

Government Polytechnic

Udupi

Name: Rekha K

Register No.: 145CS20014

Task Report: 3

1. Command Execution Vulnerability:

Command Execution or Command injection is an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell.

OS command injection is a web security vulnerability that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running an application, and typically fully compromise the application and all its data



- Home
- Instructions
- Setup
- Brute Force
- Command Execution**
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload
- XSS reflected
- XSS stored
- DVWA Security
- PHP Info

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```

PING 192.168.19.129 (192.168.19.129) 56(84) bytes of data.
64 bytes from 192.168.19.129: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 192.168.19.129: icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from 192.168.19.129: icmp_seq=3 ttl=64 time=0.013 ms

--- 192.168.19.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.013/0.021/0.027/0.005 ms
          
```

More info

<http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
<http://www.ss64.com/bash/>
<http://www.ss64.com/nt/>

2. File Upload Vulnerability:

A file upload vulnerability also called unrestricted file upload or arbitrary file upload is a potential security risk that allows an attacker to upload malicious files to a web server. It occurs when an application does not properly validate the file type or its content. In this way an attacker may be able to upload a file that could compromise the security of the server.

File upload vulnerabilities are when a web server allows users to upload files to its filesystem without sufficiently validating things like their name, type, contents, or size.



- Home
- Instructions
- Setup
- Brute Force
- Command Execution
- CSRF
- File Inclusion
- SQL Injection
- SQL Injection (Blind)
- Upload**
- XSS reflected
- XSS stored
- DVWA Security
- PHP Info
- About
- Logout

Vulnerability: File Upload

Choose an image to upload:

No file selected.

../../hackable/uploads/shell.php succesfully uploaded!

More info


http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

Username: admin
 Security Level: low
 PHPIDS: disabled

[View Source](#)
[View Help](#)

3. SQL Injection Vulnerability:

SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. SQL injection is considered a high risk vulnerability due to the fact that can lead to full compromise of the remote system.



The screenshot shows the DVWA web application interface. At the top is the DVWA logo. On the left is a sidebar menu with various security modules. The main content area is titled 'Vulnerability: SQL Injection'. It features a 'User ID:' label, a text input field, and a 'Submit' button. Below the input field, there are five lines of red text, each representing a successful login attempt with a different payload. The payloads are: 1. ID: %' or 'o'='o, First name: admin, Surname: admin; 2. ID: %' or 'o'='o, First name: Gordon, Surname: Brown; 3. ID: %' or 'o'='o, First name: Hack, Surname: Me; 4. ID: %' or 'o'='o, First name: Pablo, Surname: Picasso; 5. ID: %' or 'o'='o, First name: Bob, Surname: Smith.

Module
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info

Vulnerability: SQL Injection

User ID:

ID: %' or 'o'='o
First name: admin
Surname: admin

ID: %' or 'o'='o
First name: Gordon
Surname: Brown

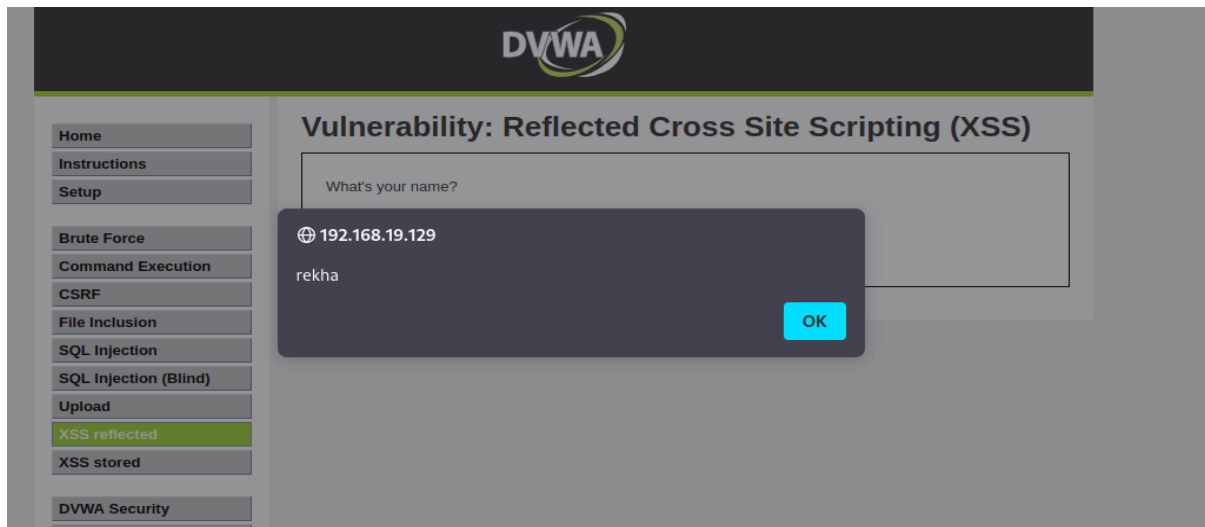
ID: %' or 'o'='o
First name: Hack
Surname: Me

ID: %' or 'o'='o
First name: Pablo
Surname: Picasso

ID: %' or 'o'='o
First name: Bob
Surname: Smith

4. Cross Site Scripting:

Cross Site Scripting (XSS) is a technique in which attackers inject malicious scripts into a target website and may allow them to gain access control of the website. XSS attack occurs when an attacker uses a web application to send malicious code or a browser-side script, to a different end user. The victim has no way of knowing that the script is malicious thus believing that it is from a trusted source.

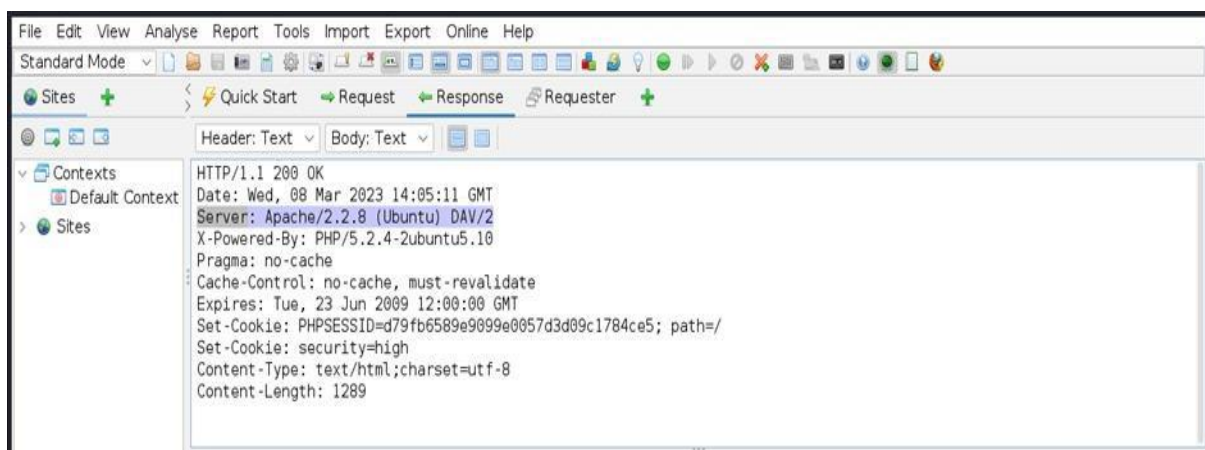


5. Sensitive information disclosure:

Information disclosure, also known as information leakage, is when a website unintentionally reveals sensitive information to its users. Depending on the context, websites may leak all kinds of information to a potential attacker, including:

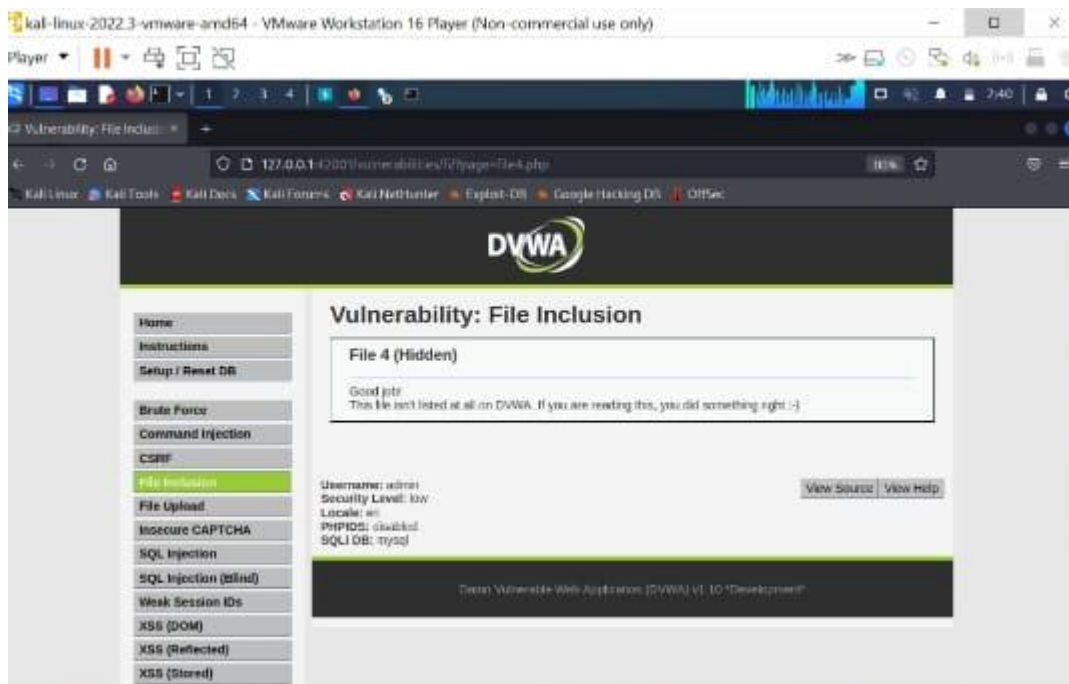
- Data about other users, such as usernames or financial information.
- Sensitive commercial or business data.
- Technical details about the website and its infrastructure.

Sensitive information disclosure happens when an application does not adequately protect sensitive information that may wind up being disclosed to parties that are not supposed to have access to it.



6. Local file inclusion:

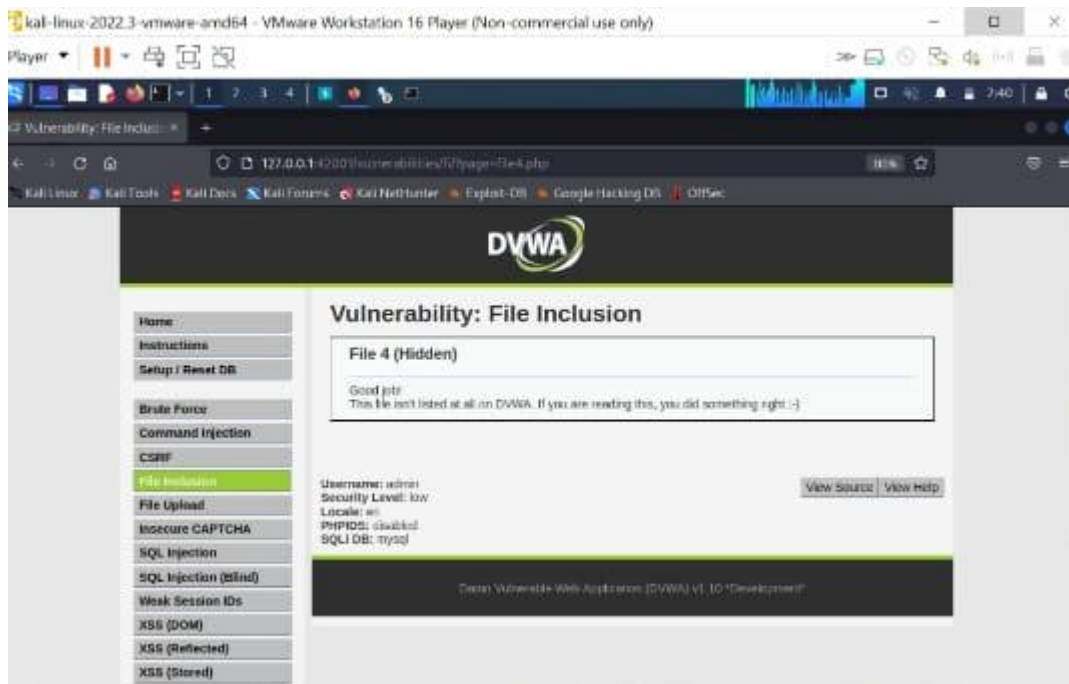
This vulnerability allows an attacker to include files stored locally on the server. An attacker can use Local file inclusion to access sensitive files on the server, such as configuration files or log files, which may contain sensitive information such as usernames, passwords, and server paths. Local File Inclusion (LFI) allows an attacker to include files on a server through the web browser.



7. Remote file inclusion:

This vulnerability allows an attacker to include files from a remote server, such as an attacker's own server. The hacker can use Remote file inclusion to execute arbitrary code on the target server, which could allow the attacker to take control of the server. If RFI is present, there is no need to rely on another vulnerability to upload and execute arbitrary files.

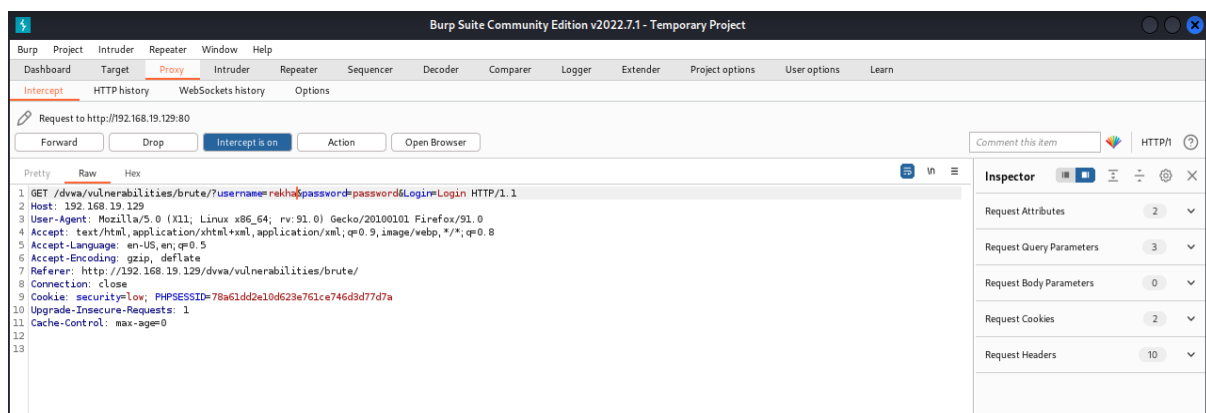
Remote file inclusion (RFI) is an attack targeting vulnerabilities in web applications that dynamically reference external scripts. The perpetrator goal is to exploit the referencing function in an application to upload malware from a remote URL located within a different domain.



8. Brute Force Attack:

A brute force attack is a type of cyber attack where a hacker uses an automated tool to guess the password of a user or system. Hackers usually perform this attack when they do not have any prior knowledge of the password or the system and are trying to gain access to a system or account.

A brute force attack is uses a trial-and-error approach to systematically guess login info, credentials, and encryption keys. The attacker submits combinations of usernames and passwords until they finally guess correctly.



DVWA

Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin

More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT-004%29
<http://www.securityfocus.com/infocus/1192>
<http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

View Source View Help

Username: admin
Security Level: low
PHPIDS: disabled

9. Forced browsing vulnerability:

Forced browsing is an attack where the aim is to enumerate and access resources that are not referenced by the application, but are still accessible. Forced browsing attack are the result of a type of security misconfiguration vulnerability. These kind of vulnerabilities occur when insecure configuration or misconfiguration leave web application components open to attack.

10.Components with known vulnerability:

Using Components with Known Vulnerabilities According to OWASP: Using Components with Known Vulnerabilities Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate server data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine the app.

11.Html injection:

Hypertext Markup Language (HTML) injection is a technique used to take advantage of non-validated input to modify a web page presented by a web application to its users. Attackers take advantage of the fact that the content of web page is often related to a previous interaction with users. When applications fail to validate user data,

an attacker can send HTML- formatted text to modify site content that gets presented to other users.

