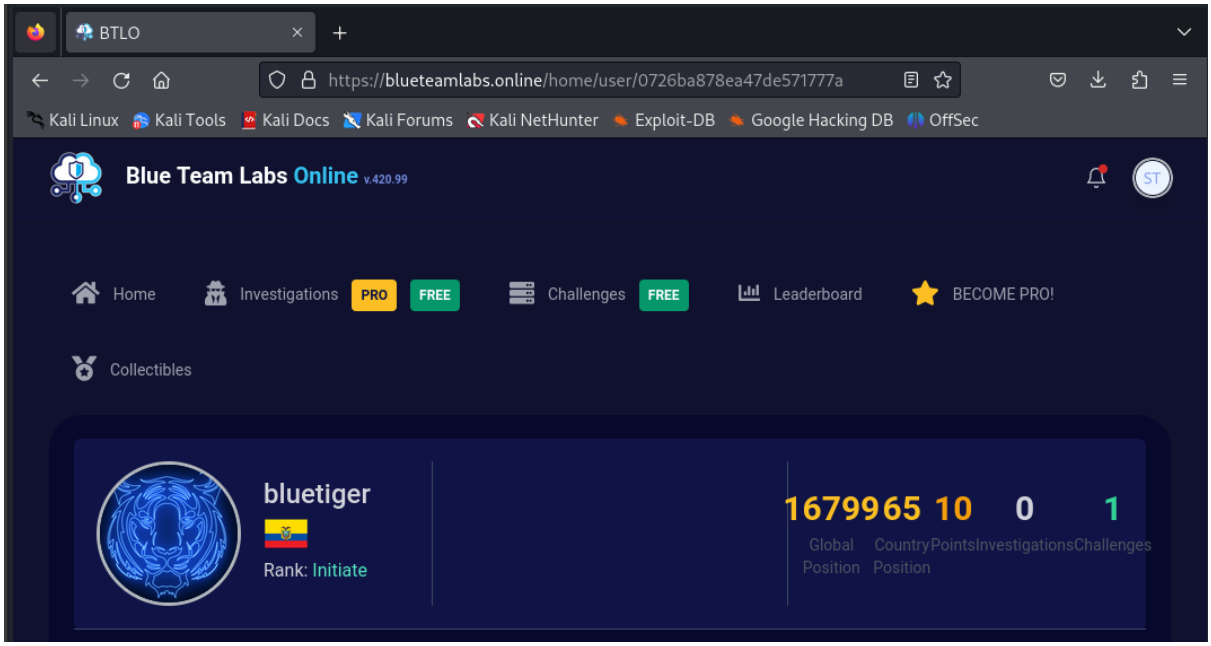
At the password prompt, I first tried “redforever” with no luck, and after double checking to see if I had missed another password (which I hadn’t), I tried no password and bingo:

- ID – 0726ba878ea47de571777a

What is the profile name of the attacker?

- Answer: bluetiger

Append the id extracted previously to blueteamlabs.online/home/user/[idHere], giving us our answer as seen on the following page:


BTLO
https://blueteamlabs.online/home/user/0726ba878ea47de571777a
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Blue Team Labs Online v.420.99
ST
Home
Investigations
PRO
FREE
Challenges
FREE
Leaderboard
BECOME PRO!
Collectibles
bluetiger
Rank: Initiate
1679965
10
0
1
Global Position
Country Position
Points
Investigations
Challenges