

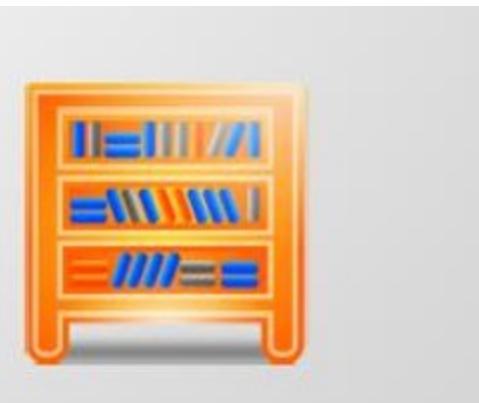
Secure Programming - SECR4483

Session 4: Password Security

Session Learning Outcome

After finish this session you have to be able to:

1. **Explaining** Password Cracking Concepts, Methodology and Password Cracking Techniques
2. **Performing** different types of password attacks
3. **Describing** methods to protect the passwords.
4. **Practicing** Secure Password Programming

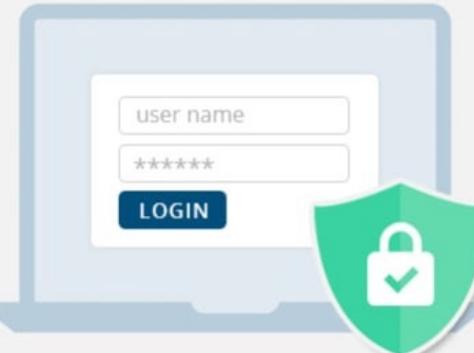


Authentication and Authorization

You can gain access to a system by following these two steps:

1. **Authentication (who are you?)**:
2. **Authorization (What can you do?)**:

Authentication



Who are you?
Validate a system is accessing by the right person

In the authentication phase, you check a username or password against a database of **valid users**. If a match comes up, you move to the second phase.

Authorization



Are you allowed to do that?
Check users' permissions to access data

In the authorization phase, you check the username or password if only passwords are used) against a database to **define how much access that user is granted**.

Authentication Techniques

- Something you know
 - Password, PIN,...etc.
- Something you have
 - Card, Key,..
- Something you are
 - Biometric

Multi factor authentication



Something
you have

Something
you are

Something
you know

Password Storage

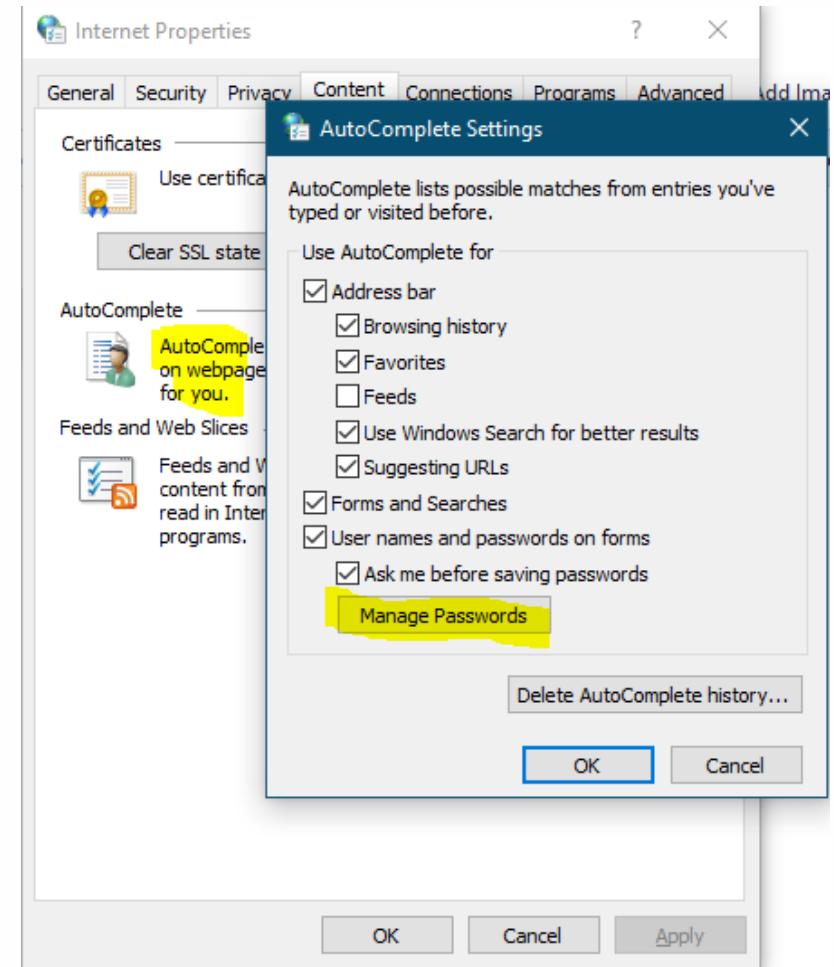
- Local and Remote Authentication
- Storage: files and databases
- Storage:
 - Plain Text
 - Password Hash
- Unix-Based OS Password File (**Demo**)
 - [/etc/passwd](#) (for users info)
 - [/etc/shadow](#) (for passwords)
- Windows Based OS Password File (DB) (**Demo**)
 - [C:/Windows/System32/config/SAM](#)
 - [C:/Windows/System32/config/SYSTEM](#)

Lab Activity (1): Finding Windows Internet Password Storage

1. Press Win + R to open Run.
2. Type **inetcpl.cpl**, and then click OK.
3. Go to the Content tab.
4. Under AutoComplete, click on Settings.
5. Click on Manage Passwords. This will then open **Credential Manager** where you can view your saved passwords.

control panel > user accounts > credential manager

For Mac user: <https://support.apple.com/en-my/HT211145>



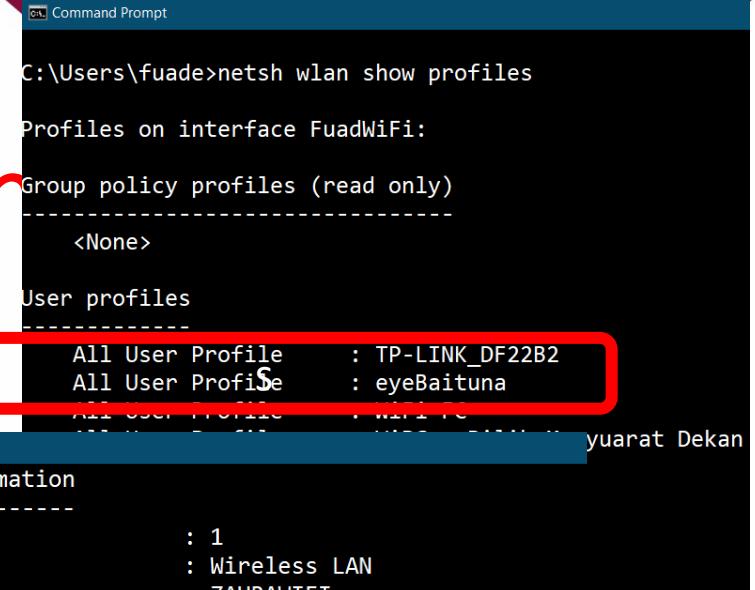
Lab Activity (2): Finding Windows WiFi Password Storage using netsh

- From command line, run the following command:

```
c:\users\fuad>netsh wlan show profiles
```

- To get the password of the specific user profile, run the following command:

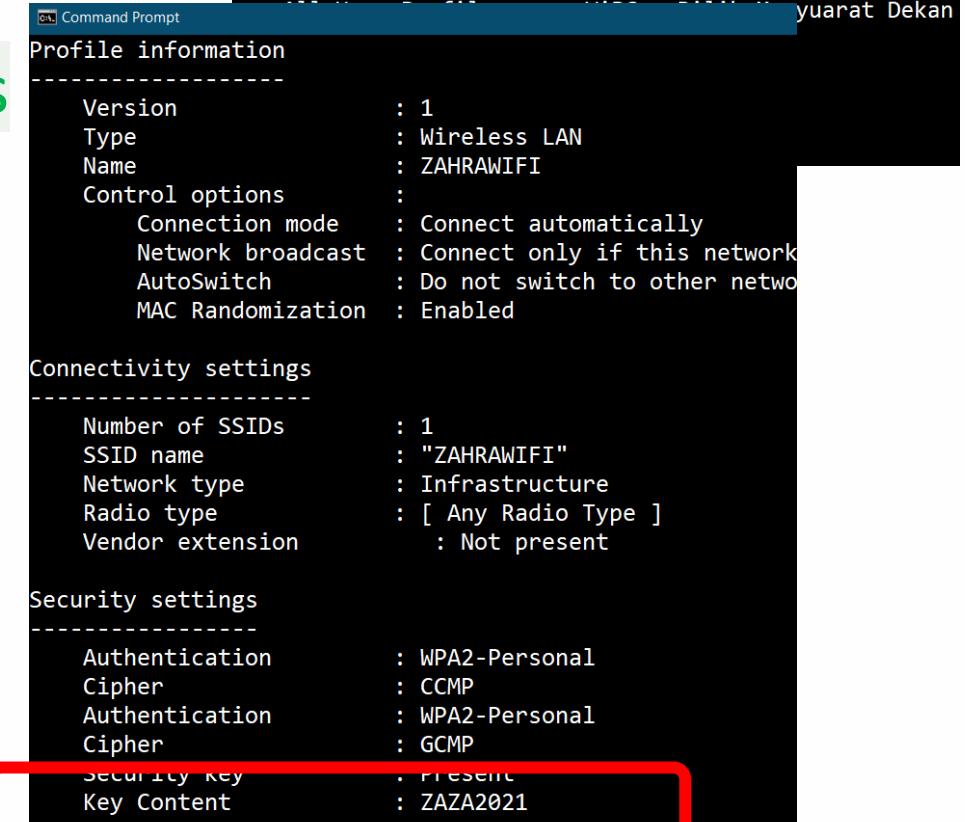
```
c:\users\fuad>netsh wlan show profiles  
name=ZAHRAWIFI key=clear
```



```
C:\Users\fuade>netsh wlan show profiles

Profiles on interface FuadWiFi:
Group policy profiles (read only)
<None>

User profiles
All User Profile      : TP-LINK_DF22B2
All User Profile      : eyeBaituna
```



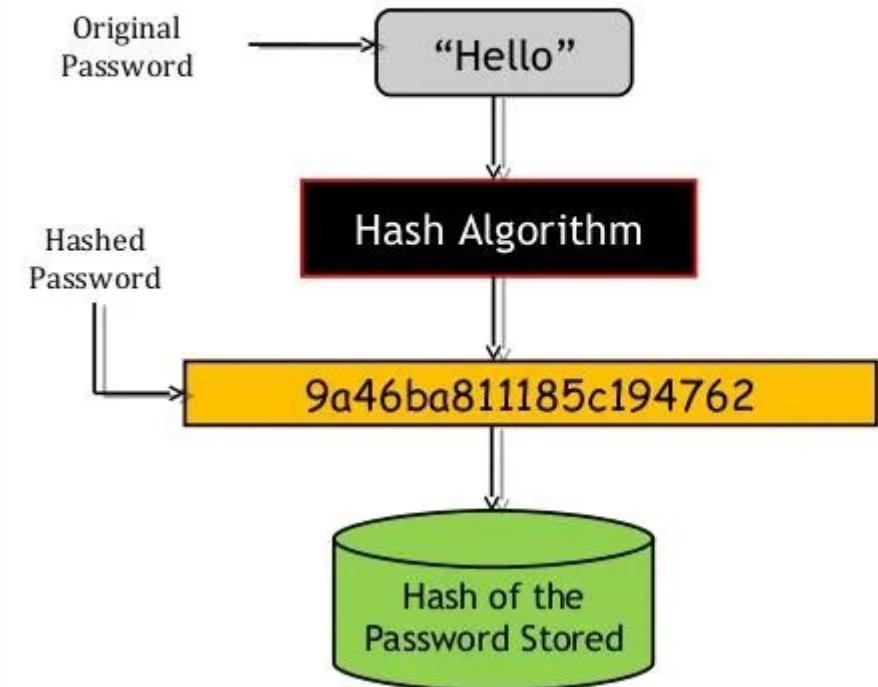
```
Profile information
Version          : 1
Type             : Wireless LAN
Name             : ZAHRAWIFI
Control options
  Connection mode   : Connect automatically
  Network broadcast : Connect only if this network
  AutoSwitch        : Do not switch to other networks
  MAC Randomization: Enabled

Connectivity settings
Number of SSIDs   : 1
SSID name         : "ZAHRAWIFI"
Network type       : Infrastructure
Radio type         : [ Any Radio Type ]
Vendor extension  : Not present

Security settings
Authentication    : WPA2-Personal
Cipher            : CCMP
Authentication    : WPA2-Personal
Cipher            : GCMP
Security key      : Present
Key Content       : ZAZA2021
```

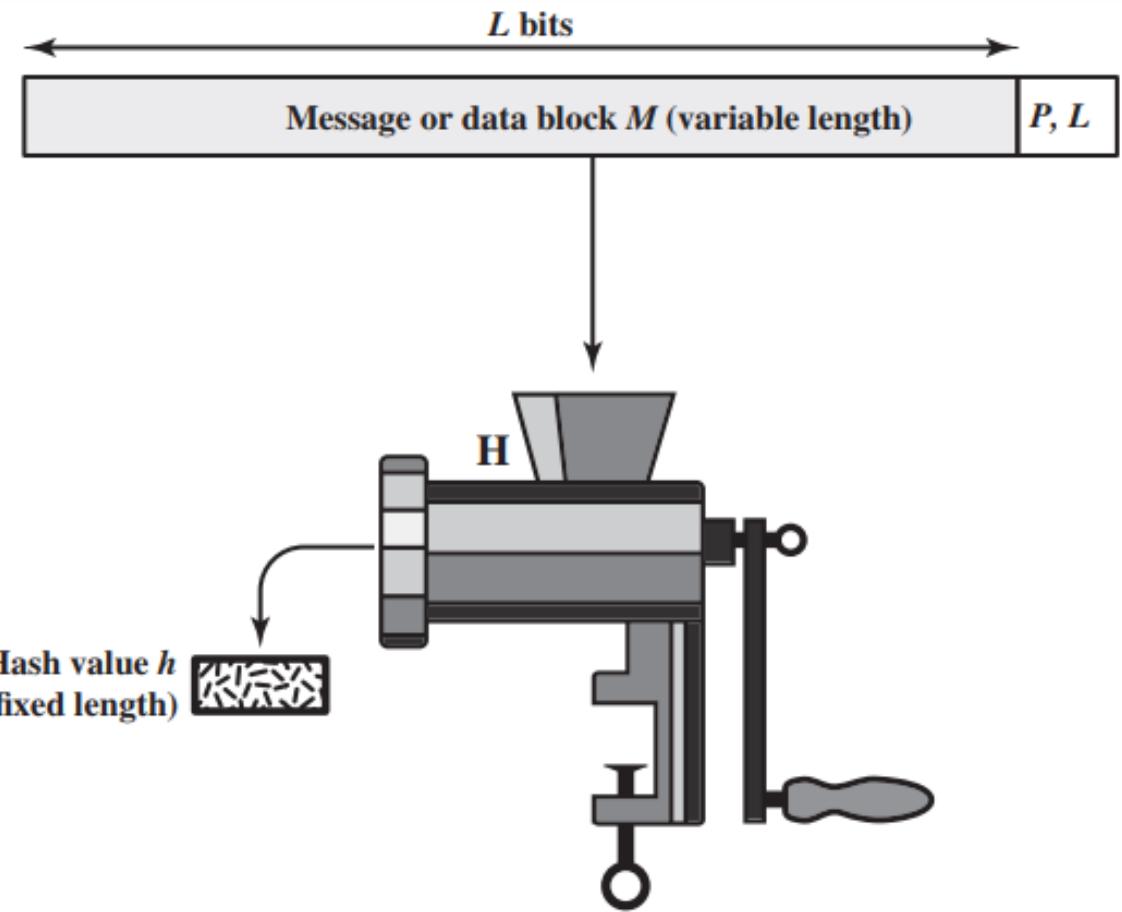
Password Hash

Operating systems store passwords in hidden (encrypted) form so that compromising the id–password list does not give immediate access to all user accounts.



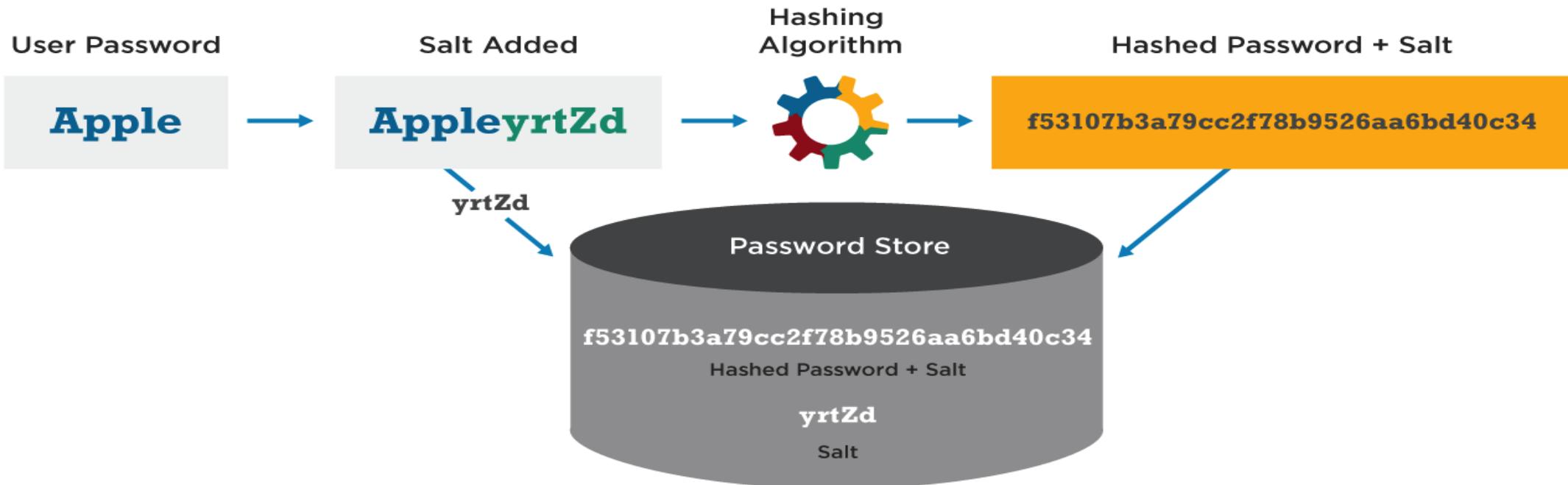
Password Hashing

- A hash function accepts a variable-size message M as input and produces a **fixed-size message digest** $H(M)$ as output.
- **Message Digest (MD)**
 - MD5 was most popular and widely used hash function for quite some years.
 - The MD family comprises of hash functions MD2, MD4, MD5 and MD6.
 - It is a 128-bit hash function.
- **Secure Hash Function (SHA)**, SHA-1 (160-bit), SHA-2 (SHA-256, SHA-384, SHA-512), and SHA-3 (1600-bit)



P, L = padding plus length field

Password Hashing with Salt



The salt serves three purposes:

1. Prevents **duplicate** passwords
2. Increases the difficulty of offline **dictionary** attacks
3. Nearly impossible to tell if a person used the **same password** on multiple systems

Salted, hashed password check



What is your username and password?

My name is john. My password is automobile.



Does
 $h(\text{automobile} | 1515)$
=
ScF5GDhW...
???

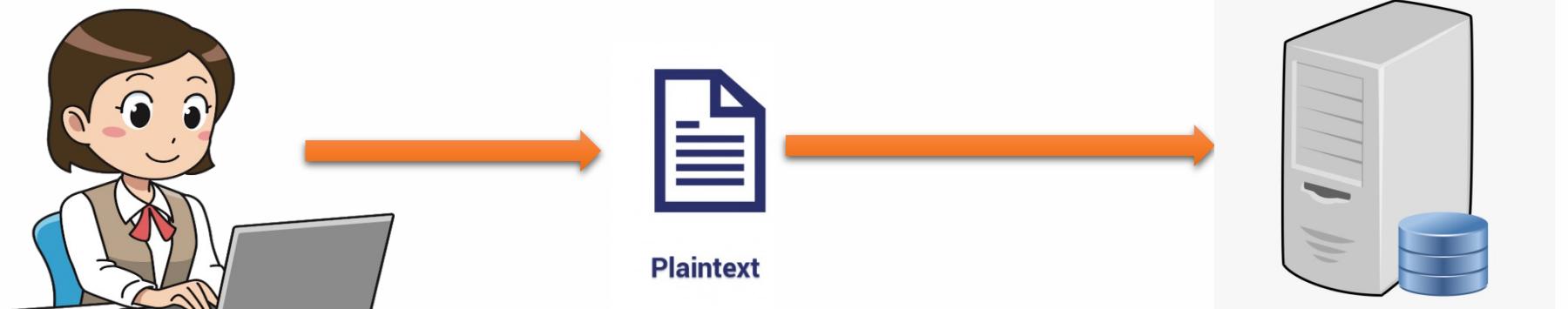
Password File:

john	ScF5GDhW...	1515
mary	9+aPuu2I...	3044
joe	zjK08IL+...	3885

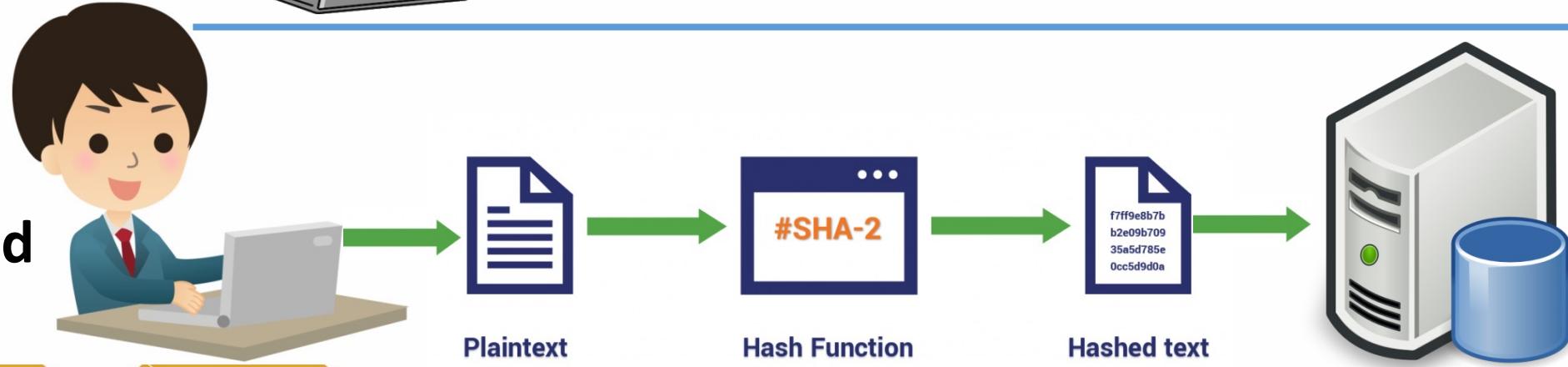
Password Transmission

Passwords are **sent to a system** in one of two ways:

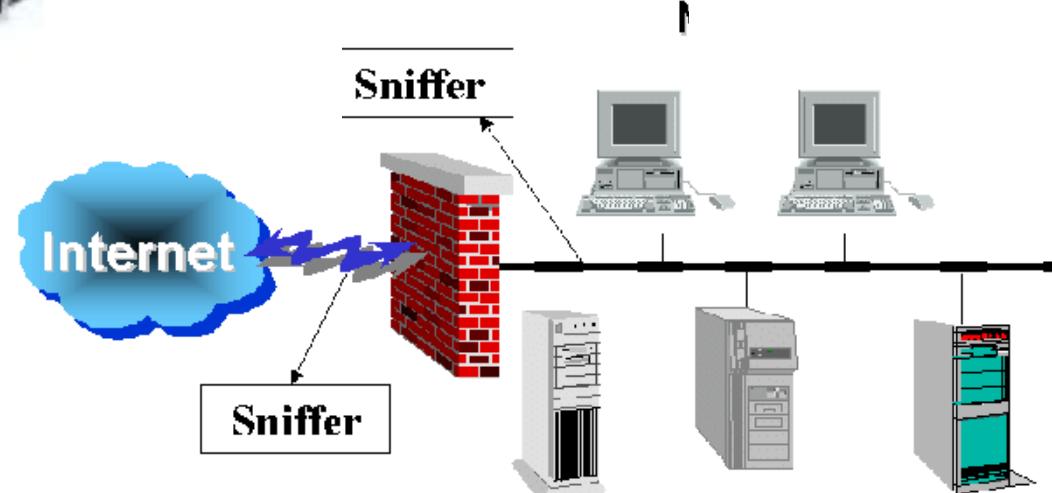
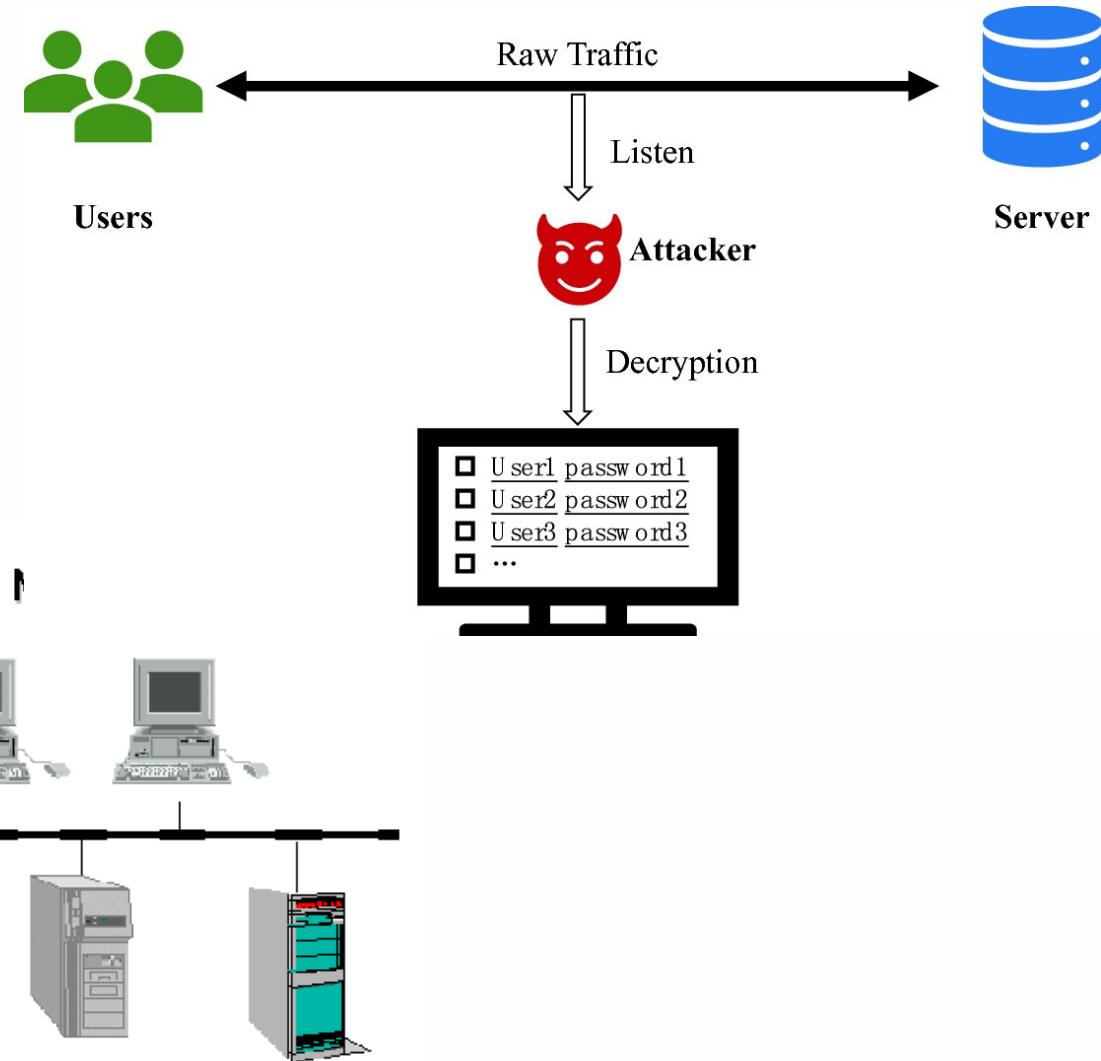
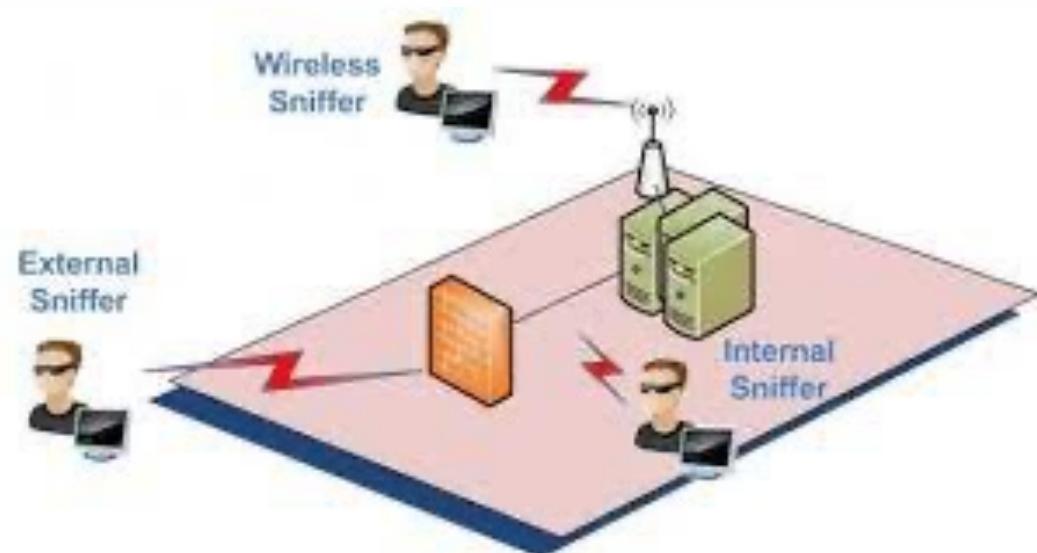
➤ **Clear-text**



➤ **Encrypted**



Password Sniffing



Secure Socket Layer (SSL) and Transport Layer Security (TLS)

HTTP vs HTTPS



Password Attacks and Defense Strategies

Type of Password Attacks

None-Electronic Attacks

- Need no Technical Knowledge

- ❖ Shoulder Surfing
- ❖ Social Engineering
- ❖ Dumpster Diving

Active Online Attacks:

- Need Direct Communication

- Password Guessing
- Dictionary Attack & Brute Forcing Attack
- Phishing
- Trojan /Spyware / Keyloggers

Passive Online Attacks:

- Without Direct Communication

- Sniffing
- Man-in- the middle attack
- Replay Attack

Offline Attacks

- Cracking the password in the attacker machine

- Rainbow Tables (Hashes)
- Distributed Network

None-Electronic Attacks



Shoulder Surfing

Attacker **looks** into users' keyboards or screens while the victim is logging



Social Engineering

Attackers **convince** victims to **reveal** the confidential information



Dumpster Diving

Attackers **search** for sensitive data in victims' trash-bins, printer trash-bins, and desks for sticky notes.

Defend Against Non-Electronic Attacks

- Password Policy
- User Awareness's
- One Time Password
- Multi-Factor Authentication
- Image Authentication



Active Online Attacks



Password Guessing



Dictionary Attacks



Brute Forcing Attack



Phishing



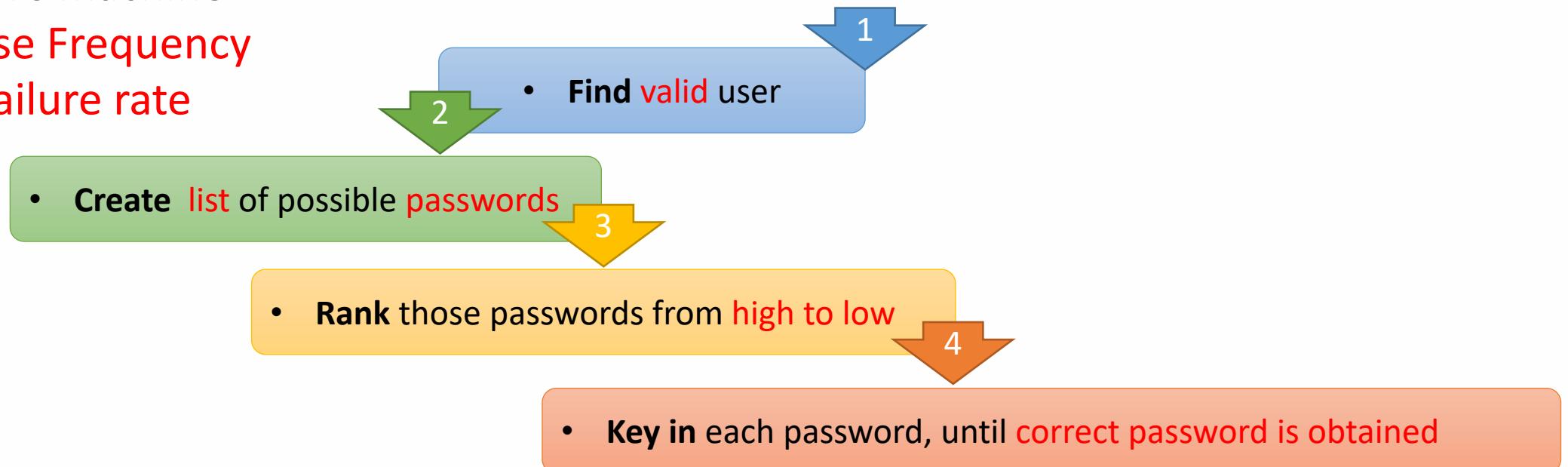
Trojan /Spyware / Keyloggers



Password Guessing

The attacker creates list of **default Passwords** or possible passwords from information collected through social engineering or other means and tries them manually on the victim's machine

- **Low Use Frequency**
- **High Failure rate**



Dictionary Attack

- 1 Most people use **real words (easy-to-remember)** passwords.
- 2 Attackers combine some **dictionary words** to create list of passwords.
- 3 Tools are used to **try login** using every password in the list. The list could be huge (Million+)

Easy to remember is **easy to guess**



Brute Forcing Attack

A brute-force attack is **repeating login attempts** using every possible letter, number, and character combination to guess a password.



The most basic form of brute force attack is an **exhaustive key search**, which is exactly what it sounds like: Trying every single possible password solution (i.e., lowercase letters, capital letters, numbers, and special characters) character by character until a solution is found.

Exhaustive key searches are the solution to **cracking any kind of cryptography**, but they can **take a very long time**.

Lab Exercise 1: Online Password Cracking – Using Burp Suite

1. Download Burp Suite Community Edition.
2. Run the Burp suite and find the Proxy tab
3. Make sure that "intercept is off" and use the embedded web browser to browse the target (victim) login page
4. Enable the intercept "intercept is on" in the Burp Suite and supply any login id and password
5. Right click on the intercepted page and select "send to intruder"
6. In the **intruder** tab select the **positions** and click the clear button on the right
7. Select the username and password variables and click add button
8. In the Payloads tab set the first payload (username) example type admin in the simple list and set brute force for the second payload
9. Start the attack and monitor the change in the response from the server

Lab Activity 1 : Online Password Cracking – Using Burp Suite

Raw Params Headers Hex

```
POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like Mac OS X
Version/5.1 Mobile/9B176 Safari/7534.48.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.16.67.136/mutillidae/index.php?page=login.php
Cookie: showhints=0; reme
dbx-postmeta=grabit=0-,1-
acopendivids=swingset,jotto,phplib2,redmine;
d5a4bd280a324d2ac98eb2c0f
```

Send to Spider
Do an active scan
Send to Intruder
Send to Repeater

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 9
Payload type: Simple list Request count: 18

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	Admin
Load ...	Admin1
Remove	Dave
Clear	User
Add	Pete
Add from list ...	Paul
	Oscar
	Harrison

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions – see help for full details.

Attack type: Cluster bomb

Start attack

Add § Clear § Auto § Refresh

POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like Mac OS X
Version/5.1 Mobile/9B176 Safari/7534.48.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.16.67.136/mutillidae/index.php?page=login.php
Cookie: showhints=0; remember_token=2NkIxJ3DG8iXL0F4vrAMBA;
tz_offset=3600;
dbx-postmeta=grabit=0-,1-,2-,3-,4-,5-,6-&advancedstuff=0-,1-,2-;
acopendivids=swingset,jotto,phplib2,redmine;
acgroupswithpersist=nada;
dsa4bd280a324d2ac98eb2c0fe58b9ed=aplaemed3d0hord07nr13fuv173;
PHPSESSID=29jrpjak954g8k8jlgek9fid23
Connection: keep-alive
Content-Type: application/
Content-Length: 57

Results Target Positions Payloads Options

username=Steats&password=§ Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length
118	Admin	ADMIN	302			39590
442	Admin	Admin	302			39590
9595	Admin	admin	302			39590
8527	User	USER	302			39593
8653	User	User	302			39593
29362	User	user	302			39593
0			200			39432
1	Admin	!@#%	200			39432
2	Admin1	!@#%	200			39432
3	Dave	!@#%	200			39432
4	User	!@#%	200			39432
5	Pete	!@#%	200			39432
6	Paul	!@#%	200			39432

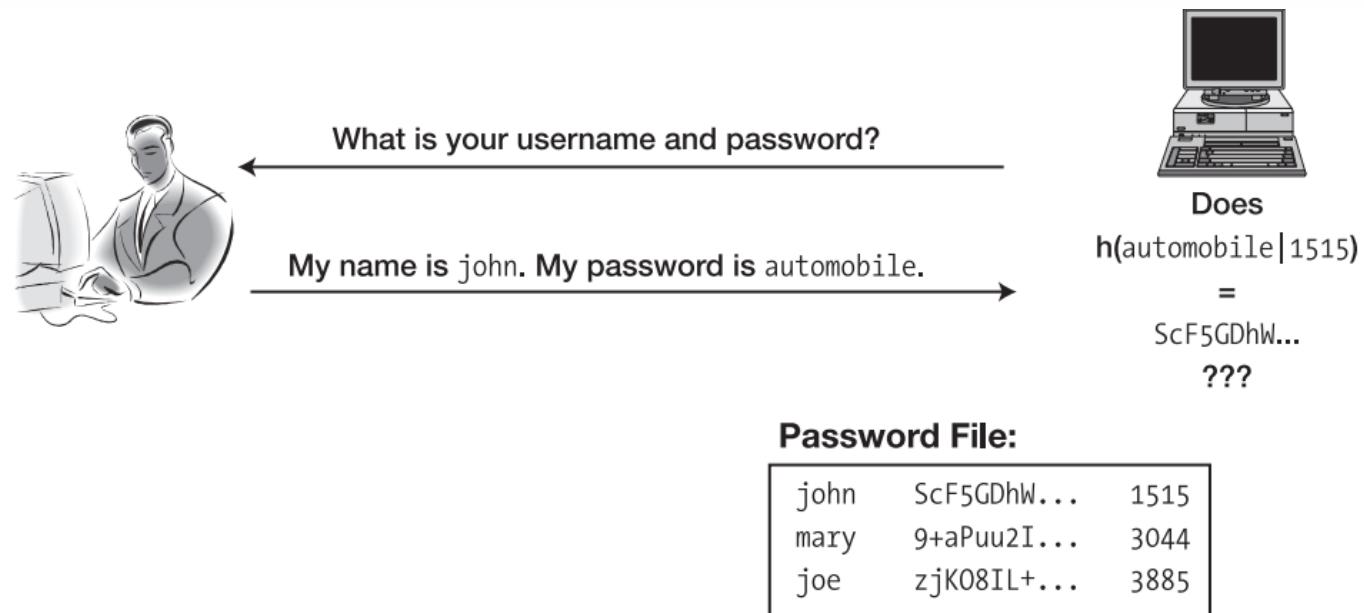
Request Response

Raw Params Headers Hex

?

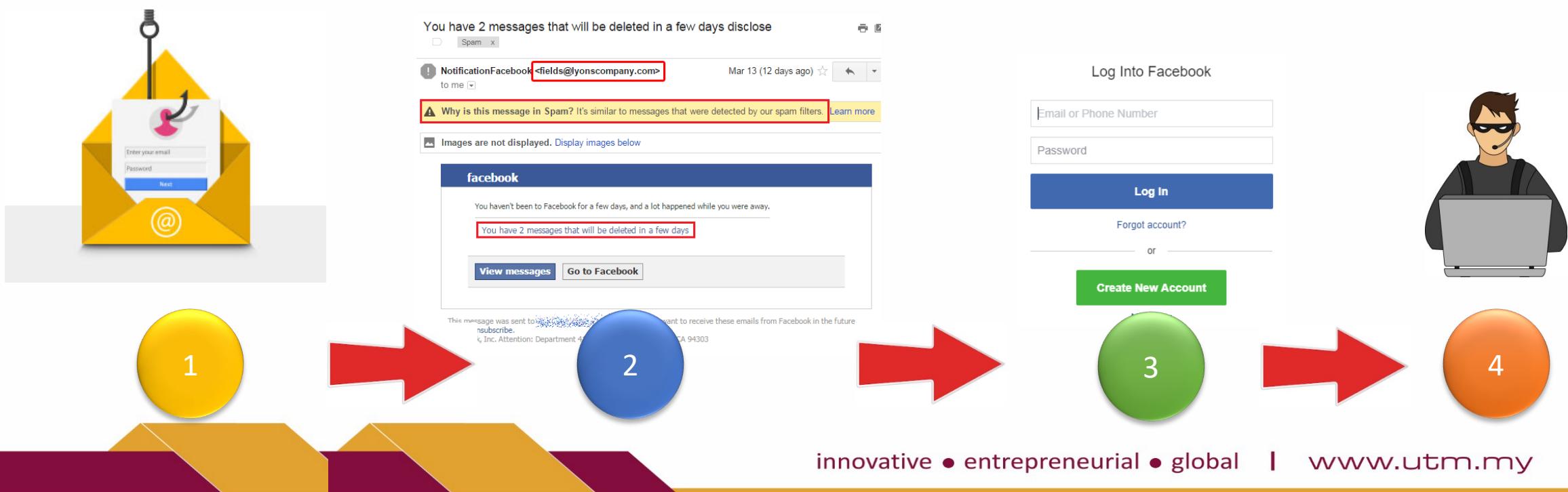
Defend Against Online Attacks: Brute Force-Attack , Dictionary Attack and Password Guessing Attack

- Strong Passwords
- Enforce Password Complexity - Password Filtering
- Use Salting
 - What is the problem?
- “Honeypot” Passwords (IDS)
 - What is the problem?
- Limited Login Attempts
 - What is the problem?
- Artificial Delays
- Last Login
- Image Authentication
- Security Word
- One-Time Passwords



Phishing Attack

- Phishing is a form of social engineering in which you simply **ask someone for a piece of information** that you are missing by making it look as if it is a legitimate request.



Defense Against Phishing Attack

- Phishing Awareness
- Enforce Password Complexity
- Multi-factor authentication
- Image Authentication
- Certificate Authority
- One-Time Passwords

Trojan/Spyware/Keylogger



Spyware is a type of malware allows attackers to **secretly gather information** about a person or organization



Trojan is a type of malware that allows the attacker to remotely gain access to victim machine **to steal stored passwords**



A keylogger is a program that runs in the background and allows remote attacker to **record every keystroke**



Trojan/Spyware/Keylogger



- 1 Attacker installs Trojan/Spyware/Keylogger on victim's machine to collect victim's **user names and passwords**.
- 2 Trojan/Spyware/Keylogger **runs in the background** and send back all user names and passwords to the attacker.

Defense Against Password Thefts

- Secure Storage (Encrypted Hash)
- Temporary Encryption Key
- Multi-factor authentication
- Change Password Regularly

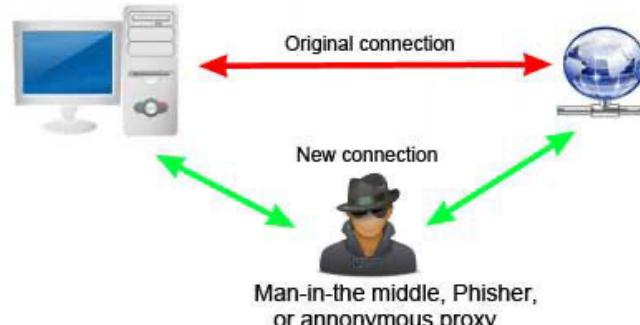
Passive Online Attacks:

Attacker performs password cracking **without direct interaction** with the target system by **eavesdropping** on network **password exchanges**.



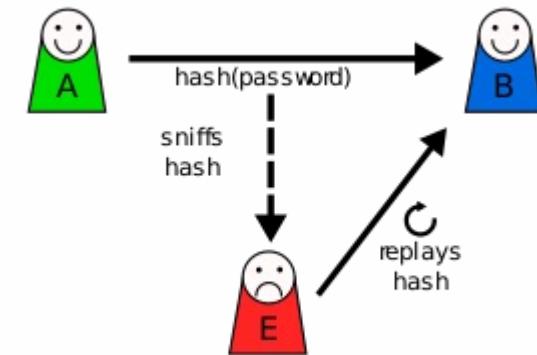
Sniffing

The password is captured during the authentication process.



Man-in-the-middle

A hacker **intercepts the authentication request** and forward it to the server .



Reply Attack

An attacker **intercepts the authentication packet** and then resend it latter to gain access.

Sniffing

Attackers run **packet sniffer tools** on the LAN to access and record the raw network traffic



The captured data may include passwords sent to remote systems during Telnet, FTP, rlogin sessions, and electronic mail sent and received



(1) Password Plain Text – (2) Precomputed Hash – (3) Replay Attack



In a MITM attack, the attacker acquires **access** to the communication channels between victim and server to extract the information

In a replay attack, packets and authentication tokens are captured using a **sniffer**. After the relevant info is extracted, the tokens are placed back on the network to gain access

Considerations

- Relatively **hard to perpetrate**
- Must be **trusted** by one or both sides
- Can sometimes be broken by **invalidating traffic**

Defense Against Sniffing Attacks

- Use Encrypted Wireless Communication
 - Wired Equivalent Privacy (WEP)
 - WiFi Protected Access (WPA), WPA2, WPA3
- SSL and TSL (TCP/IP)

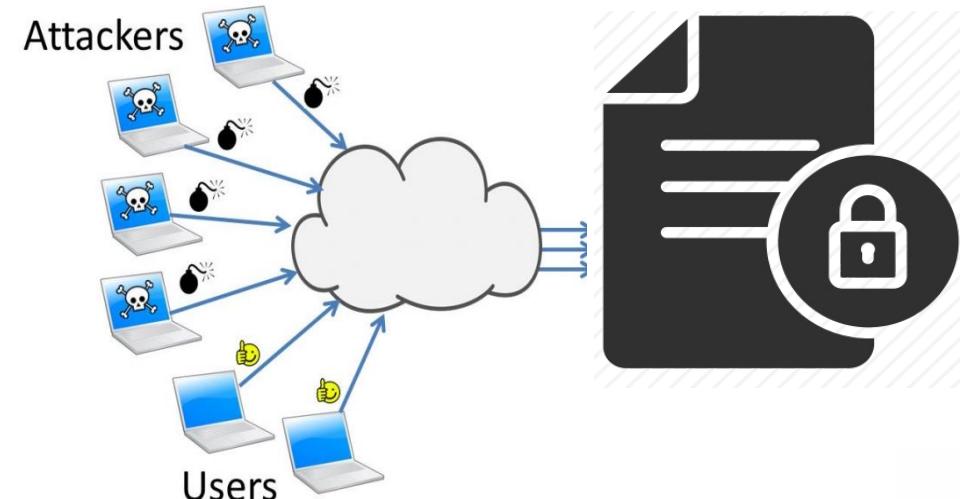
Offline Attack

Offline Password Cracking is an attempt to **recover one or more passwords from a password storage file** that has been recovered from a target system.



Rainbow Table Attack

Attacker **looks** into **users' keyboards or screens** while the victim is logging



Distributed Network Attack (DNA)

DNA uses **the power of machines across the network** or across the world to decrypt passwords.

Lab Activity 2: Offline Password Cracking – Cracking Hashed Password using **John** Tool

- 1) In Kali Linux create a user with name “pentest”

```
root@kali:~/Desktop# useradd -r pentest
```

- 2) Give the user password such as “Pentest123”

```
root@kali:~/Desktop# passwd pentest  
New password:
```

- 3) Use John Tool to crack the passwords of the shadow file

```
root@kali:~/Desktop# john /etc/shadow
```

```
Pentest123:(pentest)-password 1234 Password\ Cracking/  
1g 0:00:00:00 DONE 1/3 (2020-04-26 14:33) 1.612g/s 1058p/s 1058c/s 1058C/s pen  
test123..pentest555 to do! (Password Cracking/.zip)  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed Cracking/ (stored 0%)
```

Defense Against Offline Attack

- Use Complex Password
- Use Secure Hash Function such as SHA 1,2,3 ..etc.
- Use Salting
 - With salting, an attacker has to hash 2^{k+n} strings where n is the password length and k is the salt length. **Where it store the salt?**

Third-Party Authentication

- Web-Based Authentication vs password-based authentication
- Password-based authentication
 - Security issues and the vulnerabilities of password-based methods in untrusted environments
 - usability aspects of remembering passwords
- Web-Based Authentication
 - A single point of failure
 - Privacy Issue: Third-parties give access to a large set of sites, which share and store a large set of potentially sensitive information

Next Session

- Preventing SQL Injection Attacks