Web Browser Attack Using BeEF Framework

Samuel Agaga
Ontario Tech University
4 PUBLICATIONS **3** CITATIONS

Web Browser Attack Using BeEF Framework

Harshil Sawant, Samuel Agaga

***Abstract*— Web Browser is a tool, which connects us to the Internet. In this time of age, Internet has become a dependent factor for most of us. Therefore, it is very important for us to understand what is web browser, the architecture, and threats that come when using it. This paper illustrates the theoretical side of what is web browser, what are its components, how a browser can be a risk, and how to protect the user. Furthermore, the paper illustrates a lab that demonstrates how to exploit a web browser attack using BeEF.**

I. Introduction

WEB browser can be define in many ways. One common definition of web browser is that it is a software application that allows users to view and interact with the content available in many forms on a web page, such as text, image, music, video, games, etc. It is the most popular method for users to access the Internet. There are many examples of web browsers. The five most popular web browsers are Mozilla Firefox, Google Chrome, Internet Explorer, Safari, and Opera. Furthermore, add-ons are available as applications to extend the functionality of such browsers. Few examples of add-ons include Flash Player, Java, Adobe Reader, QuickTime Player, etc. Depending on how the developers designed the web page, specific add-ons are need to view specific content. [1]

II. Web Browser In-Depth

The main function of a web browser is to present the web resources a user requests. The browser requests the resources from the server and displays it within the browser window. The requested resource is usually an HTML document, but could be an image, PDF, or any other form of content. The user uses URL (Uniform Resource Identifier) to specify the location of the resource. Additionally, HTML and CSS specification defines the way a browser will interpret and display the HTML files. Such specification are maintained by the W3C (World Wide Web Consortium) organization. W3C is a standard organization for the web. In the past, many browsers followed a part of the specifications and developed their own extensions specific to the browser. This caused compatibility issues for web authors. Now most of the existing browsers follow the common specifications. [2]

III. Similarity Among Web Browsers

Today, a user can choose from many types of browsers. Each has few elements that are distinct from one another. There is, however, one commonality among browser that is their User Interface (UI) Elements. UI elements include address bar for inserting a URL, back/forward buttons, bookmarking options, refresh and stop buttons for refreshing or stopping the web page, and home button that takes the user to the home page. The HTML5 specification used today does not define UI elements, but include common elements, such as address bar, status bar, and tool bar. [2]

IV. Web Browser Components

There are seven main components of web browser. The components include user interface, browser engine, rendering engine, networking, UI backend, JavaScript interpreter, and data storage. [2]

1. User Interface: includes every part of the browser display, such as the address bar, back/forward button, bookmarking menu, etc., except the window where the user see the requested page.
2. Browser Engine: organizes actions between the UI and the rendering engine.
3. Rendering Engine: accountable for displaying requested content. When a user request HTML content, the rendering engine analyses HTML and CSS files, and displays the analyzed content on the screen.
4. Networking: includes network calls such as HTTP requests.
5. UI Backend: it is used for drawing basic widgets like combo boxes and windows. The backend exposes a generic interface that is not platform specific. Underneath it all uses operating system user interface approaches.
6. JavaScript Interpreter: it is used to analyze and execute JavaScript code.
7. Data Storage: It is a persistence layer. The browser needs this component to save data locally, such as cookies. Additionally, browser supports storage mechanisms such as localStorage, IndexedDB, WebSQL, and FileSystem.

The following image illustrates how each of the component interact within the system.

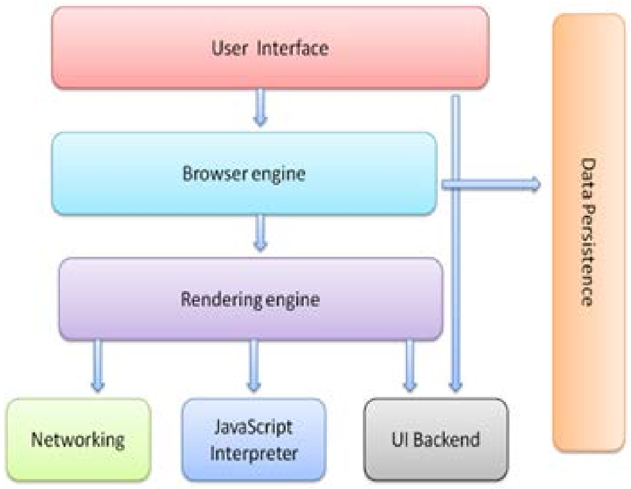


Figure 1: Web Browser Components [2]

V. WEB BROWSER RISK

According to the past studies, about 45% of people roaming the Internet are not utilizing the most secure version of their web browser. Similar to many software, without the proper security patches, web browsers are vulnerable to attack or exploit. Furthermore, even a fully patched web browser can be vulnerable to attack if the browser add-ons are not fully patched. Remember, when the user patches the browser, the add-ons are not automatically patched. [1]

Usually, browser-based attacks originated from malicious websites. However, poor security programming of web applications or vulnerabilities in the software supporting websites, let attackers to compromise trusted web sites to deliver malicious payloads to unsuspecting visitors. Hackers would add scripts that do not change a vulnerable website's appearance. These scripts can silently redirect the user to another website without him/her knowing about it. This redirect to another web site may cause malicious programs to be downloaded to your computer. Such programs are generally designed to allow remote control of the user's computer by the attacker and to capture personal information, such as credit card information, banking information, etc. [1]

VI. PROTECT USER FROM BROWSER RISK

The following are few of many practices a user must enforce to avoid unwanted browser risk. [1]

- Keep your browser(s) updated and patched.
- Keep your operating system updated and patched.
- Use anti-virus and antispyware software, and keep them updated.
- Keep your applications, such as multi-media programs used for viewing videos, updated and patched, particularly if they work with your browser.
- Make sure your computer's firewall is on.
- Block pop-up windows, some of which may be malicious and hide attacks. This may block malicious software from being downloaded to your computer.
- Tighten the security settings on your browsers. Check the settings in the security, privacy, and content sections in your browser. The minimum level should be medium.
- Consider disabling JavaScript, Java, and ActiveX controls.

It is important to note that number of these tips may limit the users from access few of the browser's content. For example, JavaScript is used to control web pages on the client side of the browser, server-side programs, and even mobile applications. If you need to use JavaScript, set your browser to prompt you before running scripts. Lower your security settings temporarily to have proper access, and then reset them. [1]

VII. WHAT IS BEEF?

BeEF is short for The Browser Exploitation Framework. It is a penetration testing tool that focuses on exploit of web browser vulnerabilities. BeEF is a browser-based exploit package that "hooks" one or more browsers as beachheads, so the attacker can launch directed command modules and further attacks against the system from within the browser context. A user can be hooked by opening a customized URL and continue to see typical web traffic, while an attacker has access to the user's session. BeEF evades network security appliances and host-based anti-virus applications by targeting the vulnerabilities found in common browsers. [4] BeEF also 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. [3]

VIII. LAB

The following experiment illustrates steps we followed to show how to execute a successful web browser attack using BeEF and how important it is to have an updated antivirus running for your computer to detect web browser attack using BeEF framework.

ATTACK

Set up victim VM and attacker VM. Make sure the network adapter for both VMs is set to Bridged Adapter.

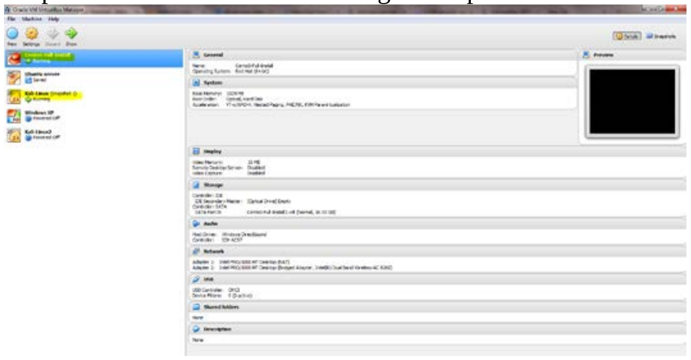


Figure 2: VMs [9]

The highlighted VM are used for the experiment. The exploit was carried out on the Kali Linux while the CentOS 7 was the victimized OS.

CentOS-Full-Install [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Desktop
7
CENTOS
Welcome to CentOS - Mozilla Firefox
05:46 PM

Figure 3: Victim's Machine CentOS 7 [8]

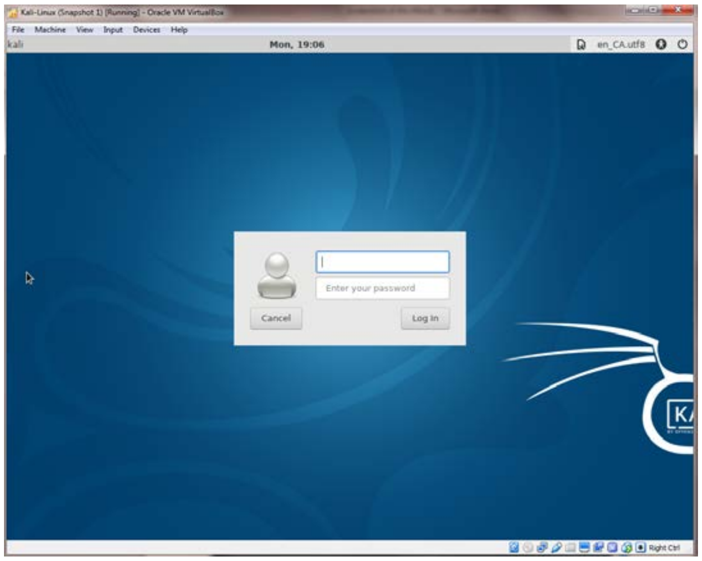
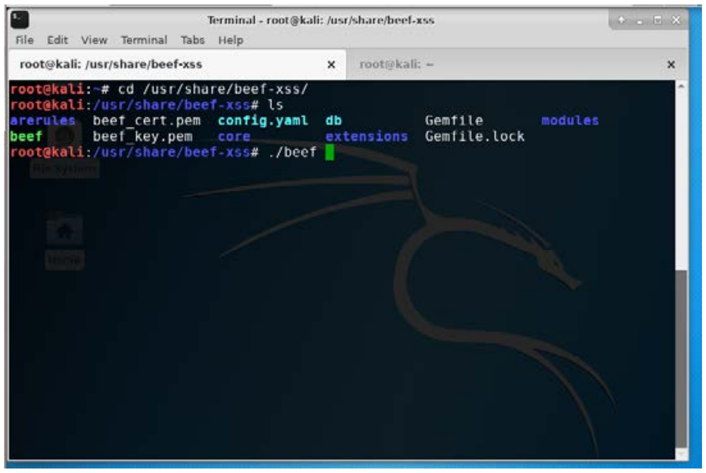

Kali-Linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mon, 19:06
en_CA.utf8
Enter your password
Cancel
Log In

Figure 4: Attacker's Machine Latest Kali Linux [7]

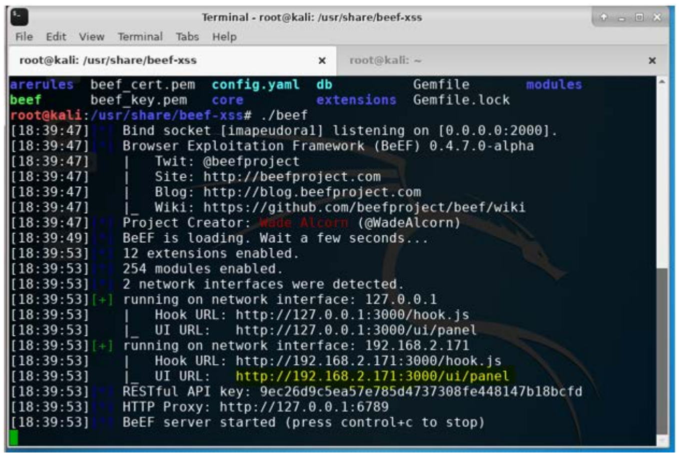
Above is the login page of Kali Linux. Note that we had created the login details during installation of the operating system.



```
root@kali:~# cd /usr/share/beef-xss/
root@kali:/usr/share/beef-xss# ls
arerules  beef_cert.pem  config.yaml  db          Gemfile       modules
beef      beef_key.pem   core         extensions  Gemfile.lock
root@kali:/usr/share/beef-xss# ./beef
```

Figure 5: Beef Directory

Use Terminal in Kali Linux to run the beef framework, the attacker will have to login into the Kali Linux and then navigate to the "beef-xss" directory and run the "beef" script as shown above.



```
arerules  beef_cert.pem  config.yaml  db          Gemfile       modules
beef      beef_key.pem   core         extensions  Gemfile.lock
root@kali:/usr/share/beef-xss# ./beef
[18:39:47][*] Bind socket [imapeudoral] listening on [0.0.0.0:2000].
[18:39:47][*] Browser Exploitation Framework (BeEF) 0.4.7.0-alpha
[18:39:47]    |   Twit: @beefproject
[18:39:47]    |   Site: http://beefproject.com
[18:39:47]    |   Blog: http://blog.beefproject.com
[18:39:47]    |_  Wiki: https://github.com/beefproject/beef/wiki
[18:39:47][*] Project Creator: Wade Alcorn (@WadeAlcorn)
[18:39:49][*] BeEF is loading. Wait a few seconds...
[18:39:53][*] 12 extensions enabled.
[18:39:53][*] 254 modules enabled.
[18:39:53][*] 2 network interfaces were detected.
[18:39:53][+] running on network interface: 127.0.0.1
[18:39:53]    |   Hook URL: http://127.0.0.1:3000/hook.js
[18:39:53]    |_  UI URL:   http://127.0.0.1:3000/ui/panel
[18:39:53][+] running on network interface: 192.168.2.171
[18:39:53]    |   Hook URL: http://192.168.2.171:3000/hook.js
[18:39:53]    |_  UI URL:   http://192.168.2.171:3000/ui/panel
[18:39:53][*] RESTful API key: 9ec26d9c5ea57e785d4737308fe448147b18bcfd
[18:39:53][*] HTTP Proxy: http://127.0.0.1:6789
[18:39:53][*] BeEF server started (press control+c to stop)
```

Figure 6: Running Beef

As can be seen from the above screenshot, beef has been successfully launched. Use the highlighted URL to open BeEF login page in the attacker browser.

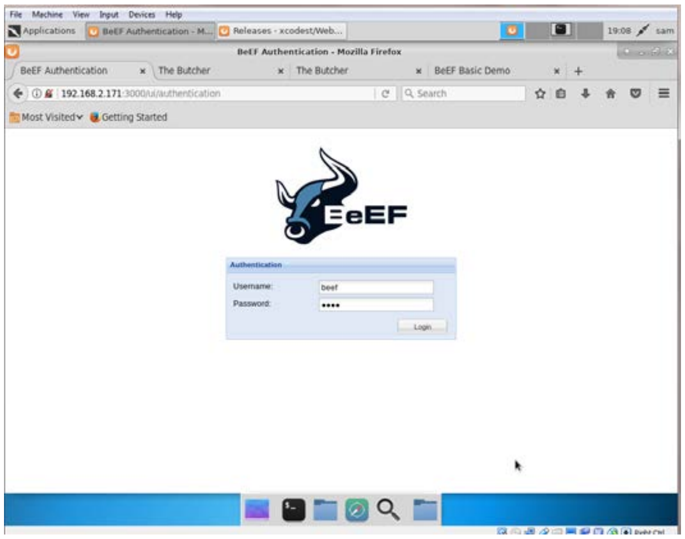
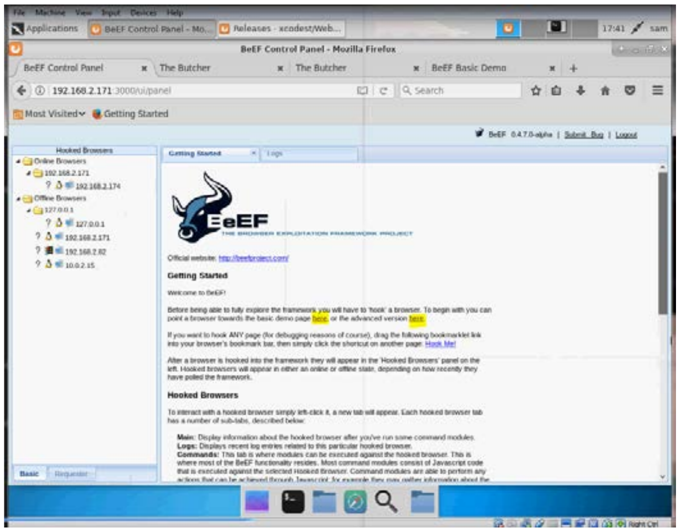

BeEF Authentication - Mozilla Firefox
192.168.2.171:3000/ui/authentication
Authentication
Username: beef
Password: ••••
Login

Figure 7: BeEF Login Page

Once BeEF has been launched, the next thing will be to login into the UI as seen above. The username and password is “beef”.



Official website: http://beefproject.com/

Getting Started

Welcome to BeEF!

Before being able to fully explore the framework you will have to 'hook' a browser. To begin with you can point a browser towards the basic demo page here, or the advanced version here.

If you want to hook ANY page (for debugging reasons of course), drag the following bookmarklet link into your browser's bookmark bar, then simply click the shortcut on another page: Hook Me!

After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

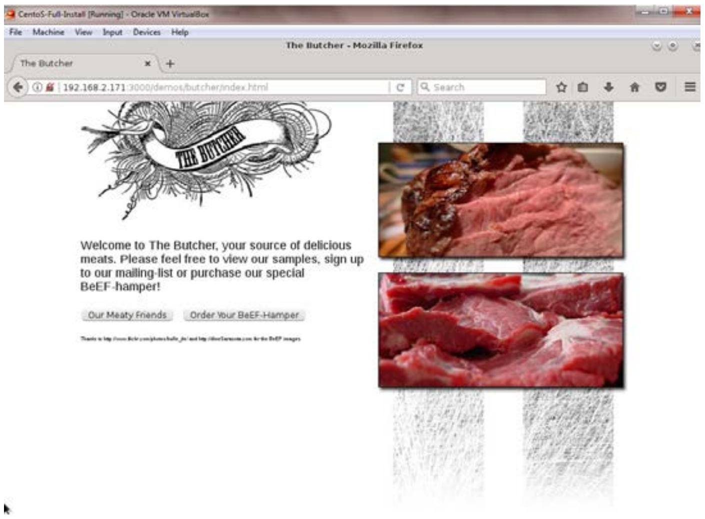
Hooked Browsers

To interact with a hooked browser simply left-click it, a new tab will appear. Each hooked browser tab has a number of sub-tabs, described below:

Main: Display information about the hooked browser after you've run some command modules.
Logs: Displays recent log entries related to this particular hooked browser.
Commands: This tab is where modules can be executed against the hooked browser. This is where most of the BeEF functionality resides. Most command modules consist of Javascript code that is executed against the selected Hooked Browser. Command modules are able to perform any actions that can be achieved through Javascript: for example they may gather information about the

Figure 8: BeEF Home Page

The above image is the BeEF home page with two highlighted links. Any one of the link can be used to hook the victim’s browser. Make sure to replace the IP address of the link from 127.0.0.1 to your attacker’s IP address (in this case 192.168.2.171). You can find you attacker’s IP address from the terminal using “ifconfig” command. Just for curiosity, the following image is one of link called “advanced version” of the html page when opened in the victim’s VM.


The Butcher - Mozilla Firefox
192.168.2.171:3000/demos/butcher/index.html
THE BUTCHER
Our Meaty Friends
Order Your BeEF-Hamper

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!

Figure 9: Malicious Link Page

At this point, from the attacker side, you can mask the malicious link using tools such as bitly.com, before baiting your victim to click on your malicious link using social engineering.

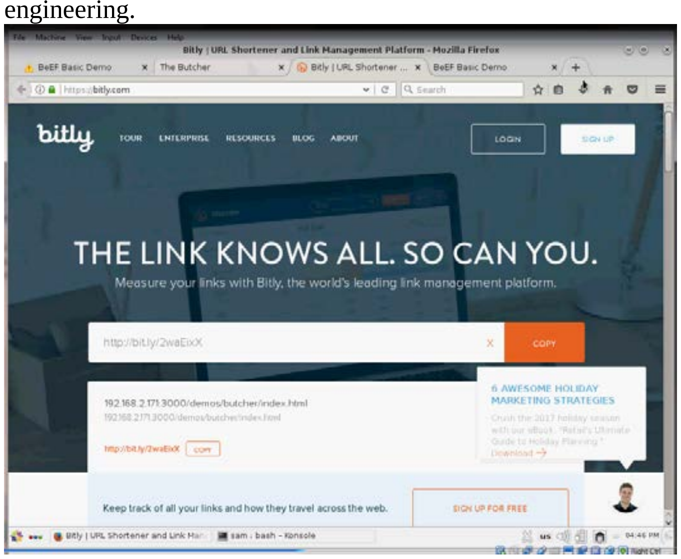
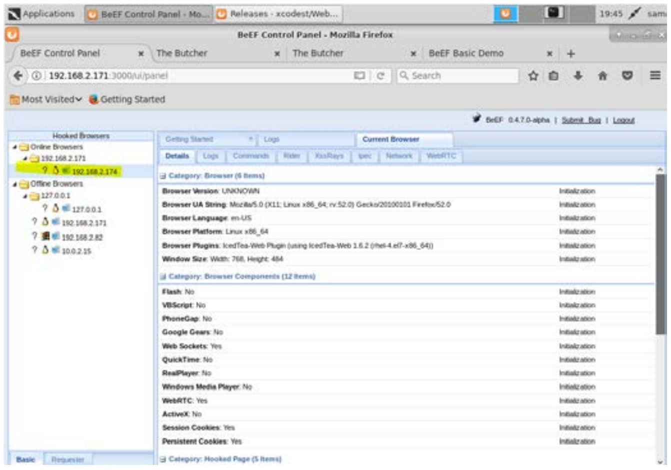

Bitly | URL Shortener and Link Management Platform - Mozilla Firefox
https://bitly.com
bitly
TOUR
ENTERPRISE
RESOURCES
BLOG
ABOUT
LOGIN
SIGN UP
THE LINK KNOWS ALL. SO CAN YOU.
Measure your links with Bitly, the world's leading link management platform.
http://bit.ly/2waEixX
COPY
192.168.2.171:3000/demos/butcher/index.html
http://bit.ly/2waEixX
6 AWESOME HOLIDAY MARKETING STRATEGIES
Keep track of all your links and how they travel across the web.
SIGN UP FOR FREE

Figure 10: Bitly.com

Suppose you are successful on bating your victim to click on you link using email or other social engineering method. The victim’s system will be “hooked” as illustrated by the following image. For this experiment, open the advanced version link in the victim’s VM to show that social engineering is successful.



Category: Browser (6 Items)

Browser Version: UNKNOWN — Initialization
Browser UA String: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0 — Initialization
Browser Language: en-US — Initialization
Browser Platform: Linux x86_64 — Initialization
Browser Plugins: IcedTea-Web Plugin (using IcedTea-Web 1.6.2 (rhel-4.el7-x86_64)) — Initialization
Window Size: Width: 768, Height: 484 — Initialization

Category: Browser Components (12 Items)

Flash: No — Initialization
VBScript: No — Initialization
PhoneGap: No — Initialization
Google Gears: No — Initialization
Web Sockets: Yes — Initialization
QuickTime: No — Initialization
RealPlayer: No — Initialization
Windows Media Player: No — Initialization
WebRTC: Yes — Initialization
ActiveX: No — Initialization
Session Cookies: Yes — Initialization
Persistent Cookies: Yes — Initialization

Category: Hooked Page (5 Items)

Figure 11: Browser Hooked

As can be seen in the above screenshot, the browser running on the victim's machine with IP address 192.168.2.174 has been hooked. The above image is shown in the attacker's VM.

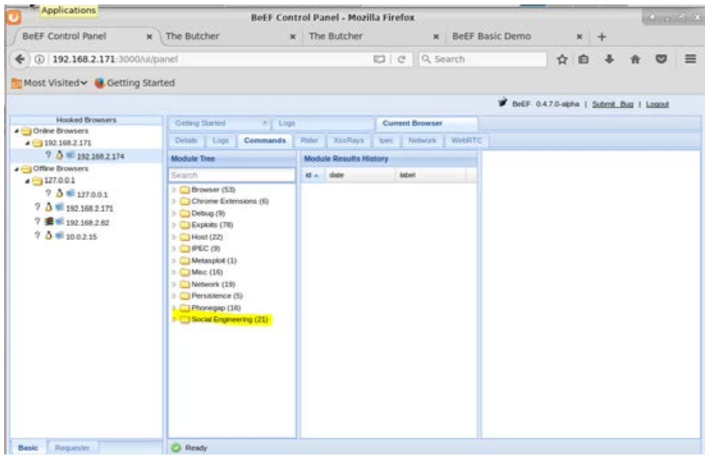
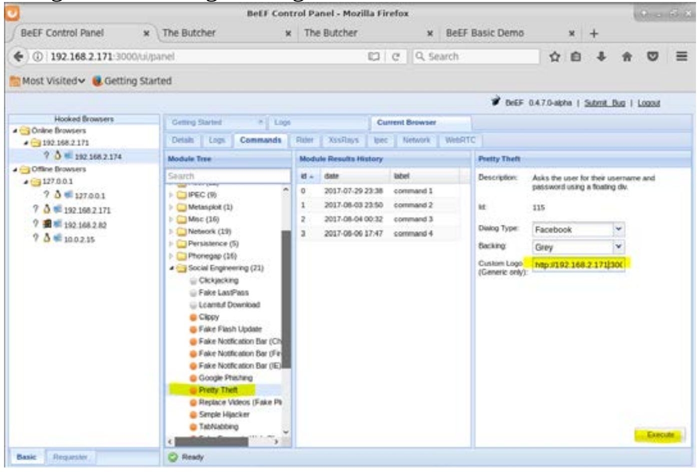


Figure 12: Social Engineering Attack

In this attack, we exploited the victim machine by means of social engineering as can been seen above. Under commands tab, go to social engineering to test the same attack we tested.



Module Results History

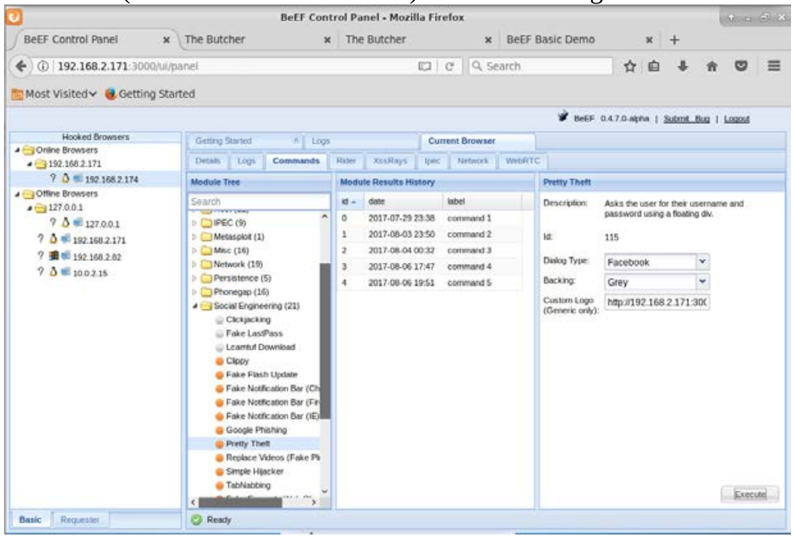
id	date	label
0	2017-07-29 23:38	command 1
1	2017-08-03 23:50	command 2
2	2017-08-04 00:32	command 3
3	2017-08-06 17:47	command 4

Pretty Theft

Description: Asks the user for their username and password using a floating div.
Id: 115
Dialog Type: Facebook
Backing: Grey
Custom Logo (Generic only): http://192.168.2.171:300

Figure 13: Selecting Attack

We are running the "Pretty Theft exploit". In Fig. 13 above, on the right is where we enter the information of the attacker's machine running the beef service. Make sure to replace the default IP address in the custom logo with the attacker's VM IP address (in this case 192.168.2.171) before executing.



Module Results History

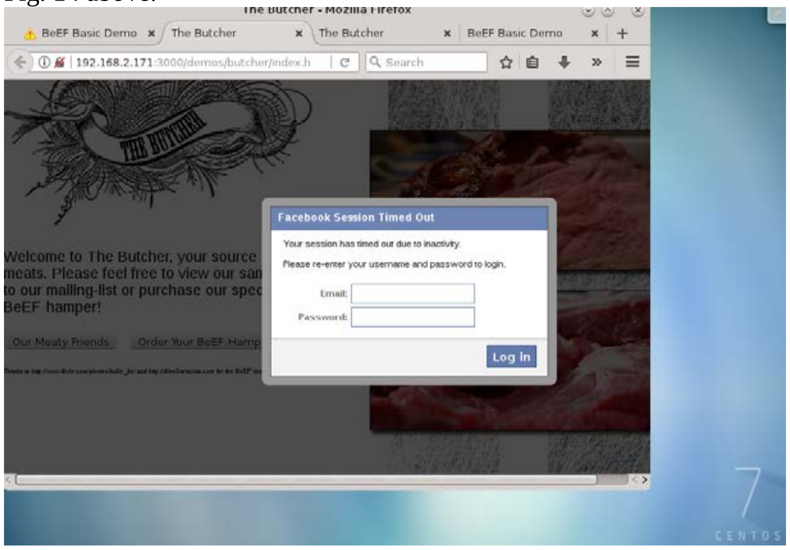
id	date	label
0	2017-07-29 23:38	command 1
1	2017-08-03 23:50	command 2
2	2017-08-04 00:32	command 3
3	2017-08-06 17:47	command 4
4	2017-08-06 19:51	command 5

Pretty Theft

Description: Asks the user for their username and password using a floating div.
Id: 115
Dialog Type: Facebook
Backing: Grey
Custom Logo (Generic only): http://192.168.2.171:30(

Figure 14: Launching the Attack

To run the attack, we simply just click on execute as show in Fig. 14 above.



Facebook Session Timed Out

Your session has timed out due to inactivity.

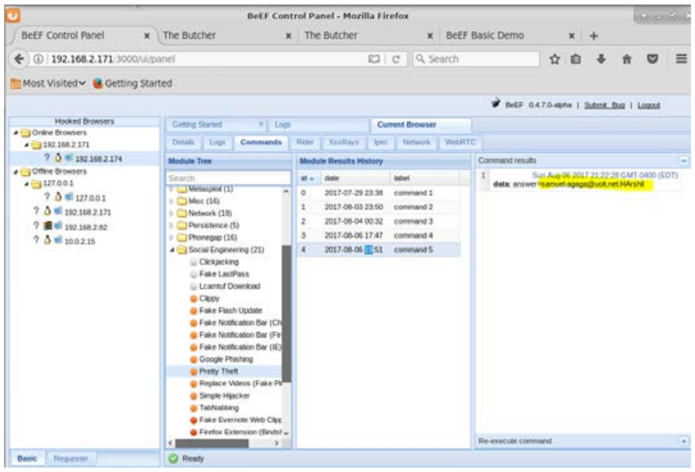
Please re-enter your username and password to login.

Email:
Password:

Log In

Figure 15: Fake Facebook Login

Once we have clicked execute, the Face Facebook authentication screen will be displaced on the victim's machine as seen in Fig. 15 above.



Module Results History

id	date	label
0	2017-07-29 23:38	command 1
1	2017-08-03 23:50	command 2
2	2017-08-04 00:32	command 3
3	2017-08-06 17:47	command 4
4	2017-08-06 [illegible]:51	command 5

Command results

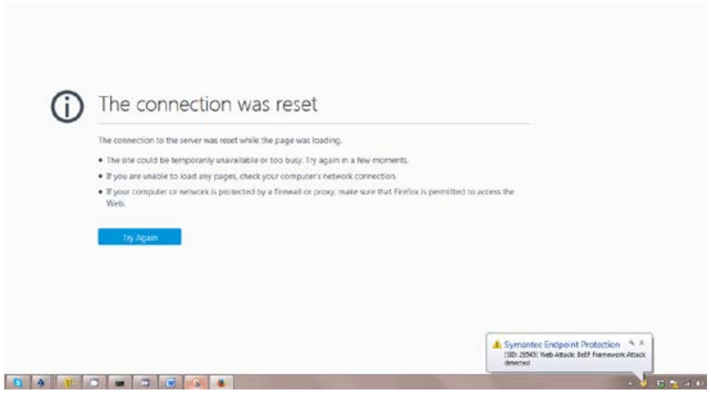
1 Sun Aug 06 2017 21:22:28 GMT-0400 (EDT)
data: answer=samuel.agaga@uoit.net:HArshil

Figure 16: Login Details Captured

Looking at the heighted portion on the right of the screenshot in Fig. 16 indicates our captured login details of the victim's username and password for Facebook.

Note. It would feel more authentic for the victim if the attacker execute his attack at the right time, for example, when the victim is on Facebook login page.

Defense



The connection was reset

The connection to the server was reset while the page was loading.

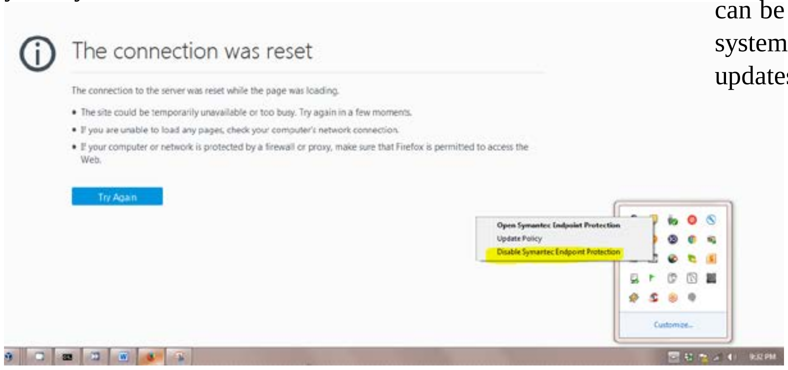
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

Symantec Endpoint Protection
[SID: 29543] Web Attack: BeEF Framework Attack detected

Figure 17: Defense by Symantec Antivirus

One of the ways to defend this attached is by having an up to date antivirus program running on your computer. In the screenshot above, when we tried this exploit against a machine running Symantec antivirus, we actually got a warning stating that there was BeEF framework attack as can be seen in the screenshot in Fig. 17. So test the same process, but turn on your Symantec antivirus.



The connection was reset

The connection to the server was reset while the page was loading.

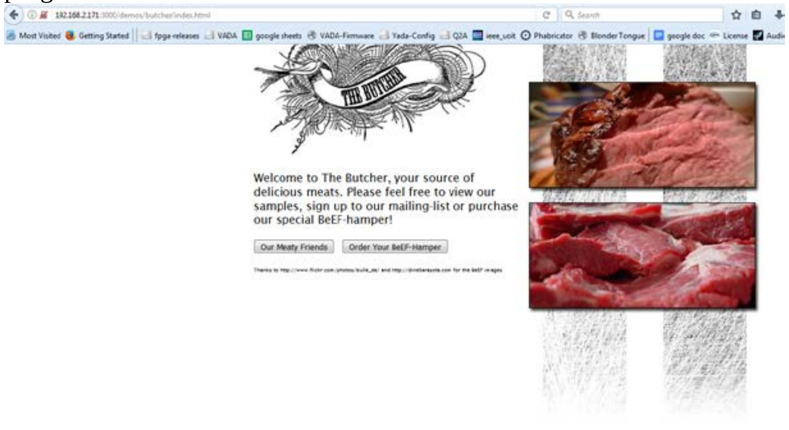
- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

Open Symantec Endpoint Protection
Update Policy
Disable Symantec Endpoint Protection
Customize...
9:32 PM

Figure 18: Disabling Symantec Antivirus

However, attack was successful after we disabled the antirust program as shown above.



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!

Figure 19: Successful Login after Disabling Antivirus

After the antivirus program was disabled, the victim's machine got "hooked" see Fig. 18 above.

IX. Future Work

In an article, it is proposed that after several successful attempts to steal credit card information or banking passwords, many companies are trying to step towards cloud-based browsers, a Java-free browser. A cloud-based browser store no data from each session and prevent any malware from networking with the user's computer. One such product is Authentic8's Silo. A separate browser that executes only after entering a password. It then executes on the cloud and calls up a list of links the user has previously entered, and can store passwords for those sites. All code executes on their remote servers, providing security against malware and privacy against tracking. [5]

X. Conclusion

In conclusion, we now know how threating it is for everyone to surf the web without using proper security practices. From the experiment, we have learned that web browser attack is a broad topic. Malicious users can execute all sort of attacks from XSS to Buffer Overflow if the user is not updating his system regularly. BeEF is a simple penetration-testing tool that can be used by anyone to test some attacks or hack someone's system, so it is necessary for everyone to keep up with the updates and patches.

REFERENCES

[1] "Web Browser Attack," *Pvamus*, Mar-2009. [Online]. Available: https://www.pvamu.edu/Include/ITS/Vol3Issue2.pdf. [Accessed: 15-Jul-2017].

[2] T. Garsiel and P. Irish, "How Browsers Work: Behind the scenes of modern web browsers," *html5rocks*, 05-Aug-2011. [Online]. Available: https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/. [Accessed: 15-Jul-2017].

[3] "BeEF - The Browser Exploitation Framework Project," *BeEF - The Browser Exploitation Framework Project*. [Online]. Available: http://beefproject.com/. [Accessed: 15-Jul-2017].

[4] J. Muniz and A. Lakhani, "Web Penetration Testing with Kali Linux," *Google Books*. [Online]. Available: https://books.google.ca/books?id=4fD7AAAAQBAJ&pg=PT5&lpg=PT5&dq=countermeasure%2Bagainst%2Bbeef%2Bkali%2Blinux&source=bl&ots=qxPZaai-Ff&sig=lyfWjc9AMfDcvjfLos9VpjVhRsA&hl=en&sa=X&ved=0ahUKEwihqdH98LzVAhWHz4MKHRwoDAAQ6AEIPzAE#v=snippet&q=beef&f=false. [Accessed: 15-Jul-2017].

[5] A. Tanner, "Why Cloud Browsers Are The Wave Of The Future," *Forbes*, 10-Mar-2014. [Online]. Available: https://www.forbes.com/sites/adamtanner/2014/03/10/why-cloud-browsers-are-the-wave-of-the-future/#150af5b9305a. [Accessed: 15-Jul-2017].

[6] "Hack Like a Pro: How to Hack Web Browsers with BeEF," *WonderHowTo*, 02-Feb-2015. [Online]. Available: https://null-byte.wonderhowto.com/how-to/hack-like-pro-hack-web-browsers-with-beef-0159961/. [Accessed: 15-Jul-2017].

[7] "Our Most Advanced Penetration Testing Distribution, Ever.," *Kali Linux*. [Online]. Available: https://www.kali.org/. [Accessed: 15-Jul-2017].

[8] "The CentOS Project," *CentOS Project*. [Online]. Available: https://www.centos.org/. [Accessed: 15-Jul-2017].

[9] "Oracle VM VirtualBox - Downloads | Oracle Technology Network | Oracle," *Oracle VM VirtualBox - Downloads | Oracle Technology Network | Oracle*. [Online]. Available: http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html. [Accessed: 15-Jul-2017].