# AN INSIGHT OF SSL SECURITY ATTACKS

**Zubair Jeelani***

**Owais***

## ABSTRACT

*Secure Sockets Layer (SSL), is a cryptographic protocol that provide communication security over the Internet. SSL encrypt the segments of network connections at the Application Layer for the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for confidentiality, and message authentication codes for message integrity. SSL secures web services such as banking, online purchases, email and remote access. SSL has been targeted with attacks from the time it was created. Most of these attacks exploit the vulnerabilities present in the services SSL use, such as digital certificates and the web browsers. Attacks on SSL itself have been successful, at least in the context of research, attacks on the services that SSL uses have been successfully exploited in an actual commercial setting; the fact that makes these kinds of attacks extremely dangerous. In this paper, we briefly explain the various attacks like SSL sniffing, MD5 collide certificate, SSL striping, SSL Null prefix, online certificate status protocol (OCSP),change cipher spec-dropping, KeyExchangeAlgorithm-spoofing, and version rollback attacks. Since most of the discussed attacks target browsers and the way they manage certificates, an evaluation on the rate of success of the SSL attacks when various browsers are used is also presented. We also discuss the origin and the conditions for the attacks to happen successfully. We further discuss in some detail the two very recent attacks BEAST (Browser Exploit Against SSL/TLS) and CRIME (Compression Ratio Info-leak Made Easy).*

***Keywords:*** *Secure sockets layer; digital certificates; SSL attacks*

*Department of Computer Science, Islamic University of Science and Technology, J&K, India.

# 1. INTRODUCTION

The recent explosive growth of the Internet and the World Wide Web has brought with it a need to securely protect sensitive information sent over this open network, such as credit card information, usernames, and passwords. Secure Sockets Layer or SSL is a protocol responsible for transmitting this sensitive information in a secure manner by achieving data confidentiality, integrity and authentication in Web transactions.SSL through a cryptographic system uses two keys to encrypt data- a public key known to everyone and a private or secret key known only to the recipient of the message. SSL works between the OSI Application and Transport layers to provide for the client and server applicable connection oriented mechanisms for secure communication channel.

Even with the presence of some attacks that are able to extract various transmitted secure credentials, SSL has been increasingly used since its introduction. Several attacks have been successfully conducted on the SSL. These attacks can either exploit the implementation of the protocol itself or may exploit the vulnerabilities in the services SSL uses. In this paper, Section 1 provides a brief overview of the SSL protocol. Section 2 explores several attacks on the SSL and Section 3 concludes the paper by providing a high-level view of the SSL protocols strengths and weaknesses.

# 2. OVERVIEW OF SSL

SSL has two distinct entities, server and client. The client is the entity that initiates the transaction, whereas the server is the entity that responds to the client and negotiates which cipher suites are used for encryption. In SSL, the Web browser is the client and the Web-Server is the Server.

SSL itself is actually composed of three protocols; the Handshake protocol, the Record Protocol and the Alert Protocol. There are discussed briefly in the following sub sections.

## 2.1 SSL Handshake

During the Handshake Protocol, the following important steps take place: the session capabilities are negotiated, meaning the encryption (ciphers) algorithms are negotiated; and the server is authenticated to the client with the help of a digital certificate, that the server send to the client.

SSL uses symmetric cryptography for the bulk data encryption during the transfer phase; however, asymmetric cryptography, (that is, PKI) is used to negotiate the key used for that symmetric encryption. This exchange is critical to the Handshake Protocol. Note that the

server may optionally ask the client to authenticate itself. However, it isnot necessary to the protocol. Table 1 give the steps of the Handshake Protocol.

**Table 1 Handshake Protocol**

| | |
|---|---|
| 1. | Client sends **ClientHello** message. |
| 2. | Server acknowledges with **ServerHello** message. |
| 3. | Server sends its certificate. |
| 4. | Server requests Client's certificate. (Optional) |
| 5. | Client sends its certificate. (Optional) |
| 6. | Client sends **ClientKeyExchange** message. |
| 7. | Client sends **Certificate Verify** message. |
| 8. | Both send **ChangeCipherSpec** messages. |
| 9. | Both send **Finished** messages. |

### 2.2 SSL Records

The encryption for all messages in SSL is handled in the Record Protocol. This protocol provides a common format to frame all Alert, ChangeCipherSpec, Handshake, and application protocol messages.[1]

SSL records consist of the encapsulated data, digital signature, message types, version, and length. SSL records are 8 bytes long. Because the record length is fixed, encrypted messages sometimes include padding and pad length in the frames.

### 2.3 SSL Alert Protocol

As mentioned earlier, the Alert protocol handles any questionable packets. If either the server or client detects an error, it sends an alert containing the error. There are three types of alert messages: warning, critical and fatal. Based on the alert message received, the session can be restricted (warning, critical) or terminated (fatal).

## 3. SSL SECURITY ATTACKS

During the past few years, different attacks were conducted against SSL. Most of them exploit the vulnerabilities of the clients' browsers deploying SSL technology. In the following sub-sections we discuss these attacks and the origin and the conditions for the attacks to happen successfully.

### 3.1 SSL Sniffing Attack

The origin of this attack is based on vulnerability present in Web Browsers that can be exploited by the attacker. There is a field in the certificates called Basic Constraints. The Basic Constraints field has two parameters; Subject Type and Path Length Constraint. The

later parameter indicates the maximum number of CA certificates above the given certificate in the certification path. Path Length Constraint is used to ensure that the holder of the certificate can only issue End Entity certificates and not CA certificates. If the browser did not check this field when it comes to validating a certificate, an attacker can use this certificate to sign a request of a forged certificate. This forged certificate will be treated by Web Browsers as a trusted certificate. When no basic ‘‘constraint field’’ validation is done, clients’ browsers will consider the chain in the forged certificate as a trusted chain, and hence establish a secure channel with attackers rather than legitimate entities.

Microsoft Internet Explorer 7 and prior versions are vulnerable to the SSL Sniffing Attack. This vulnerability is originated as a result of bypassing the “Basic Constraints” field validation. In this attack an ARP spoofing tool called *arpspoof* is used to achieve traffic redirection. As a result of using *arpspoof*, all the traffic between client and server will be redirected to the attacker machine.

When the real server replies back to the real client providing it with its certificate, the attacker intercepts this message taking by this action the client’s identity. At this point, the attacker has a secure communication with the server. Now the attacker creates a certificate on the fly imitating the server (a certificate that includes the server information) and signs it with whatever valid certificate it possesses. This newly created and signed certificate is valid because the certificate chaining leads to a valid authentic CA. This certificate is then sent to a client resulting in another secure channel between the client and the attacker. Using the MitM attack and on-the-fly certificate generation, the attacker will open two secure channels, one with the victim, and the other with the legitimate server. At this point all the secure communication is intercepted by the attacker.

**3.2 MD5 Collide Certificate Attack**

MD5 is a widely used cryptographic hash function and is employed in a variety of security applications. It is supposed to provide the following properties:

- Given message m, it is computationally easy to get the hash h = H(m).
- Given hash h, it is computationally infeasible to get m out of h, such that h = H(m).
- Given m and h = H(m), it is computationally infeasible to get another message n, such that H(n) = H(m). This property is known as collision resistance.

However, it has been shown that MD5 is not collisionresistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on thisproperty [3]. Unfortunately, some CAs still use MD5in SSL certificates and digital signatures, and hence,compromising the security of their clients. Despite thewarnings made in 2004

regarding the danger of MD5collisions, statistics in 2008 showed that 9000 certificatesare still using MD5 rather than SHA1.

To start the attack, the attacker needs a valid legitimate certificate that is signed with a real CA. The certificate should use MD5 and should be accepted by all browsers (let us call this certificate original.cer). One root CA that provides these properties is Equifax Secure Global eBusiness CA-1. Next, the attacker issues a certificate request to be signed by this CA. As expected, the Equifax authority will sign the certificate request and send it back to the client. At this point, the attacker creates another certificate request with crafted information that will help tocause the MD5 collision. However, he does not send the request to a legitimate CA, but rather he uses the same signature digest of the first legitimate certificate (original.cer) and add it to this rogue certificate (say rogue.cer). This rogue certificate is valid over the signature. Anyone who examines the rogue certificate will think that it has been signed by Equifax authority while the fact is that Equifax has never seen this certificate before.

Now, the attacker is in possession of a rogue certificate that can be used to sign other certificates. Since the rogue certificate has been created by the attacker and not by a CA, the attacker sets the Basic Constraints field to TRUE. This means that the attacker is able to use this certificate to sign other leaf certificates. From now on, the attacking scenario will be the same as the scenario described in SSL Sniffing attack discussed in Section 3.1.

All Web Browsers are vulnerable to this attack as long as they accept MD5 hash functions. Unfortunately companies kept using MD5 hash functions. Newer versions of IE including IE9 have not revoked certificates using the MD5 hash function. The best defence against this attack is to use stronger hash functions like SHA1 or SHA2. IE, FireFox v3, Opera Pre v9 and Safari are vulnerable to this attack.

**3.3 SSL Stripping Attack**

Websites usually use SSL only when transferring confidential data between client and the server. When a user types in an address like www.gmail.com in the address bar, a redirection happens that expands the website address to along URL which is hard to remember otherwise. This is called HTTP 301 permanent redirection.[2] But since this is a secure site, an another redirection happens called HTTP 302 temporary redirection. This redirection means that the connectionsession has been moved temporarily to https://www.google.com/../../../accounts and is ready to accept userinformation in a securemanner.

The attacker can exploit specifically the 302 redirections to let the secure traffic be redirected to his machine instead of forwarding it to the legitimate server. The point where the 302 redirection occurs is the point where an elusive tool called sslstrip [4] can be used to tell the users to navigate http://www.google.com/../../../accounts rather than https://www.google.com/../../../accounts. Notice that this new URL is http and not https. At this point, all the traffic exchanged between the client and the attacker is exchanged in the clear. However, the communication between theattacker and the server is completely legitimate and encrypted using the server's legitimate public key. The server's response will be decrypted by the attacker before forwarding it to the client.

This attack exploits the people's unawareness about secure communication in addition to the HTTP technical vulnerability. This attack can be checked by educating people or more appropriately by adding the https URL into the browser's bookmark. Sowhenever a user would like to access the secure loginwebpage, it will be downloaded into the browser without302 redirections.

**3.4 SSL Null Prefix Attack**

Null-prefix attack[5] is a silent MitM attack that exploits the treatment of the fields obtained from X509 certificates via most contemporary SSL/TLS implementations, and the signing process of contemporary Certificate Authorities. X509 certificates support PASCAL strings which are represented as a series of bytes mediated by another series of bytes indicating the length of the string followed by the data itself. C strings, on the other hand, are represented as a series of bytes terminated by the NULL character (''\0'').

When a user submits a Certificate Signing Request(CSR) to the Certificate Authorities, it validates the identity of the owner based on a comparison between the domain listed in the ''common name'' field retrieved from the request (www.paypal.com\0.attacker.com), and the root domain retrieved by looking up WHOIS database. The identity information is only associated with the root domain, so the CA does not care about the content of the any subdomains that might be present in the users' requests. Consequently, submitting a request for www.attacker.com or www.paypal.com\0.attacker.com to Verisign does not yield any difference as long as it can prove that you are the owner of attacker.com. Unfortunately, most contemporary SSL/TLS implementations uses ordinary C strings functions for manipulation and comparison and not PASCAL strings. A string comparison between www.paypal.com and www.paypal.com\-0.attacker.com will be identical. Therefore, the owner of www.paypal.com\0.attacker.com certificate can present his certificate for connections intended for www.paypal.com, and thus breaking the authenticity of the intended server.

Figure 9 shows the details of this attack. IE, FireFox, Chrome, Opera and Safari are vulnerable to the attack because of flaws in network security services (NSS) functions as mentioned earlier.

**3.5 OCSP Attack**

OCSP is a protocol inwhich the browser communicates with a server in real time to determine if a certificate is still trustworthy and notrevoked. Clearly, this process is very time consuming. In addition, it places a burden on the CA's servers. When the Web Browser is presented a certificate that it has never seen before, the browser contacts the certificate issuer to confirm the validity of the certificate.

OCSP attack is based on exploiting the OCSP response in which it is very hard to revoke Null-prefix certificates described in pervious sub-section.[6] There are actually two structures in an OCSP response message; responseStatus and responseBytes. In the responseBytes structure, there exists a field that requires the signature of the CA, it is called the BIT STRING field. To forge the OCSP response, the attacker first needsto avoid the ''successful'' status under the responseStatusstructure, since that would require including the CA'ssignature in the responseBytes structure; a signature theattacker cannot obtain. The ''tryLater'' is used as it doesnot require a signature and does not convey any errorconditions.

The attack is based on intercepting OCSP response whichfor instance is intended for www.paypal.com server, andgenerating a forged one with response status as ''tryLater'',which is simply the single ASCII character ''3''. The userwon't notice anything and by activating this mode insslsniff tool, it will be very hard to revoke the null-prefixcertificate, thus defeating OCSP [6].

**3.6 Change Cipher Spec-Dropping**

This attack takes advantage of the lack of protection for change cipher spec messages. We assume the special case where the negotiated ciphersuite includes only message authentication protection and no encryption. The active attacker intercepts and deletes the change cipher spec messages, so that the two endpoints never update their current ciphersuite; in particular, the two endpoints never enable message authentication or encryption in the record layer for incoming packets. Now the attacker allows the rest of the interaction to proceed, stripping off the record layer authentication fields from finished messages and session data. At this point there is no authentication protection for session data in effect, and the active attacker can modify the transmitted session data at will. The impact is that, when an authentication-only transform is negotiated, an active attacker can defeat the

authentication protection on session data, transparently causing both parties to accept incoming session data without any cryptographic integrity protection.

The simplest fix is to require that a SSL implementation receive a change cipher spec message before accepting a finished message.

**3.7 KeyExchangeAlgorithm-Spoofing**

This attack exploits a design flaw in the SSL 3.0 handshake protocol. A server can send shortlived public key parameters, signed under its longterm certified signing key, in the server key exchange message. Several key-exchange algorithms are supported, including ephemeral RSA and Diffie-Hellman public keys. Unfortunately, the signature on the short-lived parameters does not protect the field which specifies which type of key-exchange algorithm is in use. Note that this violates the Horton principle: SSL should sign not just the public parameters but also all data needed to interpret those parameters.

In the SSL 3.0 data structure from server exchange message we can analyse the *signed_params* field contains the server's signature on a hash of the relevant *ServerParams* field, but the signature does not cover the *KeyExchangeAlgorithm* value. Therefore, by modifying the (unprotected) *KeyExchangeAlgorithm* field, we can abuse the server's legitimate signature on a set of Diffie-Hellman parameters and fool the client into thinking the server signed a set of ephemeral RSA parameters.

**3.8 Version RollBack**

Although most of the Web Browsers and Servers rely on SSL 3.0 for securing the communication, there is a support in almost all the browsers for SSL 2.0. SSL 3.0 have the capability to accept the SSL 2.0 connections. This threatens to create the potentialfor version rollback attacks, where an opponentmodifies the **client hello** to look like a SSL 2.0 hellomessage and proceeds to exploit any of the numerousSSL 2.0 vulnerabilities.

Paul Kocher designed a fascinating strategy to detectversion rollback attacks on SSL 3.0. Client implementationswhich support SSL 3.0 embed some fixed redundancy in the (normally random) RSAPKCS padding bytes to indicate that they supportSSL 3.0. Servers which support SSL 3.0 will refuse toaccept RSA-encrypted key-exchanges over SSL 2.0 compatibility connections if the RSA encryption includesthose distinctive non-random padding bytes.This ensures that a client and server which both supportSSL 3.0 will be able to detect version rollbackattacks which try to coerce them into using SSL 2.0.Moreover, old SSL 2.0 clients will be using randomPKCS padding, so they will still work with serversthat support SSL 2.0.First, the success of the countermeasure dependsvitally on the assumption that SSL 2.0 will onlysupport RSA key-exchange; non-RSA public keyexchangealgorithms

will not admit the specialpadding redundancy, so version rollback attacks cannot be detected if the server supports non-RSA keyexchangemethods while operating in SSL 2.0 mode.All the browsers that support SSL 2.0 are vulnerable.

## 4. BEAST AND CRIME

BEAST and CRIME are the two very recent attacks discovered by a group of researchers. Although, there is no proof that these attacks have been conducted in a commercial setup or not, they expose some serious fractures in the SSL system to secure web transactions.

### 4.1 BEAST (Browser Exploit Against SSL/TLS)

BEAST was recently discovered by a group of researchers and it exploits a vulnerability that reside in SSL 3.0 and versions 1.0 and earlier of TLS, the successor of SSL. Although versions 1.1 and 1.2 of TLS aren't susceptible, they remain unsupported in many web browsers and websites. BEAST uses a piece of JavaScript Code that works with a network sniffer to decrypt the encrypted cookies a targeted website uses to grant access to restricted user accounts. The exploit works even against HSTS, or HTTP Strict Transport Security, which prevents certain pages from loading unless they're protected by SSL.BEAST attack was mitigated by reconfiguring web servers to use theRC4 cipher-suite rather than AES. But the newer attack CRIME enables miscreants to run in man-in-the-middle-style attacks and is not dependant on cipher-suites.

### 4.2 CRIME(Compression Ratio Info-leak Made Easy)

After BEAST was mitigated, discoverers of BEAST came up with a new attack called CRIME. All versions of TLS/SSL – including TLS 1.2, on which theBEAST attack did not work – are vulnerable. The researchers say that once they have placedthemselves in the middle of a given network, they can sniff the HTTPS traffic and launch theattack. Their chosen way to get that position is by running JavaScript code in the victim'sbrowser, but the attack doesn't rely on JavaScript.The cipher suite doesn't matter, say the researchers, noting that one workaround for BEASTattacks was to switch from AES to RC4, but for CRIME that isn't important. The feature thatCRIME is leveraging for its attack has, they say, not been a major subject for security researchin the past, but for the attack to work it must be supported at the client and server.

## 5. CONCLUSION

SSL has been long known for securing all thecommunication traffic between the client and server, thus preventing attackers from tampering withdata during transition. However,

improper design of SSL inthe various applications could lead to unexpected consequences. The above discussed attacks exploit the design flawsof SSL.

## REFERENCES


1. SSL and TLS Essentials: Securing the Web (p. 69) Thomas
2. Network Working Group. Hypertext Transfer Protocol(HTTP/1.1) RFC, June 1999.
3. Wang X, Yu H. How to break MD5 and other Hashfunctions, Eurocrypt 2005, Lecture Notes in ComputerScience, vol. 3494, May 2005; pp. 19-35.
4. Moxie M. SSLStrip: http://www.thoughtcrime.org/software/sslstrip
5. Marlinspike M. Null prefix attacks against ssl/tlscertificates, 29/7/2009, http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf.
6. Marlinspike M. OCSP attack, 29/7/2009, http://www.thoughtcrime.org/papers/ocsp-attack.pdf
7. Stevens M, Sotirov A, Appelbaum J, et al. ShortChosen-Prefix Collisions for MD5 and the Creationof a Rogue CA certificate, Proceedings of CRYPTO2009, to appear.