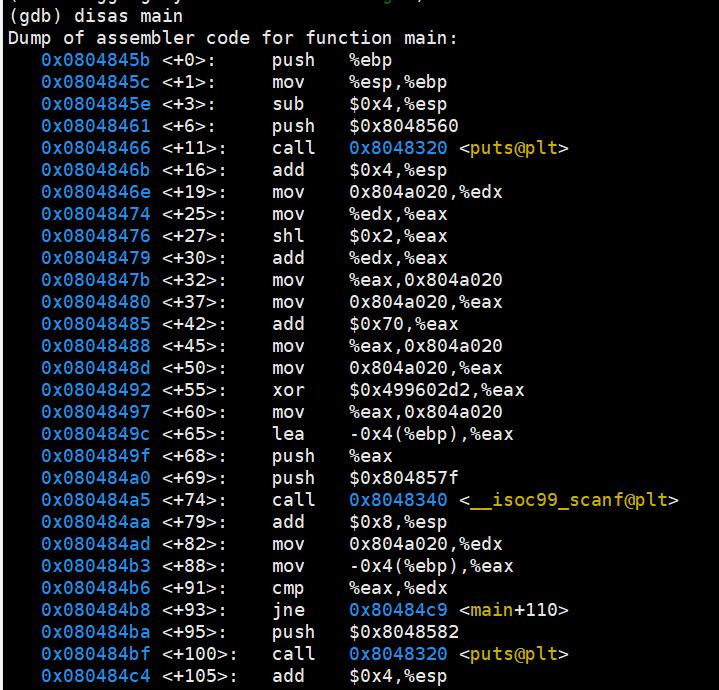
Stage1 실습

텍스트이(가) 표시된 사진

자동 생성된 설명

Gdb stage1 실행



Disas main을 통해 함수 내용을 봄.

scanf를 통해 답을 입력 / %eax에 저장 /

텍스트이(가) 표시된 사진

자동 생성된 설명

<+91>에서 cmp %eax %edx로 두개의 값을 비교하기 때문에 91번 지정주소에 브레이크를 검

X/s 를 통해 scanf에서 입력받은 값이 정수형이라는 것을 파악함

텍스트이(가) 표시된 사진

자동 생성된 설명

[ebp-0x4]의 주소가 eax에 저장됨

텍스트이(가) 표시된 사진

자동 생성된 설명

프로그램을 실행시켜서 랜덤인 정수 12345를 입력했음.

텍스트이(가) 표시된 사진

자동 생성된 설명

Edx의 값이 201821274 라는 것을 파악함.

텍스트, 장치, 게이지, 측정기이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

201821274를 입력하면 정답이 맞다는 것을 확인함

Stage2 실습

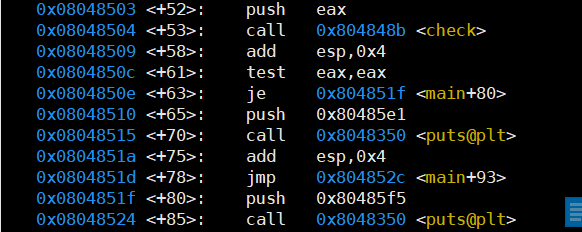


Gdb stage2를 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

Scanf로 받는 값이 정수형이라는 것을 파악



Main+53 의 check함수의 반환값으로 main+61의 test로 eax가 0인지 아닌지 확인함

텍스트이(가) 표시된 사진

자동 생성된 설명

Main+85에서 pw를 정확하게 입력할 시 정답이 출력된다는 것을 확인함

텍스트이(가) 표시된 사진

자동 생성된 설명

disas check를 통해 check 함수 내용을 어셈블리 코드로 봄

텍스트, 표지판이(가) 표시된 사진

자동 생성된 설명

Main+49에서 cmp로 비교하기에 브레이크 포인트를 걸었음

텍스트이(가) 표시된 사진

자동 생성된 설명

I r로 eax의 값이 980303 라는 것을 알았음

텍스트이(가) 표시된 사진

자동 생성된 설명

Stage3 실습



Gdb로 stage3 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

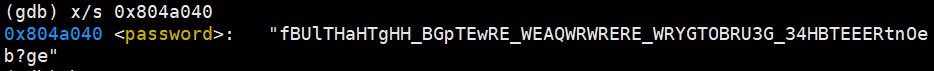
Password가 %s로 문자열이라는 것을 확인

텍스트이(가) 표시된 사진

자동 생성된 설명

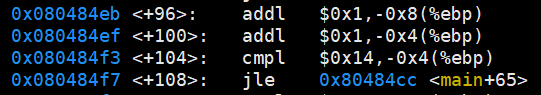
Main+85에서 eax에서 0x804a040만큼 떨어져 있는 곳의 값을 저장함

Main +79 ~ +83의 식을 통해 3의 배수에 있는 값들을 저장한다는 것을 확인



x/s를 통해 0x804a040을 확인함

맨 처음에 반복문이 돌면 0x804a040의 0번째인 f가 비밀번호임을 확인

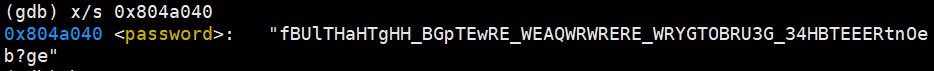


addl에서 $0x1로 1바이트씩 추가해서 저장한 후에 Main+108에서 main+65로 돌아가서 반복문을 진행함

텍스트이(가) 표시된 사진

자동 생성된 설명

Main +79 ~ +83의 식을 통해 3번째, 6번째 … 값을 가져옴



flag\_pw\_ARE\_YOU\_HERO?

비밀번호 확인



**#1 리눅스 특수권한 조사 및 실습**

**SetUID :** 나머지 사용자가 파일을 실행할 때 소유자의 권한으로 접근할 수 있게 해주는 권한

적용된 파일을 실행하는 동안 해당 파일의 소유자의 UID로 인식

일반적으로 실행 파일에 사용

소유자 권한 부분의 x 자리에 s로 표기

실행 권한이 없을 경우 대문자 S로 표시

해당 파일 소유자의 권한을 잠시 사용

**# chmod 4000 [파일명]**

**# chmod u+s [파일명] (제거 # chmod u-s [파일명])**

**<검색> # find / -perm -4000 -print**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Set-GID :** 실행하는 동안 해당 파일의 소유자 그룹 권한으로 인식

일반적으로 디렉터리에 설정

사용자가 파일이나 디렉터리를 생성하면 사용자가 속한 그룹에 상관없이 디렉터리 소유 그룹으로 만들어짐

그룹 권한 부분의 x 자리에 s로 표기

실행 권한이 없을 경우 대문자 S로 표시

**# chmod g+s [파일명]**

**# chmod 2777 [파일명]**

**<검색> # find / -perm -2000 -print**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Sticky-Bit :** 특정 디렉토리를 누구나 자유롭게 사용할 수 있게 하기 위해서 사용

일반 사용자들이 디렉터리에서 파일 및 디렉터리 생성 가능

자신이 생성한 것 이외에는 수정 및 삭제가 불가능

디렉터리에만 적용

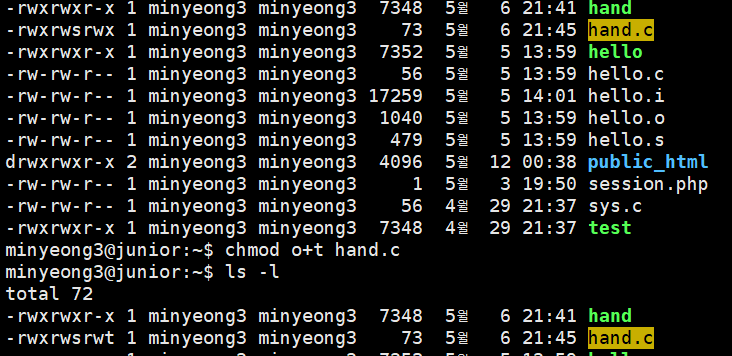
시스템 기본 폴더인 /tmp 디렉터리에 설정

설정하면 other 계층 권한 부분의 x 자리에 t 로 표기되며 실행권한이 없는 경우 대문자 T로 표기

**# chmod 1000 [파일명]**

**# chmod o+t [파일명] #chmod u+t [파일명]**

**<검색> # find / -perm -1000 -print**



**#2 함수 프롤로그, 에필로그 조사**

**프롤로그 = 스택 프레임 생성 = 함수 실행 준비 과정**

테이블이(가) 표시된 사진

자동 생성된 설명

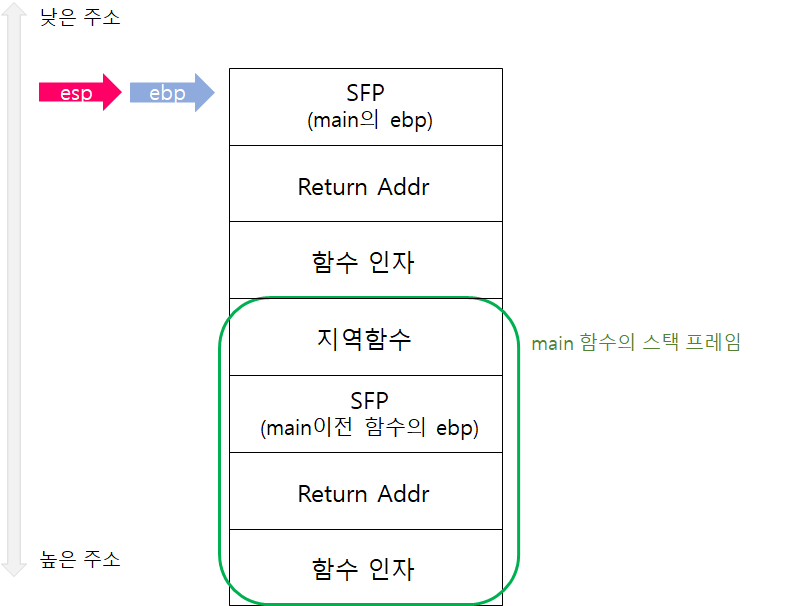
**1) push ebp : 함수가 종료된 후 ebp를 이전 함수의 ebp로 재설정하기 위해 스택에 이전 함수의 ebp를 push함**

**2) mov ebp, esp : 호출된 함수의 시작을 알리기 위해 현재 esp 값을 ebp에 복사함.**

→ 프롤로그를 통해 스택프레임을 형성하고 형성된 스택프레임을 통해 함수를 구분지을 수 있게함.

Esp가 가리키는 곳을 ebp도 가리키게 됨.

**에필로그**

****

**함수의 기능을 수행하고 난 후, 호출한 곳으로 돌아오기 위해 스택을 복원하는 과정**

**1) leave : 함수를 종료하고, ebp를 이전함수의 ebp로 재설정해주는 과정 (프롤로그와 반대로 진행)**

**1- mov esp, ebp :** 함수를 종료하기 위해 ebp값을 esp에 복사

**2- pop ebp :** 현재 esp가 가리키고 있는 sfp를 pop하면 이전 함수(현재 함수를 호출했던 함수)로 돌아갈 수 있게됨

**2) ret : 스택에 미리 저장해놓았던 return address를 eip에 넣어두고(pop) 그 주소로 이동해 (jmp) 함수를 호출한 후의 명령을 계속 수행할 수 있도록 해주는 과정**

**1 - pop eip :** esp가 가리키는 return 주소를 eip에 넣어 다음에 실행할 명령어를 지정해줌

**(pop: 스택에 가장 최근에 들어간 데이터를 꺼내 eip에 저장)**

**2 – jmp eip :**  다음에 실행할 명령어로 이동.

**(특정한 곳으로 분기한다, 점프)**

**★ 중요한 레지스터 암기**

**Ax: 산술 연산에 사용 및 함수 리턴 값에 쓰임**

**Sp: 스택의 최상단을 가리키는 포인터**

**Bp: 스택의 베이스를 가리키는 포인터**

**Ip: 다음에 실행되어야할 명령어의 주소를 가리킴**

**# 공유 라이브러리 만들기 및 실습**

1) 소스 코드인 c파일 생성 – vi test.c

텍스트이(가) 표시된 사진

자동 생성된 설명

2) 소스코드를 오브젝트 파일로 바꿔줌 - gcc -c test.c

Ls를 하면 test.o 라는 오브젝트 파일이 생긴 것을 확인

텍스트이(가) 표시된 사진

자동 생성된 설명

3) 오브젝트 파일을 .so 형식인 공유 라이브러리로 바꿔줌 - gcc -shared -o libtest.so test.o

-o 뒤의 인자는 반드시 lib”소스코드이름”.so 형태여야함

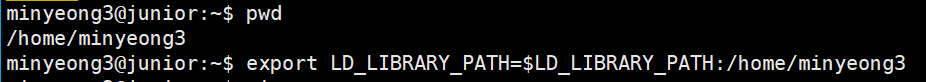
실행파일인 초록색을 나타내는 libtest.so가 생겼음을 확인

텍스트이(가) 표시된 사진

자동 생성된 설명

4) LD\_LIBRARY\_PATH 를 이용해 라이브러리 경로를 추가함 –

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/home/minyeong3



5) sum.c 라는 이름의 소스코드 컴파일 – vi sum.c

이때 맨처음에 만든 소스코드 속 add 함수를 포함해서 코드 작성

텍스트이(가) 표시된 사진

자동 생성된 설명

6) 파일 컴파일 - gcc -o sum sum.c -l file1 -L .

-ㅣ 옵션은 컴파일시에 포함할 동적 라이브러리 이름, - L 옵션은 경로를 지정



7) 파일 실행 - ./sum

실행파일인 초록색을 나타내는 sum이 생겼음을 확인.

텍스트이(가) 표시된 사진

자동 생성된 설명