Prevention Techniques for SSL Hacking Threats to E-Government Services

Rıdvan TEKDOĞAN *, Ahmet EFE **‡

* Computer Engineering, Faculty of Natural Sciences, Yıldırım Beyazıt University Ankara/Turkey

** Ankara Development Agency, Internal Auditing Executive, 1322 Cadde No:11 Çankaya Ankara/Turkey

‡ Corresponding Author; Address: Tel: +90 312 3100300/172, Fax: +90 312 3093407, e-mail: icsiacag@gmail.com

Abstract- Since security threats increase over time, security of internet commination has to be achieved with cryptography and ciphering techniques in order to ensure confidentiality. The use of SSL protocols secures our communication between intended hosts and clients. As a legislative requirement of Turkey, it is obligatory for all government organizations and e-government applications to use SSL secure socked layer which ensures encrypted information transmission. Without use of SSL anyone who captures the IP packet can observe the communication since communication channels transmit bare information. Whilst securing the communication preserves secrets, sometimes it threatens business, human rights and government sovereignty. Employees can transfer confidential data to third parties over a secure channel using their credentials and authorized certificates. Therefore to secure business and e-government, decryption of SSL is obligatory to detect malevolent users. Considering the ever widening usage of internet infrastructure for e-government services and e-government applications, securing sensitive and critical information from unauthorized and malicious parties via SSL protocols which are hardened against MITM attacks became a major concern. Furthermore, usage of blockchain technology and increasing volume of cryptocurrencies are some of other issues related to security of assets over internet. In this paper, we have surveyed the SSL hacking methods and prevention techniques to have a clear vision of threats and remedies. Since hacking have several stages from ARP poisoning to fake certificate, prevention techniques are focused on different levels.

Keywords- SSL/TLS, Hacking, MITM, HTTPS, E-Government Security

1. Introduction

Communication over a secure channel is fundamental need in cyber environment in particular for e-government sites that require more confidentiality in comparison with private sector. The government institutions that use PPI (personally private information) of citizens and strategic information produced by intelligence services and security forces. Even procurement documents, tender dossiers and evaluations for projects or programs that requires secrecy is being threatened by hackers.

While transporting our confidential information in secure manner, someone can also use this secure channel for transporting harmful data. Also for the government, it is obligatory to inspect the communication of criminals. As a result, in some cases it is needed to decrypt encrypted data for security reasons.

SSL/TLS protocols are using public key infrastructure in order to establish secure tunnel between client and server. Both client and server use each other’s public key for encrypting data before sending. And receiver decrypts data by its own private key. Public keys are public to everyone and it is computationally infeasible to gather private key from public key.

Man in the middle (MITM) is well known attack type for SSL/TLS communication. Definition of MITM attack is when A wants to communicate with B, the attacker C intervene the communication and behaves like B for A and like A for B. So C become man in the middle. A and B are not aware of C sniffing while communicating.

There are commercial firewall solutions for securing private businesses. What the firewall is doing is exactly the same as attacker. Since all the traffic of business network goes through same gateway, firewall generates a fake certificate for each session and decodes the data as in MITM attack. This is the way of active hacking. Also for the government, all traffic can be captured, stored and can be decrypted when needed. Without knowing the public and private key pairs it is hard to decode stored data. This passive hacking is out of our scope.

In the rest of paper, background information about MITM and stages will be given in Background section. Literature survey for attack types will be mentioned in Section 3. Detection and prevention techniques for MITM attack will be discussed in Section 4 and paper will be concluded with Section 5.

2. Background

As mentioned in introduction, the most known attack type for decrypting SSL traffic is MITM attack. There are various stages to establish this attack. Liu Wu et.al [6] demonstrates sample MITM attack for the standard SSL/TLS protected web site.

There are three main stages of MITM attack which are ARP poisoning, DNS spoofing and distributing forged certificate to client.

In Ethernet and Wireless LAN every client using the same medium receives the same signals. In MAC layer it is checked whether the packet is belong to me or not. If the packet is destined to its own address it is dropped. This part of the protocol is a backdoor for attackers. ARP protocol is address translation protocol used for determining hardware interface address for given IP address. By advertising host IP address with attacker MAC address with ARP message, client ARP table is poisoned and every packet destined to host is received by attacker.

DNS protocol is used for address resolution for a given http address. Machines on the internet are address with IP address but memorizing IP address is hard. Therefore domain names are given to each site and the addresses of these sites are queried from DNS servers. If someone can manipulate the DNS server, he can easily direct traffic to another destination. This attack is called DNS spoofing.

There are two parts in SSL protocol which are handshake and record. In handshake a SSL/TLS client and server arrange a stateful connection by utilizing a handshaking method. Amid this handshake, client and server concede to different parameters used to build up the connection's security.

The handshake starts when a client interfaces with a TLS-enabled server asking for a secure connection, and presents a rundown of upheld CipherSuites. From this rundown, the server picks the most grounded cipher and hash work that it likewise bolsters and tells client of the choice. The server sends back its recognizable proof as an advanced certificate. The certificate for the most part contains the server name, the trusted certificate authority (CA), and the server's public encryption key. Client may contact the server that issued certificate (the trusted CA as above) and affirm that the certificate is authentic before continuing.

Keeping in mind the end goal to produce the session keys utilized for the secure connection, client encrypts a random number (RN) with the server's public key (PbK), and sends the outcome to the server. Just the server ought to have the capacity to decrypt it (with its private key (PvK)): this is the one actuality that makes the keys escaped outsiders, since just the server and client approach this information. Client knows PbK and RN, and the server knows PvK and (after unscrambling of client's message) RN. An outsider may just know RN if PvK has been bargained. From the random number, both sides produce key material for encryption and decryption. This finishes up the handshake and starts the secured connection. In the event that any of the above strides falls flat, the TLS handshake comes up short, and the connection is not made. Sample decoding HTTPS using SSL MITM has following steps [4]:

1. Promoting the gateway-router that attacker-machine will be client-machine.

2. Publicizing those client-machines that the attacker-machine will be gateway-router.

3. Turn on packet routing on attacker-machine.

4. Running DNS Spoof should authorize those customer will unite with HTTP/HTTPS port at assailant machine.

5. Publishing fake certificate to the client.

6. Communicating with the client utilizing fake certificate.

7. Communicating with HTTPS web-site utilizing genuine certificate gotten from the HTTPS web-site.

8. Transmitting data between client and the HTTPS web-site.

9. Recording information exchanged the middle of two end-hosts.

10. Deciphering information.

3. Related Work

As stated in earlier sections, when talking about hacking the SSL, generally papers discuss MITM attack. Stages of MITM attack are discussed in [4]. Published paper can be categorized according to these stages. First category aims to route the traffic to attacker machine. [4] and [3] explains the ARP poisoning and prevention techniques. Second category is publishing fake certificate and preventing as genuine host for client and vice a versa. [1] and [5] mentions possible attack types and provides solutions. The last category is application layer attack where attacker intrudes malware to browser and captures the information. This type of attack is named Man in the Browser (MITB). [2] and [11] explains how to overcome authentication phases and install malware into victim’s computer and steal information.

There are many analysis regarding the technical reasons of such attacks and solutions offered by researchers. For instance, according to [21] such attacks arise since most users are often unable to verify server certificates properly and even worse, the implement of client authentication is typically decoupled from TLS session establishment. Typically [22] proposes a technique to bypass the system’s default certificate validation as well as built-in SSL/TLS validations performed in iOS apps. Furthermore, [23] introduces the notion of SSL/TLS session-aware user authentication to protect SSL/TLS-based e-commerce applications against MITM attacks and proposed an implementation based on impersonal authentication tokens. [24] reveals the causes and preference of forged certificates, as well as several significant differences from the benign ones and discovers several IP addresses used for MITM attacks by forged certificate tracing and deep behavior analysis. İn addition [25] proposed another solution that uses a soft-token based approach for user authentication on top of the SSL/TLS’s security features.

Since the aim of this paper is to survey the existing hacking and prevention techniques, details will be mentioned in the next section.

4. SSL Hacking Techniques

SSL is cryptographic protocol designed for secure communication over TCP/IP. Man in the Middle attack is a third who wants to intercept the communication between client and server, behaves like server for the client and client for the server. So, both client and server assume that they are communicating with genuine participant at the end point.Brief information about the existing MITM attack types will be given below.

4.1. Renegotiation Attack[1][15]

When the client intends to continue with previous session, renegotiation is needed. In this attack, attacker delays the client's m1 message and starts a new session with server and sends m0 message. After the session created between attacker and server, attacker concatenates its own m0 message with genuine client's m1 message and relays to server. Attacker replaces the cookie information with its own data.

4.2. Version Rollback Attack[1]

In this attack, attacker captures the client hello message and changes the supported SSL protocol version to lower protocol version to easily decrypt.

4.3. Freak [1]

Similar to version rollback attack, in FREAK (Factoring RSA Export Keys) attacker intercepts HTTPS traffic and makes the server accept the lower grade RSA with small key size. After attacker modifies the HelloClient message of the client, server responds with 512 bit long-term RSA export key. Attacker recovers the decryption key and resolves master secret. The key is valid until the server restarts.

4.4. BERserk Attack[1]

BERserk attack is public key signature forgery attack by incomplete ASN.1 length decoding of public key signatures in SSL implementation. There are two types of ASN.1 encodings used for encoding sequences which are BER and DER. In light of specific byte fields are skipped throughout parsing when utilizing BER encoding type, the MITM attacker can forge the public key signature between the client and the server [13]. There are various sources that give details of implantation of ASN encoding of signatures and vulnerabilities.

4.5. Man in the Browser[2]

In this type of attack, fraudulent installs malware to users' browser to modify the content submitted to server. Client establishes session with genuine server but the attacker alters the information submitted to server. By this way shopping list, account numbers can be altered. [11] It also mentions how two factor authentications fails for the Man id the Browser attacks.

4.6. ARP Poisoning[4]

This type of attack is the well-known ordinary attack. Attacker tries to modify remote computer ARP-cache by advertising wrong IP, MAC pair in ARP messages. By this way client's packets are forwarded to attacker's machine. Attacker machine becomes a proxy for the client for outgoing connections. A simple example for ARP poisoning using KALİ Linux is as follows [16]: First it is required to find the IP address of victim the target and then by checking the ARP table, it will show only router's mac address.

```
#ifconfig
…
#arp
…
```

After pinging the victim machine and again by checking ARP table, it will be seen that it is added to the table.

```
#ping Victim-IP
#arp
```

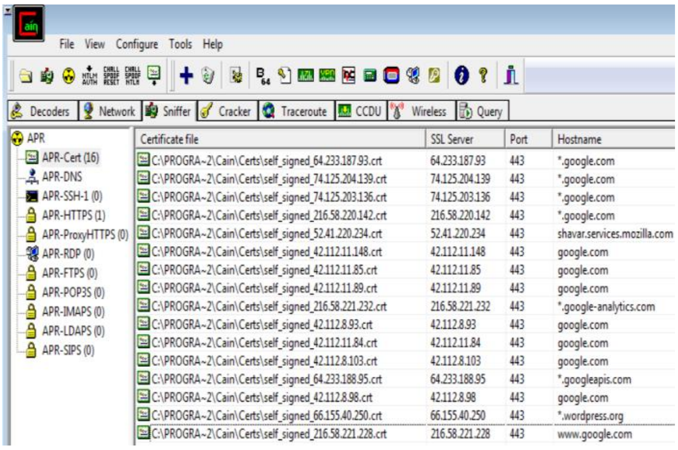
For ARP poisoning or spoofing it is required to set up IP forwarding.

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```

After checking your default gateway and start to find network interface, then it is possible to begin ARP poisoning where "-i" is for interface, "-t" is for target and "–r" is for default gateway.

```
#ip route
#ifconfig
#arpspoof -i eth0 -t 172.16.212.128 -r 172.16.212.2
```

Apart from the tools used for ARP poisoning such as Scapy that uses Phyton language and Wireshark programs working on various Linux platforms, there are also some automatic tools that are being used for ARP poisoning and decryption in the windows such as Cain & Abel which is a password recovery tool for Microsoft Operating Systems and allows easy recovery of various kinds of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols. It covers some security aspects/weakness present in protocol's standards, authentication methods and caching mechanisms. The latest version of Chain is faster and contains a lot of new features like APR (Arp Poison Routing) which enables sniffing on switched LANs and Man-in-the-Middle attacks. The sniffer in this version can also analyze encrypted protocols such as SSH-1 and HTTPS, and contains filters to capture credentials from a wide range of authentication mechanisms. The new version also ships routing protocols authentication monitors and routes extractors, dictionary and brute-force crackers for all common hashing algorithms and for several specific authentications, password/hash calculators, cryptanalysis attacks, password decoders and some not so common utilities related to network and system security [17]



Certificate file	SSL Server	Port	Hostname
C:\PROGRA~2\Cain\Certs\self_signed_64.233.187.93.crt	64.233.187.93	443	*.google.com
C:\PROGRA~2\Cain\Certs\self_signed_74.125.204.139.crt	74.125.204.139	443	*.google.com
C:\PROGRA~2\Cain\Certs\self_signed_74.125.203.136.crt	74.125.203.136	443	*.google.com
C:\PROGRA~2\Cain\Certs\self_signed_216.58.220.142.crt	216.58.220.142	443	*.google.com
C:\PROGRA~2\Cain\Certs\self_signed_52.41.220.234.crt	52.41.220.234	443	shavar.services.mozilla.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.11.148.crt	42.112.11.148	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.11.85.crt	42.112.11.85	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.11.89.crt	42.112.11.89	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_216.58.221.232.crt	216.58.221.232	443	*.google-analytics.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.8.93.crt	42.112.8.93	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.11.84.crt	42.112.11.84	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.8.103.crt	42.112.8.103	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_64.233.188.95.crt	64.233.188.95	443	*.googleapis.com
C:\PROGRA~2\Cain\Certs\self_signed_42.112.8.98.crt	42.112.8.98	443	google.com
C:\PROGRA~2\Cain\Certs\self_signed_66.155.40.250.crt	66.155.40.250	443	*.wordpress.org
C:\PROGRA~2\Cain\Certs\self_signed_216.58.221.228.crt	216.58.221.228	443	www.google.com

Fig.1. Arp Poisoning attack results using Chain

As is demonstrated in the figure 1, Chain can transfer all packets, passwords and certificates to the attacker that want to compromise all SSL and encryption measures.

5. Detection and Prevention Techniques

There various solutions for protecting MITM attack. According to the technologies used and applicability of the solution, we will analyze the solutions. Today's mobile environment also brings some challenges for configuring switches and other network elements. Therefore, solution should consider mobility to satisfy today's needs.

5.1. Static ARP[3]

Keeping ARP table static prevents ARP poisoning attacks. Since ARP cache table manually configured by user, ARP messages will be discarded and there won't be any risk. However, it is not applicable for most cases because many clients are dummy and mobile.

Although e-government sites provide preliminary protection using HTTPS, a hacker could use MITM to capture confidential information and resolve the password. In this case, however, the pirate must be connected to the same Local Area Network as the victim. There are some tools that the Pirate uses over Kali Linux to capture / resolve information, such as ARP Spoof, DNS Spoof, Sniffing, Wireshark and SSL Dump. The solution to this risk is to interrupt the MITM application, especially when disrupting the ARP Spoof during processing. You can change the interruption using Static ARP and use the ARP Watch feature to alert the system administrator. Moreover, Anti-Sniff can provide a function to scan a machine that catches information. In the study by Chomsiri (2007), it has been found that Static ARP application on the switch gives the best performance [20].

5.2. ARP Watch[3]

When attacker uses ARP Spoof, values in ARP table of gateway router and victim machine is changed. Therefore, monitoring the change in ARP table in time may give clue about SSL MITM attack. ARP Watch application detects the abnormal changes in ARP table. Running this application on victim gives alert about ARP Spoof attacks. This application does not work when the NIC card is changed or user is mobile and changes access point.

5.3. Anti Sniff[3]

Decoding HTTPS can be accomplished by running ARP Spoof and data capturing. Thus, a hacker needs to use some specific software (for example Sniffer and Ethereal). We can, by this, detect a machine that is running software to capture the information by checking if a machine is working on Promiscuous mode. Anti-Sniff is a generative application detecting a machine that is running in Promiscuous mode. The program can be installed on any one-machine in the network. The program, again, indicates some IP Address that is currently run data-capturing software. By this, the administrator can block the IP Address consequently. This technique is flexible since the administrator does not need to be in one particular machine to scan the machine. However, the drawback is that scanning and monitoring must be run all time [3].

Since attacker uses sniffer application to capture the traffic that does not belongs to him. To accomplish this, network card should run in promiscuous mode. Anti-Sniff application detects whether any node in the same network sniffs the network. This application should run on computer continuously to detect attacker. Once it is detected, IP of the attacker can be blocked.

5.4. Perspectives[1]

In order to manipulate HTTPS session, attacker should publish its self-signed certificate. When client receives this untrusted certificate, browser generally warns the user. Most of the users accept the untrusted certificate. Perspectives use notary for deciding whether the certificate is trusted or not. Notary monitors and collects the public key data of sites and when client receives a public key of a server it checks with notary. This solution is still prone to attack because attacker can also behave like notary.

5.5. DVCert[1][5]

Direct Validation of Certificates (DVCert), a productive and simple to convey convention that gives more grounded testament approval and compelling recognition of MITM assaults without utilizing third-parties. DVCert originates from a straightforward perception – clients have effectively settled secrets (e.g., passwords) with their most essential web applications. DVCert permits web applications to utilize these secrets to straightforwardly and safely bear witness to for the legitimacy of their authentications without presenting those insider facts to disconnected assaults. After a solitary round-trip DVCert transaction, a browser gets the data required to approve every one of the endorsements that could be utilized during a session with the web application, including certificates from different domains. Accordingly, to execute a MITM assault, an attacker needs to trade off a CA as well as each focused on web space.

5.6. Channel ID[1]

Strong client authentication prevents from MITM even if the attacker impersonates the server. Google Channel ID is one of this approach. When client connects to server, server generates TLS Channel ID and client stores this information in cookie cryptographically. Every server generates different Channel ID and does not contain any user information.

This approach requires first connection should be made between genuine client and server. If the attacker interferes the first connection, he can capture all traffic.

5.7. Voting-Based Security Enhanced ARP[4]

In security improved ARP, an endeavor is made to give decency in voting by depending on the uniform transmission rate in the Ethernet. Be that as it may, the voting plan is refined further to give decency in voting in a more summed up condition contrasted with other voting-based plans.

There are some assumptions for this technique to work correctly. One of them is the number of voting-cognizant good nodes should be larger than that of the malicious nodes in the same subnet. The other is all the nodes should be connected through wire.

6. Blockchain Hacking

One of the famous cryptography using algorithms that have been spreading rapidly is the blockchain methodology. This has been used for bitcoin like virtual money systems and also “*proof of existence*” functions like notary and land registration. The questions like “Can this systems be hacked with the same methodology for SSL hacking?” still remains vague. However, there is evidence that cryptocurrencies have been hacked in some cases. Any SSL cryptographic application, the cryptographic algorithm is almost always more vocal than the program that implements it. Generally, blocking is affected by any security vulnerabilities or weaknesses for any encryption solution. An SDLC programming error or a lack of good private key security can result in serious security vulnerabilities. This requires software developers to implement Secure Development Life Cycle (SDL) operations to reduce the most inaccuracies before using it in a crypto currency or in a blockchain project.

There may be situations where information pirates manipulate crypto-currency software to steal money. However, in a recent event, hackers caused a coding error that spoiled everyone's wallet if it could not play any value. The thief could not make money but caused others to lose their money [18]. There is evidence that hacking of cryptocurrencies have resulted hundredths of millions of Dollars. This is primarily because hackers have mostly targeted online wallets and cryptocurrency exchanges instead of a cryptocurrency’s blockchain [19].

A good crypto makes cryptotext, which is obtained with SSL, random in a meaningless way. However, with any blockbuster technology, it is possible to understand the format of the blocks. Some letters, characters, or numbers are usually in the same place. Just as each block is a function encoded with the hash value of the previous block; all blocks are in this relationship with the previous and the next. In this case, the general protection of the basic cryptographic system may be affected by the attenuator bits. Encryption is not a big problem if it is not normally weak, but it gives the attackers a hint of anyway.

Many cryptographers are wondering if SHA-256, which contains the same mathematical weaknesses as the SHA-1 example, is a concern for bitcoin and blockchain, but this has not yet been found. SHA-256 encryption appears to be strong enough for the foreseeable future. Even more importantly, because it is the world's financial transaction and HTTPS transactions are protected by SHA-256, when someone breaks it can only cause much greater damage than the bitcoin and block chains. Although "crypto-agility" is a very important skill to change passwords and stay in the basic program for a crypto currency or block chain. One of the most common hacking areas surrounding Bitcoin can be applied to any block-chain project. It can be determined how often the centrally-controlled website is being attacked.

7. Preventing SSL Intrusion

In order to be protected from such an attack, the user must first be more careful. If you are going to log in to a critical application (which is your important information), make sure that the address passed through the URL is HTTPS.

An intrusion detection system (IDS) can be used to prevent internal MiTM attacks. IDS will basically follow the network traffic closely. If someone tries to miss the traffic flow and intervene, IDS will prompt them instantly. However, the disadvantage of IDS is that it can generate false positive warnings many times. This causes the users to disable the IDS. Tools and techniques that use advanced address resolution protocols such as XARP or ARPOn, and measures such as dynamic host configuration protocol (DHCP) surveillance on switches can limit or prevent ARP spoofing. This can also help prevent MiTM attacks.

Another solution to block MiTM attacks is to use a virtual private network (VPN). The use of such cryptographic tunnels creates additional secure layers when you access corporate secret networks over connections such as Wi-Fi. Projects such as Public Net must be implemented via VPN. In addition, companies should have an appropriate process control and monitoring process to be aware of staff activities [28].

Banks do not provide HTTP support in order to take such countermeasures against Internet banking applications. When attempting to access the critical application via the HTTP link, a 404 or 403 error page is displayed that is related to the absence of page access. Exposure to an attack like SSLStrip is blocked on this site.

It is possible to close this shortcoming of ARP with some measures taken on switching devices. IP-MAC address mappings in switching devices must be port-based. If done so, it will not be possible to have a different MAC address from the corresponding port. This prevention; It is called "Dynamic ARP Inspection" or "Dynamic ARP Protection". At the same time, ARP requests from IP addresses that are not valid, such as 0.0.0.0 or 255.255.255.255, will also be blocked.

In an environment without a DHCP server, IP address - MAC address mappings (where IP addresses are static) must be manually performed on the switching devices.

There are some features of switching devices that can be used for prevention. Dynamic ARP Inspection (DAI) is a security feature that is available on Cisco Catalyst 6500 Series switches running Cisco IOS Software or Cisco Catalyst OS. Dynamic ARP inspection helps prevent ARP poisoning and other ARP-based attacks by intercepting all ARP (Address Resolution Protocol) requests and responses, and by verifying their authenticity before updating the switch's local ARP cache or forwarding the packets to the intended destinations. Note that on Cisco Catalyst 6500 Series switches, Dynamic ARP requires Supervisor 2, Supervisor 32, or Supervisor 720. As previously stated, a Supervisor 720-3B was used in these tests.

The DAI verification consists primarily of intercepting each ARP packet and comparing its MAC address and IP address information against the MAC-IP bindings contained in a trusted binding table. DAI discards any ARP packets that are inconsistent with the information contained in the binding table. The trusted binding table is dynamically populated by DHCP snooping when this feature is enabled. In addition, DAI allows the configuration of static ARP ACLs to support systems that use statically configured IP addresses and that do not rely on DHCP.

DAI can also be configured to drop ARP packets with invalid IP addresses, such as 0.0.0.0 or 255.255.255.255, and ARP packets containing MAC addresses in their payloads that do not match the addresses specified the Ethernet headers [25].

Another important feature of DAI is that it implements a configurable rate-limit function that controls the number of incoming ARP packets (default is 15pps on “untrusted” interfaces). This function is particularly important because all validation checks are performed by the CPU, and without a rate-limiter, the switch would be much more at risk to Denial-of-Service (DoS) attacks. When the rate of incoming ARP packets exceeds the configured limit (15pps default), the switch places the port in an “error-disabled” state.

The port remains in this state until you intervene. You can use the “errdisable recovery” global command on the switch to enable error disable recovery so that the ports will automatically emerge from the “error-disabled” state after a specified timeout period [25]

8. Assessment

Various hacking methods and prevention techniques were discussed in previous section. For the most of attacks there is a way to prevent. Solution is valuable if it is easy to apply and suits the existing protocols. Many of the solutions require introducing new authority and new protocols to understand the genuineness of the CA. Sometimes solution is easy for experts. For example for ARP poisoning using static ARP table is simple solution but not flexible. End users are not expert generally and do not love complex solutions and configuring the machine for different networks. Therefore for the prevention purposes the best solution is educating the users for cyber threats, because for the most scenarios there is an indication of weird and unusual things happen. The second solution is enhancing the existing protocols for security purposes and enforcing host and clients to upgrade the newest protocol. Also national authorities should take precaution for internal and external threats. For the organizations IT department should regulate authorization of employees. Nobody should have administrator rights on personal computer except experts.

In case of legal hacking scenarios, firewall companies legally do MITM attack and the inside business network every employee should accept the fake certificates. For the passive hacking, the most common way of decrypting the data is brute force attack. Since today's computation power comes from the cloud, decryption can be made by distributing key ranges to VMs over cloud. Another approach is using quantum computers. The computation power of quantum computers 1000 times faster than today's most advanced traditional computer poses a prospective threat to decode SSL encryptions in a very short time. Therefore computationally infeasible cryptographic protocols can be decrypted with quantum computers. This is a prospective challenge that needs to be delved into by researchers.

9. Conclusion

Although, e-commerce and e-government sites have preliminary protections on the information using HTTPS, hackers are able to capture and decode the confidential information by using SSL MiTM attacks. There found to be many cases that MiTM attacks have targeted e-government sites. In a report released last week, the Open Technology Fund (OTF)—a U.S. Government program funded to support global Internet freedom technologies—stated that the JingWang app, which the Chinese government has forced citizens of Xinjiang province to install on their Android devices, does not protect users' private information; and, besides that, it is vulnerable to man-in-the-middle attacks. [26].

In addition to e-government sites, there are many cases that MiTM attacks have targeted e-voting applications. The network-based nature of distance e-voting makes it prone to MITMs, and especially the ones that take place locally on a user's inadequately secured PC. In that context is very easy for malicious software to hijack an e-voting session and replace a legitimate vote with a fraudulent one [27].

The hacker has to hook up to the same Local Area Network with a victim. There are some tools that the hacker uses to capture/decode the information, for instance, ARP Spoof, DNS Spoof, Sniffing and SSL Dump. The solution brought to this situation is to interrupt SSL Man In The Middle while it is processing, especially interrupting ARP Spoof. The interruption can be encouraged by using Static ARP at the switch and use ARP Watch to alert the administrator. More to the point, Anti-Sniff provides a function to scan a machine that is capturing information [3].

SSL/TLS protocol is the standard cryptographic protocol that establishes the secure session between the client and the server. Session between client and server is established during handshake protocol phase that Originate from misapplication of cryptographic primitives and talk about in detail each of the flaws, the basic security principle violations, and the resulting assaults to demonstrate the shortcoming of SSL/TLS conventions. Nonetheless, the certificate model is not adequate to distinguish and anticipate MITM attacks. It is also mentioned that even having two factor authentications, it is possible to hack communication with MITB attacks.

There are various tools and techniques to be used by hackers and script kiddies that can easily use a published script in YouTube and other open internet sources. Even for an ARP spoofing attack there is no need for further sophistication and deep knowledge of hacking due to widespread information and techniques designed and guided in a fool statement mode. This situation poses a very big threat particularly for e-government sites and institutions that needs further sophisticated counter measures and prevention techniques.

Various prevention techniques were mentioned in this paper. Some of the solutions are easy to apply by manually configuration and some of them require amendment to existing system to make it more secure. There are tradeoffs for the provided solutions. Some manual configuration of ARP table is easy for experts but you need to adjust when you relocate. Therefore many protocols designed for making our life easy. It is important to use the tools carefully with being aware of threats. Education is first part of protection because many of attacks need get confirmation by user. Second important part of protection is enhancing the existing protocols. Many of given solution may be adapted to newest version of SSL handshake.

In case of protection of business and government, the only existing solution is doing MITM attack with by taking permission of employees. We can also decrypt stored encrypted data with brute force attack but it is computationally infeasible. Only using quantum computers, it might be feasible but there is not any existing solution in literature.

References

[1] H. Seung-Woo, et al. "*A survey on MITM and its countermeasures in the TLS handshake protocol.*" Ubiquitous and Future Networks (ICUFN), 2016, Eighth International Conference on. IEEE. 11-12.

[2] O. Eisen, "*Catching the fraudulent Man-in-the-Middle and Man-in-the-Browser.*" 2010, Network Security.

[3] T. Chomsiri, "*HTTPS Hacking Protection*." Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on. Vol. 1. IEEE.

[4] N. Seung Yeob, S. Djuraev, and M. Park. "*Collaborative approach to mitigating ARP poisoning-based Man-in-the-Middle attacks*." Computer Networks 57.18 2013, pp. 3866-3884.

[5] D. Italo, M. Ahamad, and P. Traynor. "*POSTER: Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties.*" pp 199-216

[6] L. Wu, et al. "*SSL-DP: a rootkit of network based SSL and TLS traffic decryptor.*" Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second. IEEE.

[7] D. Jiang, X. Li, and H. Huang. *"A study of man-in-the-middle attack based on SSL certificate interaction*." Instrumentation, Measurement, Computer, Communication and Control, 2011, First International Conference on IEEE.

[8] O. Rolf, R. Hauser, and D. Basin. "*SSL/TLS session-aware user authentication–Or how to effectively thwart the man-in-the-middle.*" Computer Communications 29.12, 2006, 2238-2246.

[9] O. Rolf, R. Hauser, and D. Basin. "*SSL/TLS session-aware user authentication revisited.*" Computers & Security 27.3, 2008, 64-70.

[10] D. Manik Lal, and N. Samdaria. "*On the security of SSL/TLS-enabled applications.*" Applied Computing and informatics 10.1, 2014: 68-81.

[11] J. Sharp, "*Man in the browser attacks: worse than viruses"*, 2008,_Latest Access Time for the website is 20 June 2018

[12] Entrust, "*Winning the Fight Against Man-in-the-Browser - Entrust IdentityGuard Mobile Now Available*" 2010, Newswire

[13] Intel, "*berserk analysis*", 2010,_Latest Access Time for the website is 20 June 2018

[14] P. Mutton, "*95% of HTTPS servers vulnerable to trivial MITM attacks*" Netcraft, 2016

[15] P. Sec, *"Renegotiation Attack"*, 2017 Latest Access Time for the website is 20 June 2018

[16] Blackhatinside, *"How to do arp poisoning / spoofing with Kali Linux 2016.2 | arpspoof | wireshark | steal passwords | sniff packets"*

[17] Oxit, "*Cain & Abel*" , 2010, Latest Access Time for the website is 20 June 2018

[18] R. Grimes, "*Hacking bitcoin and blockchain*", 2017, Latest Access Time for the website is 20 June 2018

[19] R. Sharma, "*Bitcoin Mining Market Hacked: $70M Stolen from NiceHash"*, 2017

[20] C. Thawatchai. 2007, *HTTPS hacking protection.* IEEE. 2. 590-594. 10.1109/AINAW.2007.200.

[21] W. Yang, X. Li, Z. Feng and J. Hao, "TLSsem: A TLS Security-Enhanced Mechanism against MITM Attacks in Public WiFis," *2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Fukuoka, pp. 30-39.doi: 10.1109/ICECCS.2017.24

[22] C. J. D'Orazio, K. Kwang, R. Choo, "*A technique to circumvent SSL/TLS validations on iOS devices"*, Future Generation Computer Systems, Volume 74, 2017, Pages 366-374, ISSN 0167-739X

[23] Arnbak, A & Asghari, Hadi & Eeten, Michel & Eijk, Nico. 2014, *Assessing legal and technical solutions to secure HTTPS*. 12. 1-15. 10.1145/2668152.2673311.

[24] L. D. Manik & S. Navkar. "*On the security of SSL/TLS-enabled applications*". Applied Computing and Informatics. 2014, 10. 10.1016/j.aci.2014 .02.001.

[25] J. King, K. Lauerman, *"ARP Poisoning (Man-in-the-Middle) Attack and Mitigation Techniques*", 2016, A CSSTG SE Residency Program, CISCO White Paper

[26] R. Leandro, "*Chinese government surveillance app is vulnerable to MITM attacks",* 2018, Sidechannell. Latest Access Time for the website is 20 June 2018

[27] D. Mitropoulos, D. Spinellis *"Securing e-voting against MITM attacks"*, 2009, 13th Panhellenic Conference on Informatics, Corfu, Greece, September, pp. 1-5

[28] S. Hidayatullah, "*Man in the middle attack prevention strategies*" 2018, Computer Weekly, Latest Access Time for the website is 20 June 2018