2025.10.03

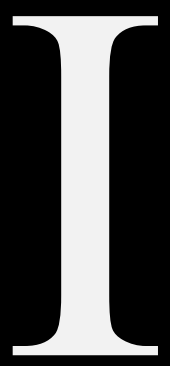
Format String Bug

가디언 시스템 보안 & 취약점 분석 세미나 4.1

임준서

Format String

printf, vprintf, ···



Format String bug

Vulnerability

잘 사용하면 문제 X

```
printf(buffer);
```

사용자가 format-string을 조작하면 AAR/AAW 가능

Format String bug

PoC

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

void win(){
    system("/bin/sh");
}

int main() {
    char str[40];
    printf("Enter some text: ");
    scanf("%40s", str);
    printf(str);
    return 0;
}
```

Format String bug

PoC

아래 코드를 입력해보자

hi%13$n

Format String

%[argument$][flag][width][.precision][length][specifier]

specifier

d,u,s,x,o 등

%n

현재까지 출력된 내용을 argv에 저장

UNIX 시스템 및 POSIX 확장 계열에만 해당

C99 Spec: cplusplus.com/reference/cstdio/printf/

Unix Spec: man7.org/linux/man-pages/man3/printf.3.html

Format String

%[argument$][flag][width][.precision][length][specifier]

width : 출력할 총 길이

%2d처럼 사용한다.

length : 변수의 실제 크기

hh=char, h=short, l=long, ll=longlong

UNIX 시스템 및 POSIX 확장 계열에만 해당

C99 Spec: cplusplus.com/reference/cstdio/printf/

Unix Spec: man7.org/linux/man-pages/man3/printf.3.html

Format String

%[argument$][flag][width][.precision][length][specifier]

parameter : 참조할 argv의 인덱스

$를 구분자로 사용한다.

%0$x

what will happen?

UNIX 시스템 및 POSIX 확장 계열에만 해당

C99 Spec: cplusplus.com/reference/cstdio/printf/

Unix Spec: man7.org/linux/man-pages/man3/printf.3.html

Format String

%[argument$][flag][width][.precision][length][specifier]

```
Enter some text: %0$x
%0$x
```

Calling Convention

```
rdi, rsi, rdx, rcs, r8, r9, [rsp], [rsp+8], …
 0 ,  1 ,  2 ,  3 , 4 , 5 ,   6  ,    7   , …
```

Format String

%[argument$][flag][width][.precision][length][specifier]

```
pwndbg> stack -if
07:0038|+008 0x7fffffffd878 —▸ 0x7ffff7de1ca8 (__libc_start_call_main+120)
06:0030|  rbp 0x7fffffffd870 ◂— 1
05:0028|-008 0x7fffffffd868 —▸ 0x7fffffffd900 ◂— 1
04:0020|-010 0x7fffffffd860 ◂— 0
03:0018|-018 0x7fffffffd858 —▸ 0x7ffff7fe4900 (dl_main) ◂— push rbp
02:0010|-020 0x7fffffffd850 ◂— 0
01:0008|-028 0x7fffffffd848 ◂— 0
00:0000|  rsp 0x7fffffffd840 ◂— 0
```

%13$x

[rsp]=%6$

Format String

%[argument$][flag][width][.precision][length][specifier]

```
pwndbg> c
Continuing.
Enter some text: %13$x
f7de1ca8[Inferior 1 (process 874) exited normally]
```

[rsp+7*8]의 값을 hex로 출력해주세요

Format String

Stack leak

%{arg_idx}$x 로 stack leak

이후 배울 AAW 과 결합하면 AAR 가능

Format String

AAW

%{숫자}$n

{숫자}번째 argv가 가리키는 포인터에 지금까지 출력된 문자 수를 저장

stack을 조작해서 addr 셋업 후 write

[argument$]은 UNIX Spec이지만 %n 은 C99 Standard이다.

사용자 입력이 스택 어딘가에 저장된다는 가정하에 셋업할 수 있다.

실습 [fsb]

주어진 바이너리

Homework [fsb2lib]

주어진 바이너리

2025.10.03

Q&A

질문이 있다면 하십시오

임준서

2.5cm-2.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

3.5cm 떨어진 내용 1 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

3.5cm 떨어진 내용 2 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

3.5cm 떨어진 내용 3 32px

좌측으로 0.5cm 떨어진 내용 하단의 설명 18px

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

2.5cm-3.5cm 떨어진 제목 36px

제목 하단의 부제목 18px

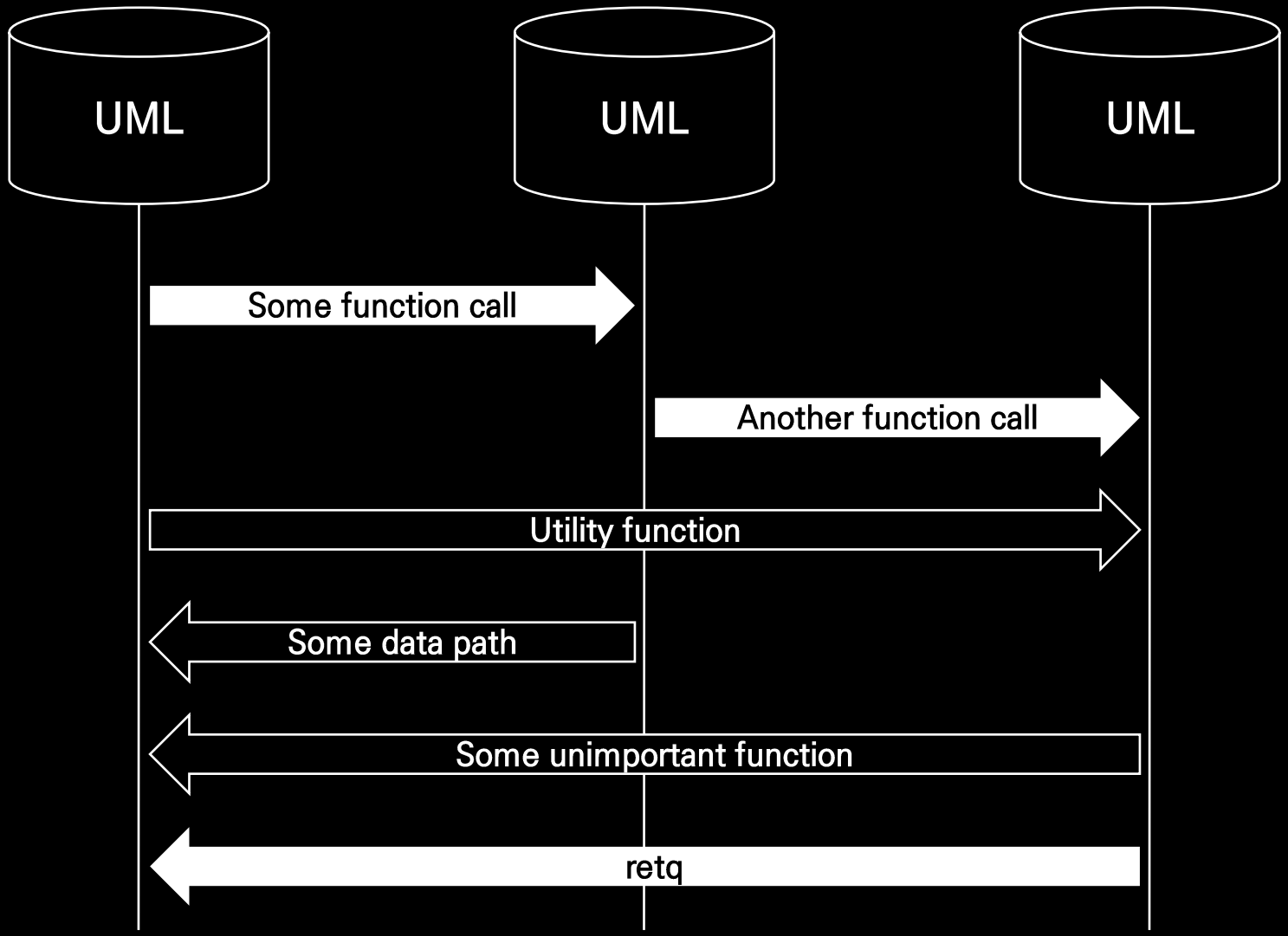
3.5cm 떨어진 내용 1 32px

```
Git init
Git status
Git add text.txt
Git add .
Git commit
Ctrl+C
Git commit -m “genesis”
Git log
Git log --oneline
```

```
Git add .
Git reset .
Git commit -m “add README”
Git log --oneline -n 3
Git commit –a –m “hello”
```

1cm-1cm 떨어진 주석 12px

1cm-1cm 떨어진 주석 12px

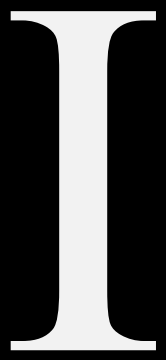


Guardian 2025

UNUSED SLIDE

Signal Oriented Programming

SROP



Signal UNUSED SLIDE

x86-64 ABI

프로그램의 비정상적 종료

Caught unexpected signal: SIGSEGV (11)