# Analysis on Man in the Middle Attack on SSL

Pushpendra Kumar Pateriya
Asst. Prof.
Computer Science Department
Lovely Professional University,
Punjab, India

Srijith S. Kumar
M.Tech. Student
Computer Science Department
Lovely Professional University,
Punjab, India

## ABSTRACT
Man-In-The-Middle attack is the major attack on SSL. Some of the major attacks on SSL are ARP poisoning and the phishing attack. Phishing is the social engineering attack to steal the credential information from the user using either fake certificates or fake web-pages. Same in the case of ARP Poisoning, where in the attacker act as middle-man in the client-server communication channel. MITM attack makes the users difficult to understand that whether they are connected to original secured connection or not. Since the certificate that is being passed during the connection setup is insecure, attacker can easily modify the information in the certificate and leave the approval of the certificate to the user. Since many users are not well educated about the whereabouts of the forged certificates and their corresponding attacks, they accept the certificates making way for the attackers to implement the attack. To deal with such attacks, two approaches have been proposed: one for the ARP poisoning; and other for phishing attack.

## General Terms
Security, Network Security, SSL

## Keywords
SSL, TLS, Man in the Middle attack, Security, ARP, Phishing

## 1. INTRODUCTION
As the internet is growing tremendously and the data being passed is becoming crucial for the organization, we need to provide the customers as well as the organization some level of privacy and authentication so that all the users in this internet-world can assure that they are contacting the right person. To provide the user and the organization this level of security Netscape came up with a solution called Secured Socket Layer (SSL) in 1994. They never released the 1st version of SSL because of some limitations in the system [2]. SSLv2 was released in 1994, and later IETF standardized SSL and released SSLv3.0 strengthening the protocol by adding more secured algorithms and handle the credential information more securely. They renamed the next upgraded version of the SSLv3.1 as TLSv1.0 (Transport Layer Security) [1][2][3].

SSL was developed keeping in mind to provide information privacy and security, but still this protocol has some limitations. First of all, the SSL follows the weak model of PKI: web-model[8]. Web-model is best for large scale implementation for the SSL, but the problem is that there are 2 types of Trust Roots in web model: CA, and User. Here user can also judge whether or not they should allow any other certifications that are not in CA. Secondly, the attacks that the SSL face are majorly from MITM attack, mainly ARP poisoning, wherein the attacker can hijack the secured session and can get the secured credentials.

In this paper, we have discussed a scheme to strengthen the SSL using a Firefox add-on which can detect any spurious SSL certificates and a bash shell script which can be run on any Linux system to counteract against RP-poisoning.

The paper is divided in 3 parts. 1st part will discuss about general information about the SSL, about its structure and how it works. Then in the 2nd section we will be discussing about the possible attacks over SSL and how much they will be harmful. And the last section we will discuss about the solution that we have proposed.

## 2. SSL:
SSL have three protocols under it: Handshake Protocol; Record Protocol; and Alert Protocol. Handshake protocol is used to establish the secure connection between the client and the server using the cipher suites and other parameters that both have agreed upon. Record Protocol is used to encrypt the data that is to be sent through the network using the key that have been established during the handshake protocol. Alert protocol is used to send the custom messages to other whenever they detect any intrusion in the system. As I need to show the defects in the SSL methods, handshake protocol (see Figure 1) need to be discussed first[1]. It is as follows:

**Step 1:** Client Sends a **ClientHello** message to the server he wishes to contact. This message contains the Version No of the SSL which client can support with a 32-byte random no. this message also contains the Cipher Suites and the Compression Method that the client can support.

**Step 2:** Now the Server sends a **ServerHello** message to the client. This message is the complement to the Client Hello message. This message contains the version of SSL both the party will support, 32-byte random no., Session ID and the cipher suite and the compression method that it will support.

**Step 3:** Server then sends the **ServerKeyExchange** message to the client. This message contains the public key information itself, for e.g.: the Public Key in case of RSA. Then to authenticate the client, server requests for the client's certificate information, if it has one.

**Step 4:** After all the information have been passed to the client, server sends a **ServerHelloDone** indicating the client that server's phase of initial negotiation have been done and now its clients turn.

**Step 5:** Now the client will send its key information to the server with **ClientKeyExchange** message encrypted with the server public key so that the legitimate server only can access client's information.

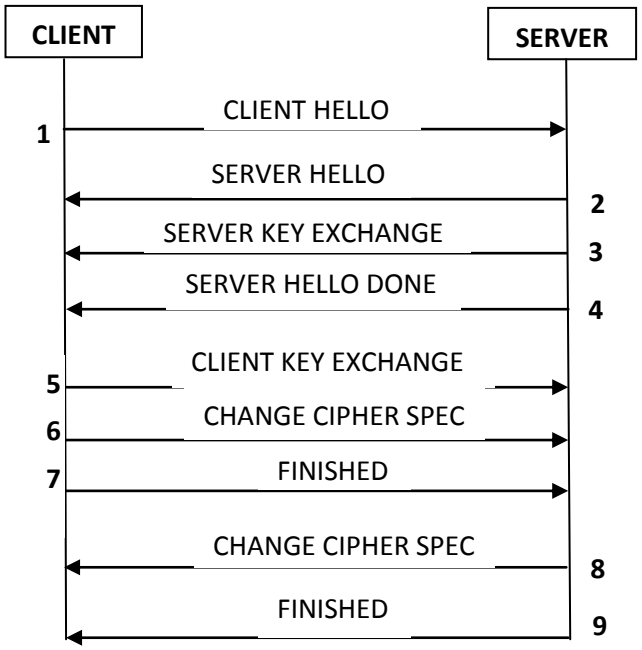


**Figure 1: SSL Handshake**

**Step 6:** Now as both the client and the server have sent their key information and other parameters, Client sends a **ChangeCipherSpec** message to the server to notify all the parameters of the secured connection and activate the same.

**Step 7:** Then the client sends the **Finished** message to the server to let it check the newly activated options.

**Step 8:** The server sends the same **ChangeCipherSpec** to the client to notify all the options in the secured connections and then send the **Finished** message to client to verify all the options.

Next to the Handshake Protocol is the Record Layer Protocol. This layer encapsulates all the data into a frame format of size 5bytes preceding other protocol messages. This protocol provides a single frame format for Alert, ChangeCipherSpec, Handshake, and Application Data[1].

## 3. ATTACKS ON SSL:

Since the inception of SSL, attackers and the researchers have been trying to find as many flaws in the structure of SSL. Till dated many flaws and solutions for them have been found in the structure of the SSL.

The major type of attack on the SSL is Man-In-The-Middle (MITM) attack. It can be either ARP poisoning attack, Algorithm Rollback Attacks[3], Cipher Suite Rollback Attack[9], Compelled Certificate Creation Attacks, sslstrip[5]etc. The main problems that we are dealing with are: ARP Poisoning; and Fake Certificate Attack. First we would have a brief discussion over these types of attacks.

### 3.1. ARP Poisoning:

ARP (Address Resolution protocol) is the protocol method to detect the MAC address, i.e. hardware address, of a particular node when the sender knows the IP address, i.e. logical address, of the destination node. The destination sends a broadcast packet asking for the MAC address of the known IP address so that it can locate the device in the network. Once it gets the IP/MAC mapping for a device, it stores this mapping into its ARP cache.

The possible attack that can occur is ARP poisoning. In this attack, the attacker first of all tries to capture some packets to get knowledge about which is the gateway and which all devices are connected to it[4]. When he finds the victim and the gateway IP addresses, it sends an ARP reply to victim stating that the gateway MAC address is now the MAC of the attacker and a similar packet to gateway stating that the victim's MAC address is now changed to that of attacker's MAC.

With this attack, the attacker can hijack the session even if it is secured by SSL/TLS as shown in Figure 2. Here the victim contacts the server through the gateway via attacker. Attacker gets all the packets that are travelling through the network and can see all the data.

### 3.2. Fake Certificate Attack (Phishing Attack):

Phishing[10] is the type of attack in which the user is forced to enter user credentials into the fake websites. These web sites look similar to the real and authentic websites and make fool out of users to enter their information. But in a more sophisticated manner, attacker can hijack the session between the client and the server.

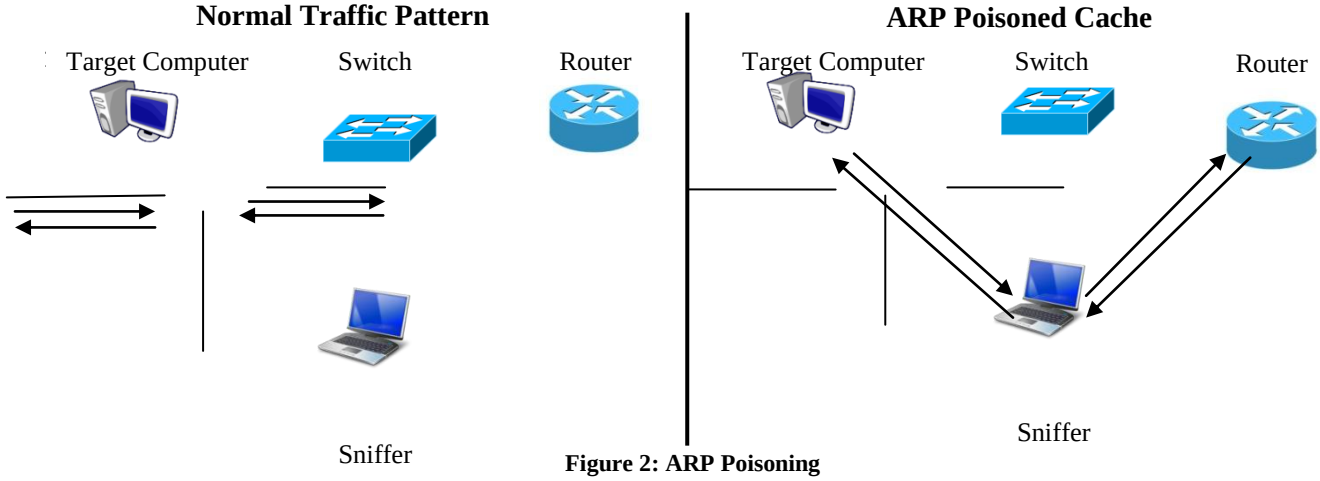


**Figure 2: ARP Poisoning**

As it can be visible from the handshake protocol of the SSL that initial negotiations wherein the server sends its Public-Key information and this information isn't secured. So the attacker can capture this message and change the details in the certificate. The attacker can change the public-key value to that of the attacker and then send it to the victim [11][12].

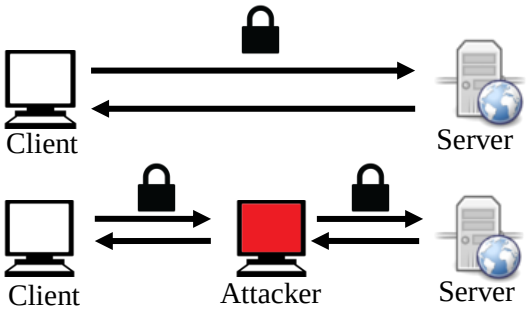


**Figure 3: MITM Attack**

For victim, it will be a secured connection as he is securely connected to the victim's computer, which in turn is securely connected to the server. This is main disadvantage of the PKI model that the SSL supports. This type of attack gives a user a warning about the certificate being forged and gives the user to either accept the certificate or reject the connection. Many users ignore the warning and moves ahead with the connection.

# 4. PROPOSED SOLUTION:

## 4.1. For ARP Poisoning:

The solution we have proposed is a Linux shell script. Earlier in [6] the author proposed a solution for ARP poisoning in which the Shell Script lacked the capability to check the ARP table without the manual entry of the gateway address. The main thing which is being considered here is that Gateway is never changed within a network and thus the IP assigned would never change until the gateway goes down. So the IP/MAC combination of the gateway would remain same. The working for this shell script is as follows:

**Step 1:** First, the script will check for the Gateway IP address and the corresponding MAC address using route –n and arp –a.

**Step 2:** Redirect the IP-MAC mapping from arp –a output to a file at regular intervals

**Step 3:** After every two successive redirection check for the same values of MAC for IP addresses using the awk.

**Step 4:** If the search for the same MAC for different IP is found, then there would be a notification sent to the user that that the ARP cache have been compromised.

**Step 5:** As the user is being reported that the ARP table has been modified, the script will modify the ARP cache using the original values that are stored in the variables used in the first step.

This script will help in overcoming the ARP poisoning attack by checking the ARP cache of the system regularly. The interval between which the ARP cache of the system is set to an optimal time interval so that the system won't be busy checking the ARP cache at smaller intervals, and not at large intervals that important packets are leaked.

## 4.2. For Fake Certificate Attack:

For counterattacking the Fake Certificate Attack, we have proposed a Firefox add-on. Blake et. al. [7] proposed a solution to mitigate the MITM phishing attack using the plug-in called PwdHash. But Yogesh Joshi et. al. in their paper [10] proposed a solution to mitigate the same problem. In that solution they hashed the user password with the fingerprint of the web server and then send the password to the server side. This is shown in Figure 4 Then the server needs to hash the password in the database and then check for correctness in the password. This solution had to change the password verification technique on the server side.

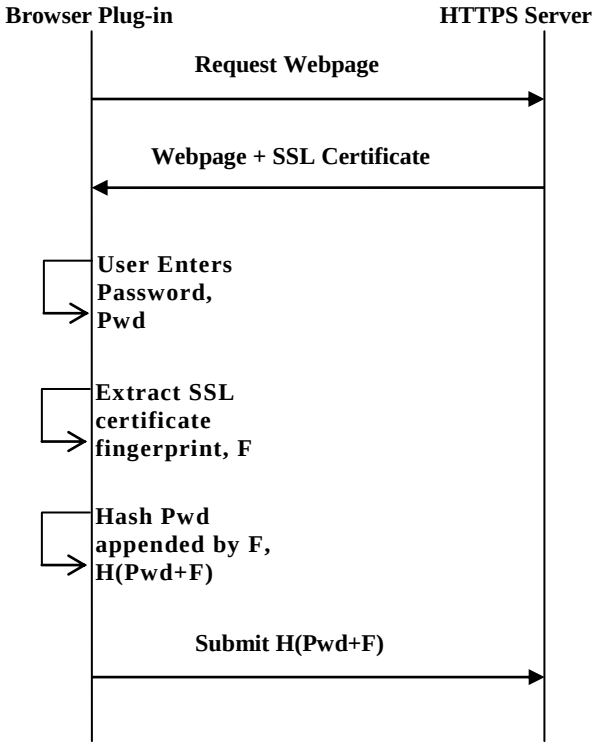


**Figure 4: Solution Provided by Yogesh Joshi et. al.**

In our solution, we provide a different solution in which we pre-store the sha-1 public key of each secured website. The main strategy behind storing the public-key value of each web server is that they aren't changed so much, so storing and then checking the public-key of the web server is the optimized way to counteract the MITM attack. This is demonstrated in Figure 5

The working of the plug-in is as follows:

**Step 1:** As the user connects to a web server, the add-on will collect the secured server information such as the common name and the SHA-fingerprint.

**Step 2:** The add-on will check for the common name in its database to get the fingerprint of the server.

**Step 3:** Then it will check whether the public key value gathered and the in the database are same or not. If it is same, then it will continue. If it is not, then it will show a warning notice to the user to whether to accept or not the connection.

With this solution that we have proposed, we have an edge over the solution provided by Yogesh Joshi et. al. They needed

to hash the user password with the SSL Certificate fingerprint and then send it to the server for verification. The problem was that this solution can come into effect if and only if the server side too uses the same way to validate the password, i.e. verifying the user name and the corresponding hashed password. So this solution needed two-side changes.

But our solution doesn't require any change to the server side; rather it verifies that the server that the user is contacting is a legitimate server and not a forged server. We have given the user the ultimate decision to allow the forged certificate or not because in many cases the server certificate may have been updated to a new class making the fingerprint change.

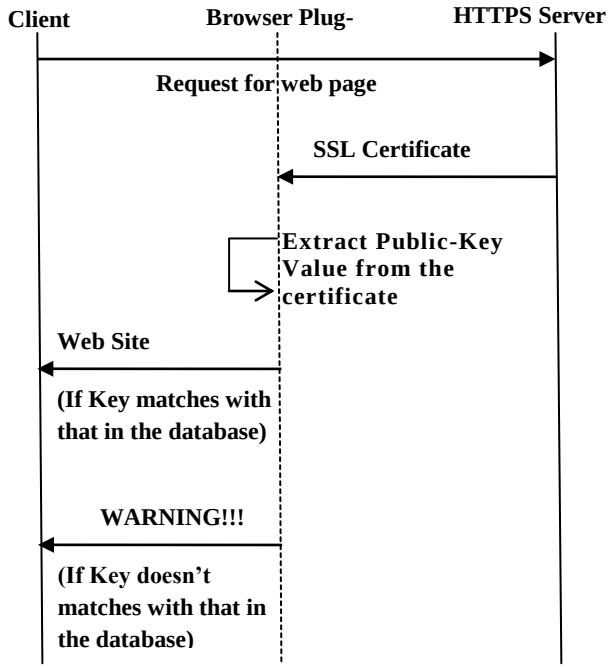


**Figure 5: Proposed Solution for Certification Attack**

## 5. CONCLUSION

In this paper solutions for the ARP poisoning attack and the Fake Certification Attack over SSL have been provided. The shell script will check for the ARP-IP of the gateway and other network devices and check for any modifications made into the ARP cache. The other browser plug-in may not be a successor to as that in [10], but will provide a better client side protection to the user.

## 6. REFERENCES

[1] Thomas, S. 2000. SSL and TLS Essentials: Securing the Web. Wiley.

[2] Introduction to Secured Socket Layer. White Paper Cisco System.

[3] McKinley, H.L. 2003. SSL and TLS: A Beginners Guide. SANS Institute.

[4] Wagner, R., Bryner, J. 2006. Address Resolution Protocol Spoofing and MITM Attacks. SANS Institute.

[5] Marlinspike, M. 2009. New Tricks for Defeating SSL in Practice. In Proceedings of the Black Hat Technical Security Conference.

[6] Nayak, G.N. 2010. A Defence Strategy for Attacks on SSL Based Connection: A Pragmatic Approach. Master's Thesis. Motilal Nehru NIT, Allahabad.

[7] Ross, B., Jackson, C., Miyake, N., Boneh, D., Mitchell, J.C. 2005. Stronger Password Authentication Using Browser Extension. In Proceedings of the 14th Usenix Security Symposium '05.

[8] Huawei, Z., Ruixia, L. 2009. A Scheme to Improve Security of SSL. In Proceedings of the Pacific-Asia Conference on Circuits, Communications and System, PACCS '09.

[9] Lee, Y., Hur, S., Won, D., Kim, S. 2009. Cipher Suite Setting Problem of SSL Protocol and It's Solutions. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, WAINA '09.

[10] Joshi, Y., Das, D., Saha, S. 2009. Mitigating Man in the Middle Attack over Secure Sockets Layer. In Proceedings of the International Conference on Internet Multimedia Services Architecture and Applications, IMSAA '09

[11] Cheng, K., Gao, M., Guo, R. 2010. Analysis and Research on HTTPS Hijacking Attacks. In Proceedings of the Second International Conference Networks Security Wireless Communications and Trusted Computing, NSWCTC '10.

[12] Jiang Du, Xinghui Li, Hua Huang. 2011. A Study of Man-in-the-Middle Attack Based on SSL Certificate Interaction. In Proceedings of the 2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control, IMCCC '11.