Chapter 2

Background

This chapter presents a detailed discussion of the important theoretical concepts on the subject to provide an understanding of the project undertaken.

2.1 Internet Scanning

Internet Scanning is the process of using network scanning techniques to conduct scans on a large scale. Network Scanning is defined as the process of identifying hosts on a network by using features in the network protocol to ping devices or hosts and of analysing information based on the data received by those pings. The basic concept of network scanning is to identify all hosts connected to a network and map them to their IP addresses. This process is achieved by sending packets to addresses and in turn discover what is on the network based on the data received. All the active hosts in the network will respond to this ping while the inactive hosts will have no response. The feedback from this scan gives information about how the hosts behave with internal and external components of a network. Another technique called "Port Scanning" can be used to gather information about open ports that can receive or send information for the identified hosts in the network.
Performing network scans can have both good and bad implications. While some entities use network and port scanning to identify weaknesses, some may use it to exploit the weakness in the network to gain access to information they are not privy to [24].

2.1.1 MaxMind Geolocation IP Databases and Services

Maxmind provides services packaged in APIs and databases that provide accurate IP intelligence data. The web services give the most accurate IP geolocation data and can be accessed through APIs in almost every programming language. The GeoIP2 databases provide in-depth information for IPv4 network blocks and are locally maintained for high volume, low latency purposes and provide the user with unlimited internal use. Since the GeoIP2 databases are now commercial, the project uses the free version called GeoLite2 databases that are slightly less accurate than the former and are updated weekly. The databases include information about the entire IPv4 address space for all countries. These APIs and databases are used for a variety of purposes like detecting network vulnerabilities and fraud detections [17]. The program for this project required a dataset containing IPv4 addresses and their associated country name and country codes for this project. Multiple Maxmind datasets were used to gather this information, and Python was used to generate the dataset we required by combining information from different datasets. To get access to Maxmind, an account was made through the service in order to obtain the license key required to use the services mentioned above.

2.1.2 ZMap and ZGrab

ZMap is an open-source fast single packet network scanner that is capable of scanning the entire IPv4 address space in under 45 minutes from a single mid-range machine. It provides the user with various probe modules including TCP SYN scans, ICMP, DNS queries, UPnP BACNET and can also send UDP probes. Compared to other tools in the market ZMap achieves a better performance due to its architecture which is optimised for carrying out internet-wide surveys [5]. This project makes use of the SYN module for identifying open ports. A TCP SYN scan involves the sending of an SYN packet to open a connection with a host on a specific port. If there is an SYN/ACK response from the host that indicates there is an open TCP/IP port. In case there is an RST response instead of an ACK, that indicates the particular port is closed [10].

ZGrab is ZMap's sister project and an open-source fast application-layer network scanner designed to perform extensive internet-wide surveys. It works in combination with ZMap, but can also be used independently. It provides detailed information about the network handshakes and captures most of the meta-data during a TLS negotiations like TLS certificates and banner information. It is built using Golang and is capable of carrying out the scanning process for all standard protocols like HTTP, SSH, IMAP and more. It provides the output in JSON format for each IP scanned for the selected protocol [6].

2.1.3 Ethical Considerations

As stated above, network and port scanning techniques are carried out for various purposes. While network administrators or academics use these techniques to identify vulnerabilities or weaknesses, cyber attackers can use the same techniques to gain access to systems they are not privy to. Since we are carrying out active scanning to understand the extent of public key reuse, it is important to consider whether there are any ethical implications for the hosts we are scanning. To carry out the scanning process, Durmeric et al. [5] have summarised some practices one needs to consider as it is next to impossible to get permissions from all hosts we scan in advance. Some of these practices include:

- Ensuring scans will not overwhelm the network.
- Providing the nature of the scans in the form of webpages and DNS entries of the scan source.
- Having a clear scope of the project and explain why you need to carry out the scans.
- Limiting scanning when possible.
- Having a simple opt-out method.

In considering the points mentioned above, Dr Farrell carried out these scans using his Virtual Private Server and had a DNS TXT record that indicated the nature of the scans. Also, the project scans hosts that are mail servers and hence, it is less likely to come across sensitive information since individuals do not run most mail servers. The scan rate for the ZMap and ZGrab tools was limited to not cause any disruptions to the active services. The default blocklists that were used and provided by the ZMAP included: local, reserved, and multicast IPv4 addresses.
Another factor that was considered besides the ones stated above was the secure storage and reporting of the data collected. All data was securely stored locally on my machine, and the analysis was also carried out on the same machine to report the key reuse scenarios. All IP addresses were anonymised in this report, and no domain names were released.

2.2 Public Key Infrastructure

Public Key Infrastructure is a framework that comprises of a set of policies, guidelines and technologies that different enterprises, vendors and other entities can use to establish and maintain authentication and confidentiality while communicating over the internet. The PKI works on the core concept of public key cryptography also known as asymmetric encryption that comprises of a public and private

key or more commonly reffred as a key-pair. Consider, for example Bob wants to send a message to Alice securely. Both of them have knowledge about respective their key-pair. The following figure demonstrates how Bob would send a message to Alice using asymmetric encription. A key over here

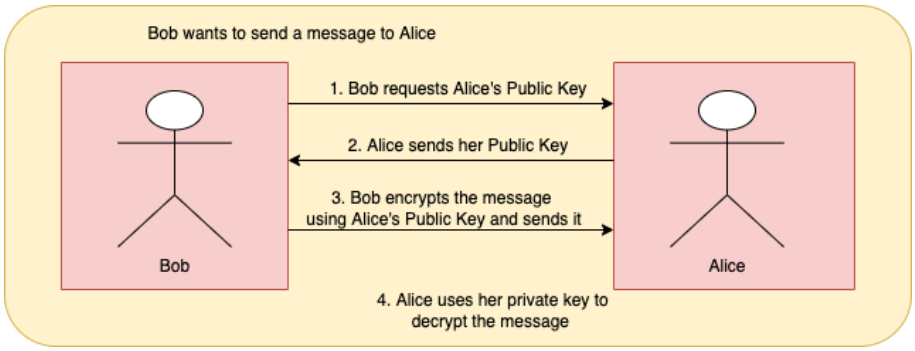

Figure 2.1: Asymetric Encryption

is defined as a randomly generated sequence of bits and the public and private key are closely related to each other by some mathematical operation. But the question that rises here is that how does Bob know that was Alice who sent her public key. How can Bob authenticate the identity of Alice? This is where Digital Certificates play a crucial role as they help in associating a public key with a person or an entity that helps with authentication. These certificates are issues by Certification Authorities or commonly known as CAs. They are usually a third party organisation (Eg: Digicert) that are responsible for issuing, revoking and distribution of certificates. The CA is trusted by all parties involved in the PKI, in this case that would be Bob and Alice.

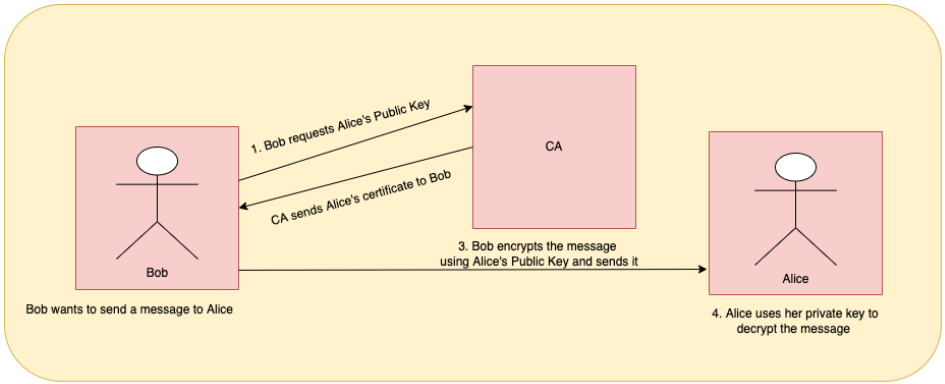

Figure 2.2: PKI

From the figure above we observe how Bob is now asking the CA for the Alice's certificate that in turn contains information about the Alice's public key and since Bob's trusts the CA and the CA is voucing for Alice, this way Alice's identity could be authenticated.

The PKI is deployed in a variety of environments over the internet to secure them, some instances

are web browers, emails, file security etc [20].

2.3 Transport Layer Security

Transport Layer Security (TLS) is a cryptographic protocol used for the secure transfer of data between applications over the internet. In today's day and age, TLS is one of the widely adopted protocols as it is majorly used in web browsers to ensure a secure session has been established. It can also be used for the safe transfer of data for different applications such as e-mail, file transfers, voice-over-IP, as well DNS. TLS does not secure data on the end systems, but is used to facilitate secure data transfer from one point to another over the internet such that no attacker can tamper or eavesdrop while the data is in transit. It is usually implemented over protocols like TCP/IP or UDP layer of the OSI model to secure application layer protocols like HTTP, IMAP, POP3, SMTP etc. TLS was built on the Secure Socket Layer (SSL) protocol and was designed to be its replacement. It is a multilayered protocol that consists of:

- **Handshake Protocol:** Authenticates the parties involved and negotiates an encryption algorithm and parameters.
- **Record Protocol:** Ensures data is not tampered with when in transit using parameters negotiated during the handshake protocol.

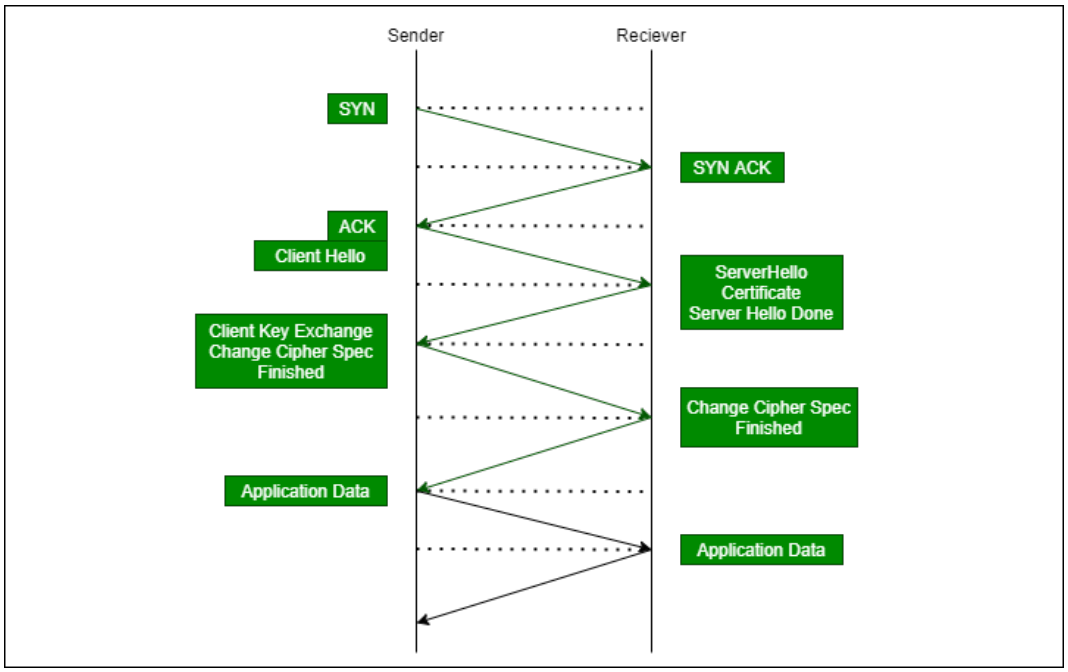

Figure 2.3: TLS Handshake Process

[21]

Since the inception of TLS, it has been deployed in most of the web services on the internet because it protects sensitive information such as passwords, card details, emails, online chats, browsing habits, etc. Since most websites work on a client-server model deploying TLS between the two endpoints protects sensitive information from attackers. TLS also makes use of the technique of asymmetric cryptography which involves a key pair. In this scenario, the public key is used to encrypt the data by the sender and the receiver uses their private key to decrypt the data [28].

2.3.1 Implicit TLS vs Opportunistic TLS

Around the time when TLS was invented, plain text protocols like SMTP, POP3, and IMAP were already deployed heavily over the internet. While many services supported the usage of the *STARTTLS* command to upgrade the connection on the plain text ports, if a client did not support the same information would be transmitted in plain text before encryption was standardised, *STARTTLS* or Opportunistic TLS was used to upgrade plain text connections to a secure ones. Here the connection is upgraded after making the initial connection. To upgrade these plain text protocols, new ports were decided upon and the difference here was to a TLS connection was immediately negotiated between the server and the client. If a server or client did not support TLS and the connection was not established no information would be exchanged between the two. This is known as Implicit TLS. The use of Implicit TLS is preferred over the former in an effort to encourage consistency of how TLS is used [22]. The table below shows the ports used for each protocol using Implicit or Opportunistic TLS as decided upon by IANA [4].

Protocol	Standard Port - No encryption	Implicit TLS Port	Opportunistic TLS Port
SMTP	25	587	25
POP3	110	995	110
IMAP	143	993	143
HTTP	80	443	-

Table 2.1: Implicit TLS vs Opportunistic TLS Ports

2.3.2 TLS Certificates

TLS certificates verify the ownership of a public key and are essential to secure connections and transactions over the internet. They are usually issued by some Certification Authority (CA) by signing the certificates indicating that the CA have verified the ownership. Whenever a user tries to

connect to a server, the server sends them a certificate and then the user verifies the server's certificate using the CA certificate present on the user's machine to establish a TLS connection. The certificates usually contain the following fields of information [3]:

- Subject Domain Name
- Subject Organisation
- Issuing CA
- Anternative Subject Name
- Date of Issue
- Expiry Date
- Public Key
- Digital Signature by the issuing CA

2.3.3 TLS Cipher Suites

A cipher suite is defined as a set of cryptographic algorithms that are used by TLS to encrypt the information. It provides crucial information about securing data when using different network protocols like SMTP, HTTPS, POP3, etc. A cipher will dictate what algorithm is best suitable to make a secure and reliable connection to the server. A cipher suits provides the following information to a server:

- **Key Exchange Algorithm:** Data over the internet is encrypted using a key. This provides the client and server with which Algorithm to use for encryption/decryption of data.
- **Authentication Algorithm:** The server needs to verify the identity of the client before sending or receiving any data. This field specifies that Algorithm.
- **Bulk Data Encryption:** This is to ensure secure transfer of data.
- **Message Authentication Code Algorithm:** A MAC algorithm is sent along the with data to verify the contents of the data.

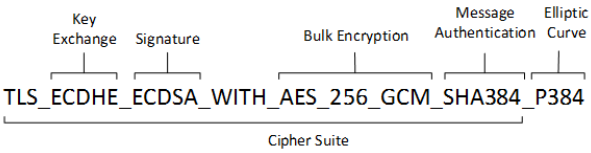

Figure 2.4: TLS Cipher Suite

[19]

2.3.4 Fingerprint SHA-256

The concept of a key-pair was introduced above. A fingerprint is defined as a short sequence of bytes used to identify a longer public key and are generated by applying a hash function on a public key. SHA stands for Secure Hash Algorithm that is used to shorten a data into a smaller sequence. The resulting output cannot be cracked unless a brute force attack is used, and this is where hashing differs from encryption. SHA256 is a popular cryptographic Algorithm that if applied on a sequence of number of "n" bits will return a 256 bit value. It is widely used to in digital certificates and signatures.

2.4 Application Layer Protocols

This section describes the ports we scan and the protocols associated with the ports.

Port	Protocol
22	SSH
25	SMTP
110	POP3
143	IMAP
443	HTTPS
587	SMTP Submit
993	IMAPS

Table 2.2: Ports Scanned

2.4.1 Secure Shell Protocol

The Secure Shell (SSH) is a network communications protocol that enables two computers to communicate and share data remotely. Communication between the two machines is encrypted, meaning

this protocol can be used over an insecure network to make it encrypted. SSH consists of three layers:

- **Transport Layer:** Establishes safe connections between the server and the client for communications after the authentication process has been validated. Oversees data encryption, decryption, integrity, and provides caching and compression if needed.
- **Authentication layer:** Conducts the authentication process i.e. verifies the identity of the user.
- **Connection Layer:** Manages the communication between the two machines once authentication is completed, handles the opening and closing of each session, and also allows for multiple sessions for a user.

SSH requires a login from the user to start performing operations on the remote machine and can be used for the safe transfer of data. It works on a client-server model, i.e. the client will initiate the process by pinging the server, and in turn, the server responds to the client prompting them to finish the authentication process. The SSH server listens on some TCP/IP ports designated for SSH and listens for clients that make contact with this port. Usually, TCP/IP port 22 is reserved for SSH servers and clients contact the server on this port to start the connection process [16].

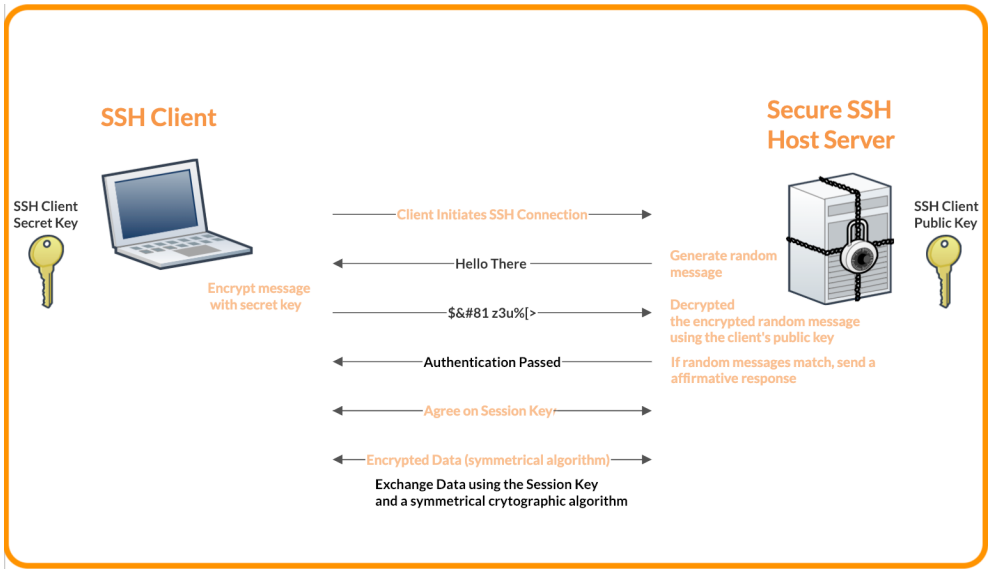

Figure 2.5: SSH Process

[32]

SSH uses asymmetric cryptography similar to the PKI for the authentication process between the client and the server, and uses symmetric encryption and other hashing algorithms to encrypt the data transfer between the client and the server, ensuring privacy and integrity. Each SSH server should have at least one host key to ensure that the client is communicating with the correct server during the key exchange. Ideally, each host should have a unique key, as key sharing between hosts

can leave the host susceptible to man-in-the-middle attacks. However, in some cases, key sharing may be acceptable and even practical (for instance, multiple hosts sharing keys but all under one entity - an Autonomous System). The Secure Shell Protocol is widely adopted these days and used for various purposes by individuals and Corporations. Some use cases are remote access to machines, port forwarding, virtual private networks, and many more.

2.4.2 SMTP/ (S)

Simple Mail Transfer Protocol is used to transfer mail reliably and efficiently. When an SMTP client wants to transmit a message, a two-way connection is established with an SMTP server. The main objective of this protocol is to transfer mail messages to an SMTP server or to report failure to incase it fails to do so. Traditional SMTP operates over assigned port 25 and usually does not provide encryption meaning that the client and server communicate over the internet in plain text [13]. This can be considered a major flaw as all communications susceptible to eavesdropping or man-in-the-middle attacks while emails are in transit. SMTP over TLS or SMTPS was introduced to encrypt these communications. By using SMTP over TLS, one is wrapping SMTP commands inside a TLS connection. Usually, port 587 is used for SMTPS as compared to port 25 for SMTP to distinguish between the two.

With the use of SMTP over TLS, the client waits for the *STARTTLS* keyword from the SMTP server. After that, the TLS handshake protocol is completed. After the handshake is completed the client and the server decides whether to continue ahead with the session based on the current privacy achieved. Some client-servers may decide to continue ahead even if no TLS authentication was achieved as traditionally SMTP operates without any encryption while others may only decide to continue with the session based on certain authentication and privacy achieved. The decision to believe the authenticity of the server-client during a TLS negotiation is a local issue but there are some general rules laid out [12].

2.4.3 POP3 / (S)

Post Office Protocol Version 3 or POP3 is a standard mail protocol that is used by mail servers and their clients to receive emails from a remote server and send it to a local client. It is a client-server protocol in which email is received and stored on a mail server and a recipient or an email client can download emails from that server which enables the client to view the email offline. POP3 is built into most email clients and once the email is on the client then POP3 can be configured to delete the email from the server or to save the email for a specific period enabling clients to download the

mail as many times in that period. The standard port assigned to POP3 is port 110, but usually communication is over plain text on this port but a secure connection can be established using the *STARTTLS* command on this port. In case, the client wants to connect securely POP3 over TLS is used that is assigned to port 995 [29].

2.4.4 IMAP/ (S)

The Internet Message Protocol over IMAP is a standard mail protocol that is used to download mail messages by email clients. The port assigned to the IMAP protocol is 143, but it does not support encryption over that port. To enable encryption the IMAPS protocol is used where the "S" stands for secure and is assigned port 993 [18].

2.4.5 HTTPS

HTTPS or HTTP over TLS is an alternative to simple HTTP over TCP. The difference here is that the HTTP client should also act as a TLS client. The client should initiate the connection, but before making an HTTP request, it should establish a TLS session by initiating the TLS handshake protocol. Once the handshake is completed, the client can start making HTTP requests to the server. All data exchanged between the client and the server is sent as TLS application data. To distinguish between HTTP over TCP and HTTP over TLS, both protocols have been assigned different port numbers. HTTP operates over port 80 while HTTPS operates over port 443 [27].

2.5 Technology Evolution

The development of the surveying program was previously done in 2017/18 using Python2. However, Python3 has gained popularity and has been widely adopted by corporations, individuals, and others. There are few notable differences between the two, and specific Python3 versions provide better performance than Python2 versions. Since January 1, 2020, support for Python2 has been no longer supported. That means there will be no further improvements for Python2 even if a significant bug or security flaw is found [1]. Popular libraries like Pandas, NumPy and many more have officially stopped supporting Python2 versions which makes even more crucial to migrate the current code to Python3 [2].

The previous program used ZGrab1 to run the scans back in 2017/18, and since then, ZGrab2 has been released which is depreciating the previous version. It contains a revamped framework that has simplified the tools' usage and allows individuals to add custom protocols over various ports by building them on their own using Golang. It has integration tests available which can help the development process and can be efficiently run using Docker.

There were also some minor changes to how Maxmind ships the data needed for the program to work. Additional scripts and modifications to current ones needed to be carried out to get the countrywide data to carry out the scans.

2.6 Code Refactoring

Code refactoring can be defined as restructuring code to improve code readability, reduce complexity, and improve the maintainability and efficiency of the program. The refactoring process should contribute to the above factors, but the program's functionality should not be compromised. Refactoring enables developers to gain an in-depth understanding of the program and enables them to expand the program quickly by integrating more features efficiently and reducing the amount of technical debt.

Technical debt, also known as code debt, refers to when development is rushed or when code delivery is prioritised over writing quality code. To ensure code refactoring has yielded benefits, one needs to define a few metrics. Techniques like unit testing and functionality tests need to be performed regularly in order to ensure the refactoring process is beneficial [15]. Refactoring a program should be started by analysing the current code and checking whether refactoring is required. There are a few standard practices defined in the software engineering community to carry out the process of code refactoring. Some of the practices used in the project include:

- **Inline:** Simplifying code by eliminating unnecessary elements.

- **Extract:** Break down code into smaller fragments and then move these fragments into a different method.
- **Abstraction:** Reduce the amount of duplicate code by identifying points of similarity. [9]