HTB Machine Write Up: Bashed

Write up by Ori Iankovitch

Start: Enumeration

Started with an Nmap scan and found that ports 80 (http) was open, running an apache server.

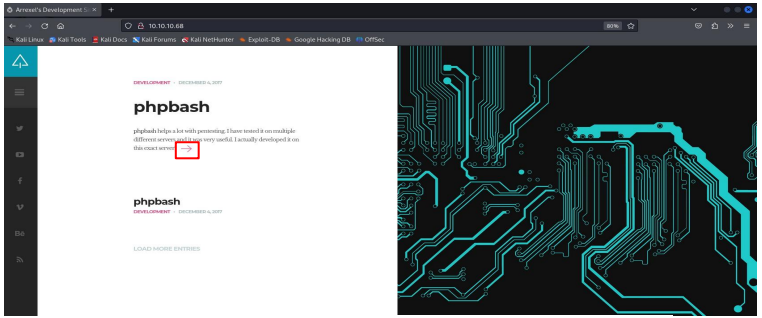
Opened the web page and found this website designed to help with pentesting.

After clicking on the arrow in the middle of the page it took me into the “/single.html” page, containing this link to GitHub “https://github.com/Arrexel/phpbash”.

The GitHubs readme file explains that this is a standalone semi-interactive web shell based on kali linux, that its purpose is to assist in penetration tests. The GitHub page also contained two php files.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -sC -sV 10.10.10.68
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 18:26 EDT
Nmap scan report for 10.10.10.68
Host is up (0.069s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Arrexel's Development Site
|_http-server-header: Apache/2.4.18 (Ubuntu)

Service detection performed. Please report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 11.02 seconds
```

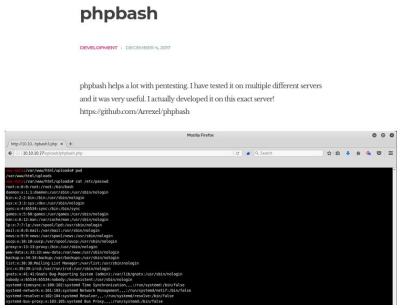


LICENSE	Initial commit	7 years ago
README.md	spelling fix, no content changes	6 years ago
phpbash.min.php	Patch XSS vuln	6 years ago
phpbash.php	Patch XSS vuln	6 years ago

README Apache-2.0 license

phpbash

phpbash is a standalone, semi-interactive web shell. It's main purpose is to assist in penetration tests where traditional reverse shells are not possible. The design is based on the default Kali Linux terminal colors, so pentesters should feel right at home.

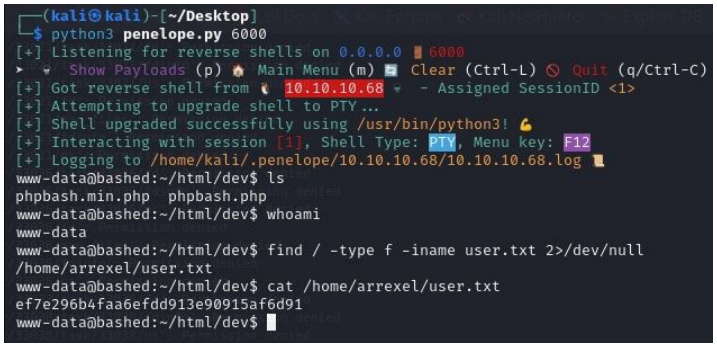


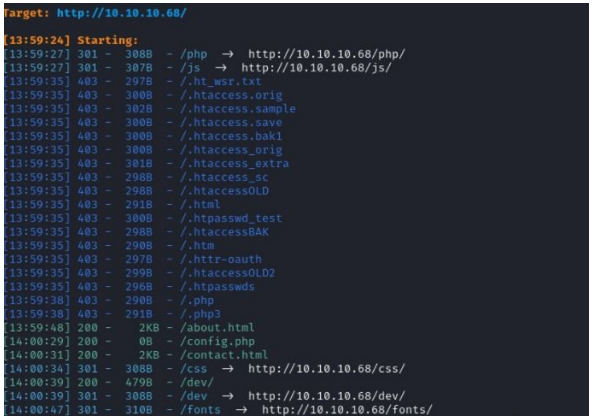
Next step is to run “Dirsearch” and look for useful directories in the site.

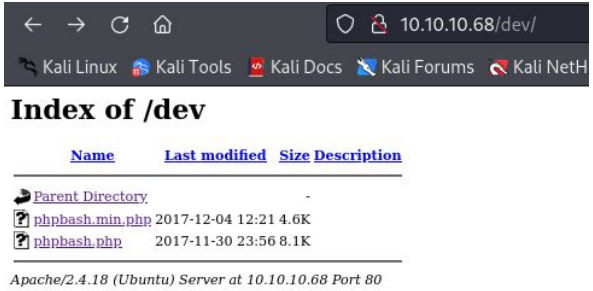
After looking at the results and checking the different directories, I came across “/Dev” page witch had a link to the web shell.

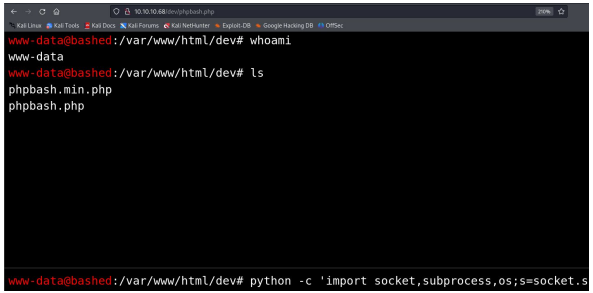
I opened a listener on my machine, in the server I ran a python reverse shell command, and got a shell with the user “www-data”.

After obtaining a shell I used the “find” command to get the first flag.









Privilege Escalation:

Next, I used the sudo -l command to identify the commands that the default user can execute with root privileges.

The output shows that the user can run any command using “scriptmanager” without password.

After some researching I run the command “sudo -u scriptmanager bash -i” and got the “scriptmanager” users shell.

“sudo -u scriptmanager”: Runs the command as the scriptmanager user.

“bash”: Launches a new shell session as the scriptmanager user.

“-i”: makes the shell interactive, so it loads the user's settings.

```
www-data@bashed:~/html$ sudo -l
Matching Defaults entries for www-data on bashed:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/u

User www-data may run the following commands on bashed:
    (scriptmanager : scriptmanager) NOPASSWD: ALL
www-data@bashed:~/html$ sudo scriptmanager bash -i
[sudo] password for www-data:
www-data@bashed:~/html$  sudo -u scriptmanager bash -i
scriptmanager@bashed:/var/www/html$
```

```
scriptmanager@bashed:/$ ls -la
total 92
drwxr-xr-x  23 root          root          4096 Jun  2  2022 .
drwxr-xr-x  23 root          root          4096 Jun  2  2022 ..
-rw-------   1 root          root           174 Jun 14  2022 .bash_history
drwxr-xr-x   2 root          root          4096 Jun  2  2022 bin
drwxr-xr-x   3 root          root          4096 Jun  2  2022 boot
drwxr-xr-x  19 root          root          4140 Aug 31 08:04 dev
drwxr-xr-x  89 root          root          4096 Jun  2  2022 etc
drwxr-xr-x   4 root          root          4096 Dec  4  2017 home
lrwxrwxrwx   1 root          root            32 Dec  4  2017 initrd.img →
drwxr-xr-x  19 root          root          4096 Dec  4  2017 lib
drwxr-xr-x   2 root          root          4096 Jun  2  2022 lib64
drwx------   2 root          root         16384 Dec  4  2017 lost+found
drwxr-xr-x   4 root          root          4096 Dec  4  2017 media
drwxr-xr-x   2 root          root          4096 Jun  2  2022 mnt
drwxr-xr-x   2 root          root          4096 Dec  4  2017 opt
dr-xr-xr-x 255 root          root             0 Aug 31 08:04 proc
drwx------   3 root          root          4096 Aug 31 08:06 root
drwxr-xr-x  18 root          root           520 Sep  1 06:25 run
drwxr-xr-x   2 root          root          4096 Dec  4  2017 sbin
drwxrwxr--   2 scriptmanager scriptmanager  4096 Sep  1 11:59 scripts
drwxr-xr-x   2 root          root          4096 Feb 15  2017 srv
dr-xr-xr-x  13 root          root             0 Sep  1 15:39 sys
drwxrwxrwt  10 root          root          4096 Sep  1 16:49 tmp
drwxr-xr-x  10 root          root          4096 Dec  4  2017 usr
drwxr-xr-x  12 root          root          4096 Jun  2  2022 var
lrwxrwxrwx   1 root          root            29 Dec  4  2017 vmlinuz → boo
```

Running the command “ls -la”, shows that the only directory under the user “scriptmanager” is the “scripts” directory.

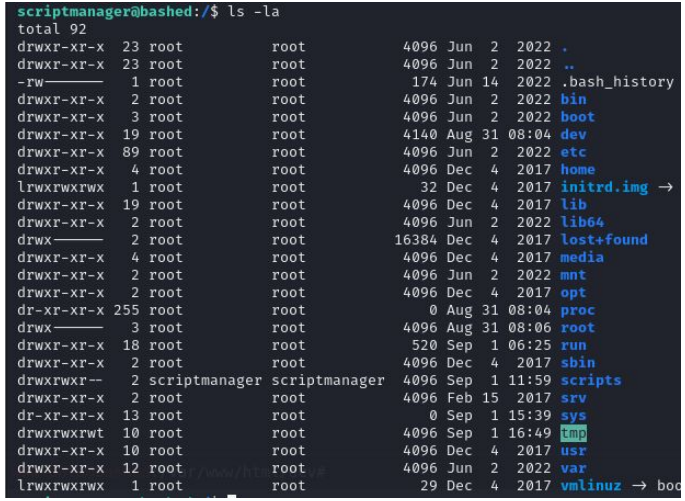
I navigated to that directory and found two files: test.txt and test.py.

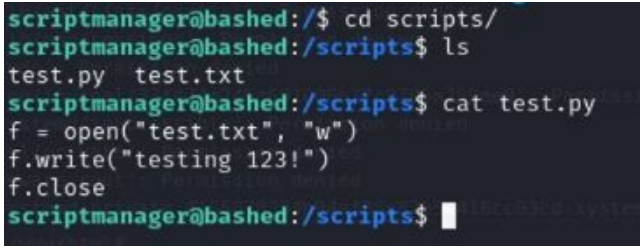
The test.py contains a simple python script that write into the test.txt file.

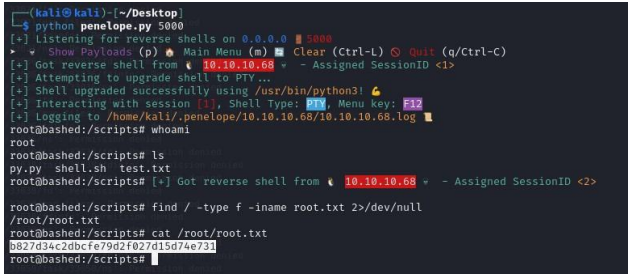
I was able to see that the python script is executed repeatedly about every one minute, creating and editing the text.txt file.

Because the script was executed repeatedly, I assumed that the files in this directory are being executed by a root privileged user, so I made a new python script, using a reverse shell command.

I Opened a new listener on my machine, waited about a minute, got a root shell and found the missing flag.







Bashed has been Pwned!

Congratulations **shokoyanko**, best of luck in capturing flags ahead!

#29161	01 Sep 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE