[Hack CTF] Basic\_BOF #1

텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명

Gdb로 basic\_bof 파일을 분석해본다.

사용된 함수 중 주목할 것은 fgets함수가 call되는 부분, cmp로 문자열 비교하는 부분과 system함수가 call되는 부분이다. 나머지는 그닥 신경쓰지 않아도 된다.

Main+17 부분을 보면 [ebp-0xc] 주소에 0x4030201의 값을 넣는다.

이후 바로 fgets함수의 인자들이 push되며 그 값이 저장되는 주소는 [ebp-0x34]위치이다.

처음에 값이 저장된 위치와 fgets로 읽는 값이 저장되는 위치 사이는 40byte의 공간이 있다.

근데 fgets로 44byte를 읽어오니 40byte를 채우고 [ebp-0xc]위치에 4byte를 덮을 수 있다.

이후 여기에 저장된 값이 0x4030201, 0xdeadbeef와 비교를 하여 함수들이 실행된다.

[ebp-0xc]의 값과 각 두 값이 같은지 보고 같다면 main+120으로 점프 둘 다 다르면 je 명령을 무시하고 진행한다. 근데 무시하고 진행해도 실행되는 건 puts함수로 사실상 여기서 값을 비교하는 건 무시해도 된다.

system함수가 실행되는 부분을 보면 위쪽에 [ebp-0xc]의 값과 0xdeadbeef를 비교하여 main+177로 점프할지 말지 정하는 부분이 있다.

Cmp를 통해 앞의 값과 뒤의 값이 같은지 확인하고 같다면 jne 명령을 무시하고 진행하게 된다. 즉 [ebp-0xc]에 0xdeadbeef를 덮어씌워야 한다.

Python pwntools 라이브러리를 이용해서 익스플로잇 할 것이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

접속 후 fgets에 40바이트 채우고 4바이트를 deadbeef로 채우기 위해 sendline함수를 써서 a로 40바이트 채우고 리틀엔디안임을 유의해서 서순에 맞게 deadbeef를 입력한다.

쉘이 실행되면 cat flag로 플래그 확인이 가능하다