Exercise-2: Cross-Site Scripting

1. Reflected XSS attack when a user is not logged in:

    1. An attack that fakes the login form to attack users:

    We are using Altoro Mutual website which is a purposely vulnerable website application used for testing. Cross-site scripting is a basic vulnerability of a web application that attackers use to exploit the victims.

    We first tried to test whether the website is susceptible to HTML injection using script tags. HTML injection is a weakness where the website returns the user input back onto the web page. By injecting the script tags the attacker can create a space in which they can execute their customized codes and programmatically control the website at the client end.

    Once we identified that the website is prone to HTML injection , we are using a tool called BeEF in Kali Linux system which is also abbreviated as – Browser exploitation framework, a penetration testing tool that is used to exploit vulnerabilities in a web browser.

    In order to hook the website, we need Hook script, Hook script is obtained during the initialization of the BeEF tool. Once the script is obtained, we will have to note the IP address by using the command **ifconfig** the below screenshot shows the hook script and the IP address obtained.

    - To check if website is vulnerable to XSS attacks, we run the script within the website's search bar.

    The below screenshot shows that the website is susceptible for XSS attacks :

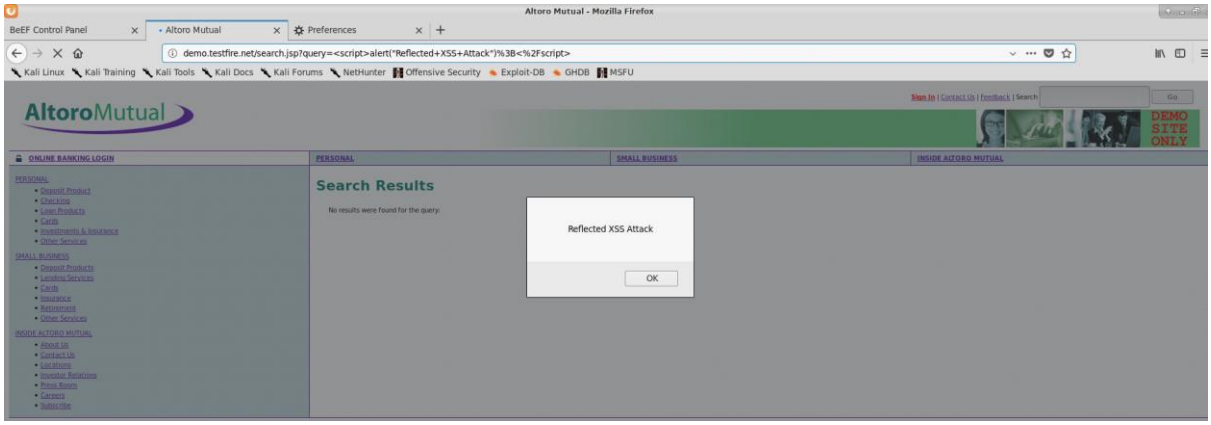

Fig1: Shows the website is vulnerable to XSS attack.

- **To obtain Hook script:**

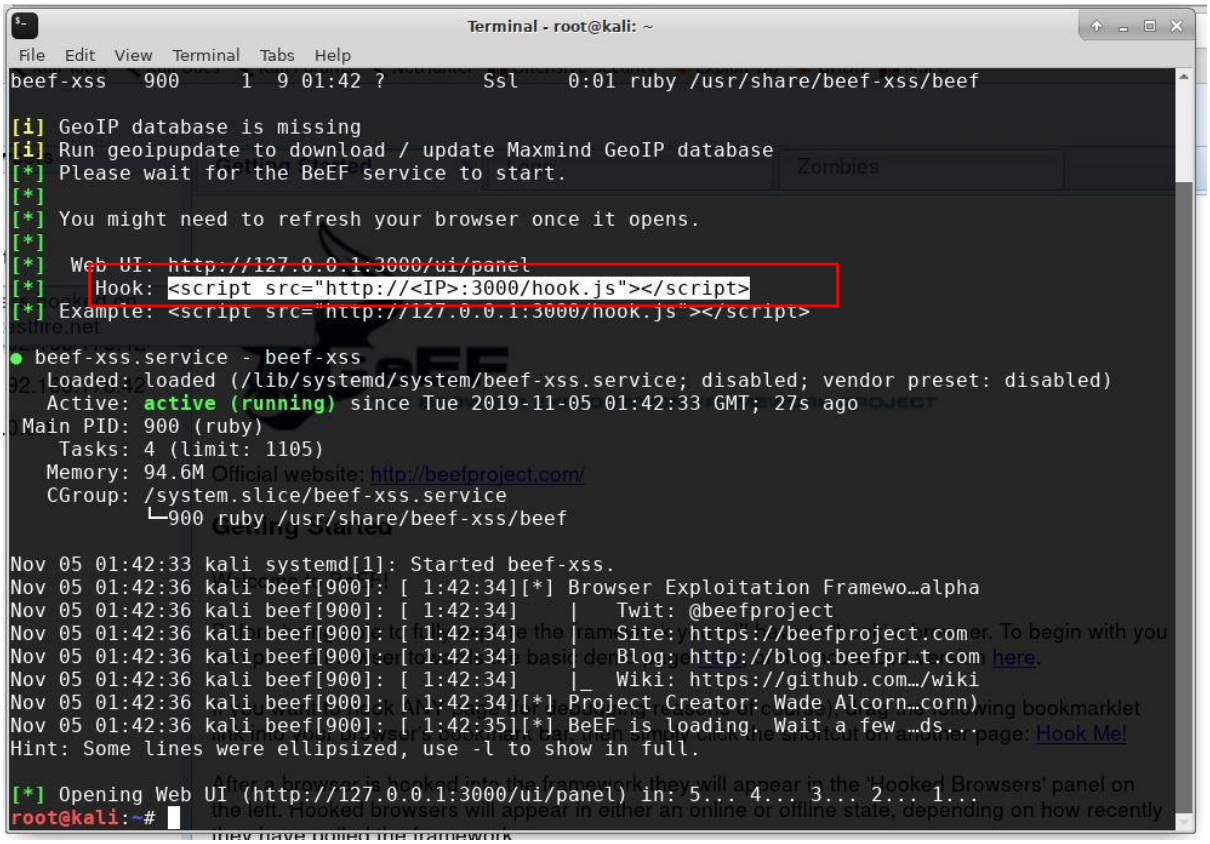

Fig2: Shows Hook script

- **To obtain IP address:**

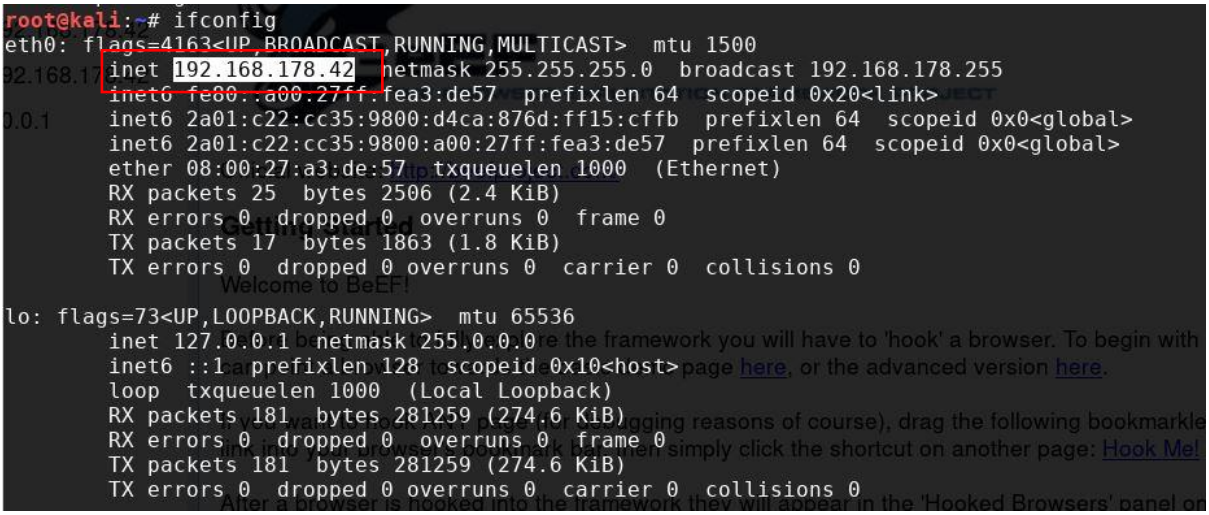

Fig 3: IP address

The hook script is executed by pasting it into the search box of Altoro mutual website to intercept the client. Then, we will refresh the BeEF tool, it displays BeEF control panel where we can execute and exploit the victim as it has picked up the client.

We can see the Online Browser IP address in the below screenshot within the BeEF control panel.

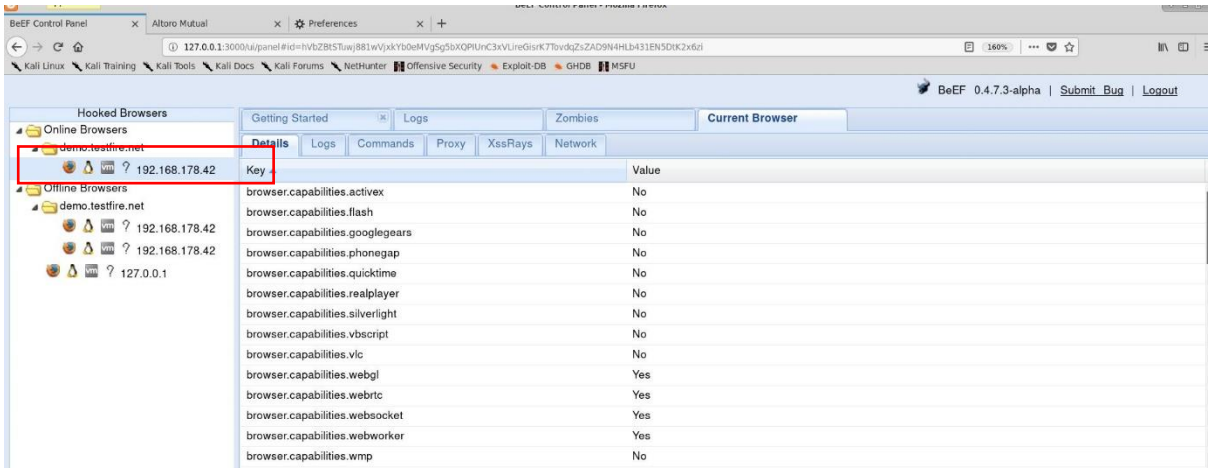

Fig 4: Showing the online browser

We then copy the URL of the login page from the client side into iFrame source within the iFrame event logger and click execute in the BeEF tool. This action refreshes and gives the login page to the user at the client's browser. Now, forcing the user to login to access his account. Once the victim types the username and password and logs in. We go to our attacker page i.e. BeEF tool where we notice that it has logged the activities of the Victim's browser. We can now see the keystrokes and the complete username and password of the victim and have gained access to victim's user account without his knowledge.

The below screenshot shows iFrame event logger and iFrame source in BeEF tool as stated above.

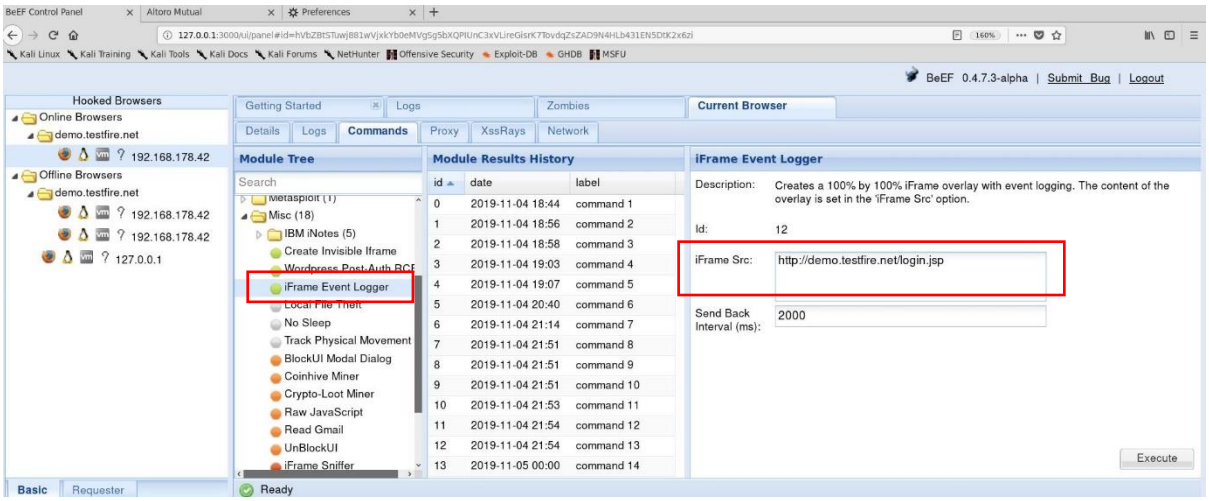

Fig 5: iFrame Event logger and iFrame source

Once the above screen is executed, It refreshes the login page of the victim's browser as discussed above and forces the user to login to access his account once the user logs in the BeEF tool captures all the key strokes as shown in the below screenshot and reveals the user's username and credentials to the attacker.

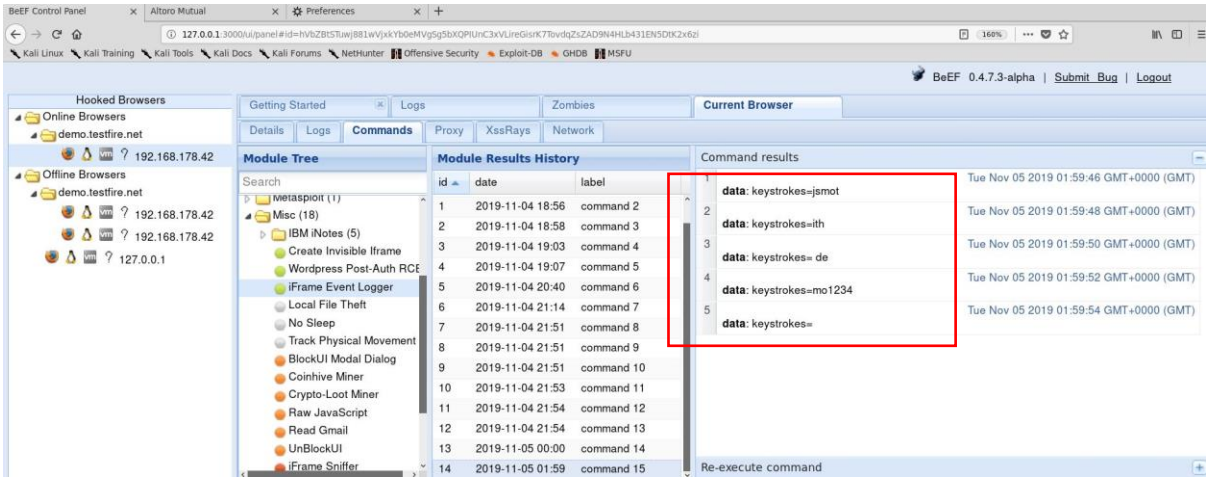

Fig 6: Keystrokes of the user details.