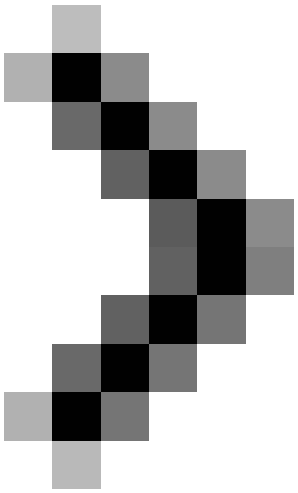
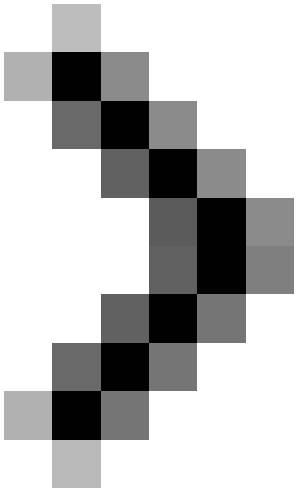
Learn about Man-in-the-Middle attacks - Obtain SSL control

In the next part of this series, we will introduce you to SSL spoofing attacks, besides some theories under SSL connections and what is safe.

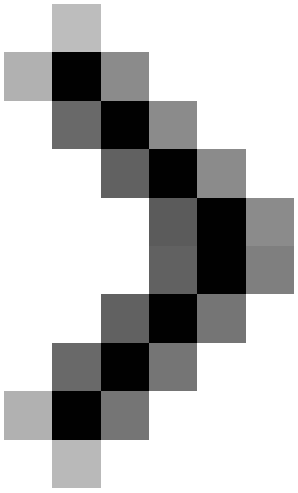
Network Administration **- In the next part of this article series, I will talk about SSL spoofing, along with some theories under SSL connections and what makes them secure. all** .



Learn about Man-in-the-Middle attacks - ARP Cache spoofing



Learn about Man-in-the-Middle attacks - DNS spoofing



Learn about Man-in-the-Middle attacks - Take over Session control
So far in this article, I have shown you about ARP cache spoofing, DNS spoofing and session hijacking attacks in this series of man-in-the-middle attacks. In this fourth part, we will talk about SSL spoofing, which is one of the most dangerous MITM attacks because it allows exploiting the services that users think are safe. We will begin this introduction by discussing some of the theories underlying SSL connections and what can make them safe, and then introduce how to exploit this type of connection. . And like the previous sections, the last part of the article will be some measures to detect and prevent this type of attack.

SSL and HTTPS

Secure Socket Layers (SSL) or Transport Layer Security (TLS) under its more modern implementation, are protocols designed to provide security for network communications by encryption method. This protocol is easily combined with the other protocols to provide a secure implementation for the service that the protocol provides. Examples here include SMTPS, IMAPS and HTTPS. The ultimate goal is to create secure channels on insecure networks.

In this section, we will focus on introducing SSL attacks on HTTP, known as HTTPS, as it is the most common use case of SSL. You may not realize it, but you're probably using HTTPS on a daily basis. The most popular email services and online banking applications rely on HTTPS to ensure communication between your web browser and their servers is securely encrypted. Without using this technology, anyone with a packet of 'sniffing' on the network can detect the username, password and anything else hidden.

The process used by HTTPS to ensure data security is to tighten the centers involved in distributing certificates between trusted servers, clients, and third parties. Take an example of a case where a user is trying to connect to a Gmail email account. This process will include a few easy steps, which have been simplified in Figure 1

below.

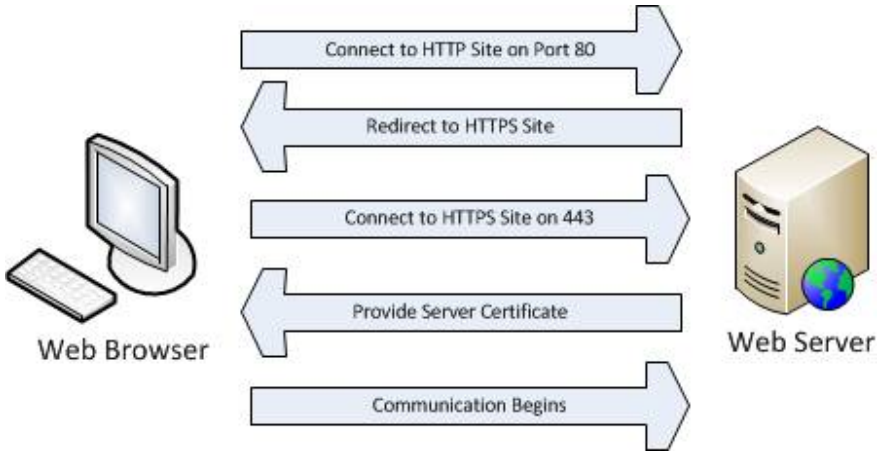


Figure 1: HTTPS communication process

The process outlined in Figure 1 is not a detailed process, but it will basically work as follows:

1. The client browser connects to http://mail.google.com on port 80 using HTTP.
2. The server redirects the client HTTPS version of this site using HTTP 302 code.
3. The client connects to https://mail.google.com on port 443.
4. The server will provide a certificate to the client including its digital signature. This certificate is used to verify the identity of the site.
5. The client uses this certificate and authenticates this certificate with its list of trusted certificate authorities.
6. Encrypted communication will happen later.

If the validation process fails, it means that the website has failed to verify its identity. At this point, users will see a certificate validation error and they can still continue with the possible risks, since there is probably no real communication with the website they think they need to access. updated.

Destroy HTTPS

This process is considered highly secure a few years ago when an attack published that it could successfully hijack the communication process. This process does not involve defeating SSL, but rather destroying the 'bridge' between unencrypted and encrypted communications.

Moxie Marlinspike, a leading security researcher, said that in most cases, SSL has never been directly attacked. Most of the time an SSL connection is initiated via HTTPS so it could be because someone redirect an HTTPS through an HTTP 302 response code or they clicked on directing them to an HTTPS site, for example. as login button. The idea is that if you attack a session from an unsecured connection to a secure connection, in this case from HTTP to HTTPS, you will attack the bridge and possibly 'man- in-the-middle 'SSL connection before it appears. To do this effectively, Moxie created an SSLstrip tool, we will use this tool below.

The process is quite simple and recalls the attacks we studied in the previous parts of the series. It is outlined as in Figure 2 below.

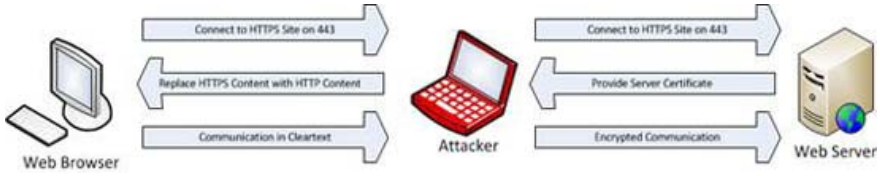


Figure 2: Take control of HTTPS communications

The process outlined in Figure 2 works as follows:

1. Traffic between the client and the first server will be blocked
2. When encountering an HTTPS URL, sslstrip will replace it with an HTTP link and will map its changes.
3. The attack machine will provide certificates for web servers and fake clients.
4. The traffic is retrieved from the secure website and returned to the client.

The process is quite good, the related server still receives SSL traffic without knowing about this difference. The only obvious difference in the user experience is that traffic will not be flagged as HTTPS in the browser, so an experienced user will be able to see it as an anomaly.

Use SSLStrip

This attack utility is called SSLstrip, you can download it here. This program only runs on Linux so you can download and install it yourself, in case you do not want to do this, you can download and run Backtrack 4, this is a pre-installed version of the utility. this tool.

Once you have access to SSLstrip, you will see there are a few privileged tasks that need to be performed. First, the Linux distribution you are using must be configured to forward IP. To do this, enter the command echo "1"> / proc / sys / net / ipv4 / ip_forward into a shell.

Figure 3: Enable IP Forwarding

When done, we must make all HTTP traffic blocked will be routed to the port where SSLstrip will listen. This is done by changing the firewall's iptables configuration. Use the command iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port.

Figure 4: Configure IPTables to properly route HTTP traffic

Obviously, you need to replace it with a certain port of your choice. After doing this configuration, we can run sslstrip and configure it to listen on the port specified by the sslstrip -l command.

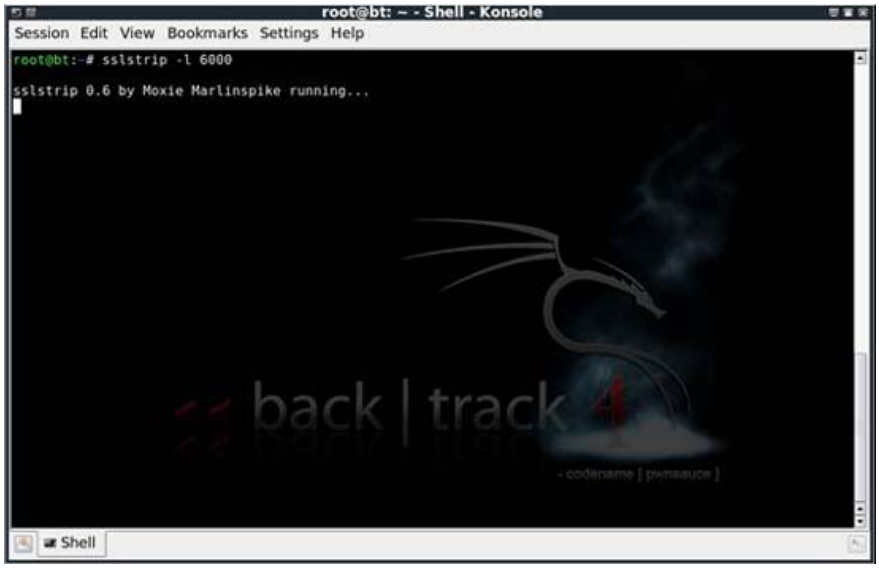


Figure 5: Using sslstrip

The final step in this process is to configure ARP spoofing to block the destination host traffic. We did this using Cain and Abel in Windows in the previous sections, but in this case, we will use the arpspoof utility, which is the utility included in Backtrack 4. The command to execute is arpspoof -i -t.

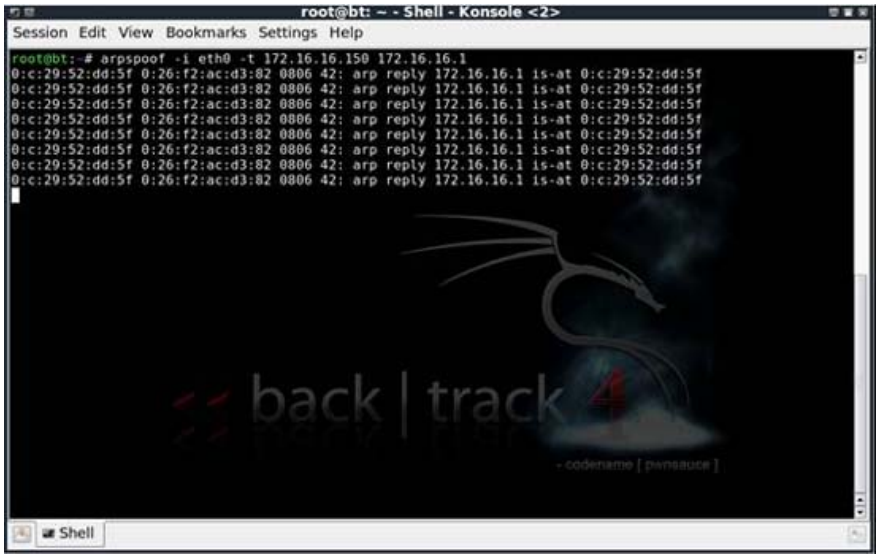


Figure 6: Configuring ARP spoofing

Using this command you need to replace with the network interface you are performing these actions (eth0, eth1, .), with the target client's IP address, which is the IP address of the gateway router whose entry is pepper is in use.

When done, you should be able to hijack any SSL connections that are being established. From here, you can launch the 'sniff' utility and collect passwords, other personally identifiable information such as credit card numbers, etc. from traffic.

Prevention measures

As introduced above, SSL hijacking in this way is virtually undetectable from the server side because the server thinks it still communicates normally with the client. It has no idea that it is communicating with a client by proxy. Fortunately, there are a few things that can be done from the client context to detect and prevent these types of attacks.

- **Using HTTPS secure connection -** When you perform the attack described here, it takes away the security aspect of the connection, which can be determined in the browser. This means that if you log into your online bank account and find that it is just a standard HTTP connection, there is definitely something wrong here. Whenever the browser you choose to use, make sure you know how to distinguish secure connections from unsafe connections.
- **Save your online bank account at home** - The chance for someone to block your traffic on your home network is much less than your workplace network. This is not because your home computer is safer (but the truth is probably less secure), but the truth is if you have only one or two computers at home, you only have to care about taking over. session control if someone in your home (big boy) starts watching and practicing hacking videos on YouTube. On the corporate network, you don't know what is going on in the

lobby or branch office many kilometers away, so the potential source of attack is plenty. One of the biggest goals for session hijacking is online banking, but this culprit can be applied to anything.

- **Secure the computers inside the network** - Don't hit a dead horse, but again, the same attacks are usually executed inside a network. If your network devices are secure, then the risk of compromising hosts is then used to launch session hijacking attacks.

Conclude

This form of MITM attack is one of the deadliest forms because it makes us think we are in a secure connection, but the truth is not. If you consider the number of secure sites that you visit every day, then consider the potential impact if all of those connections are unsafe and the data falls into the bad guys then you will Understand the level of danger that can happen to you or your organization.