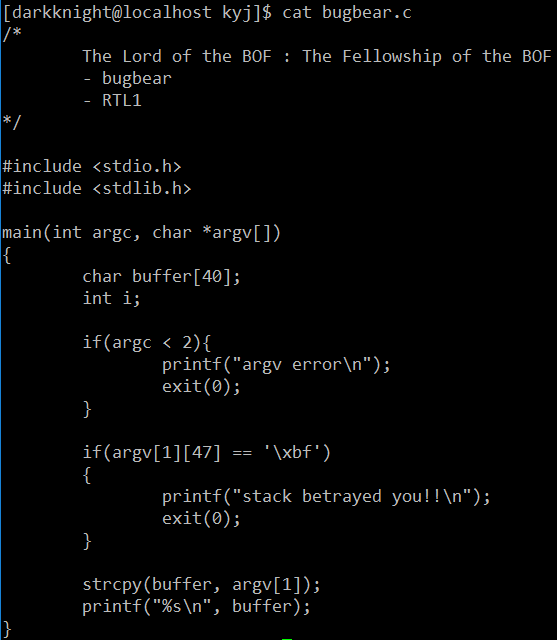
|  |
| --- |
| LOB Bugbear |
| 동아리마크  2018. 03. 08 (thur)  Dong-a .univ  COLONY  Young jin kim |

목차

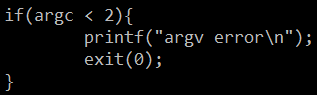
1. 소스코드 이해
2. 소스코드 분석
3. 공격
4. **소스코드 이해**



[ bugbear.c의 소스코드 ]

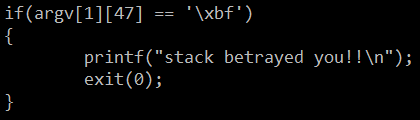
|  |  |
| --- | --- |
| Buffer[40]변수 선언  i변수 선언  인자가 2개이하이면 종료  첫번째인자 48번째값이 \xbf이면 종료  Strcpy로 buffer=argv[1]  Buffer에 있는값 출력 | RTL : Return To Library   * 스택에서의 버퍼플로우가 다 막히자 * 공유 라이브러리의 함수를 통해 * 공격을 하는 방법 |

1. **소스코드 분석**



[ 조건 1 ]

첫번째, 인자는 2개 이상 들어와야한다.



[ 조건 2 ]

첫번째, 첫번째인자 48번째 데이터는 \xbf이면 안된다.

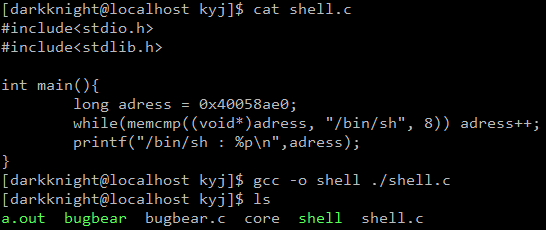
1. SFP, RET는 조작이 가능하다
2. Buffer에 무언가를 넣을 수 있다.
3. 하지만 조건에서 \xbf 는 쓸 수없다.
4. 스택영역이 아닌 다른 영역에 메모리를 넣어서 공격해야 할 것 같은 생각을가지고
5. RTL 공격에 대해 찾아보았다.

**2.소스코드 분석**



[ System함수의 메모리상 위치 ]

첫번째, Ret주소를 0x40058ae0로 돌려주어서 시스템 함수를 호출한다.



[ /bin/sh 주소 찾기 ]

첫번째, system함수가 실행시킬 코드 /bin/sh가 저장된 메모리 찾는 코드를 만들었다.



[ /bin/sh 주소 찾기 2 ]

첫번째, /bin/sh가 저장된 메모리의 주소값은 0x400fbff9



[ 확인 ]

첫번째, /bin/sh가 저장된 메모리 확인.

**RTL에 대한 간단한 정리,**

Return To Library의 약자이며 공유 라이브러리는 메모리에 프로그램이 실행되면서 올라간다.

이때 이 공유 라이브러리의 함수를 이용하는데,

이 공격 방식은 스택에서의 버퍼 오버플로우가 막히면서 등장한 방법이라고 한다.

먼저, 메인함수가 돌아가다가 프로그램이 끝날 때 ret를 수행한다.

Ret를 조작하여서 공유 라이브러리 메모리상에 있는 system함수의 위치를 gdb로 찾아내어

Ret 시켜준다.

Ret다음에 메모리에는 argc argv가 있다.

시스템 함수를 실행시킬 인자값이 필요한데. 이 인자는 “/bin/sh”이고,

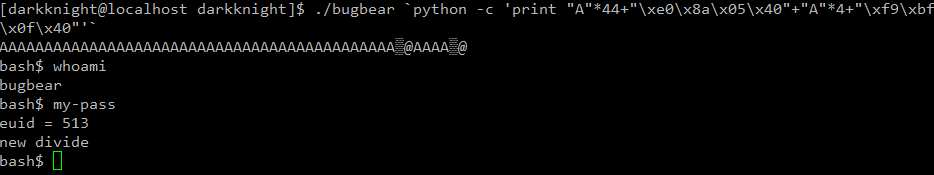
이 인자를 argv위치에 넣어준다.

**3.공격**

공격을 시작해보자.

나의 페이로드는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Buffer[40] | Sfp[4] | Ret[4] | argc | argv |
| A\*40 | A\*4 | 0x40058ae0 | A\*4 | 0x400fbff9 |



[ 익스플로잇 ]

PW : new divide