

2025.09.10

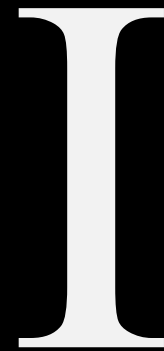
# Orientation

가디언 시스템 보안 & 취약점 분석 세미나 1.1

임준서

# What we learn

Introduction



# 시스템 해킹

Your computer is mine.

## RCE & LPE

Remote Code Execution & Local Privilege Escalation

## 다른 컴퓨터의 제어권을 탈취

원격으로 Random 한 코드를 실행하자 & 권한을 (강제로) 승격받자

Pwnable[포너블] 이라고도 한다.

# 시스템 해킹

종류

User-land

Kernel

Other system (V8, windows, ...)

# 시스템 해킹

세미나

세미나에서는

“Linux”의 “Userland”

# 시스템 해킹

Userland



```
1  What is you name
2  >> AAAAAAA} ' ' a.
3  $
```

# Environment

## Setup

# II

# GDB

Main debugger

프로그램을 디버깅 할 때 쓴다

근데 기능이 다소 아쉬움

1986년 초도 개발

Gaurdian 2025



# Pwndbg

Main debugger

## GDB + 확장기능

이름을 보면 알겠지만 보통 Pwnable 할 때 씬

## 다른 디버깅 툴도 있지만, 굳이?

<https://github.com/pwndbg/pwndbg>

# Pwndbg

## Main debugger

```
Use the errno (or errno <number>) command to see the name of the last or provided (libc) error
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA

[ REGISTERS / show-flags off / show-compact-regs off ]
RAX 0xffffffffffffe00
RBX 0x7f372e8c2740 ← 0x7f372e8c2740
RCX 0x7f372e954687 (__internal_syscall_cancel+103) ← pop rbx
RDX 1
RDI 0
RSI 0x7f372e9ac963 (_IO_2_1_stdin_+131) ← 0xaae7c00000000000
R8 0
R9 0
R10 0
R11 0x202
R12 0x7f372e9aae80 (__io_vtables) ← 0
R13 0x5567429f1040 (stdout@GLIBC_2.2.5) → 0x7f372e9ad5c0 (_IO_2_1_stdout_) ← 0xfbad2887
R14 0xffffffff
R15 0x63
RBP 0x7f372e9aafd0 (_IO_file_jumps) ← 0
RSP 0x7ffdb92c533f0 → 0x7f372e9ac8e0 (_IO_2_1_stdin_) ← 0xfbad208b
RIP 0x7f372e954687 (__internal_syscall_cancel+103) ← pop rbx

[ DISASM / x86-64 / set emulate on ]
► 0x7f372e954687 <__internal_syscall_cancel+103>      pop     rbx      RBX => 0x7f372e9ac8e0 (_IO_2_1_stdin_)
0x7f372e954688 <__internal_syscall_cancel+104>      ret
+
0x7f372e9546ad <__syscall_cancel+13>                pop     rdx      RDX => 0
0x7f372e9546ae <__syscall_cancel+14>                pop     rcx      RCX => 0x7ffdb92c533c0
0x7f372e9546af <__syscall_cancel+15>                movsxd  rdx, eax  RDX => 0xffffffffffffe00
0x7f372e9546b2 <__syscall_cancel+18>                cmp     eax, 0xffffffff000 0xffffffff000 - 0xffffffff000  EFLAGS => 0x206 [ cf PF af zf sf IF df of ac ]
0x7f372e9546b7 <__syscall_cancel+23>                jae     __syscall_cancel+40 <__syscall_cancel+40>
+
0x7f372e9546c8 <__syscall_cancel+40>                mov     rdx, qword ptr [rip + 0x157709]  RDX, [GLOBAL_OFFSET_TABLE_+632] => 0xffffffffffff78
0x7f372e9546cf <__syscall_cancel+47>                neg     eax
0x7f372e9546d1 <__syscall_cancel+49>                mov     dword ptr fs:[rdx], eax  [0x7f372e8c26b8] <= 0x200
0x7f372e9546d4 <__syscall_cancel+52>                mov     rdx, 0xffffffffffffff  RDX => 0xffffffffffffff

[ STACK ]
00:0000 | rsp 0x7ffdb92c533f0 → 0x7f372e9ac8e0 (_IO_2_1_stdin_) ← 0xfbad208b
01:0008 | 0x7ffdb92c533f8 → 0x7f372e9546ad (__syscall_cancel+13) ← pop rdx
02:0010 | 0x7ffdb92c53400 ← 0
03:0018 | 0x7ffdb92c53408 → 0x7ffdb92c533c0 ← 0xffffb258a66a00000
04:0020 | 0x7ffdb92c53410 ← 0
05:0028 | 0x7ffdb92c53418 → 0x7f372e9c8ea6 (read+22) ← add rsp, 0x18
06:0030 | 0x7ffdb92c53420 ← 0
07:0038 | 0x7ffdb92c53428 → 0x7f372e9ac8e0 (_IO_2_1_stdin_) ← 0xfbad208b

[ BACKTRACE ]
► 0 0x7f372e954687 __internal_syscall_cancel+103
1 0x7f372e9546ad __syscall_cancel+13
2 0x7f372e9c8ea6 read+22
3 0x7f372e94f861 _IO_file_underflow+337
4 0x7f372e951beb _IO_default_uflow+43
5 0x7f372e91e7ba _vfprintf_internal+1866
6 0x7f372e91e0be _isoc99_scanf+174
7 0x5567429d78bf calculator()+74

pwndbg> 
```

# Pwndbg

Main debugger

```
git clone  
https://github.com/pwndbg/pwndbg && \  
cd pwndbg && \  
./setup.sh
```

<https://github.com/pwndbg/pwndbg>

# IDA

Decompiler

## Assembly → C

## 세미나에선 IDA 사용 예정

[아이다]

<https://hex-rays.com/ida-free>

Gaurdian 2025

# Python – pwntools

Tool

Pwnable을 위한 패키지

페이로드 보낼 땐 거의 필수

```
pip install pwntools
```

<https://github.com/Gallopsled/pwntools>

Gaurdian 2025

# Docker

Tool

서버와 동일한 환경 제공

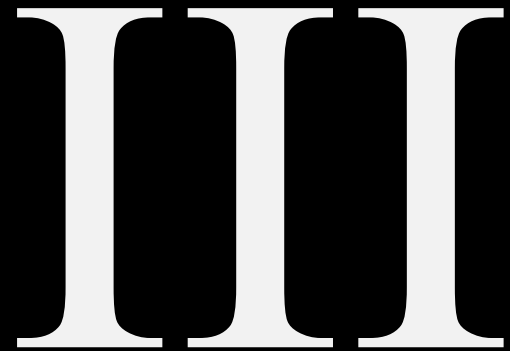
Dockerfile을 통해 build, run

<https://docs.docker.com/engine/install/debian/>

Gaurdian 2025

Before we start

Tips



# 윤리의식

왜 가져야 하는가

**지정된 서버 외에는 공격하지 마세요.**

돈도 안 받고 그런 짓 하면 재능 아까워요

**예전에는**

어디 해킹해서 어디 입사하고 그런 게 통했음

**요즘은 알짜없음**

즉시 등록 후 레드든 블루든 업계 못 들어오게 함.

인성 좋은 그저 그런 해커 >>>> 인정 더러운 천재 해커



# 삽질도 힘이다

열심히 디버깅 하기

왜 안될까?

GDB를 활용해서 확인하기

왜 서버에서 안 되지?

주로 lib 버전차이 + 정렬문제

# 풀이의 다양성

여러가지 시도하기

문제의 풀이는 여러가지

풀고 다른 풀이도 공부

# 시스템 해킹 못 함

```
# 0x20 pc_low
# 0x28 pc_high
# 0x30 load_base
# 0x38 p_gh_frame_hdr (pointer)
# 0x40 p_dynamic (pointer)
# 0x48 frame_hdr cache element (pointer)
```

```
/sourceurl.py
[x] Starting local process './prob
[+] Starting local process './prob
[+] pid 37659
[DEBUG] Wrote gdb script to '/tmp/
    b __cxa_throw
        b _Unwind IteratePhdrCallk
[*] running in new terminal: ['/us
'-x', '/tmp/pwnlib-gdbscript-7732
[DEBUG] Created script for new ter
#!/home/zirajs/miniconda3/envs
import os
os.execve('/usr/bin/gdb', ['/u
, '-x', '/tmp/pwnlib-gdbscript-773
[DEBUG] Launching a new terminal:
[+] Waiting for debugger: Done
[DEBUG] Sent 0x2 bytes:
    b'+\n'
[DEBUG] Sent 0x2 bytes:
    b'-\n'
[DEBUG] Sent 0x2 bytes:
    b'\n'
[DEBUG] Received 0x71 bytes:
    b'Simple C Calculator\n'
    b'Usage: [number] [operator] [number]\n'
    b'Operators: + - * /\n'
    b'"Enter \'q\' as operator to quit.\n"'
    b'\n'
    b'>> '
[DEBUG] Received 0x33 bytes:
    b'Num 1 : Num 2 : 55d0389beb59 + 1 = 55d0389beb5a\n'
    b'>> '
[DEBUG] Sent 0x1 bytes:
    b'q'
[DEBUG] Received 0x14 bytes:
    b'Exiting calculator.\n'
[DEBUG] Sent 0x30d bytes:
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |....|
*
00000130 00 00 00 00 00 00 00 00 ff ff ff ff ff ff ff ff |....|
00000140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |....|
*
000002c0 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 |....|
000002d0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |....|
000002e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |....|
*
00000300 00 00 00 00 00 00 00 00 00 00 00 00 0a          |....|.
0000030d
```



2025.09.10

# 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