**[BeEF - The Browser Exploitation Framework Project](https://beefproject.com/)**

Shyam Manohar. B, Pagerathan. V.M and Venkata Surendra .M

Amrita Vishwa Vidhyapeetham, Chennai TN 601103, INDIA

Abstract:

BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on the web browser.

Amid growing concerns about web-borne attacks against clients, including mobile clients, BeEF allows the professional penetration tester to assess the actual security posture of a target environment by using client-side attack vectors. Unlike other security frameworks, BeEF looks past the hardened network perimeter and client system, and examines exploitability within the context of the one open door: the web browser. BeEF will hook one or more web browsers and use them as beachheads for launching directed command modules and further attacks against the system from within the browser context.

**Keywords:** Browser, Linux, Terminal.

**INSTALLATION**:

Open terminal in the system. Run the following commands to install BeEF.

We can download BeEF by running the below command:

git clone https://github.com/beefproject/beef beef

cd ~/beef

Bundler is essential for tracking and installing the missing gems in ruby application. Run bundler to install gems in project directory.

gem install bundler

Next, run the install script in the BeEF directory:

./install

To start BeEF, simply run:

./beef

We can update BeEF by running the following command

git pull

The default login credentials for BeEF are beef and beef . The credentials can be changed in the configuration file config.yaml

**EXPLANATION:**

1. **GMAIL CREDENTIALS PHISING**

A phishing link is sent to user. Once the user clicks on the link, the user is presented with a fake login page for gmail. When the user enters credentials they are sent back to beef, and they are redirected to the real gmail login page (where they are prompted for their login again). These credentials can be seen on the beef GUI.

STEPS:

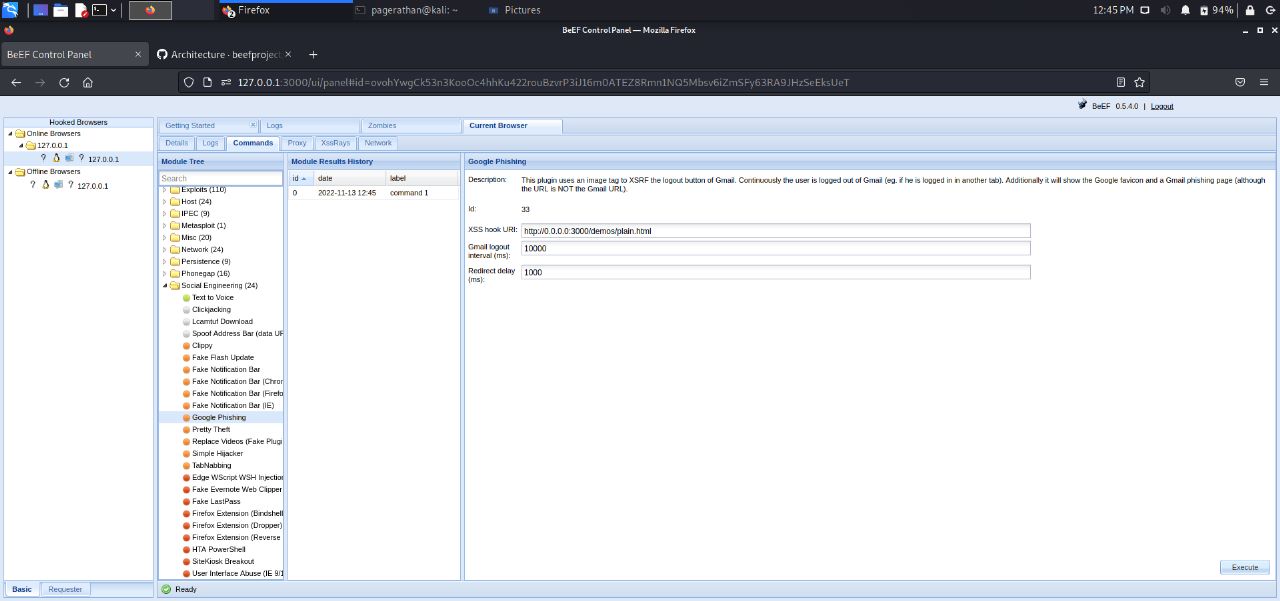
1. When you find a module named "Google Phishing" module in the "Social Engineering" folder.

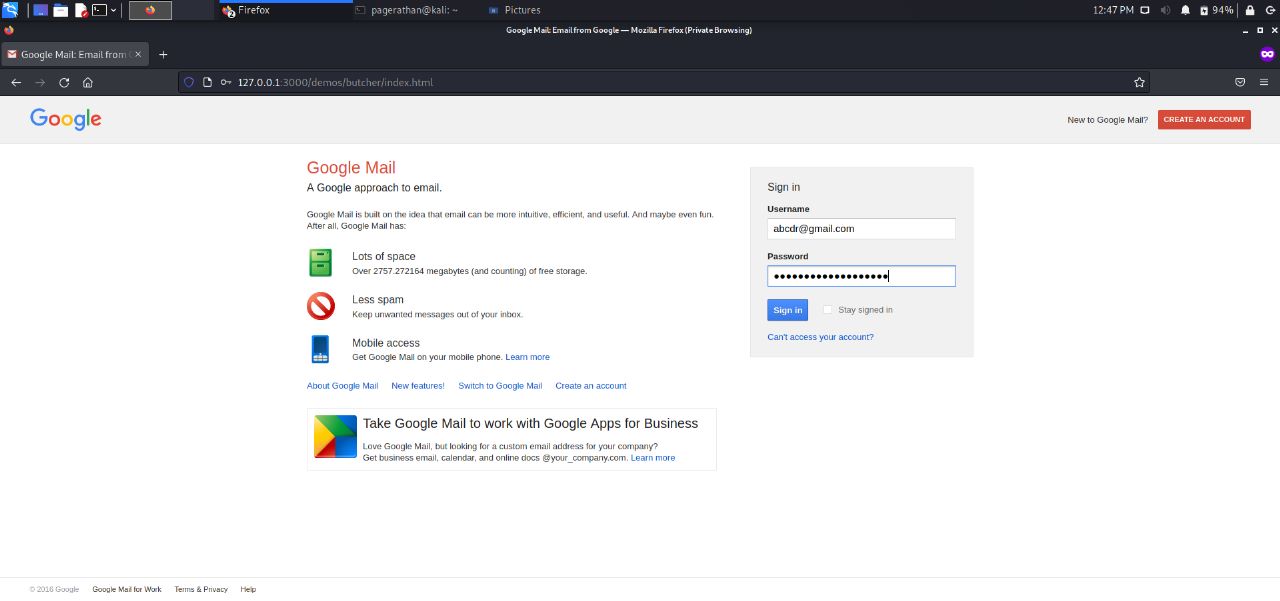
2. Click "Execute" under its description.

3. After executing it, a fake Gmail login page will appear in the hooked browser. The user may not think twice about inserting their username and password, and once they do, we log it.

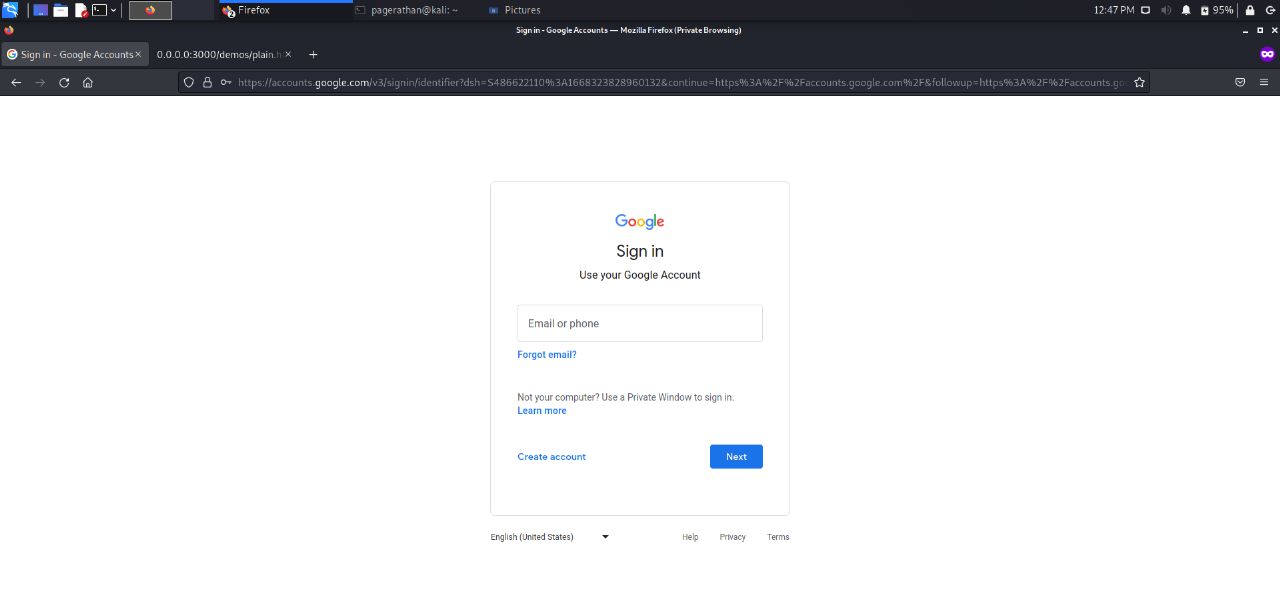
4. Afterward, they are directed back to Google's site as if they logged in regularly.

5. To find the username and password we logged, just click on the command in the Module Results History column.

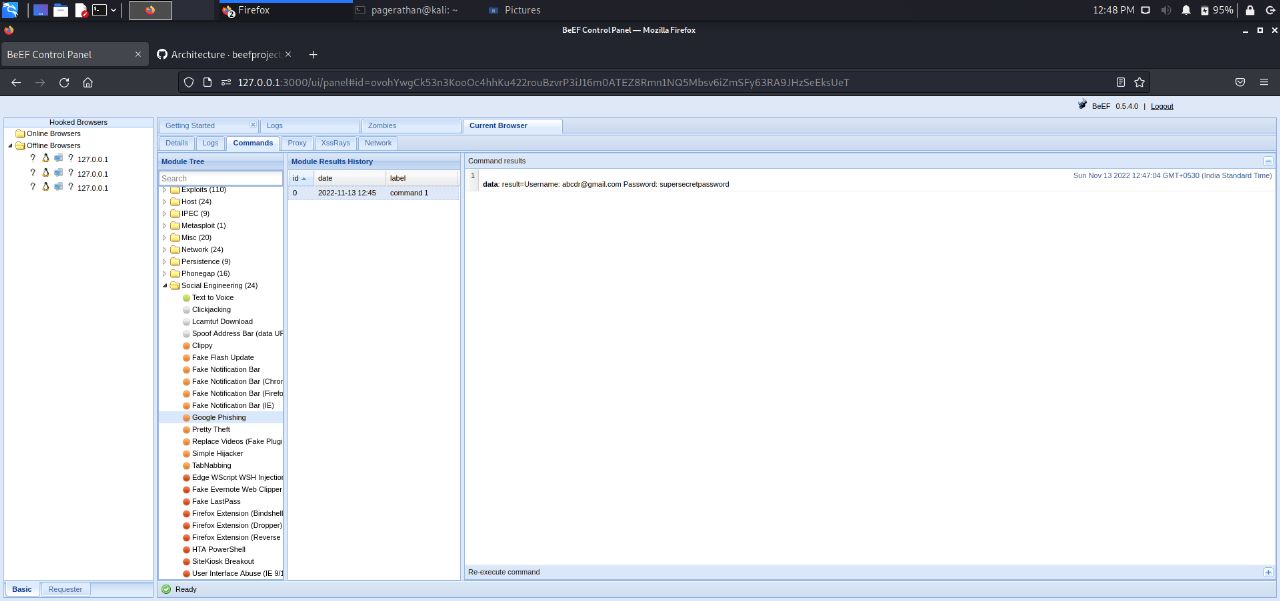


DIAG 1.1: GOOGLE PHISING INPUTS

DIAG 1.2: FAKE GOOGLE LOGIN PAGE



DIAG 1.3: REDIRECTED TO ACTUAL GOOGLE LOGIN



DIAG 1.4: STOLEN CREDENTIALS

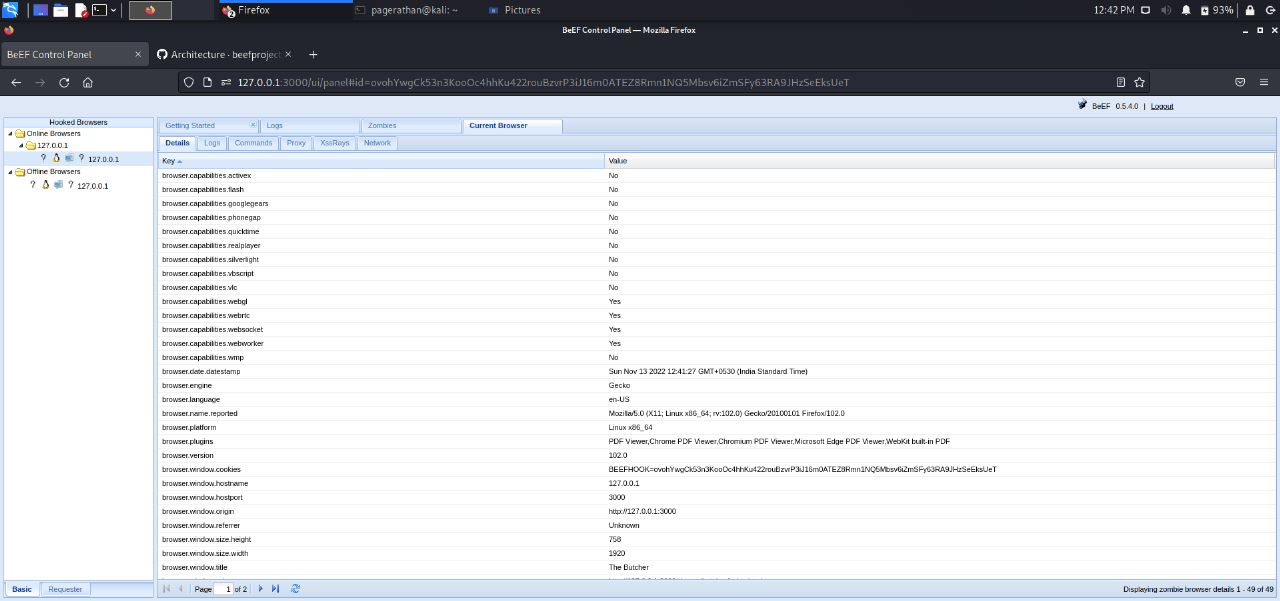
**2. HOOKING A BROWSER**

A BeEF hook is a JavaScript file, used to latch on to a target’s browser to exploit it while acting as a command-and-control between it and the attacker. This is what is meant by a “hook” in the context of using BeEF. Once a web browser is hooked by BeEF, you can proceed to inject further payloads and begin with post-exploitation.

STEPS:

1. To target a web browser, you will first need to identify a webpage that the victim to-be likes to visit often, and then attach a BeEF hook to it.
2. Deliver a javascript payload, preferably by including the javascript hook into the web page’s header. The target browser will become hooked once they visit this site.

The screenshot below is what the attacker will know once the target clicks on the link.

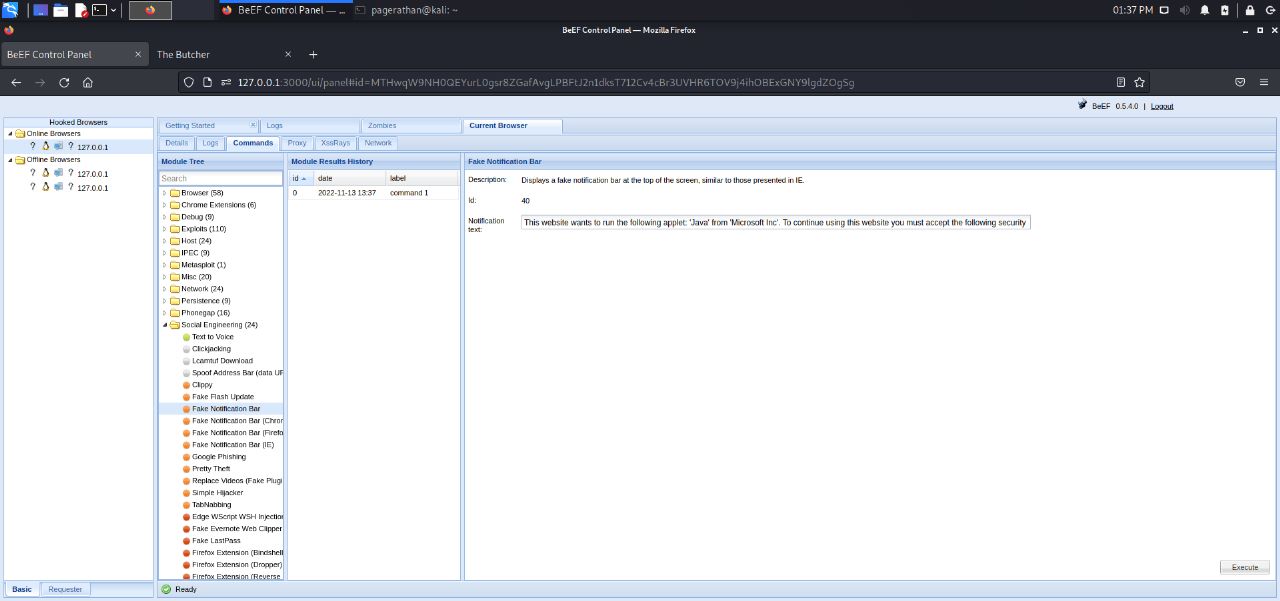


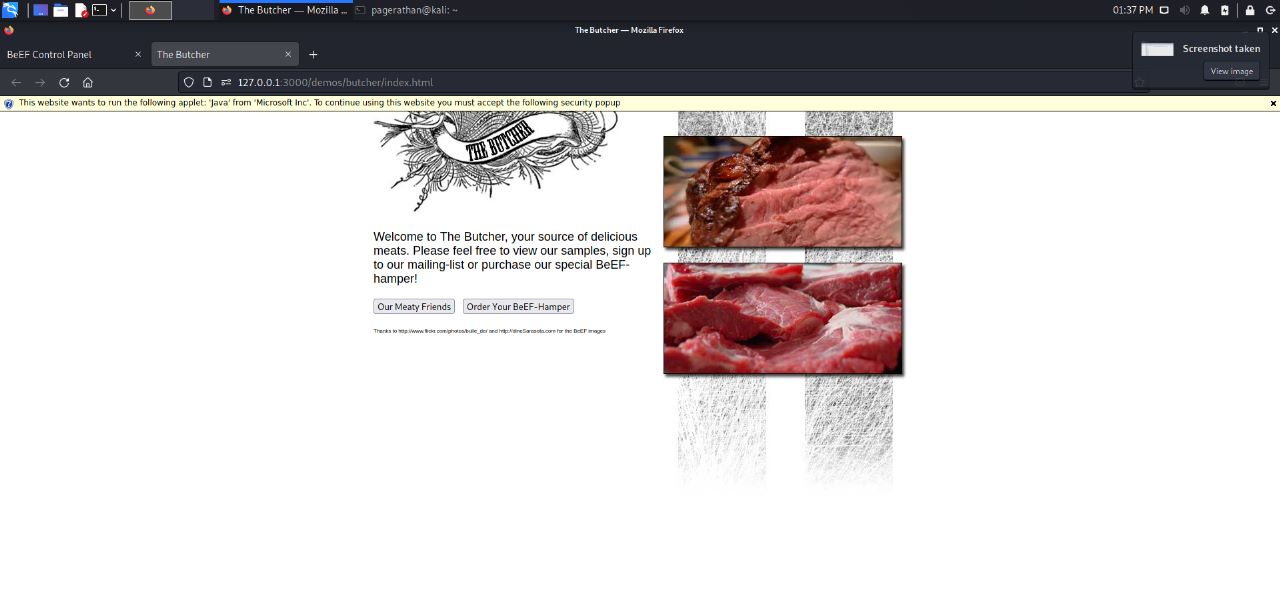
DIAG 2.1: VARIOUS ASPECTS OF TARGET BROWSER

**3. FAKE NOTIFICATION BAR**

Get user to download and run malicious file through a notification. Displays a fake notification bar at the top of the screen, similar to those presented in IE. The file will need to be hosted on the path specified on the file system. Once clicked on, the attacker will get all aspects of information on the target

STEPS:

1. When you find a module named “Fake notification bar " module in the "Social Engineering" folder.
2. Enter the desired text in the notification text box
3. Click "Execute" under its description.
4. Afterwards the target is displayed a website with the desired notification bar.

DIAG 3.1: INPUT SCREEN BEFORE STARTING THE ATTACK

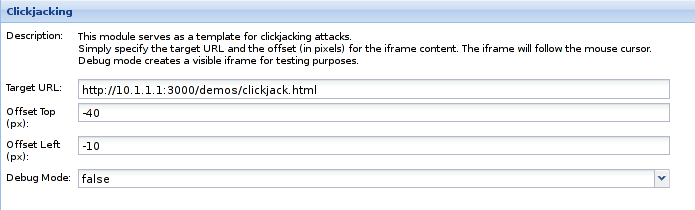
DIAG 3.2: MALICIOUS FAKE NOTIFICATION BAR BEING DISPLAYED ON A WEBSITE

**4. CLICK JACKING**

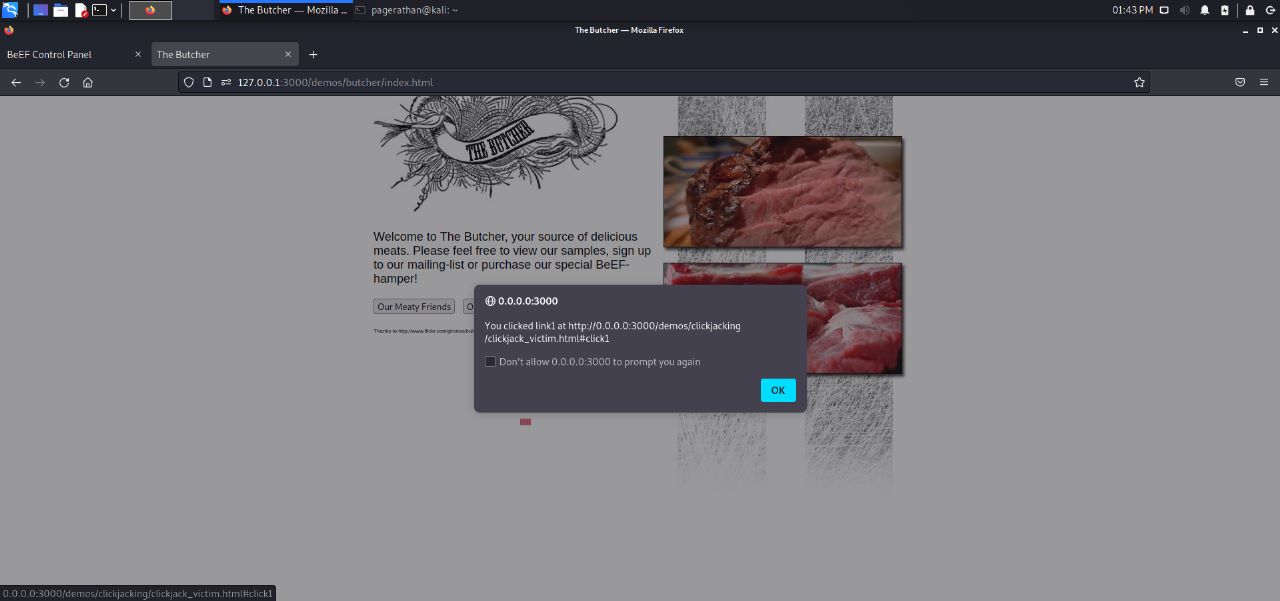
This performs basic multi-click clickjacking. The iframe follows the mouse, so anywhere the user clicks on the page will be over x-pos,y-pos. The optional JS configuration values specify local Javascript to execute when a user clicks, allowing the page can give visual feedback. The attack stops when y-pos is set to a non-numeric values (e.g. a dash).

STEPS:

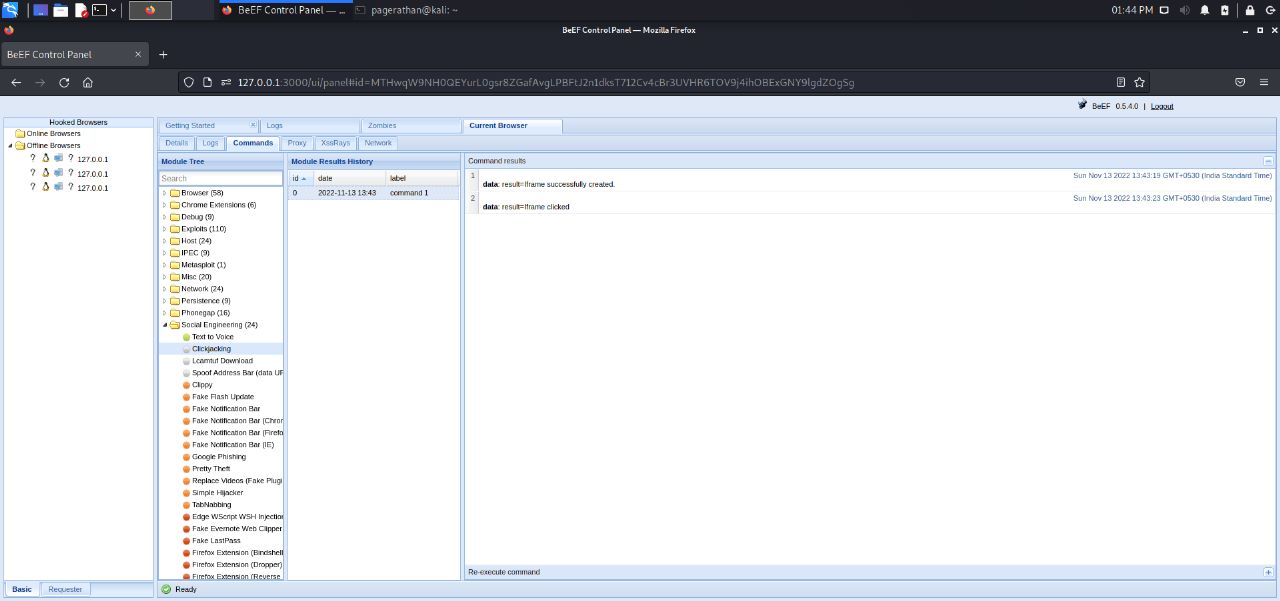
1. When you find a module named “Clickjacking " module in the "Social Engineering" folder.
2. Enter the target URL and choose the appropriate values for the offsets.
3. Click "Execute" under its description.
4. When you click anywhere on the target website, the command will be executed and a message will pop up. This will give access to control the mouse movement.



DIAG 4.1: VARIOUS INPUTS FOR CLICKJACKING



DIAG 4.2: THE POP UP

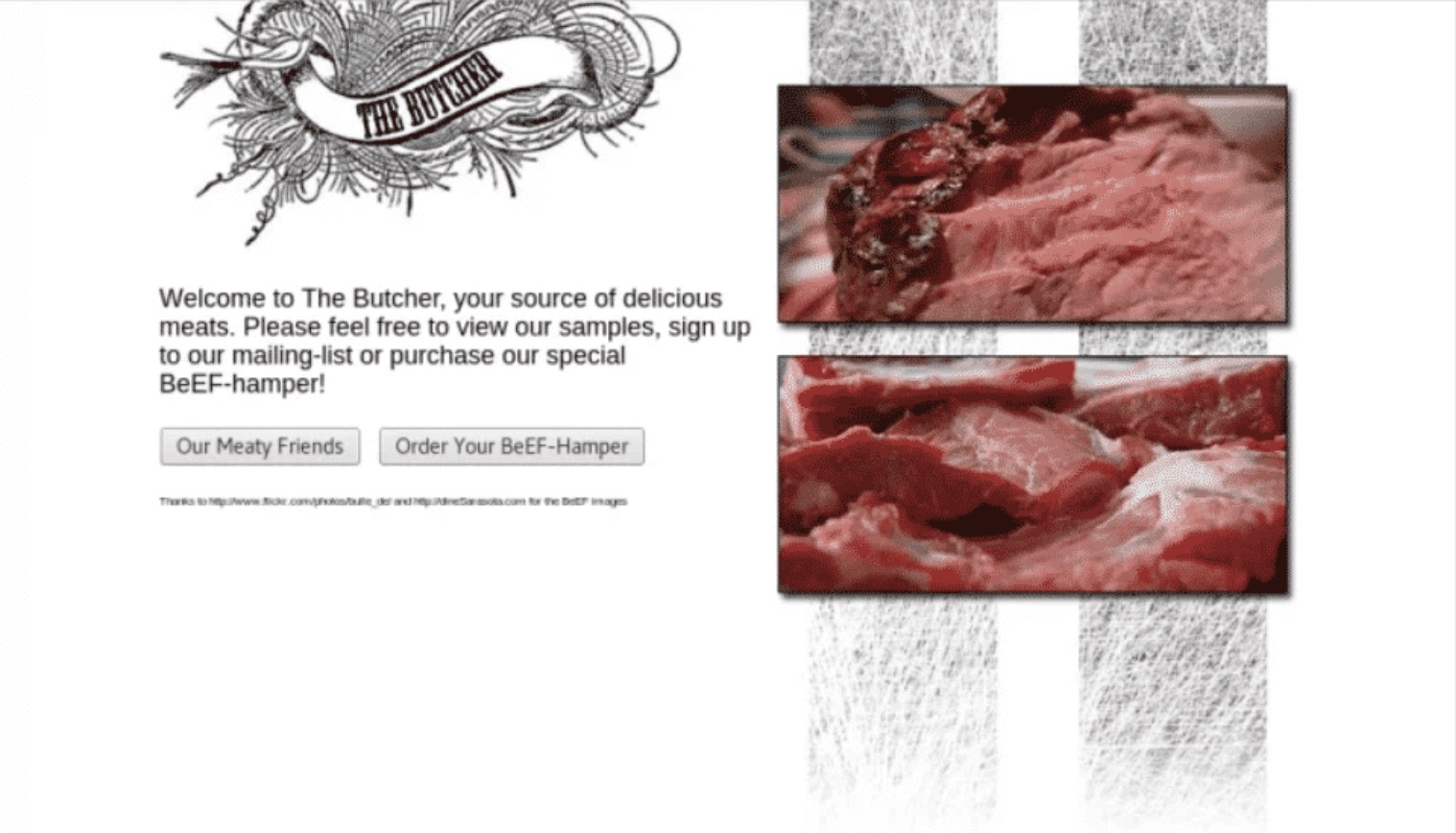


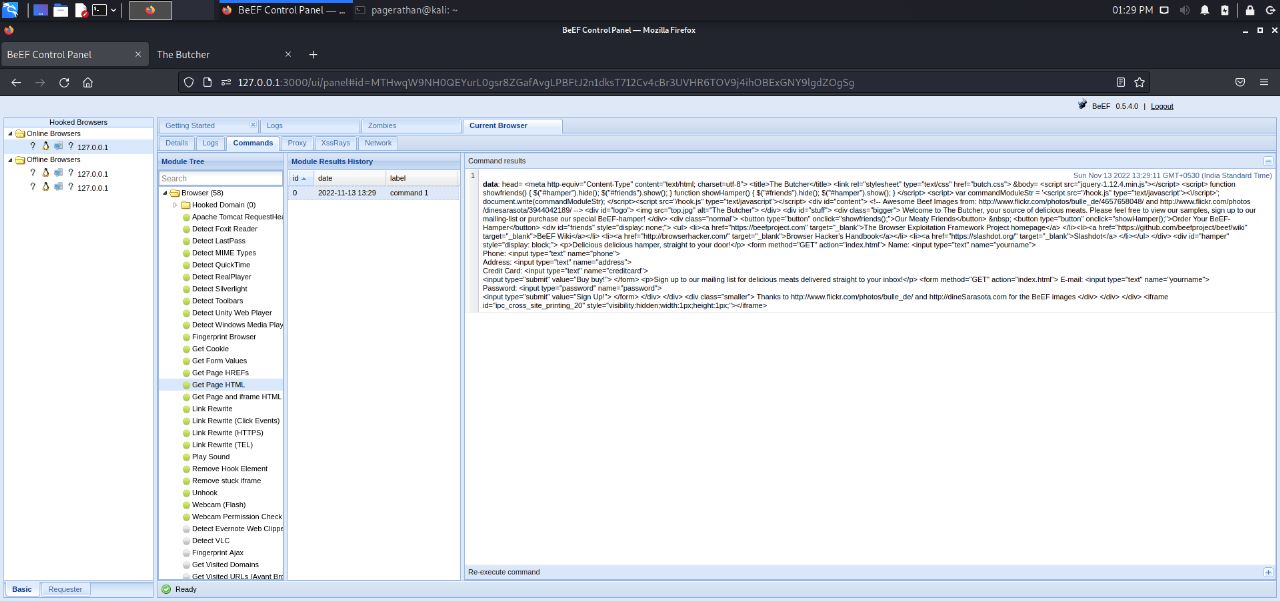
DIAG 4.3: THE OUTPUT SCREEN WHICH TELLS WE’VE SUCCESSFULLY COMPLETED

**5. GET PAGE HTML**

This command will retrieve the HTML from the current page. This module returns the HTML of the header and body of the hooked web page

STEPS:

1. When you find a module named “Get page HTML " module in the “Browser" folder.
2. Click "Execute" under its description.
3. Afterwards, it’ll display the target’s current website in html format.

DIAG 4.1: THE WEBSITE THE TARGET IS BROWSING

DIAG 4.2: THE TARGET’S WEBSITE IN HTML FORMAT AVAILABLE TO THE ATTACKER

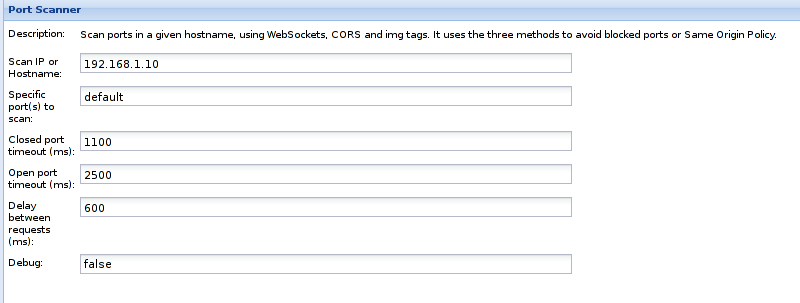
**6. PORT SCANNER**

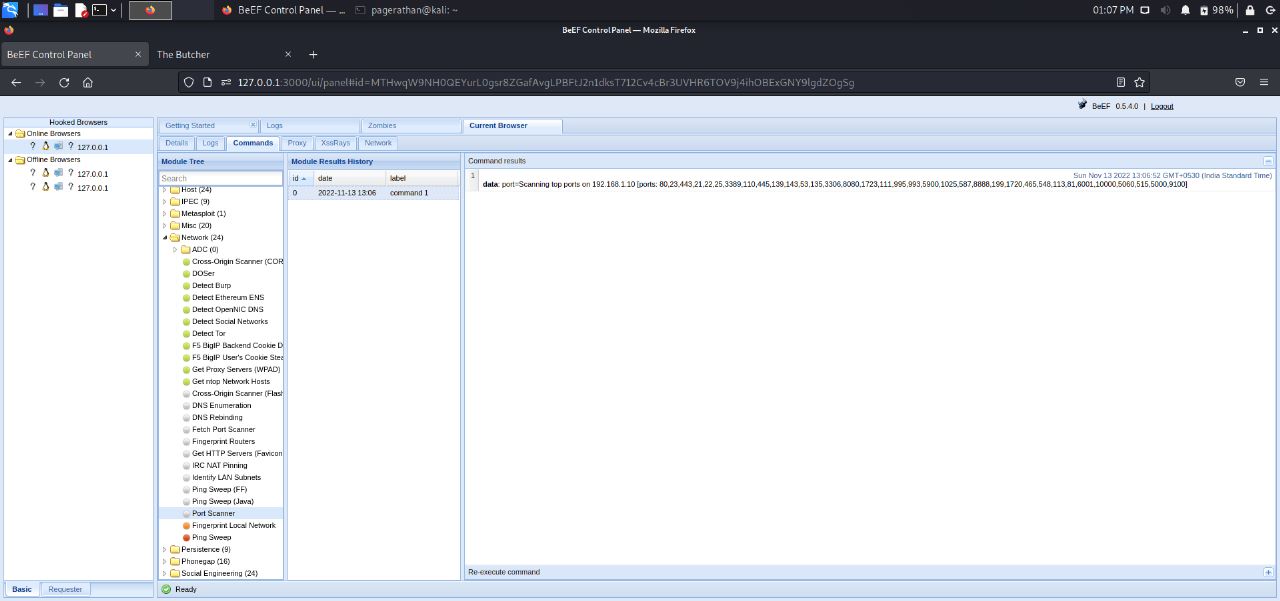
Now that we know the IP address of the hooked system and several hostnames, it would be interesting to launch port scanning. Several security researchers have found that it is possible to use the same timing hack to scan ports by loading images into the browserScan ports in a given hostname. It uses the three methods to avoid blocked ports.

Parameters :

* IP or Hostname : IP or hostname of the target to be scanned (only one per command)
* Specific ports to scan : list of ports to be scanned
* Closed port timeout : Time-out to detect closed port (in ms, default is 1100)
* Open port time-out : Time-out to detect open port (in ms, default is 2500ms)
* Delay between requests : in ms, default is 600ms
* Debug : Debug mode set to false by default

STEPS:

1. When you find a module named “Port scanner " module in the “Network" folder.
2. Enter desired values for the above mentioned parameters
3. Click "Execute" under its description.
4. After executing, the ports which are open and are being used will be displayed.

DIAG 6.1: VARIOUS INPUTS FIELDS FOR PORT SCANNER

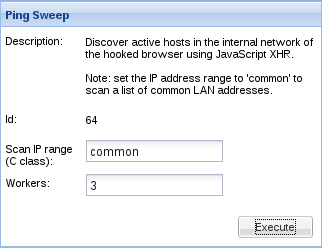
DIAG 6.2: THE OUTPUT SCREEN SHOWING THE PORTS WHICH ARE OPEN

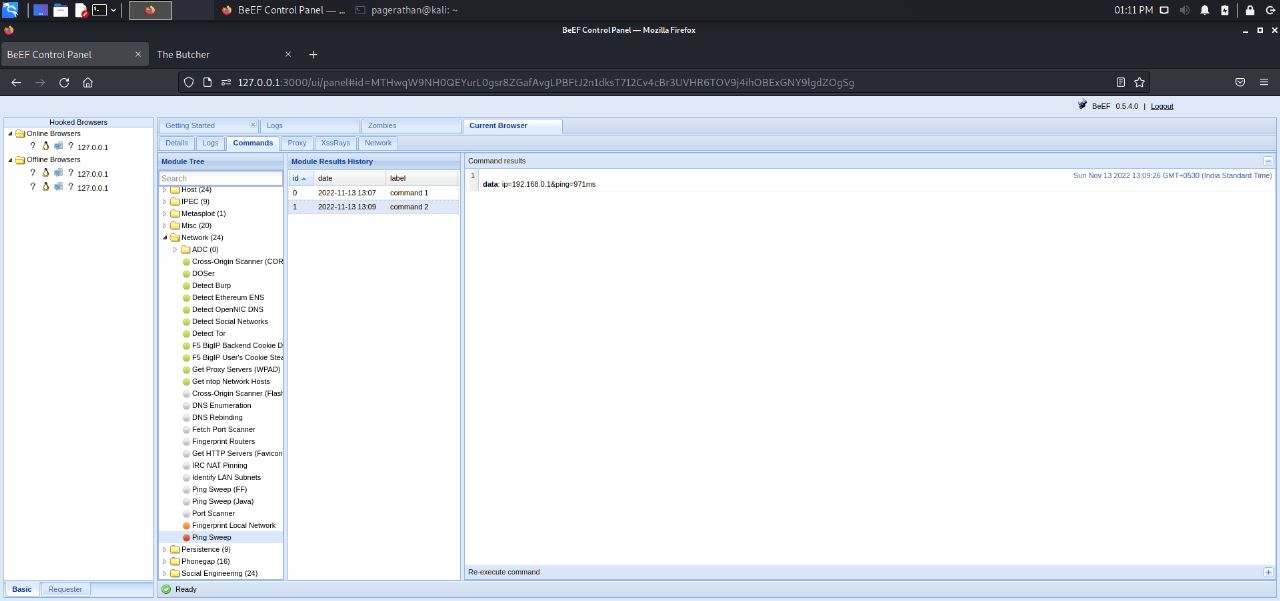
**7.PING SWEEP**

It performs a ping sweep of the network via Java.It discover active hosts in the internal network the hooked browser using JavaScript XHR. The Browser needs to support CORS (Cross Origin

Resource Sharing). Set the IP address range to 'common' to scan a list of common LAN addresses. The default number of workers (3) should be sufficient. Increasing the number of workers is likely to result in false negatives due to hitting the browser's maximum connection cap.

STEPS:

1. When you find a module named “Ping Sweep" module in the “Network" folder.
2. Enter desired IP range to be identified.
3. Click "Execute" under its description.

DIAG 7.1: THE INPUT FIELDS FOR PING SWEEP

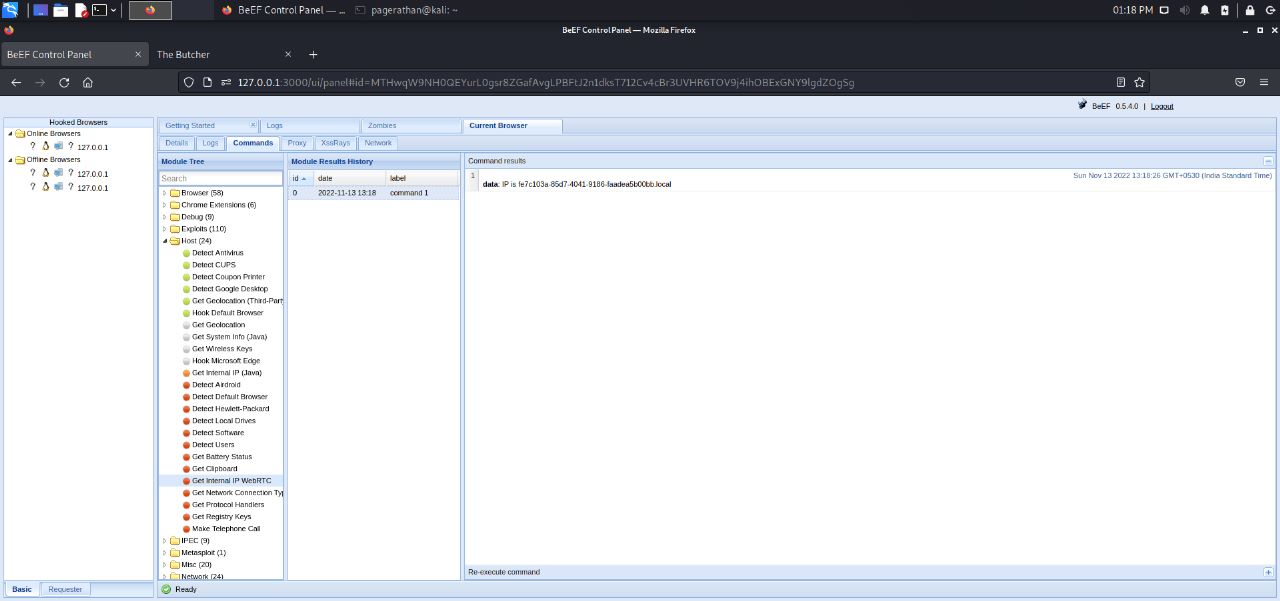
DIAG 7.2: THE OUTPUT SCREEN DISPLAYING THE PING

**8. INTERNAL WEB RTC**

It gets the internal IP address using a nice WebRTC hack. Based on WebRTC behaviour implemented in the target browser, it was supposed to prevent the internal IP addresses from being leaked to a web page without calling getUserMedia() and causing a user prompt.

STEPS:

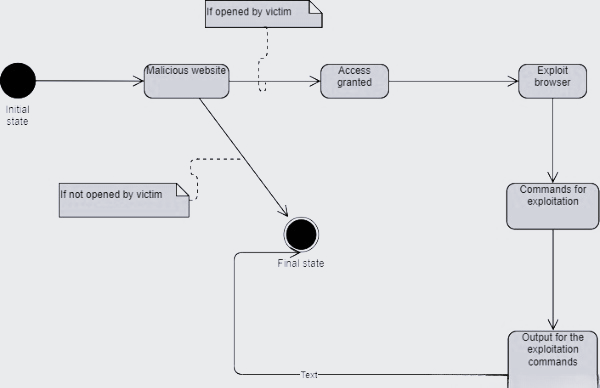
1. When you find a module named “Get Internal IP WebRTC" module in the “Host" folder.
2. Click "Execute" under its description.
3. Afterwards, it’ll show the ip address of the target system.



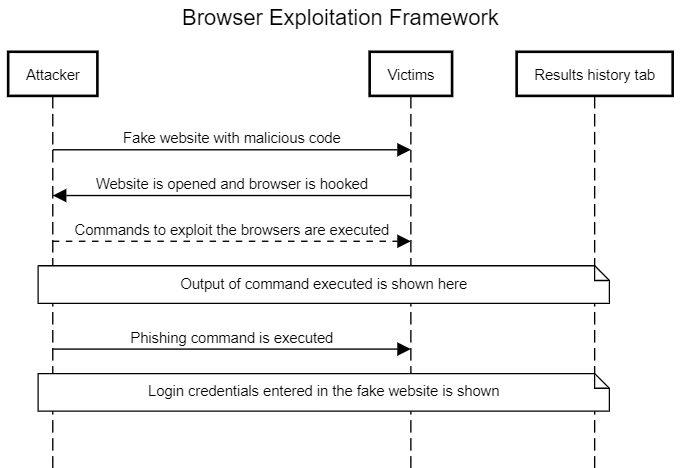
DIAG 8.1: THE OUTPUT SCREEN DISPLAYING THE IP ADDRESS

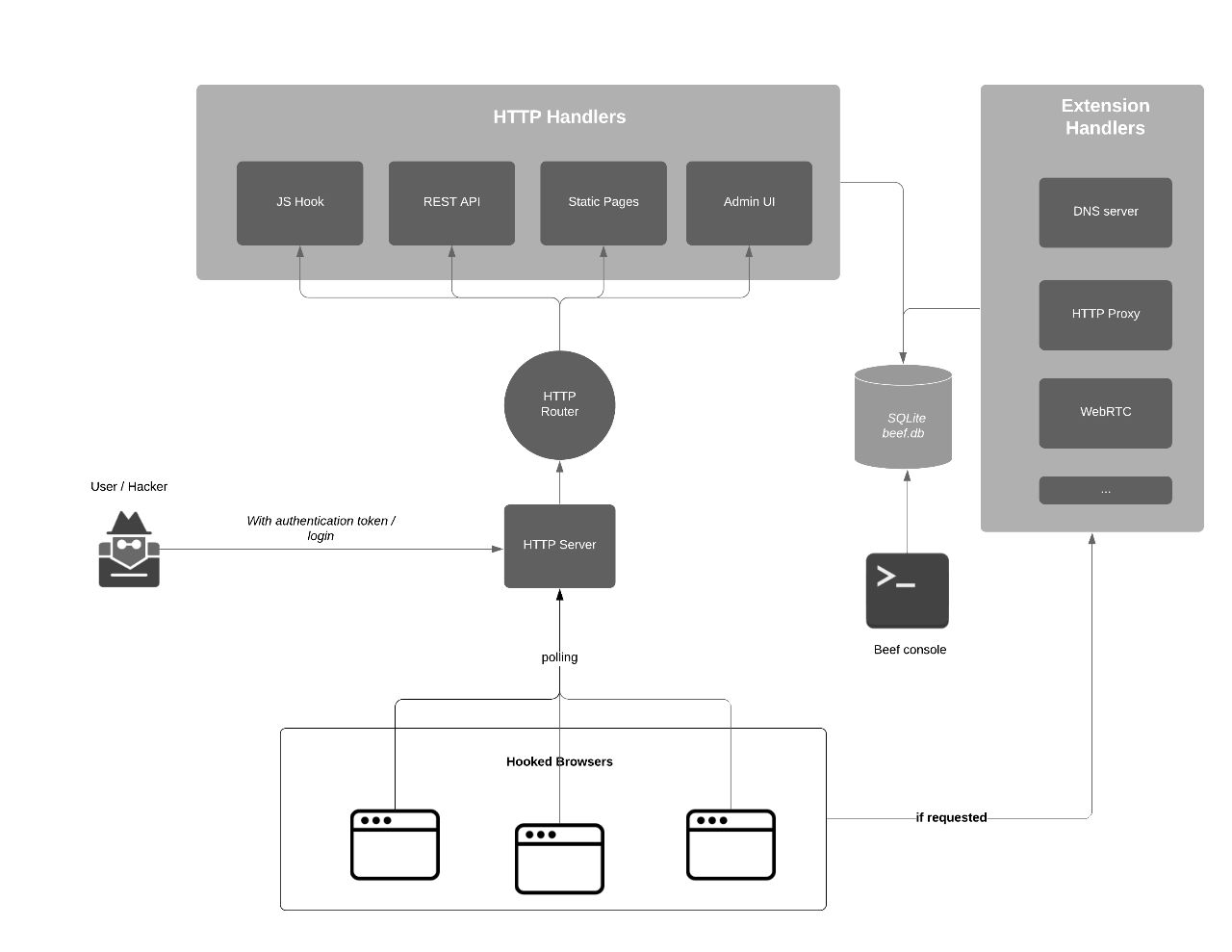
**DIAGRAMS:**

*STATE CHART DIAGRAM*



*SEQUENCE DIAGRAM*



*ARCHITECTURE DIAGRAM*

References

1. <https://github.com/beefproject/beef/>
2. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjf2d6Jyun7AhWm4TgGHfEZCE0QFnoECBQQAQ&url=https%3A%2F%2Fbeefproject.com%2F&usg=AOvVaw2cOveWm_neOz2YMtyp3ghS>
3. <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjf2d6Jyun7AhWm4TgGHfEZCE0QFnoECDYQAQ&url=https%3A%2F%2Flinuxhint.com%2Fhacking_beef%2F&usg=AOvVaw2PST738V72Jxs1i-W0MTmp>