STRICT TRANSPORT SECURITY IMPLEMENTATION ON HTTP/HTTPS CONVERSION

CENG 3544, COMPUTER AND NETWORK SECURITY

Ahmet Gurbuz
ahmetgurbuz2@posta.mu.edu.tr
Murat Gun
muratgun@pota.mu.edu.tr

Monday 14th June, 2021

Abstract

This project has been prepared in order to emphasize the importance of Strict Transport Security Implementation within the scope of the ceng3544 Computer and Network Security course. This report contains theoretical and applied information. What is HTTP,HTTPS,HSTS? What do they do? What are the differences between them? What does it have to do with data security? How are these protocols hacked? We will look for answers to your questions.

1 Introduction

Http, which is the basis of the web in today's world, has been used for years. Http is simply the protocol that enables data flow between web sites and web browsers. However, since this protocol was created in the early days of the internet, it is very weak in terms of security. So how to fix this security vulnerability? In this document, we will explain the working methods of http/https and hsts protocols. In addition, we will present the demo work on the attacks that led to the development of these protocols in our report.

2 Fundamentals

2.1 HTTP

HTTP is an application-level communication protocol for hypermedia information systems distributed from a source and open to common use.[1]HTTP is the protocol that provides communication between web browser and web server in general.

2.2 HTTPS

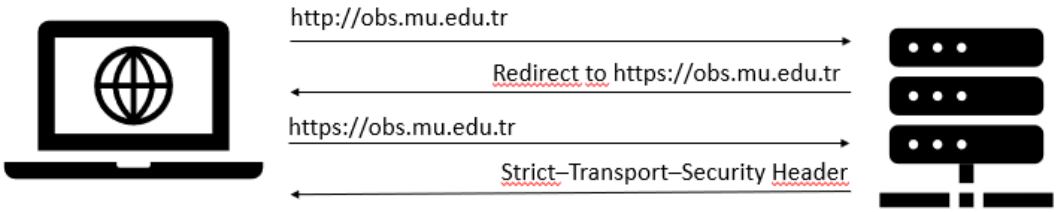
HTTPS is the protocol that provides communication between the web browser and the web server, as in HTTP, using SSL or Tls encryption protocols. In short, it is an encrypted http protocol.

2.3 SSL/TSL

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are encryption protocols designed to provide secure communication over a computer network.[2]

2.4 HSTS

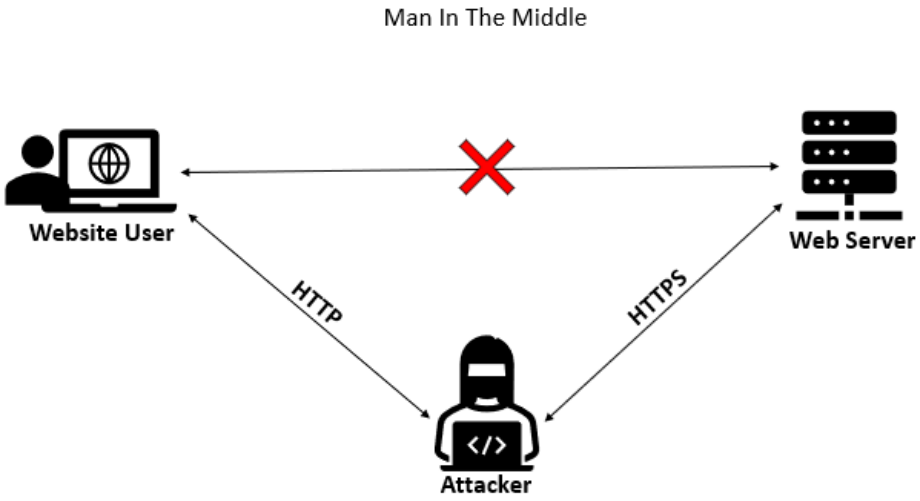
HSTS is a web security policy mechanism that helps protect websites against downgrade attack and session stealing attacks. Web servers are the mechanism to specify that requests sent to it must be HTTPS only.[3]



HSTS Mechanism (Figure 1).

2.5 Man In The Middle Attack

A man-in-the-middle attack is a type of attack in which the attacker secretly transmits or changes the communication between two parties communicating directly with each other. Data packets circulate freely over the communication network. Especially broadcast packets can be seen by all devices connected to the same network. In principle, machines that receive a packet that does not have its own IP at its destination should not take any action on these packets. However, if desired, they can interfere with these packages or learn their contents. Man-in-the-middle attack can be summarized as capturing and manipulating packets on the network.



Man-In-The Middle (Figure 2).

2.6 Downgrade Attack

A downgrade attack or rollback attack is a type of cryptographic attack that occurs when a computer system or communication protocol is forced to abandon its high-quality mode of operation to an older, lower-quality mode of operation, which is usually provided for backward compatibility with legacy systems. An example of such a flaw was found in OpenSSL that allowed an attacker to negotiate the use of a lower version of TLS between the client and server. This is one of the most common types of downgrade attacks. Another example is to intercept web traffic and redirect the user from a secure, HTTPS version of a website to an unencrypted HTTP version.[4]

3 Theory

3.1 Why is HTTP weak ?

The main reason why the HTTP protocol remains weak in terms of security is that it transmits information in simple texts without encryption. When an information packet is sent to a web server, the packets can be intercepted by people on the same network. For this reason, user data can be easily stolen in in-network attacks.

3.2 What is HTTPS , How did it originate ?

The fact that the HTTP protocol is very weak in terms of security necessitated the development of this protocol and the HTTPS protocol was developed. HTTPS was created simply by combining the HTTP protocol with the SSL/TLS protocols. It provides sending or receiving data encrypted with HTTPS, SSL or TLS protocol with HTTP protocol.

3.3 What are the Attack Types, How is the Data Stolen?

The most well-known of the types of in-network attacks are man in the middle attacks. The purpose of this type of attack is to listen or manipulate network traffic. In this type of attack, the attacker gets between the victim computer and the network. In this way, it is ensured that all communication between the victim computer and the server takes place over the attacker's computer. This allows an attacker to access and manipulate packets on the network.

3.4 Measures taken against attacks

At the beginning of 2010, some types of attacks were seen quite often, especially in e-commerce sites and websites where credit card information is used. Over the years, ways have been sought to eliminate all of these situations. Basically, it has been determined that there are 2 steps to be the target of this attack. The first is to be on the same network as the attacker, and the second is to send an http request to the connected server. It was realized that by avoiding the second step, the user could be protected against all these attacks, and for this reason, a mechanism was developed. This system is HSTS (Strict Transport Security).

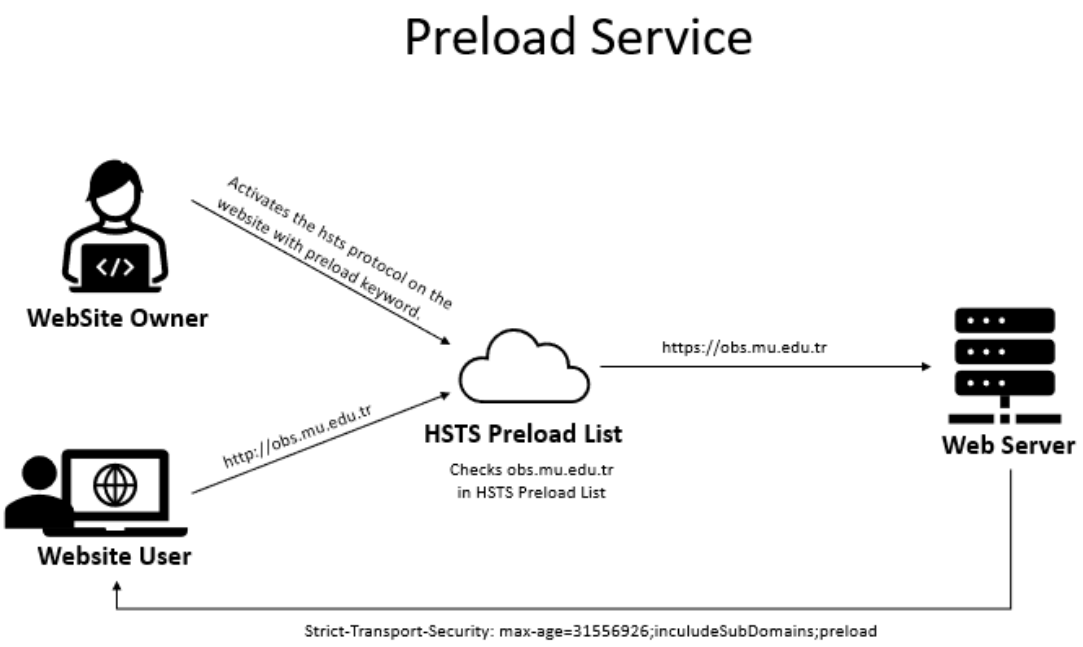
3.5 What is HSTS?

HSTS is generally a mechanism that forces web browsers to send all network packets via HTTPS. Simply put, when you want to make a connection to a website over http, the server sends you a

warning and tells you that it does not accept http requests. Then it requests the web browser to send the request again with HTTPS. The web browser converts the http request to HTTPS and transmits it back to the server. As a result of this forwarded request, the requested data packet and the HSTS header are transmitted to the web browser. However, resolving all of these requests in response on the server constantly puts a serious workload on the server. In order to prevent this, this process is requested to be only once. Therefore, the parameter called max-age is added to the hsts header. This parameter tells the web browser how valid this rule is in seconds. In other words, for a parameter that says max-age:300, the web browser automatically converts the links from http to https for 5 minutes without making a second request to the web server. And in this way, the workload is reduced and at the same time security is provided.

3.6 What is the Preload Service?

No matter how secure the HSTS connection is, the first request can be HTTP. For this reason, google and web browser developers created a service called preload in 2018. Websites that fulfill the necessary conditions can register for this service. Here's how Google's preload service works. The user sends an http request, the web browser checks the preload list. If the site is included in this list, it automatically converts the request from HTTP to HTTPS without establishing the first connection. In this way, even for the first single request, hacking is prevented. With this method, the possibility of the user being attacked on a site that has activated HSTS security is eliminated.



Man-In-The Middle (Figure 3).

4 IMPLEMENTATION-DEMOS

4.1 Man In The Middle Attack With Kali

In the application part of our report, we simulated an attack with 2 virtual machines that we installed on VMware. The machine we use for the attack has the Kali distribution. Our victim machine is a virtual machine with Windows 10 operating system. Before we started the attack, we ran two machines on the same network because we wanted to simulate an in-network attack. The reason we chose Kali for the attack is that many tools for cyber security are pre-installed. Since the type of attack we wanted to do was Man In The Middle Attack, we had to choose an appropriate tool. We used Bettercap, one of the most famous tools. Our priority was to detect the victim machine(s) on the same network. Within Bettercap, there is a plug-in that scans devices on the network, and we can run this plug-in with the 'net.probe on' command. Then we use the 'net.show' command to display the scan results. After selecting the target machine(s) from the displayed list, we can start our main attack, the arp spoof attack. This attack sends certain packets to the victim machine, allowing it to detect the attacking device as a gateway. From the moment the attack starts, the victim machine's internet access passes over the attacking machine. This attack can be performed on the Bettercap tool. To perform it, firstly, the IP address of the victim computer is selected with the 'set arp.spoof.targets' command. With the 'set arp.spoof.fullduplex true' command, the victim machine's internet traffic is allowed to pass through the attacking machine at the time of the attack. Finally, the attack is started with the command 'arp.spoof on'. From this moment on, the attacking device has literally become man-in-the-middle and can control all the traffic of the victim machine. Many tools can be used to listen to the traffic of the victim machine, but the traffic of the victim machine can be monitored simply with the 'net.sniff on' command on Bettercap. From now on, all the attacker has to do is wait.

4.2 Downgrade Attack With Kali

Although the data of the sites communicating with the HTTP protocol can be accessed in the attacks made in step 4.1, the victim data cannot be accessed because the data is transmitted encrypted on the websites communicating with the HTTPS protocol. In such cases, attackers try to downgrade the HTTPS connection to HTTP. This type of attack is called Downgrade Attack. The main purpose of this attack is to reduce the HTTPS connection to the HTTP connection, to ensure that user data is sent unencrypted, and to easily capture packets in this way. This downgrade forces the server to use a weaker protocol than the TLS protocol between the client. In this way, HTTP protocol can be used instead of HTTPS protocol on some sites. In our example, we used the caplets called 'hstshijack' on Bettercap. This caplet is written in javascript. If we run this file with the 'hstshijack/hstshijack.js' command before starting the apr spoof attack, we can downgrade the victim with man in the middle attack.

5 CONCLUSION

As a result, in this report, we explained the concepts of HTTP , HTTPS, SSL, TLS, HSTS , Man in the middle , Downgrade Attack in a way that anyone without technical knowledge can understand. We also demoed the attacks on these protocols and explained step by step how these attacks can be done. In general, the aim of this report was to reveal what these concepts are, to eliminate the lack of knowledge in this field and to report our project for the ceng3544 Computer and Network Security course. With this report, we have achieved this goal.

References

[1] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext transfer protocol–HTTP/1.1.

[2] Dierks, T., & Rescorla, E. (2008). The transport layer security (TLS) protocol version 1.2..

[3] Mozilla, "Strict-Transport-Security", www.developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security, Retrieved 15.06.2021

[4] Praetorian,"Man in the Middle TLS Protocol Downgrade Attack", www.praetorian.com/blog/man-in-the-middle-tls-ssl-protocol-downgrade-attack/ Retrieved 15.06.2021