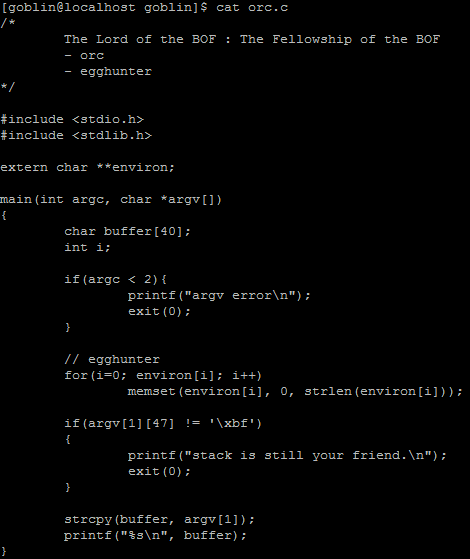
|  |
| --- |
| LOB goblin |
| 동아리마크  2018. 01. 10 (wed)  Dong-a .univ  COLONY  Young jin kim |

목차

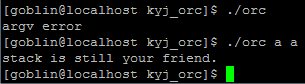
1. 소스코드 이해
2. 소스코드 분석
3. 공격
4. **소스코드 이해**



[ orc.c의 소스코드 ]

|  |  |
| --- | --- |
| 외부변수 extern선언  buffer 배열 선언[40byte]  I 변수 선언  인자값이 1개 이하이면 에러출력 후  프로그램 종료  memset함수로 2차원 배열 한줄 씩 초기화  argv[1][47]의 값이‘\xbf’와 다르면 프로그램 종료  buffer배열에 첫번째 인자값 복사  버퍼 값 출력 | memset(초기화할 변수, 초기화 값, 변수의 길이)  extern : 다른소스파일에 선언된 변수를  사용하기위한 코드  Environ : unistd.h에 속한 변수지만 프로그램 만들 때,  자동으로 라이브러리에 포함됨 |

1. **소스코드 분석**



[예외 종료 이미지]

첫번째, 인자값이 없는경우 에러

두번째, 인자값 2번째배열 48번째칸에 ‘\xbf’ 값이 있지 않아 오류 출력



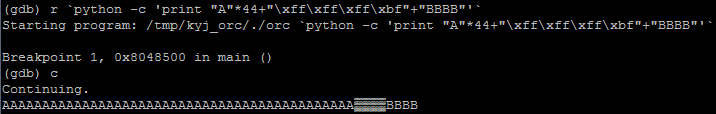
[환경변수 초기화]

첫번째, 환경변수 값을 초기화해서 쉘코드를 넣어두는 것을 막음



[디버깅]

첫번째, buffer[40]+int i[4]를 합쳐 44바이트가 할당 됨

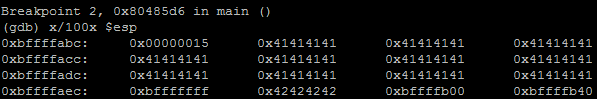


[디버깅(주소값 찾기)]

첫번째, 48번째 인자에 “\xbf”가 들어가야 한다.

(bf를 강제함으로써 다른 메모리에 쉘 코드를 넣는 것을 방지) [ ex)eggshell ]

1. **소스코드 분석**



[디버깅(주소값 찾기2)]

|  |  |  |  |
| --- | --- | --- | --- |
| Int i[4] | Buffer[40] | Ebp[4] | Ret[4] |
|  | “A”\*44 | | “\xbf………” |

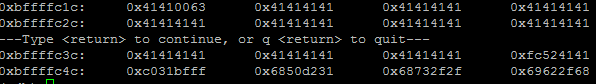
첫번째, 위 내용을 종합하면 우리의 소스코드는 인자값이 2개 이상이고

두번째, 환경변수 사용 불가하며

세번째, ebp포함 44바이트 만큼 데이터를 채우고,

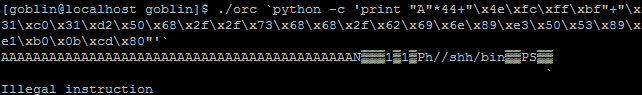
네번째, ret첫번째 값은 \xbf(쉘코드의 위치 강제함)가 들어가야 함

다섯째, 쉘코드가 들어갈 자리는 0xbfffaf1



[코어파일 디버깅]

첫번째, 코어파일 분석결과 주소값은 \xbffffc4e



[원본과 사본의 스택 주소차이로 인한 오류]

첫번째, 사본은 쉘 따는데 성공을 했다.

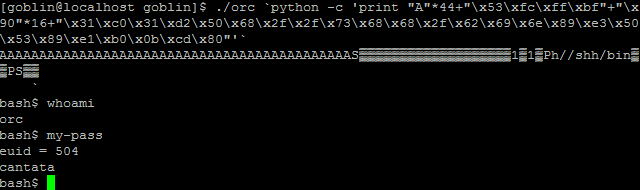
두번째, 원본은 쉘 따는데 실패 했다.

세번째, 문제의 길이 차이로 인해 메모리 값이 다를 수 있다.

Ex] 원본 : /home/goblin/orc 사본 : /tmp/kyj\_orc/orc

네번째, ret주소를 뒤쪽으로 살짝 밀고, 쉘코드 앞에 놉16byte를 추가 하였다.

1. **공격**



[놉16bte추가, ret주소 수정 후 공격 시도]

Pw : cantata