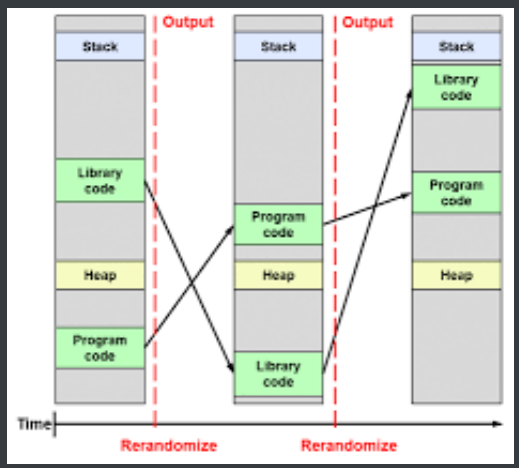
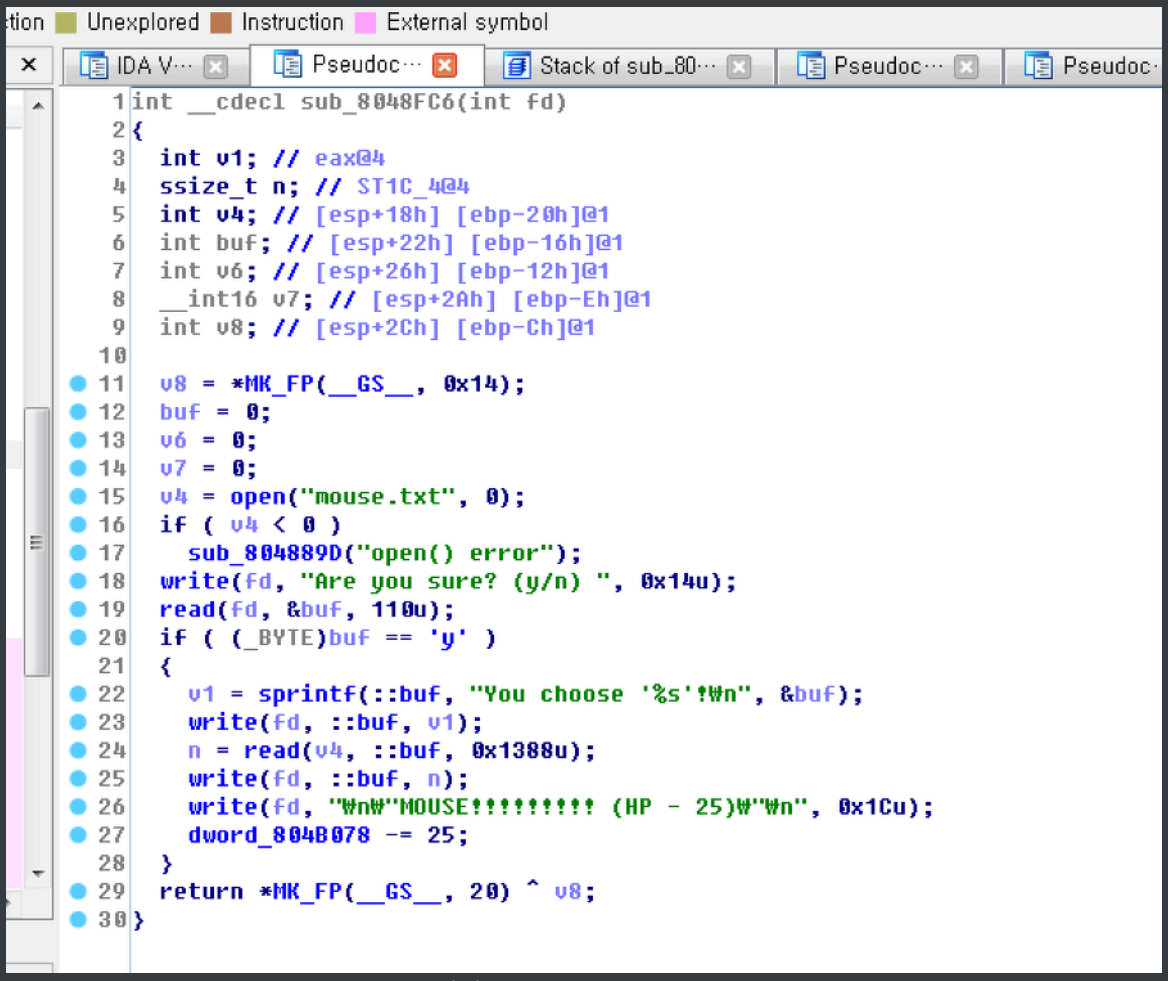
2019_05_04_5차시

BOF취약점 방지를 위한 보호기법 알아보기

1. NX(Non-eXecutable)
    1. 메모리 영역에서 실행권한을 뺌.
    2. 즉 메모리 영역에서 실행이 불가능함.
    1.
```
m0nday@trust ~/challenges/wargames/ropbaby $ checksec ropbaby
[*] '/home/m0nday/challenges/wargames/ropbaby/ropbaby'
    Arch:     amd64-64-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
    FORTIFY:  Enabled
```
2. ASLR (Address Space Layout Randomization)
    1. 메모리 영역들의 주소 공간 배치를 랜덤화하여 공격을 방해.
    2. 직접적인 메모리 참조가 힘들어짐.
    3. 환경변수와 같은 곳을 직접참조가 힘들다.
    4. 매번 실행할때마다 메모리의 위치가 바뀐다. 하지만 프로그램을 그대로이므로 (SBP라고 하는 stack base point만 변한다.)
    1. 

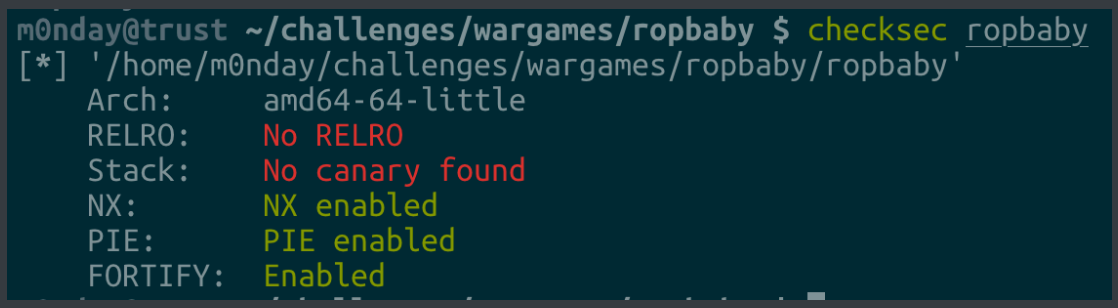
3. ASCII Armor
    1. 공유 라이브러리 영역의 상위 주소에 0x00을 포함시키는 기법
    2. RTL 공격을 대응하기 위함(00값으로 인해 문자열의 널바이트로 인식시켜 인자 전달을 불가능하게 함)
        1. RTL 공격 : DEP를 우회하기 위해 라이브러리 함수를 사용하는 공격이다.
4. CANARY
    1. Buffer 와 RET 사이의 값 변조 모니터링하는것이다.
    2. CANARY 값이 변조될 경우 경고 후 프로그램 종료된다.



3. gcc -fno-stack-protector —> SSP 해제

5. PIE(Position Independent Executable)

    1. 전체가 위치 독립 코드로 이루어진 실행 가능한 바이너리
    2. 모든 심볼 주소를 상대적으로 작성하고 base address를 랜덤화해서 main()같은 함수들의 주소를 실행할 때마다 랜덤화하는 것이다.

    1. 

    2. gcc -no-pie —> PIE 해제