Lazenca RTL(x86)

RTL (Return To LIBC)

- Return Address 영역에 공유 라이브러리 주소로 변경하여 해당 함수를 호출하는 방식
- 버퍼를 채우고, RET 영역에 LIBC (즉, LIBC 영역에서 공격자가 사용할 함수의 주소)를 넣어주는 것
- 이 기법으로 NX Bit (DEP)를 우회할 수 있음

  Lazenca - Technote

- 공격이 가능한 이유
  - 프로그램에서 printf( ) 라는 함수를 사용하고 있다고 하자.
  - 그리고 이 printf( )라는 함수는 공유 라이브러리인 /lib/libc.so.6 이라는 파일 안에 들어 있다.
  - 그렇다면, 이 프로그램을 실행시키기 위해서 (즉, printf( )를 사용하기 위해) LIBC (공유 라이브러리) 영역에 printf( )만이 아닌 /lib/libc.so.6 파일이 올라가게 된다.
  - 따라서 /lib/libc.so.6에서 우리가 사용할 함수의 주소를 RET에 넣게 되면 RTL 기법을 사용할 수 있는 것이다.
  - NX Bit (DEP)는 스택 안에서 공격 코드 실행을 막는 방어 기법이므로, LIBC 영역을 이용하는 RTL 기법으로 우회할 수 있다.

Cdecl (C언어의 함수 호출 규약)

Calling convention features

Aa 특징	설명
인자 전달 방법	Stack 이용
인자 전달 순서	오른쪽 → 왼쪽
함수의 반환 값	EAX
Stack 정리	호출한 함수가, 호출된 함수의 Stack 공간을 정리

```
int a, b, c, d;
int ret;
```

```
ret = function(a, b, c, d);
```

```
push    d
push    c
push    b
push    a
call    function
mov     ret, eax
```

위의 C Code는 Cdecl에 의해 아래와 같은 Assembly Code로 변환됨.

function(a, b, c, d)는 d부터 a로 인자가 push되는 것을 확인할 수 있으며, function의 return값은 eax에 저장되어 ret에 보내지는 것을 알 수 있음