Exploring the Vulnerabilities and Countermeasures of SSL/TLS Protocols in Secure Data Transmission over Computer Networks

Fidan BOZKURT
Department of Computer Engineering
National Defence University
Istanbul, Turkey
fbozkurt@hho.msu.edu.tr

Mustafa Kara
Department of Computer Engineering
Air Force Academy, National Defence University
Istanbul, Turkey
mkara@hho.msu.edu.tr

Muhammed Ali Aydın
Department of Computer Engineering
Istanbul University-Cerrahpaşa
Istanbul, Turkey
aydinali@iuc.edu.tr

Hasan Hüseyin Balık
Department of Computer Engineering
İstanbul Aydın University
Istanbul, Turkey
balik@aydin.edu.tr

***Abstract*— The expansion of computer networks and the increase in information sharing over the internet platform have raised the issue of data security during transmission. SSL (Secure Socket Layer) and its more secure and updated version, TLS (Transport Layer Security), are encryption and authentication protocols designed for secure communication over computer networks. Unfortunately, the SSL/TLS protocol structure established to ensure the security of the transmitted data has become a target for malicious attackers. This article will discuss the versions of the SSL/TLS protocols, their operational structure, weaknesses in the SSL/TLS protocol, the attacks carried out through these weaknesses, and the measures that can be taken against them.**

Key words—Computer Network, TLS, Secure Data Transmission, Network Attacks

I. INTRODUCTION

TLS (Transport Layer Security) is a widely-used security protocol designed to protect the confidentiality and integrity of data transmitted over the internet. TLS establishes a secure connection and ensures that people send and receive data encrypted for transmission over the network. TLS protocol is scrambled and unreadable to anyone who might try to intercept it. TLS secures many internet communications, including email, file transfers, and online transactions. It is vital for protecting sensitive information, such as passwords and financial data. In addition to protecting the confidentiality of sending/receiving data, TLS also helps to ensure the integrity of the information people send and read over the internet. This means it helps prevent someone from altering the data in transit, ensuring that people receive the same data. Overall, TLS is an essential tool for protecting internet communications and helping to keep people's sensitive information safe.

Today's developments in the internet world enable the fastest possible access to information, both in individual and institutional terms, while providing end-to-end communication and online transactions. However, they also bring certain security risks. Data transmitted over the internet infrastructure can contain confidential and sensitive content. The SSL/TLS protocol is used to protect data transmission. SSL stands for Secure Sockets Layer and is a cryptographic encryption protocol that enables the encryption and authentication process between the client and server. The TLS protocol, previously known as SSL, encrypts communication between clients and servers on the web [1]. TLS, which stands for Transport Layer Security, can be defined as an advanced version of the SSL protocol [2]. However, both protocols still commonly use SSL as a generic term. The TLS protocol provides encryption, authentication, and data integrity. The security of the TLS protocol is based on robust, well-established cryptographic algorithms [3].

SSL and its successor, TLS, are widely used protocols for securing communication over the internet. However, despite their widespread use and the high level of security, SSL/TLS has several weaknesses that can leave encrypted communication vulnerable to attacks. One such weakness is the susceptibility of SSL/TLS to man-in-the-middle (MITM) attacks, in which an attacker intercepts the communication between two parties and alters it. Another weakness is the lack of proper certificate validation, which can result in an attacker obtaining a valid SSL/TLS certificate for a malicious website and stealing sensitive information from unsuspecting victims. Additionally, the use of weak encryption algorithms and outdated versions of SSL/TLS can also result in vulnerabilities. Furthermore, using SSL/TLS does not guarantee the website's authenticity, as an attacker can mimic the website and steal sensitive information.

II. RELATED WORK

When the literature is examined, A. Aayush et al. [4] explained how the communication between the client and server is performed based on the HTTPS protocol and discussed the cryptographic vulnerabilities of the system. On the other hand, O. Ivanov et al. [5] extensively discussed the TLS protocol versions and the differences between these versions. Vulnerabilities in the TLS protocol, such as BEAST, CRIME, BREACH, and DROWN, were identified, and the measures that can be taken were explained. P. Sirohi et al. [6] chronologically arranged the attacks on the SSL/TLS protocol in the last 22 years. It was emphasized that there are current attacks on the SSL/TLS protocol and that security measures need to be taken. Another study by T. Radivilova et al. [7] revealed vulnerabilities by discussing the tools and methods that analyze SSL/TLS traffic and check whether the traffic is malicious. J. Ćurguz [8] analyzed

the SSL/TLS protocol vulnerabilities in their study. These studies highlight the importance of exploring the vulnerabilities and countermeasures of SSL/TLS protocols in secure data transmission over computer networks. Berbecaru et al. [9] propose a TLS-Monitor tool to detect and prevent TLS attacks. This tool monitors network traffic, detects potential vulnerabilities, and prevents attacks by taking the necessary measures. The tool has been applied and tested for selected attack scenarios. Platenka et al. [10] created Padding Oracle attack models using various SSL/TLS protocol versions with the CipherCAD application. These models have been tested on randomly selected servers to identify weak points that might enable attacks due to poorly implemented protocols. Kumari and Mohapatra [11] examined the performance of the TLS 1.3 protocol, its advantages compared to TLS 1.2, and its potential for improvement. Although TLS 1.3 demonstrated successful efficiency, security, and interaction performance, they emphasize that specific vulnerabilities still exist in current techniques and that improvements are needed to align TLS 1.3 with the targeted standards.

III. SSL/TLS WORKING MECHANISM

TLS is a protocol that provides secure communication over a computer network and is widely used to secure internet communications, including email, file transfers, and Voice-over IP (VoIP). The TLS protocol establishes a secure connection between two parties, a client and a server. The connection is established using a handshake process, during which the parties exchange messages to establish the details of the connection, such as the protocols and cryptographic algorithms used. Once the connection is established, the parties can communicate securely over the network by exchanging encrypted messages. The encryption keys used to encrypt and decrypt the messages are generated during the handshake process and are unique to that connection. In summary, the TLS protocol provides a mechanism for establishing a secure connection between two parties and exchanging encrypted messages. Correspondingly, it helps to ensure the exchanged messages' confidentiality, integrity, and authenticity.

A. SSL/TLS Protocol Structure

The SSL/TLS protocol is composed of two layers and several protocols. As shown in Figure 1, the SSL/TLS protocols work between the application layer and the transport layer [12]. The SSL/TLS protocol consists of the handshake layer and the record layer. The handshake layer contains the Handshake, Change Cipher Spec, and Alert protocols, while the record layer contains the Record protocol. The SSL/TLS protocol is widely used in web sites and web applications along with the HTTP protocol. In addition, it is used by many other services and protocols such as email (SMTP, POP, and IMAP protocols), FTP, virtual private networks (SSL/TLS VPNs), and network devices.

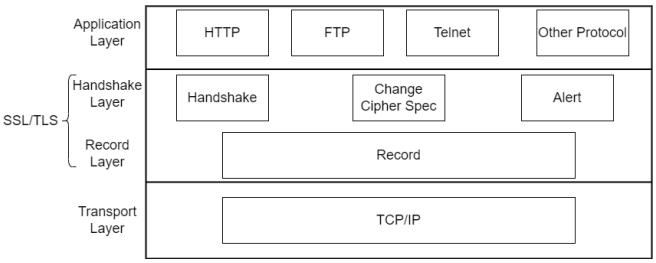

Figure 1: SSL/TLS Protocol Layers [13]

The Record protocol ensures the secure connection by encapsulating various protocols. The Handshake protocol is the protocol where the client and server authentication is performed. Before data communication, it allows encryption keys to be used by using encryption algorithms. Communication between the server and client using encrypted data is achieved through the SSL/TLS Handshake protocol [15].

The handshake process is explained in the following steps [13]:

- The client sends a Hello message to the server. The Hello message contains information such as RandomNumber, SessionID, Ciphersuite, ClientTLSversion, and CompressionMethod.
- The server receives the Hello message from the client and sends a modified Hello message to the client, changing the CipherSuite and RandomNumber information.
- The client responds with a key exchange message to establish communication for the session.
- Sends the ChangeCipherSpec message to the server to indicate that future messages will be encrypted and authenticated by the client.
- Secure communication between client and server starts with the Finished message.

B. Session Resumption

Session renewal may be requested by the server or the client.

1) The server that wants to refresh the session:

When responding to the ClientHello message, the server conveys the SessionID information assigned to the session when it was first established to the client through the ServerHello message. This information allows the server and client to quickly start the session by using the key sharing information used in a previous handshake scenario.

2) If the party that wants to renew the session is the client: The client sends the SessionID information to the server in the ClientHello message. If the server returns the same SessionID information in the ServerHello message, it means that the session has accepted to be renewed. The server and client create a new key value by using the previously agreed master secret value. This way, the network load is reduced and handshaking is performed faster [16].

C. Cipher Suites

A cipher suite is a structure that enables secure communication between client and server, by mutually supporting it. Figure 2 shows the main categories for cipher suites in TLS 1.2 version. The working sequence for the main categories is: Authentication, Key Exchange, Bulk Cipher-Encryption, Message Authentication Code (MAC) - Hashing Algorithm, in that order.

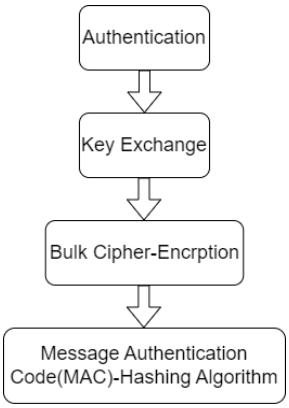

Figure 2: Working Order of Cipher Suites Main Categories

Authentication specifies how the server and client will perform authentication. *Key Exchange* determines the way symmetric keys will be exchanged. *Bulk Cipher-Encryption* specifies which symmetric key algorithm will be used to encrypt actual data. *Message Authentication Code (MAC) – Hashing Algorithm s*pecifies the method to be used to perform data integrity checks for the connection.

Table 1 shows the algorithms used in the Cipher Suite.

Table 1: SSL/TLS Cipher Suite Algorithms

Function	Algorithm
Key Exchange	RSA, Diffie-Hellman(DH), ECDH, ECDHE, SRP, PSK
Authentication	RSA, DSA, ECDSA, PKCS#1 v1.5, RSA-PSS(PKCS#1 v2.1.)
Batch Encryption	RC4, 3DES, AES, CBC
Message Authentication	HMAC-SHA256, HMAC-SHA1, HMAC-MD5

The TLS 1.2 protocol process is indicated to start with ECDHE, where the keys are temporarily changed through the Elliptic Curve Diffie Hellman (ECDHE) algorithm during the handshake. RSA manages the authentication algorithm process. AES_128_GCM is the bulk encryption algorithm with a 128-bit key size running the AES in Galois Counter Mode (GCM), and SHA-256 is stated to be the hashing algorithm used—the TLS 1.3 Cipher Suite structure. There are only two algorithms in this cipher package, encryption, and hash. During the TLS handshake, less communication between the client and server results in a faster handshake. There are two ways to combine MACs and CBC mode ciphers. The plain text is encrypted first, then the encrypted text uses MAC, or the plain text uses MAC first, and then everything is encrypted. In TLS 1.3, RSA-PSS (PKCS#1 v2.1) instead of AEAD and PKCS#1 v1.5, which combines encryption and integrity in a single seamless process, increases the security of the cipher package.

IV. VULNERABILITIES AND PRECAUTIONS ON THE SSL/TLS PROTOCOL

Although the SSL/TLS protocol protects data, security vulnerabilities in the protocol make it weaker and can lead to man-in-the-middle attacks. The security vulnerabilities in the SSL/TLS protocol and the attacks that exploit these vulnerabilities are discussed below, along with the measures taken. There are a few known cryptographic weaknesses in SSL/TLS, the protocols used to secure internet communications. Some of the most notable ones include such as Poodle, Heartbleed, Freak, Logjam.

A. Version Rollback Policy

It is generally not a good idea to allow version rollback, as it can introduce system vulnerabilities. When a new software version is released, it typically includes security fixes and other updates that address known issues. Allowing users to roll back to an older software version means they will no longer have access to these security updates, potentially leaving their systems open to attack. Instead of allowing version rollback, it is generally recommended to encourage users to update their software. This ensures that they have the latest security fixes and other updates, which can help to protect their systems against known vulnerabilities. Examples of this attacks include the Poodle, Freak, Beast, and Sloth attacks.

*Poodle: (*Padding Oracle On Downgraded Legacy Encryption) This vulnerability allows an attacker to decrypt certain parts of an SSL/TLS session by repeatedly sending specially crafted packets to the server and exploiting a weakness in the way the server handles padding bytes. An attack that aims to downgrade the TLS connection to SSL 3.0 is carried out. After the connection is downgraded, an attacker only needs to make 256 requests over SSL 3.0 to break one byte of encrypted communication. This vulnerability led to the widespread removal of SSL 3.0 from use on the internet [5].

Sloth: That is carried out by forcing the client or server in the TLS 1.2 client authentication process to downgrade to weaker hash algorithms such as MD5 and SHA1 [17].

Freak: (Factoring Attack on RSA-EXPORT Keys) This vulnerability allows an attacker to force a client and server to use a weaker version of the RSA cipher, which can then be easily broken. Attackers allow the server to switch from a standard RSA encryption packet to an export-class encryption packet, instead of fully downgrading the protocol version. This is targeted towards SSL and TLS applications that allow export-grade encryptions, which use RSA encryption more frequently. When the server switches to a less secure encryption packet, attackers can access the

packet's decryption key, decrypt the packet, and inject traffic [8].

Logjam: The Diffie-Hellman key exchange is a widely used method for securely exchanging cryptographic keys [8]. This attack is launched against servers using TLS with Diffie-Hellman key exchange. Attackers using the man-in-the-middle approach force the server to use the 512-bit Diffie-Hellman key exchange algorithm in the TLS protocol.

Beast: The BEAST (Browser Exploit Against Advanced Evasion Techniques) attack is a vulnerability that affects the SSL 3.0 and TLS 1.0 protocols. This attack aims at network security vulnerabilities in TLS 1.0 and SSL versions. The attacker exploits the cipher-block chaining (CBC) mode encryption to allow decryption of the content of an SSL or TLS encrypted session. It allows an attacker to decrypt certain parts of an encrypted communication by forcing the client and server to use a vulnerable version of the TLS protocol.

B. Cipher Suite/ Legacy Cipher Suites

The older versions of the SSL/TLS protocol have weaker encryption algorithms. This leads to the weakening of the protocol over time [4]. The RC4 attack was carried out exploiting this vulnerability.

RC4: The RC4 (Rivest Cipher 4) attack is a vulnerability that affects the RC4 cipher, a widely used cryptographic algorithm. The attack allows an attacker to recover the key used to encrypt data by analyzing patterns in the encrypted data. The vulnerability was discovered in 2013 and affected the RC4 cipher when used in conjunction with the SSL 3.0 and TLS 1.0 protocols. While it has been largely mitigated by using newer versions of TLS, such as TLS 1.1 and TLS 1.2, which do not use RC4, it is still a concern for systems that continue to use SSL 3.0 or TLS 1.0. The symmetric encryption used in SSL version 3.0 is the RC4 stream encryption method. The RC4 method is considered insecure because it leaks information in the case of multiple encryptions of the same message.

C. Compression

Compression algorithms work by removing repetitions. If a character in the main text repeats two or more times, the compressed text includes the symbol of the character, its frequency in the original text, and its location. In compression algorithms, frequently repeating characters are replaced with shorter symbols (less than one byte), while rare characters are replaced with larger symbols (more than one byte) [18]. The Crime and Breach attacks were carried out exploiting this vulnerability.

Crime Attack: The SSL/TLS protocols use data compression formats such as DEFLATE and gzip [6]. Crime attacks are brute force attacks that exploit a leak in compressed SSL/TLS communications.

Breach Attack: The Breach attack takes advantage of compression algorithms used in the SSL/TLS protocol, similar to the Crime attacks. The difference between Crime attacks and Breach attacks is that Breach attacks target HTTP compression, attacking HTTP responses [17]. TLS is commonly used to secure web traffic and other types of internet communication but some attack such as TLS breach, which is a type of attack on a computer network that targets the security protocols that protect internet communications can be done.

D. Time Delay Response

The Timing Attack is an attack that looks at how long it takes a system to perform an operation and uses statistical analysis to find the correct password decryption key and gain access. The only information the attacker needs is the timing information generated by the application's algorithms. The attacker can guess the correct input by providing various inputs to the application, timing the process, and analyzing the information statistically. The Lucky 13 attack was carried out taking advantage of this vulnerability. The Lucky 13 attack is a timing attack that can be used against implementations of the TLS protocol using the CBC mode of operation. In a session with a cipher packet created using CBC, small timing differences occur during the decryption process. The Lucky13 attack exploits these timing differences.

E. Data Padding

The data padding problem for SSL/TLS protocols refers to a security vulnerability that arises from how data is padded or expanded before being encrypted by the protocol. In SSL/TLS protocols, data is padded to ensure that encrypted data has the same length, making it harder to detect patterns in the data and thus making it more difficult for attackers to mount specific attacks. However, if padding is implemented in a predictable or vulnerable manner, an attacker may exploit the vulnerability to uncover information about the encrypted data.

HeartBleed: This vulnerability is a security flaw found in version 1.0.1 and 1.0.1f of the widely used OpenSSL library for TLS implementation [19]. It is a feature that helps control the connections between client and server. The attacker can fool the other party by claiming that the data being sent has a false length, causing the responding party to attempt to send data of the length specified by the attacker, thereby revealing sensitive information such as user names and passwords from the responding party's RAM memory [5,16].

Robot: An attack that allows a TLS server to perform RSA decryption and signature operations with its private key [5]. It is a vulnerability that allows an attacker to perform RSA decryption and signature operations with the private key of a TLS server. This attack exploits weaknesses in the implementation of the SSL/TLS protocols, specifically in the handling of error messages, to perform a man-in-the-middle attack and steal sensitive information.

Drown: Decrypting RSA with Obsolete and Weakened eNcryption is a type of cyber attack that targets servers using the SSL or TLS protocols. The attack takes advantage of a vulnerability in SSL version 2.0, which is an outdated version of the SSL protocol. DROWN works by exploiting

the use of outdated encryption ciphers, weak encryption keys, and poor implementation practices, which can allow an attacker to break the encryption used by the server and steal sensitive information. DROWN attacks can be prevented by disabling SSL v2.0, using stronger encryption ciphers, and implementing best practices for secure configuration of SSL/TLS protocols.

V. The countermeasure for TLS attacks

There are several measures that can be taken to protect against TLS attacks:

Use strong, up-to-date encryption: The use of robust and up-to-date encryption refers to employing cryptographic algorithms with a high level of security to protect data transmitted over the internet. Encryption is a method of converting plaintext data into an unreadable ciphertext format, which helps to protect the confidentiality and integrity of the data. Using robust and up-to-date encryption makes it much harder for attackers to intercept or alter data in transit, even if they can gain access to the network.

Table 2 shows attacks against the SSL/TLS protocol, and the precautions that can be taken against these attacks are mentioned.

Table 2:Attacks and Precautions

Attack	Precautions
Poodle	Disabling support for SSL 3.0
Sloth	Discontinuing use of weak hash algorithms like MD5 and SHA1
Freak	Rejecting RSA_EXPORT cipher suites at the server side
Logjam	Discontinuing use of DHE_EXPORT algorithm
Beast	Not using SSL 3.0 or lower versions or TLS 1.0 at the server side and CBC algorithm does not support block ciphers
RC4	Not using RC4 encryption algorithm
Crime	Disabling SSL/TLS compression
Breach	Disabling HTTP compression
Lucky 13	Adding random time delays to counter cipher-block chaining mode decryption to prevent statistical analysis and Using AEAD ciphers like AES-GCM
HeartBleed	Not responding to Heartbeat messages by the server/client and Discontinuing use of OpenSSL library versions 1.0.1 and 1.0.1f
Robot	Discontinuing use of RSA encryption algorithm
Drown	Disabling support for SSL 2.0

Use trusted SSL/TLS certificates: Using trusted SSL/TLS certificates ensures that the website or server the computers are communicating with is authentic and not an impostor. Trusted SSL/TLS certificates are issued by third-party certificate authorities, who have verified the identity of the website or server owner. Using a trusted SSL/TLS certificate, the computers can be assured that the website or server they are communicating with is whom they say they are and that their sensitive information, such as passwords and financial data, is transmitted securely and protected from malicious actors. In addition, trusted SSL/TLS certificates also provide encryption of the transmitted data, which helps to protect the confidentiality of the information being sent and received over the internet.

Enable HTTPS: HTTPS (Hypertext Transfer Protocol Secure) is a protocol for secure internet communication. It provides a secure and encrypted connection between the user's web browser and the website they are visiting. The encrypted connection ensures that any information transmitted between the browser and website, such as login credentials or sensitive data, is protected from eavesdropping or tampering by third parties. HTTPS also helps to prevent man-in-the-middle attacks, as the encryption ensures that a third party cannot intercept and alter the data.

Keep software and security protocols up to date: Keeping software and security protocols up-to-date is a critical step in maintaining computer system security and includes updating the operating system and any other software installed on the system, such as web browsers and other applications. In addition, updating software and security protocols can help to address any newly discovered vulnerabilities that attackers may have exploited to gain unauthorized access to sensitive information.

Train employees to recognize and report potential threats: Educating employees about the importance of internet security and how to recognize potential threats can help to prevent attacks from occurring.

Use a firewall: A firewall is a barrier between a private internal network and the public Internet. By implementing a firewall, the system admin can restrict access to the network from unauthorized sources and prevent attackers from accessing sensitive information transmitted through the TLS protocol.

Use a VPN: A virtual private network (VPN) can help to secure your internet connection by encrypting all traffic and routing it through a secure server.

VI. End to End Protection via TLS

TLS is a secure communication protocol that provides end-to-end protection for data transmission between two hosts. It is important to differentiate between "hop-by-hop" protection and end-to-end protection. Hop-by-hop protection refers to the protection of data between routers as it passes through different networks, whereas end-to-end protection refers to the protection of data from the source to the final destination, bypassing any intermediate network nodes. TLS operates over a TCP connection between two end hosts, providing a secure channel for data transmission, making it an end-to-end secure protocol. With TLS, data is encrypted and decrypted at the source and destination, respectively, ensuring that sensitive information is kept confidential and protected against eavesdropping and tampering.

TLS is a widely used protocol for ensuring secure communication over the internet. It provides a secure channel between two endpoints, such as a client and a server. One of the key features of TLS is the use of certificates from trusted third-party organizations, known as TTPs, as an authentication mechanism. These certificates serve as proof of the identity of the parties involved in the communication and allow them to verify each other's authenticity. This helps prevent man-in-the-middle (MITM) attacks, where an attacker intercepts and potentially modifies communication between the parties. By using certificates, TLS ensures that an attacker should not be able to forge a valid certificate and impersonate one of the parties involved in the communication. This makes TLS an essential component in securing online transactions, such as online banking and e-commerce, where the protection of sensitive information is of utmost importance. Third-party certificates as an authentication mechanism certificates are issued by a trusted third-party (TTP), who acts as a trusted authority and confirms the identity of the parties involved in the communication. In cases where the TTP is offline, there is no bottleneck or single point of failure issue. This ensures the secure flow of information between two parties, even in cases where the TTP is unavailable. The use of TTPs in TLS authentication helps to mitigate security risks and provides a robust defense against cyber-attacks like MITM.

By using a clear address or other identifier as the subject of the certificate, the certificate can be validated during the authentication process. The use of a Public Key Infrastructure (PKI) enables a more scalable trust model compared to other methods like Pre-Shared Key (PSK) or Secure Remote Password (SRP). Despite its benefits, relying solely on a PKI system also leads to centralization and delegates all security responsibilities to a third party, which can lead to potential key escrow problems. To prevent attacks, some form of authentication mechanism based on shared secrets or private/public keys is necessary. However, when it comes to P2P communication, the requirement of having pre-established shared information between peers can be a challenge. Digital certificates, Certification Authorities (CAs), and PKI can help overcome this challenge but still result in centralization issues.

VII. Analyzing Vulnerabilities On SSL/TLS Protocol

There are the tools *Sslscan* and *Testssl*, which are command-line tools that return a comprehensive list of protocols and ciphers accepted by an SSL/TLS server on a specified target, as well as some other information that is useful in a security test. The sslscan tool is an open-source software and a part of the OpenSSL library, and it is command-line-based. By providing detailed reports on scan results, sslscan provides important information about the security of servers' SSL/TLS connections. Sslscan can also export scan results in XML format, so that the results can be analyzed later. The target address *www.atlascentral.co.uk* was identified and used for testing purposes on a Kali Linux machine. An example command and output executed on the Kali Linux machine is shown in Figure 3. When the command *'sslscan www.atlascentral.co.uk'* is executed, information about the security level of the server's SSL/TLS connections is obtained. Upon examining the screenshot in Figure 3, different colors can be seen in the results obtained with sslscan. Green Color indicates that the SSL/TLS connection is secure and the SSL/TLS certificate is valid. Yellow Color indicates that the validity of the SSL/TLS certificate has not yet been verified, but no security vulnerability has been detected. Red Color indicates that the SSL/TLS connection is insecure, the SSL/TLS certificate is invalid, or there is another security vulnerability.

A. *Sslscan Tools*

This tool can run from the command line by specifying the target server's hostname or IP address. For example, *"sslscan server.example.com"*. SSLscan will then perform a scan of the server's SSL/TLS configuration and report the results. The output will include information about the ciphers and protocols that are supported, as well as any vulnerabilities or issues that were detected.

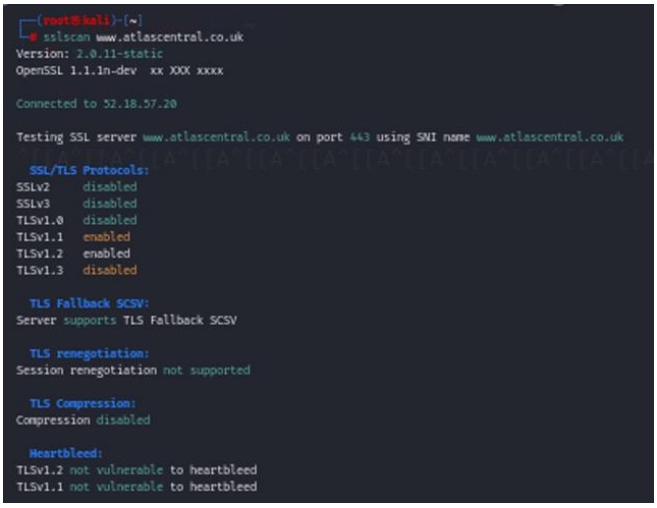

Figure 3: SSLscan Command Output

B. *Testssl Tools*

When the command in Figure 4 is run on the Kali Linux machine, the screen outputs shown in Figure 5 and Figure 6 are obtained. This tool provides information on the SSL/TLS version of the target web address, Cipher Suites information, and vulnerability information on it.

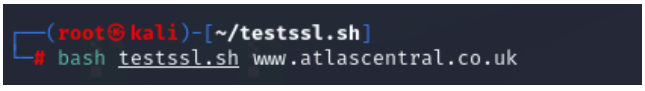

Figure 4: Testssl Command

Testssl tools provide information on the SSL/TLS version information of the target web address, the category information in which the encryption algorithms that can be used in the SSL/TLS connection are grouped, and possible vulnerability information on it. For example, Figure 6 shows that the target website is vulnerable to SWEET32, LUCKY13, and LOGJAM attacks. The tools mentioned above have demonstrated successful results regarding the SSL/TLS protocol version, cipher suite information, and potential vulnerabilities associated with a domain or IP address. It is evident, especially from the Testssl tool outputs, that all possible attacks against the site (if any) have been detected.

Figure 5: Testssl Output-1

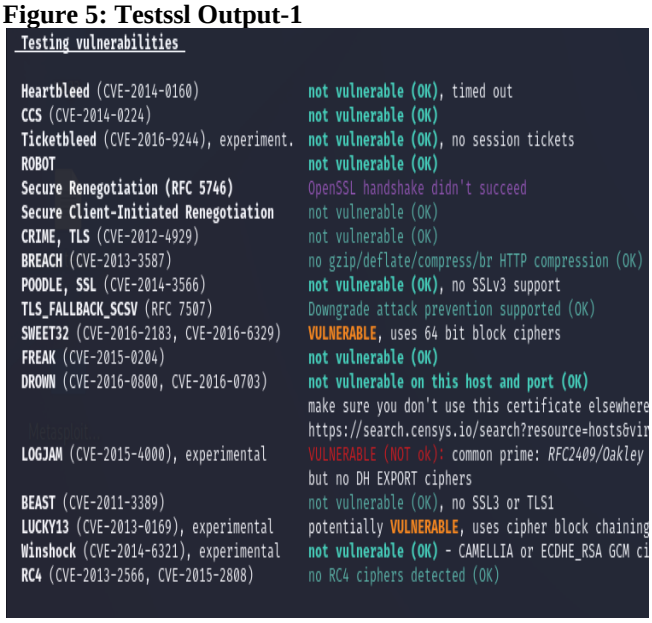

Figure 6: Testssl Output-2

VIII. Conclusion

SSL/TLS has become a standard for authentication and encrypted communication between clients and servers. However, as technology advances, the amount of data shared on the internet increases even more. This increase paves the way for an increase in vulnerabilities in the protocol as well. For this reason, there may be various security vulnerabilities and weaknesses in the SSL/TLS protocol. Therefore, it is essential to test the security of SSL/TLS connections on servers and to detect security vulnerabilities. Also, it should be ensured that the latest versions are effectively used between clients and servers. Tools that analyze vulnerabilities in the SSL/TLS protocol are used to test the security of servers' SSL/TLS connections. These tools detect the SSL/TLS protocols, encryption algorithms, key lengths, and security vulnerabilities servers support. This increases the security of servers and ensures information security in internet traffic. In this study, the tools SSLscan and Testssl have been examined, and analyses have been made for secure internet communication. It was determined that the website tested with SSLscan and Testssl uses the TLSv1.2 version, and three attack vulnerabilities were found out of the 18 attack analyses conducted. By testing the security algorithms, we can assess the average security level of the site. However, this outcome may vary depending on each site's security level.

References

[1] Bhargavan, K., Fournet, C., & Kohlweiss, M. (2016). mitls: Verifying protocol implementations against real-world attacks. *IEEE Security & Privacy*, *14*(6), 18-25.

[2] Manfredi, S., Ranise, S., & Sciarretta, G. (2019, July). Lost in TLS? no more! assisted deployment of secure TLS configurations. In *IFIP Annual Conference on Data and Applications Security and Privacy* (pp. 201-220). Springer, Cham.

[3] Kehret, O., Walz, A., & Sikora, A. (2016). Integration of hardware security modules into a deeply embedded TLS stack. Computing, (15, Issue 1), 24-32.

[4] Aayush, A., Aryan, Y., & Muniyal, B. (2022, April). Understanding SSL Protocol and Its Cryptographic Weaknesses. In *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 825-832). IEEE.

[5] Ivanov, O., Ruzhentsev, V., & Oliynykov, R. (2018, October). Comparison of modern network attacks on TLS protocol. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)* (pp. 565-570). IEEE.

[6] Sirohi, P., Agarwal, A., & Tyagi, S. (2016, October). A comprehensive study on security attacks on SSL/TLS protocol. In *2016 2nd international conference on next generation computing technologies (NGCT)* (pp. 893-898). IEEE.

[7] Radivilova, T., Kirichenko, L., Ageyev, D., Tawalbeh, M., & Bulakh, V. (2018, May). Decrypting SSL/TLS traffic for hidden threats detection. In *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (pp. 143-146). IEEE.

[8] Ćurguz, J. (2016). Vulnerabilities of the SSL/TLS protocol. Computer Science & Information Technology, 245.

[9] Berbecaru, D. G., & Petraglia, G. (2023, January). TLS-Monitor: A Monitor for TLS Attacks. In 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) (pp. 1-6). IEEE.

[10] Platenka, V., Mazalek, A., & Vranova, Z. (2021, June). Attacks on devices using SSL/TLS. In 2021 International Conference on Military Technologies (ICMT) (pp. 1-6). IEEE.

[11] Kumari, N., & Mohapatra, A. K. (2022). A comprehensive and critical analysis of TLS 1.3. Journal of Information and Optimization Sciences, 43(4), 689-703.

[12] https://learn.microsoft.com/en-us/windowserver/security/tls/transport-layer-security-protocol Erişim Tarihi:14.11.2022.

[13] Alkazimi, A., & Fernandez, E. B. (2014, September). Cipher suite rollback: a misuse pattern for the SSL/TLS client/server authentication handshake protocol. In *Proceedings of the 21st Conference on Pattern Languages of Programs* (pp. 1-9).

[14] https://www.natro.com/blog/tls-1-3-nedir-neden-tls-1-3-kullanilmali/ Erişim Tarihi:17.11.2022.

[15] Kim, S. M., Goo, Y. H., Kim, M. S., Choi, S. G., & Choi, M. J. (2015, August). A method for service identification of SSL/TLS encrypted traffic with the relation of session ID and Server IP. In *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 487-490). IEEE.

[16] Çakmak, A. (2018). Web güvenliğinde SSL/TLS kriptografik protokolü: açıklıklar, saldırılar ve güvenlik önlemleri (Master's thesis, Fen Bilimleri Enstitüsü).

[17] Ozden, D. U. Y. G. U. (2016). Analysis of recent attacks on SSL/TLS protocols. *student reports*.

[18] Ristic, Ivan. Bulletproof SSL/TLS and PKI: The Complete Guide to Securely Using SSL/TLS and PKI in Infrastructure Deployment and Web Application Development. Londra: Feisty Duck, 2014.

[19] Momani, E. M. H., & Hudaib, A. A. Z. (2014). Comparative Analysis of Open-SSL Vulnerabilities & Heartbleed Exploit Detection. *International Journal of Computer Science and Security*, *8*(4), 159-176.

CERTIFICATE OF ATTENDANCE

is awarded to

Fidan BOZKURT

for participation at the

2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-2023)

Dortmund University of Applied Sciences and Arts
Dortmund, Germany
September 7 - 9, 2023

Anatoliy Sachenko
IDAACS-2023
Co-Chairman

Carsten Wolff
IDAACS-2023
Co-Chairman