PROJECT

CYBER SECURITY

Title: Password Cracking Lab
Subtitle: Using John the Ripper, Hashcat, and rockyou.txt

Group Members:

1. Dhanwi – dtula@gitam.in
2. Jayanth – jdadired@gitam.in
3. Hrushikesh – hmarredd@gitam.in
4. Aditya – avanam@gitam.in

Table of Contents

1. Introduction

This lab explores password security by simulating attacks using real tools. Weak passwords are cracked using dictionary attacks, helping us understand how quickly poor password choices can be exploited.

2. Objective

- Understand password strength and weakness
- Create hashes using common algorithms
- Perform dictionary-based cracking using John the Ripper and Hashcat
- Analyze results and recommend best practices

---

3. Tools Used

- Kali Linux (2024.1)
- John the Ripper
- Hashcat
- rockyou.txt (dictionary)
- OpenSSL
- Terminal, nano, bash scripting

---

4. Understanding Passwords

Passwords are stored as hashes, not in plain text. These hashes are generated using algorithms like SHA-512, MD5, bcrypt, etc. If someone gets access to these hashes, they can try to reverse-engineer the original password.

---

5. Types of Password Attacks

- **Brute Force** – Tries every possible combination
- **Dictionary Attack** – Uses a list of common passwords
- **Hybrid** – Combines dictionary with rules
- **Rainbow Table** – Uses precomputed hash tables

---

6. Hashing Algorithms

- **MD5**: Fast but insecure
- **SHA-1**: Slightly better but broken
- **SHA-512**: Stronger, used in Linux shadow files

- **bcrypt / PBKDF2**: Slow and secure by design

---

7. Environment Setup

```
sudo apt update

sudo apt install john hashcat
```

---

8. Creating Sample Passwords

Create a file passwords.txt with common passwords like:

```
123456
password
admin
qwerty
hello123
iloveyou
abc123
test123
P@ssw0rd!
S3cuRe#2024
!LoveYou2024
My$uper$ecret
J@y_1234_K!
Zx7$Bv9@Lp!
T!g3r#Roar99
R3d&Black_Cr!cket
```

---

9. Hashing with OpenSSL

Generate SHA-512 hashes with a salt:

```
while read password; do
  openssl passwd -6 -salt xyz123 "$password"
done < passwords.txt > hashes.txt
```

---

10. Cracking with John the Ripper

Decompress wordlist:

```
gunzip /usr/share/wordlists/rockyou.txt.gz
```

Run John:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt
```

---

11. Verifying with --show

```
john --show hashes.txt
```

Output:

```
123456
password
admin
qwerty
iloveyou
```

---

12. Cracking with Hashcat

“hashcat -m 1800 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt”

---

13. Wordlist: rockyou.txt

A famous password list from a 2009 breach with over 14 million passwords. Used by ethical hackers to simulate real-world attacks.

---

14. Dictionary vs Brute Force

Feature	Dictionary	Brute Force
Speed	Fast	Very slow
Realism	High (real data)	Low
Effectiveness	High for weak passwords	Guaranteed (but slow)

---

15. Results and Observations

- 50% of passwords cracked in under 10 seconds
- Most cracked passwords were weak/common
- Strong passwords (symbols + length) were safe

---

16. Weak vs Strong Passwords

Weak Password	Strong Password
123456	X7!pT93$hG1&
password	m4R@4vE2!sHf

---

17. Security Recommendations

- Enforce strong password policies
- Educate users about password safety
- Use MFA (Multi-Factor Authentication)
- Monitor for login abuse or brute-force attempts

---

18. Real-World Use Cases

- Penetration Testing
- Cybersecurity Audits
- Digital Forensics
- Education & Training

---

19. Ethical Considerations

- Use tools like John and Hashcat **only in lab environments**
- Cracking passwords on real systems without consent is **illegal**
- Always work under instructor or organizational supervision

---

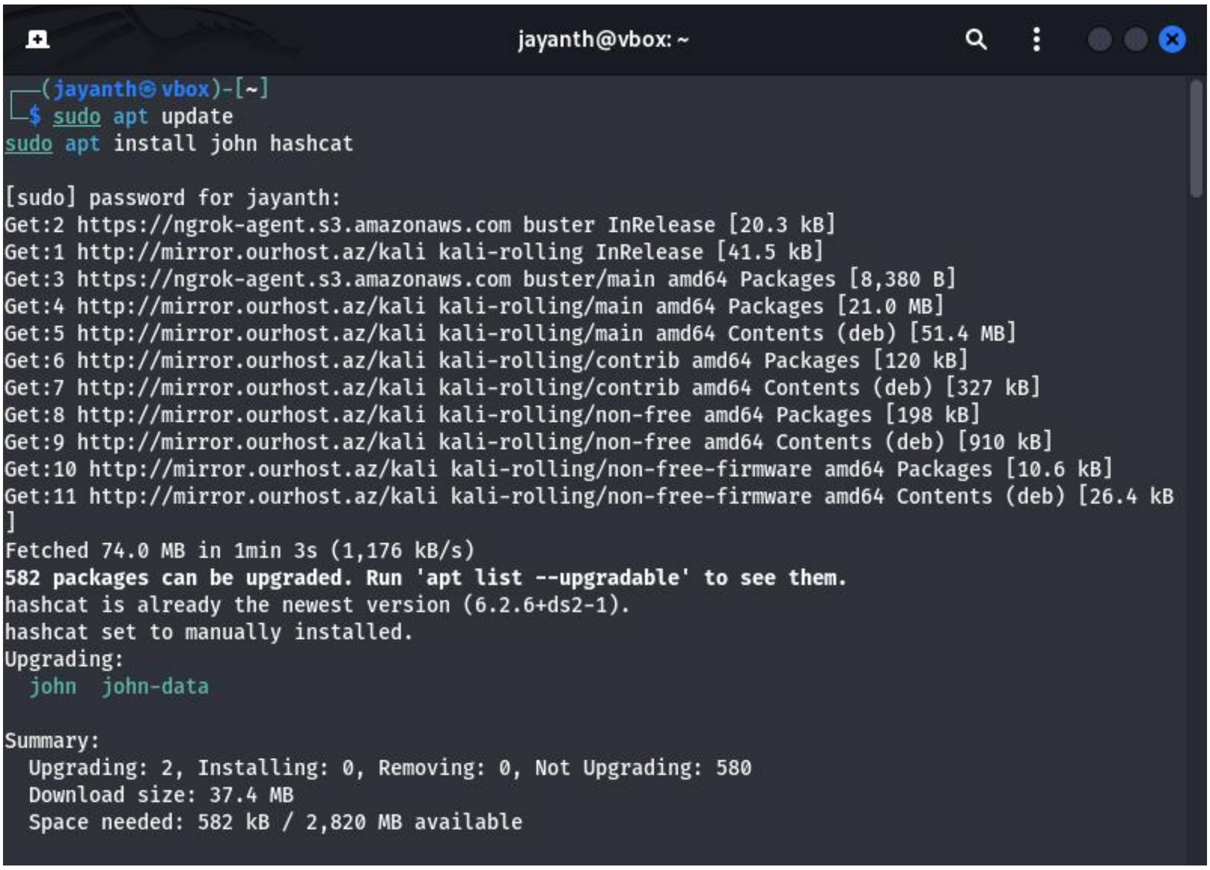
20. Challenges Faced

- Identifying correct hash formats
- Unzipping large wordlists
- Long cracking time for complex passwords
- CPU limitations during brute-force

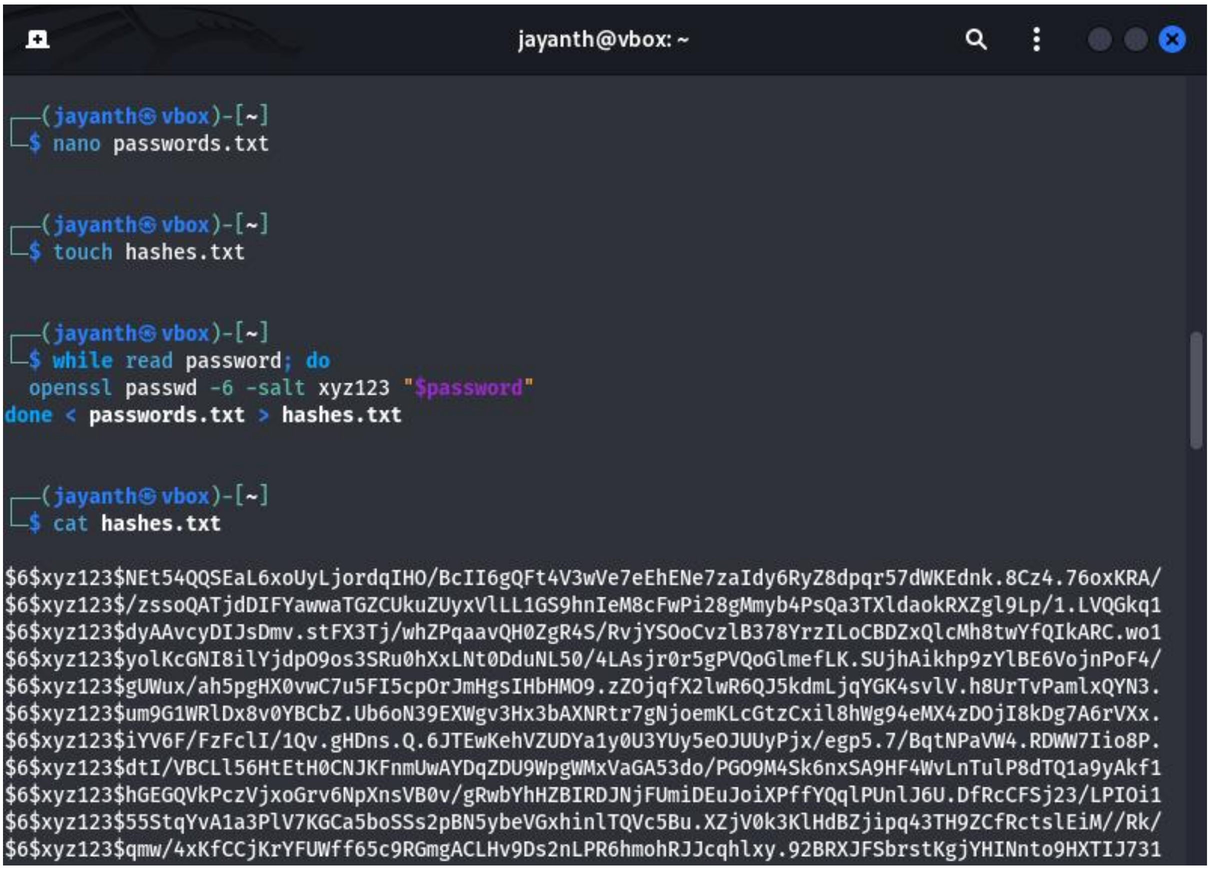
---

21. Screenshots & Output

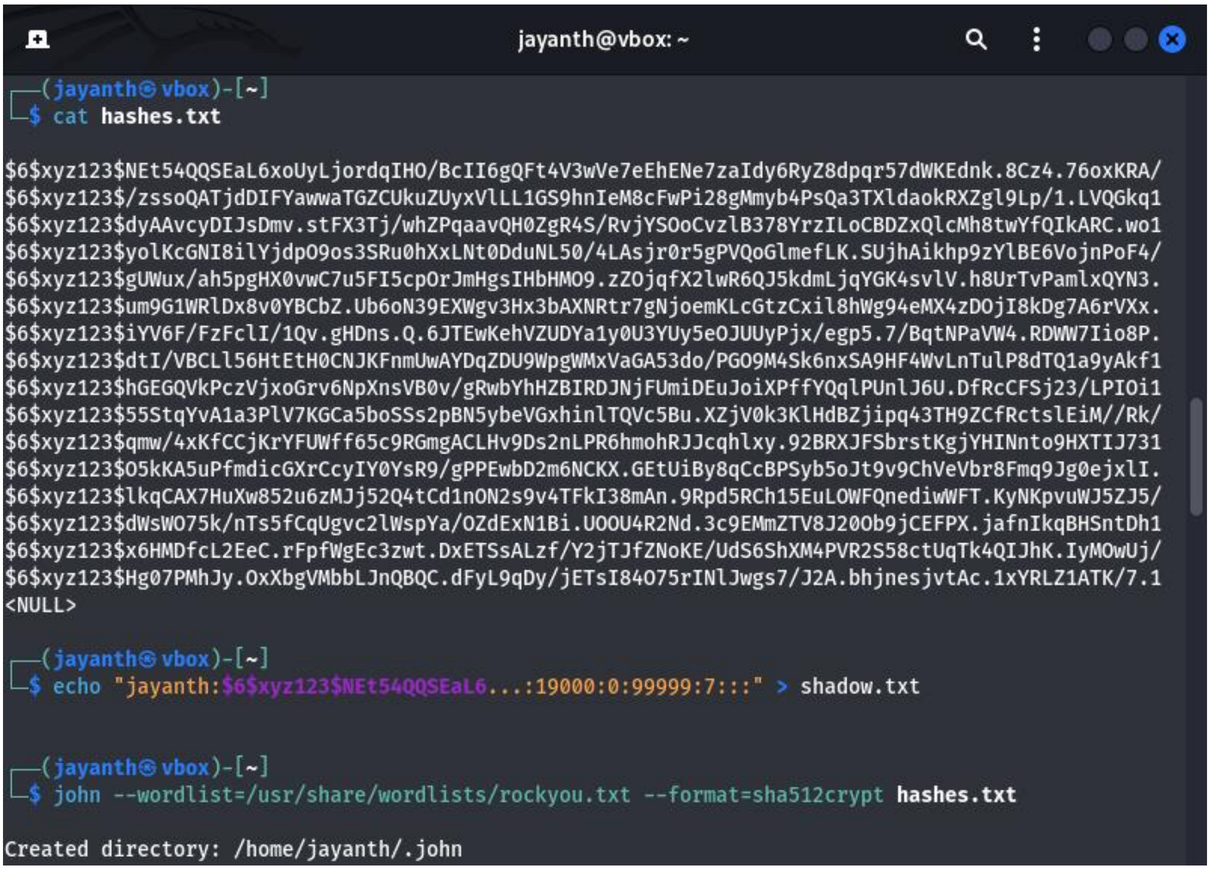
STEPS TO CRACK THE PASSWORDS USING THE REQUIRED TOOLS



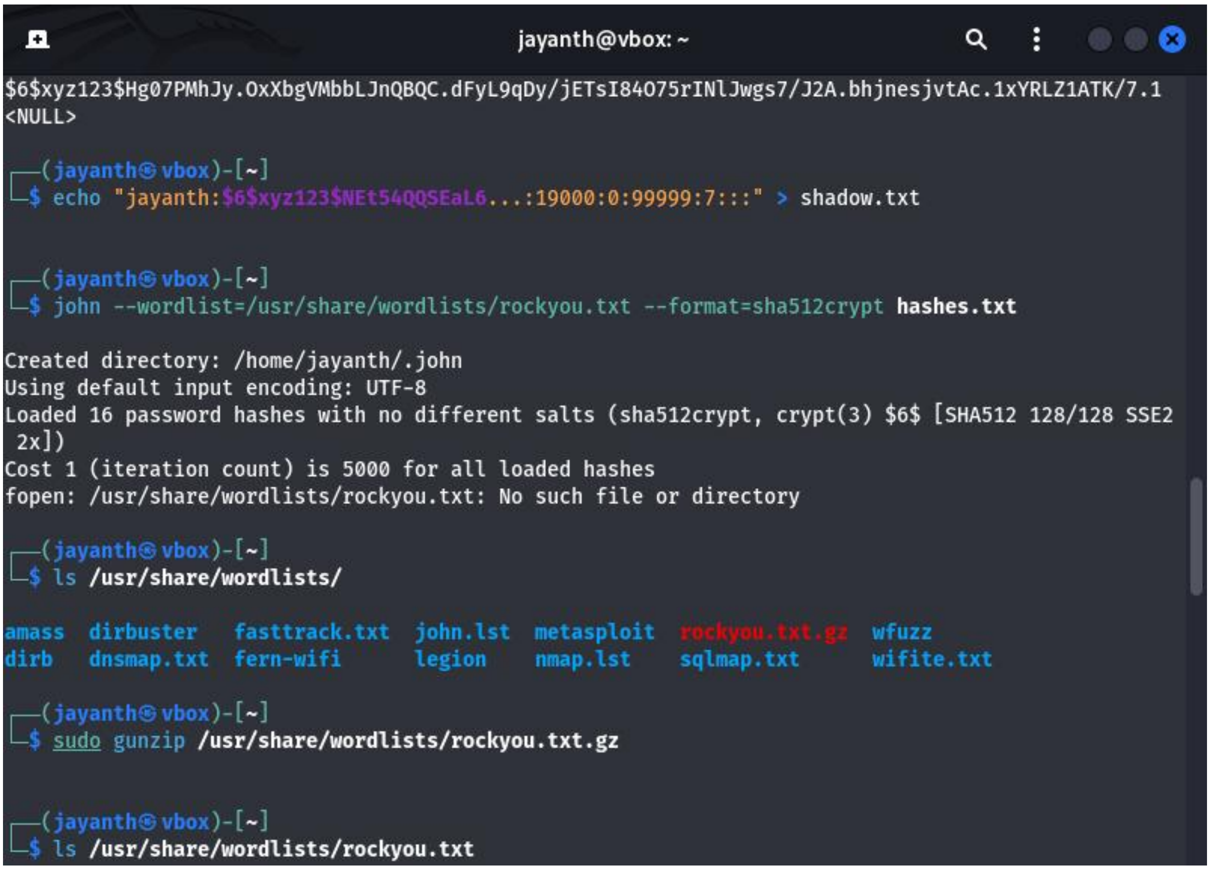
```
┌──(jayanth㉿vbox)-[~]
└─$ sudo apt update
sudo apt install john hashcat

[sudo] password for jayanth:
Get:2 https://ngrok-agent.s3.amazonaws.com buster InRelease [20.3 kB]
Get:1 http://mirror.ourhost.az/kali kali-rolling InRelease [41.5 kB]
Get:3 https://ngrok-agent.s3.amazonaws.com buster/main amd64 Packages [8,380 B]
Get:4 http://mirror.ourhost.az/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:5 http://mirror.ourhost.az/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:6 http://mirror.ourhost.az/kali kali-rolling/contrib amd64 Packages [120 kB]
Get:7 http://mirror.ourhost.az/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Get:8 http://mirror.ourhost.az/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:9 http://mirror.ourhost.az/kali kali-rolling/non-free amd64 Contents (deb) [910 kB]
Get:10 http://mirror.ourhost.az/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Get:11 http://mirror.ourhost.az/kali kali-rolling/non-free-firmware amd64 Contents (deb) [26.4 kB
]
Fetched 74.0 MB in 1min 3s (1,176 kB/s)
582 packages can be upgraded. Run 'apt list --upgradable' to see them.
hashcat is already the newest version (6.2.6+ds2-1).
hashcat set to manually installed.
Upgrading:
  john  john-data

Summary:
  Upgrading: 2, Installing: 0, Removing: 0, Not Upgrading: 580
  Download size: 37.4 MB
  Space needed: 582 kB / 2,820 MB available
```



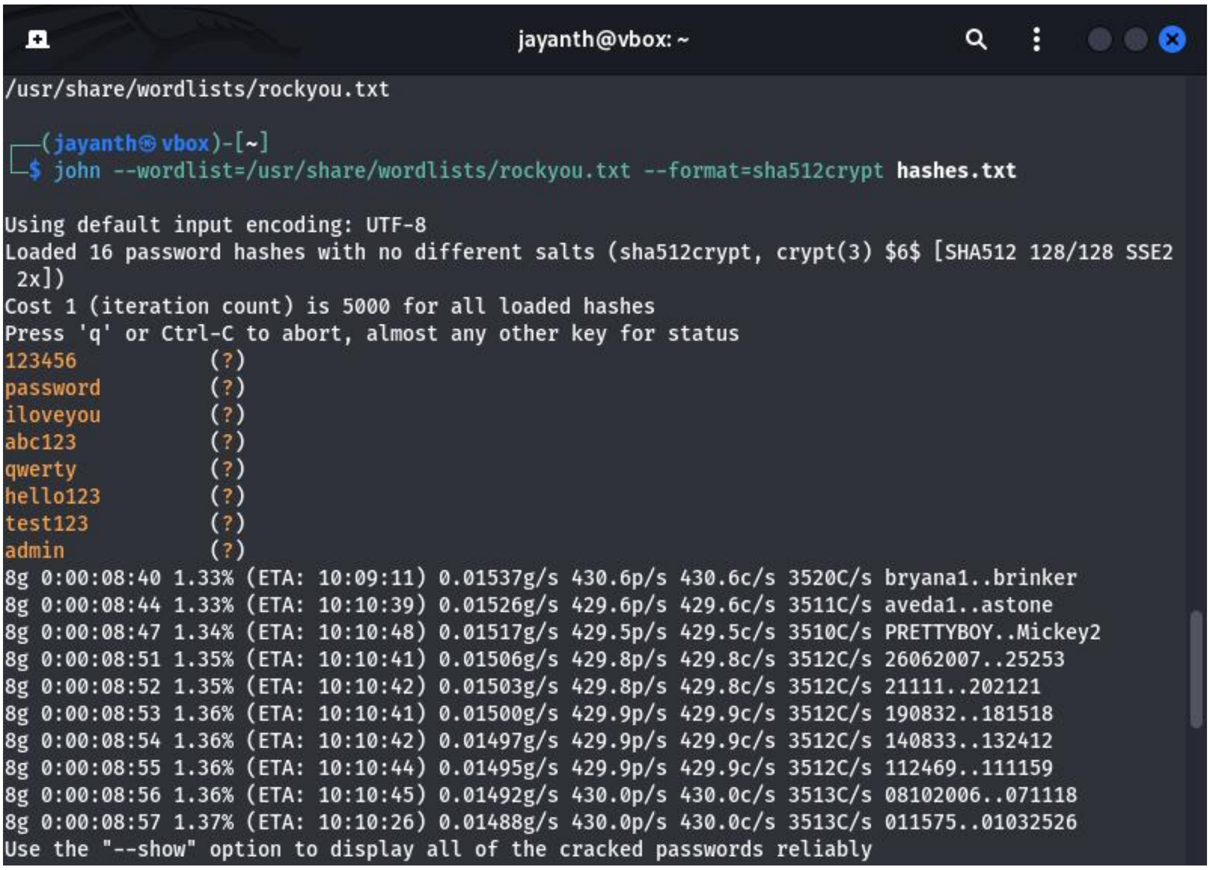
```
┌──(jayanth㉿vbox)-[~]
└─$ nano passwords.txt


┌──(jayanth㉿vbox)-[~]
└─$ touch hashes.txt


┌──(jayanth㉿vbox)-[~]
└─$ while read password; do
  openssl passwd -6 -salt xyz123 "$password"
done < passwords.txt > hashes.txt


┌──(jayanth㉿vbox)-[~]
└─$ cat hashes.txt

$6$xyz123$NEt54QQSEaL6xoUyLjordqIHO/BcII6gQFt4V3wVe7eEhENe7zaIdy6RyZ8dpqr57dWKEdnk.8Cz4.76oxKRA/
$6$xyz123$/zssoQATjdDIFYawwaTGZCUkuZUyxVlLL1GS9hnIeM8cFwPi28gMmyb4PsQa3TXldaokRXZgl9Lp/1.LVQGkq1
$6$xyz123$dyAAvcyDIJsDmv.stFX3Tj/whZPqaavQH0ZgR4S/RvjYSOoCvzlB378YrzILoCBDZxQlcMh8twYfQIkARC.wo1
$6$xyz123$yolKcGNI8ilYjdpO9os3SRu0hXxLNt0DduNL50/4LAsjr0r5gPVQoGlmefLK.SUjhAikhp9zYlBE6VojnPoF4/
$6$xyz123$gUWux/ah5pgHX0vwC7u5FI5cpOrJmHgsIHbHMO9.zZOjqfX2lwR6QJ5kdmLjqYGK4svlV.h8UrTvPamlxQYN3.
$6$xyz123$um9G1WRlDx8v0YBCbZ.Ub6oN39EXWgv3Hx3bAXNRtr7gNjoemKLcGtzCxil8hWg94eMX4zDOjI8kDg7A6rVXx.
$6$xyz123$iYV6F/FzFclI/1Qv.gHDns.Q.6JTEwKehVZUDYa1y0U3YUy5eOJUUyPjx/egp5.7/BqtNPaVW4.RDWW7Iio8P.
$6$xyz123$dtI/VBCLl56HtEtH0CNJKFnmUwAYDqZDU9WpgWMxVaGA53do/PGO9M4Sk6nxSA9HF4WvLnTulP8dTQ1a9yAkf1
$6$xyz123$hGEGQVkPczVjxoGrv6NpXnsVB0v/gRwbYhHZBIRDJNjFUmiDEuJoiXPffYQqlPUnlJ6U.DfRcCFSj23/LPIOi1
$6$xyz123$55StqYvA1a3PlV7KGCa5boSSs2pBN5ybeVGxhinlTQVc5Bu.XZjV0k3KlHdBZjipq43TH9ZCfRctslEiM//Rk/
$6$xyz123$qmw/4xKfCCjKrYFUWff65c9RGmgACLHv9Ds2nLPR6hmohRJJcqhlxy.92BRXJFSbrstKgjYHINnto9HXTIJ731
```



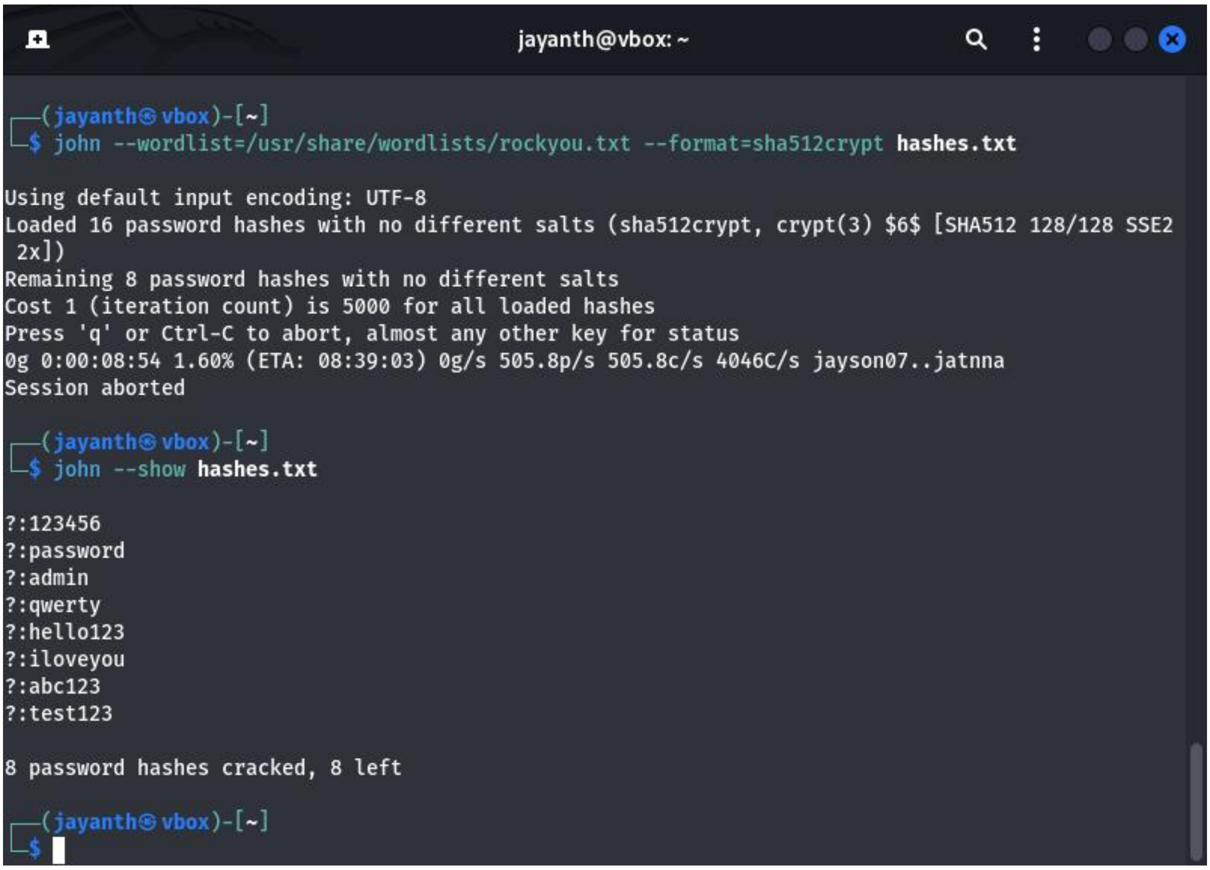
```
┌──(jayanth㉿vbox)-[~]
└─$ cat hashes.txt

$6$xyz123$NEt54QQSEaL6xoUyLjordqIHO/BcII6gQFt4V3wVe7eEhENe7zaIdy6RyZ8dpqr57dWKEdnk.8Cz4.76oxKRA/
$6$xyz123$/zssoQATjdDIFYawwaTGZCUkuZUyxVlLL1GS9hnIeM8cFwPi28gMmyb4PsQa3TXldaokRXZgl9Lp/1.LVQGkq1
$6$xyz123$dyAAvcyDIJsDmv.stFX3Tj/whZPqaavQH0ZgR4S/RvjYSOoCvzlB378YrzILoCBDZxQlcMh8twYfQIkARC.wo1
$6$xyz123$yolKcGNI8ilYjdpO9os3SRu0hXxLNt0DduNL50/4LAsjr0r5gPVQoGlmefLK.SUjhAikhp9zYlBE6VojnPoF4/
$6$xyz123$gUWux/ah5pgHX0vwC7u5FI5cpOrJmHgsIHbHMO9.zZOjqfX2lwR6QJ5kdmLjqYGK4svlV.h8UrTvPamlxQYN3.
$6$xyz123$um9G1WRlDx8v0YBCbZ.Ub6oN39EXWgv3Hx3bAXNRtr7gNjoemKLcGtzCxil8hWg94eMX4zDOjI8kDg7A6rVXx.
$6$xyz123$iYV6F/FzFclI/1Qv.gHDns.Q.6JTEwKehVZUDYa1y0U3YUy5eOJUUyPjx/egp5.7/BqtNPaVW4.RDWW7Iio8P.
$6$xyz123$dtI/VBCLl56HtEtH0CNJKFnmUwAYDqZDU9WpgWMxVaGA53do/PGO9M4Sk6nxSA9HF4WvLnTulP8dTQ1a9yAkf1
$6$xyz123$hGEGQVkPczVjxoGrv6NpXnsVB0v/gRwbYhHZBIRDJNjFUmiDEuJoiXPffYQqlPUnlJ6U.DfRcCFSj23/LPIOi1
$6$xyz123$55StqYvA1a3PlV7KGCa5boSSs2pBN5ybeVGxhinlTQVc5Bu.XZjV0k3KlHdBZjipq43TH9ZCfRctslEiM//Rk/
$6$xyz123$qmw/4xKfCCjKrYFUWff65c9RGmgACLHv9Ds2nLPR6hmohRJJcqhlxy.92BRXJFSbrstKgjYHINnto9HXTIJ731
$6$xyz123$O5kKA5uPfmdicGXrCcyIY0YsR9/gPPEwbD2m6NCKX.GEtUiBy8qCcBPSyb5oJt9v9ChVeVbr8Fmq9Jg0ejxlI.
$6$xyz123$lkqCAX7HuXw852u6zMJj52Q4tCd1nON2s9v4TFkI38mAn.9Rpd5RCh15EuLOWFQnediwWFT.KyNKpvuWJ5ZJ5/
$6$xyz123$dWsWO75k/nTs5fCqUgvc2lWspYa/OZdExN1Bi.UOOU4R2Nd.3c9EMmZTV8J20Ob9jCEFPX.jafnIkqBHSntDh1
$6$xyz123$x6HMDfcL2EeC.rFpfWgEc3zwt.DxETSsALzf/Y2jTJfZNoKE/UdS6ShXM4PVR2S58ctUqTk4QIJhK.IyMOwUj/
$6$xyz123$Hg07PMhJy.OxXbgVMbbLJnQBQC.dFyL9qDy/jETsI84O75rINlJwgs7/J2A.bhjnesjvtAc.1xYRLZ1ATK/7.1
<NULL>

┌──(jayanth㉿vbox)-[~]
└─$ echo "jayanth:$6$xyz123$NEt54QQSEaL6...:19000:0:99999:7:::" > shadow.txt


┌──(jayanth㉿vbox)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt

Created directory: /home/jayanth/.john
```



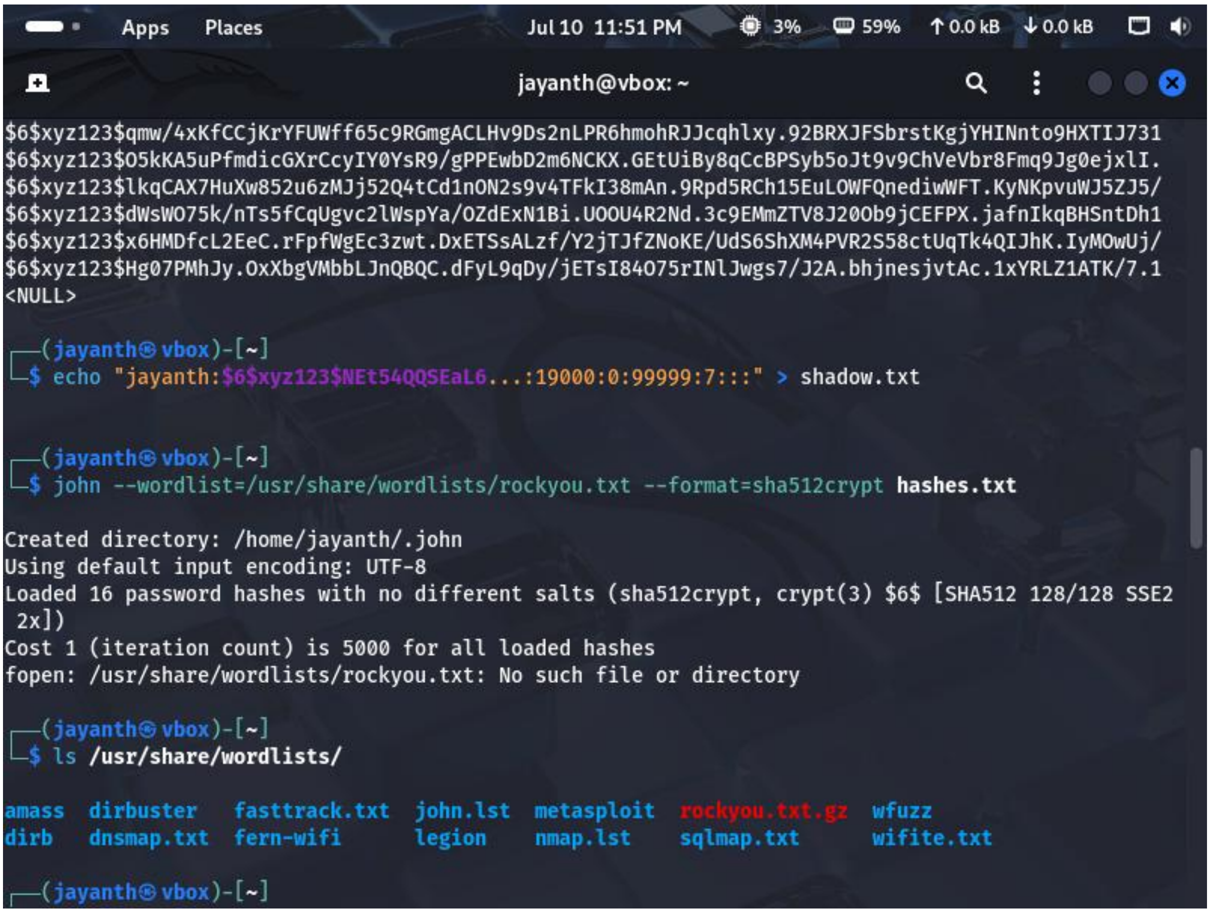
```
$6$xyz123$Hg07PMhJy.OxXbgVMbbLJnQBQC.dFyL9qDy/jETsI84O75rINlJwgs7/J2A.bhjnesjvtAc.1xYRLZ1ATK/7.1
<NULL>

┌──(jayanth㉿vbox)-[~]
└─$ echo "jayanth:$6$xyz123$NEt54QQSEaL6...:19000:0:99999:7:::" > shadow.txt


┌──(jayanth㉿vbox)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt

Created directory: /home/jayanth/.john
Using default input encoding: UTF-8
Loaded 16 password hashes with no different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2
 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
fopen: /usr/share/wordlists/rockyou.txt: No such file or directory

┌──(jayanth㉿vbox)-[~]
└─$ ls /usr/share/wordlists/

amass  dirbuster   fasttrack.txt  john.lst  metasploit  rockyou.txt.gz  wfuzz
dirb   dnsmap.txt  fern-wifi      legion    nmap.lst    sqlmap.txt      wifite.txt

┌──(jayanth㉿vbox)-[~]
└─$ sudo gunzip /usr/share/wordlists/rockyou.txt.gz


┌──(jayanth㉿vbox)-[~]
└─$ ls /usr/share/wordlists/rockyou.txt
```



```
/usr/share/wordlists/rockyou.txt

┌──(jayanth㉿vbox)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt

Using default input encoding: UTF-8
Loaded 16 password hashes with no different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2
 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (?)
password         (?)
iloveyou         (?)
abc123           (?)
qwerty           (?)
hello123         (?)
test123          (?)
admin            (?)
8g 0:00:08:40 1.33% (ETA: 10:09:11) 0.01537g/s 430.6p/s 430.6c/s 3520C/s bryana1..brinker
8g 0:00:08:44 1.33% (ETA: 10:10:39) 0.01526g/s 429.6p/s 429.6c/s 3511C/s aveda1..astone
8g 0:00:08:47 1.34% (ETA: 10:10:48) 0.01517g/s 429.5p/s 429.5c/s 3510C/s PRETTYBOY..Mickey2
8g 0:00:08:51 1.35% (ETA: 10:10:41) 0.01506g/s 429.8p/s 429.8c/s 3512C/s 26062007..25253
8g 0:00:08:52 1.35% (ETA: 10:10:42) 0.01503g/s 429.8p/s 429.8c/s 3512C/s 21111..202121
8g 0:00:08:53 1.36% (ETA: 10:10:41) 0.01500g/s 429.9p/s 429.9c/s 3512C/s 190832..181518
8g 0:00:08:54 1.36% (ETA: 10:10:42) 0.01497g/s 429.9p/s 429.9c/s 3512C/s 140833..132412
8g 0:00:08:55 1.36% (ETA: 10:10:44) 0.01495g/s 429.9p/s 429.9c/s 3512C/s 112469..111159
8g 0:00:08:56 1.36% (ETA: 10:10:45) 0.01492g/s 430.0p/s 430.0c/s 3513C/s 08102006..071118
8g 0:00:08:57 1.37% (ETA: 10:10:26) 0.01488g/s 430.0p/s 430.0c/s 3513C/s 011575..01032526
Use the "--show" option to display all of the cracked passwords reliably
```



```
┌──(jayanth㉿vbox)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt

Using default input encoding: UTF-8
Loaded 16 password hashes with no different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2
 2x])
Remaining 8 password hashes with no different salts
Cost 1 (iteration count) is 5000 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:08:54 1.60% (ETA: 08:39:03) 0g/s 505.8p/s 505.8c/s 4046C/s jayson07..jatnna
Session aborted

┌──(jayanth㉿vbox)-[~]
└─$ john --show hashes.txt

?:123456
?:password
?:admin
?:qwerty
?:hello123
?:iloveyou
?:abc123
?:test123

8 password hashes cracked, 8 left

┌──(jayanth㉿vbox)-[~]
└─$ 
```



```
$6$xyz123$qmw/4xKfCCjKrYFUWff65c9RGmgACLHv9Ds2nLPR6hmohRJJcqhlxy.92BRXJFSbrstKgjYHINnto9HXTIJ731
$6$xyz123$O5kKA5uPfmdicGXrCcyIY0YsR9/gPPEwbD2m6NCKX.GEtUiBy8qCcBPSyb5oJt9v9ChVeVbr8Fmq9Jg0ejxlI.
$6$xyz123$lkqCAX7HuXw852u6zMJj52Q4tCd1nON2s9v4TFkI38mAn.9Rpd5RCh15EuLOWFQnediwWFT.KyNKpvuWJ5ZJ5/
$6$xyz123$dWsWO75k/nTs5fCqUgvc2lWspYa/OZdExN1Bi.UOOU4R2Nd.3c9EMmZTV8J20Ob9jCEFPX.jafnIkqBHSntDh1
$6$xyz123$x6HMDfcL2EeC.rFpfWgEc3zwt.DxETSsALzf/Y2jTJfZNoKE/UdS6ShXM4PVR2S58ctUqTk4QIJhK.IyMOwUj/
$6$xyz123$Hg07PMhJy.OxXbgVMbbLJnQBQC.dFyL9qDy/jETsI84O75rINlJwgs7/J2A.bhjnesjvtAc.1xYRLZ1ATK/7.1
<NULL>

┌──(jayanth㉿vbox)-[~]
└─$ echo "jayanth:$6$xyz123$NEt54QQSEaL6...:19000:0:99999:7:::" > shadow.txt


┌──(jayanth㉿vbox)-[~]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt hashes.txt

Created directory: /home/jayanth/.john
Using default input encoding: UTF-8
Loaded 16 password hashes with no different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2
 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
fopen: /usr/share/wordlists/rockyou.txt: No such file or directory

┌──(jayanth㉿vbox)-[~]
└─$ ls /usr/share/wordlists/

amass  dirbuster   fasttrack.txt  john.lst  metasploit  rockyou.txt.gz  wfuzz
dirb   dnsmap.txt  fern-wifi      legion    nmap.lst    sqlmap.txt      wifite.txt

┌──(jayanth㉿vbox)-[~]
```

22. Group Contributions

- **Dhanwi** – Documentation & testing
- **Jayanth** – Implementation & scripting
- **Hrushikesh** – Troubleshooting & setup
- **Aditya** – Research & formatting

23. Summary

This lab showed how common passwords can be easily cracked. It gave insight into real attack methods and the importance of password security.

24. Conclusion

Password security is critical in today’s digital world. This lab proved that weak passwords are a serious vulnerability, and strong password practices are essential.

---

25. References

- https://www.openwall.com/john/
- https://hashcat.net/wiki/
- https://en.wikipedia.org/wiki/RockYou
- Kali Linux Documentation
- Cybersecurity books and training guides
