5주차 3차시. 리버스 엔지니어링 공격

【학습목표】

1. 리버스 엔지니어링 공격에 대한 개념을 알고, 바이너리 파일 수정을 통한 리버스 엔지니어링 공격과 프로그램의 검증 로직을 우회하는 공격에 대해 설명할 수 있다.

학습내용1 : 바이너리 파일 수정을 통한 리버스 엔지니어링 공격

1. 주제

바이너리 파일 수정을 통한 리버스 엔지니어링 공격

2. 참고

· 한빛미디어
· 정보 보안 개론과 실습: 시스템 해킹과 보안
· 257페이지
· 실습 5-2. 바이너리 파일 수정을 통해 리버스 엔지니어링 공격하기

3. 실습에 사용할 코드

REVERSE_2.exe

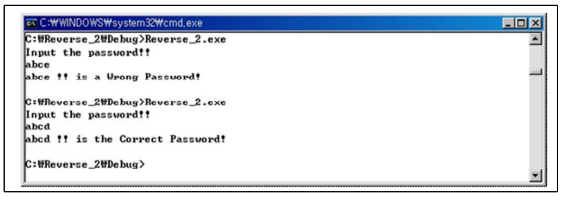
```
#include "stdafx.h"
#include "stdio.h“
#include "string.h“

int main(int argc, char* argv[])
{
        char password[12] ;
        char Correct_password[12] = "abcd";

        printf("Input the password!!₩n");
        scanf("%s",&password);
        if (strcmp(password,Correct_password)==0)
            printf("%s !! is the Correct Password!₩n",password);
        else
            printf("%s !! is a Wrong Password!₩n",password);
        return 0;
}
```

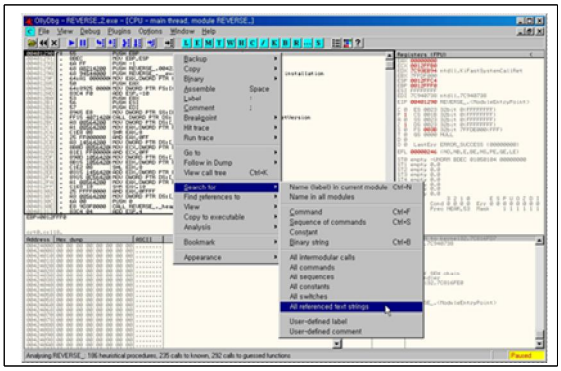
4. 실습용 테스트 파일 생성

· REVERSE_2.exe실행하면 패스워드를 입력을 요구
· abcd 입력 - 틀리면 '~ !! is a Wrong Password!' 출력- 맞으면 'abcd !! is the Correct Password!' 출력



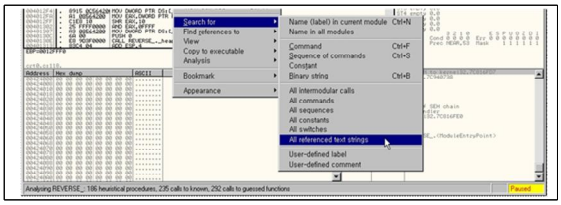
5. 올리 디버거로 파일 열기

[File]-[Open]에서 Reverse_2.exe 파일 열기

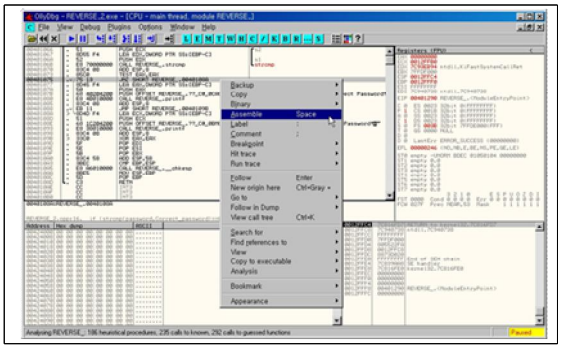


6. 리버스 엔지니어링 포인트 찾기

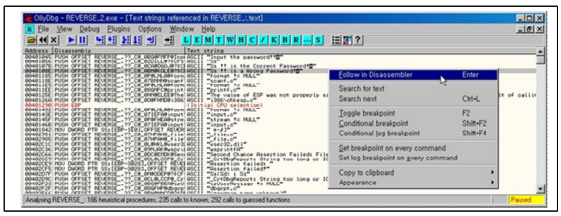
* 잘못된 패스워드 입력시 '~ !! is a Wrong Password!'메시지 출력하는 메시지 출력 지점을 시작점으로 검증 로직을 검색
* 올리 디버거 어셈블리어 영역에서 마우스 오른쪽 버튼 클릭
* [Search for]-[All referenced text strings] 메뉴 실행

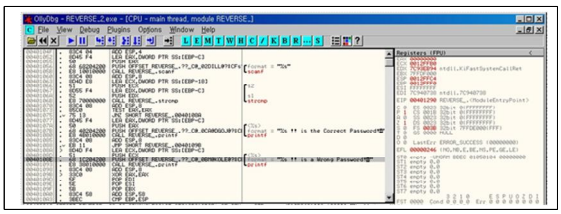


* [Search for text] 메뉴 : 파일이 큰 경우 사용



* 해당 문자열에서 마우스 오른쪽 버튼 클릭 [Follow in Disassembler] 메뉴 선택
* 어셈블리어 코드 확인





7. 프로그램 변경 포인트 찾기

순번	주소	어셈블리어 코드와 주석	
1	0040106B	CALL REVERSE_.strcmp	; Wstrcmp
2	00401070	ADD ESP,8	
3	00401073	TEST EAX,EAX	
4	00401075	JNZ SHORT REVERSE_.0040108A	
5	00401077	LEA EAX,DWORD PTR SS:[EBP-C]	
6	0040107A	PUSH EAX ;	/<%s>
7	0040107B	PUSH OFFSET REVERSE_.??_C@_0CA@DGOJ@?$CF>: |format = "%s !! is the Correct Password!"	

순번	주소	어셈블리어 코드와 주석	
8	00401080	CALL REVERSE_.printf ;	Wprintf
9	00401085	ADD ESP,8	
10	00401088	JMP SHORT REVERSE_.0040109B	
11	0040108A	LEA ECX,DWORD PTR SS:[EBP-C]	
12	0040108D	PUSH ECX ;	/<%s>
13	0040108E	PUSH OFFSET REVERSE_.??_C@_0BM@KOLE@?$CF>: |format = "%s !! is a Wrong Password!"	
14	00401093	CALL REVERSE_.printf ;	Wprintf

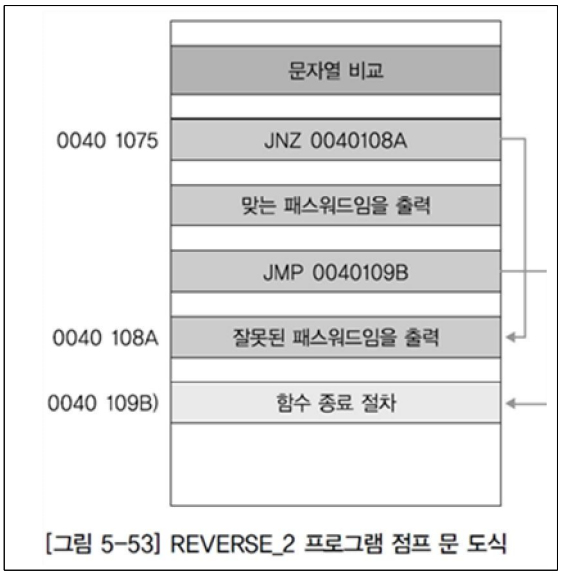
순번	주소	어셈블리어 코드와 주석
15	00401098	ADD ESP,8
16	0040109B	XOR EAX,EAX
17	0040109D	POP EDI
18	0040109E	POP ESI

· (1)번의'CALL REVERSE_.strcmp'부터 (3)번까지 : 어떤 값을 비교
· (4)번, (10)번 : 점프문
· (4)번 점프 주소 : 0040108A, 이 값은 (11)번 주소 값
· (10)번 점프 주소 : 0040109B, 이 값은 (16)번 주소 값
· 문자열을 비교(어셈블리어코드(1)~(3))
· ZF값 0이 아니면 0040108A 주소로 점프
· 0040108A 주소에'%s !! is a Wrong Password!' 출력 구문(어셈블리어 코드 (11)~(14))

· ZF 값이 0 이면 맞는 패스워드 출력 (어셈블리어 코드 (5)~(8)) 0040109B로 점프

· 함수종료

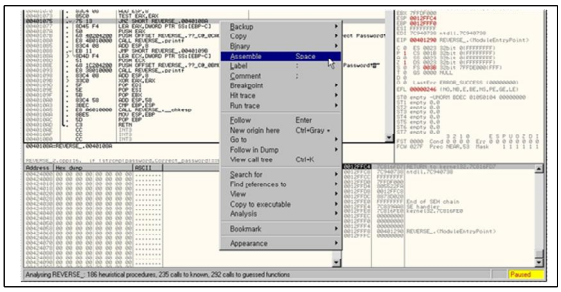
· JNZ 다음의 주소를 00401077로 바꾸면 ZF 값에 관계없이 문자열 비교 결과와 상관없이 항상 맞는 패스워드 출력, 함수 종료



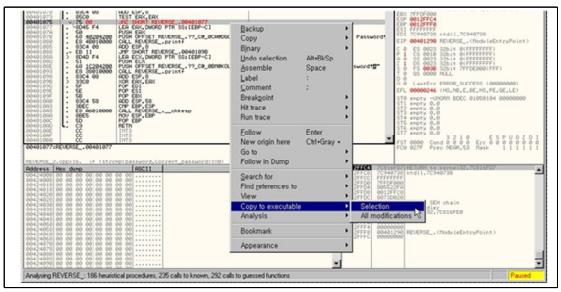
[그림 5-53] REVERSE_2 프로그램 점프 문 도식

8. 프로그램 변경하기

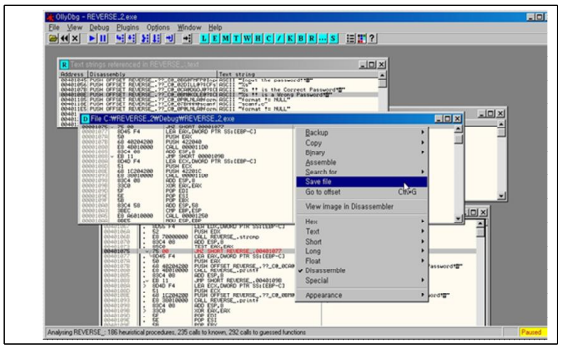
* 변경할 라인에서 마우스 오른쪽 버튼 클릭 [Assemble] 메뉴 선택
* JNZ 목적 주소지 00401077로 바꾼 뒤 <Assemble> 버튼 클릭 팝업 창 닫기



* 해당 라인에서 마우스 오른쪽 버튼 클릭 [Copy to executable]-[Selection] 메뉴 선택
* 마우스 오른쪽 버튼 클릭 [Save File] 메뉴로 파일을 다른 이름으로 저장

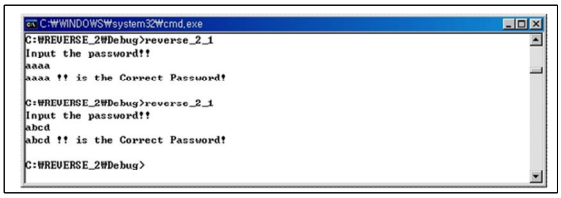


9. 변경된 프로그램 저장하기



10. 결과 확인하기

* 저장한 파일 실행



학습내용2 : 프로그램의 검증 로직을 우회하는 공격

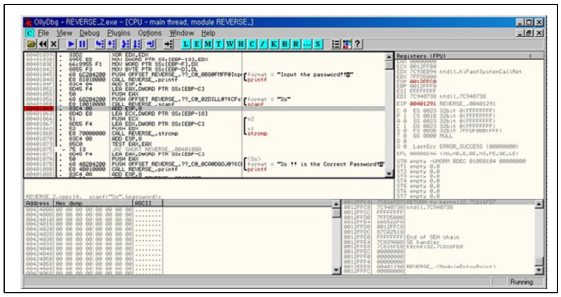
1. 주제

프로그램의 검증 로직을 우회하는 공격

2. 참고

· 한빛미디어
· 정보 보안 개론과 실습: 시스템 해킹과 보안
· 264페이지
· 실습 5-. 프로그램 로직 분석을 통해 리버스 엔지니어링 공격하기

3. 리버스 엔지니어링 포인트 찾기



4. 어셈블리어 코드와 주석

순번	주소	어셈블리어 코드와 주석
1	00401045	PUSH OFFSET REVERSE_.??_C@_0BG@FMFP@Inpu): /format =" Input the password!!"
2	0040104A	CALL REVERSE_.printf : Wprintf
3	0040104F	ADD ESP,4
4	00401052	LEA EAX,DWORD PTR SS:[EBP-C]
5	00401055	PUSH EAX
6	00401056	PUSH OFFSET REVERSE_.??_C@_02DILL@?$CFs?): /format = "%s"
7	0040105B	CALL REVERSE_.scanf : Wscanf

순번	주소	어셈블리어 코드와 주석
8	00401060	ADD ESP,8
9	00401063	LEA ECX,DWORD PTR SS:[EBP-18]
10	00401066	PUSH ECX : /s2
11	00401067	LEA EDX,DWORD PTR SS:[EBP-C] : |
12	0040106A	PUSH EDX : |s1
13	0040106B	CALL REVERSE_.strcmp : Wstrcmp
14	00401070	ADD ESP,8

순번	주소	어셈블리어 코드와 주석
15	00401073	TEST EAX,EAX
16	00401075	JNZ SHORT REVERSE_.0040108A

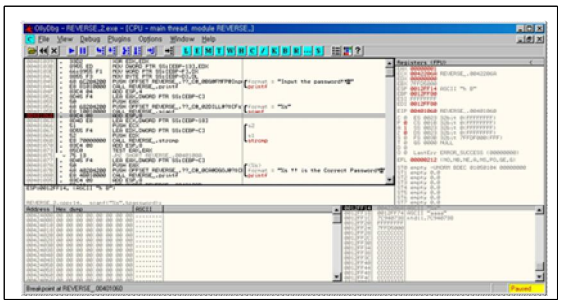
5. 로직 분석

- (1), (2)번 프로그램 실행 시 출력되는 'Input the password!!' 출력
- (3) ~ (8) 입력된 패스워드 스택 저장
- (9)~(10)번 SS : [EBP-18] 주소 값 ECX에 저장
- (11)~(12)번 SS : [EBP-C] 주소 값 EDX에 저장, (13)번에서 strcmp 함수 호출
- ECX에 저장된 SS : [EBP-18] 값과 EDX에 저장된 SS : [EBP-C] 값을 비교하는 것임

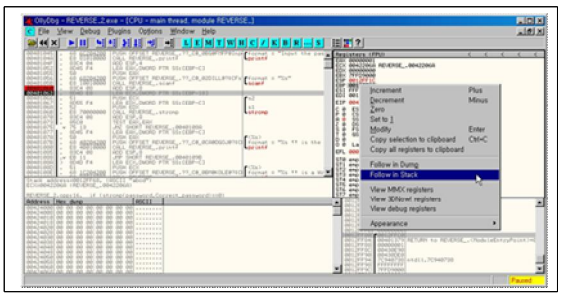
6. 로직 확인

* 00401060 주소에 브레이크 포인트 설정 → F9 눌러 REVERSE_2.EXE 파일 올리 디버거에서 실행
* DOS 창에서 'Inputthe password!!' 메시지 확인

* aaaa를 입력하고 Enter
올리 디버거에서 브레이크 포인트에서 정지 확인



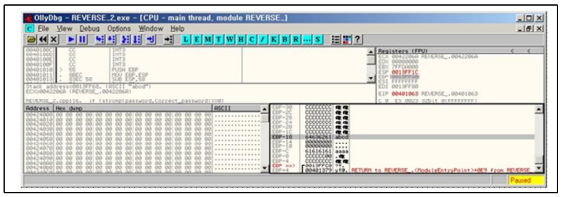
* 현재 EBP는 0x0012FF00, F7 눌러 한 단계 진행, 레지스터 창에서 EBP 선택 → 마우스 오른쪽 버튼 클릭→ [Follow in Stack] 메뉴 선택 스택 내용 확인



* 스택의 각 라인에서 마우스 오른쪽 버튼 눌러 [Show ASCII dump] 메뉴 선택

* [Address]-[Relative to EBP]'를 선택

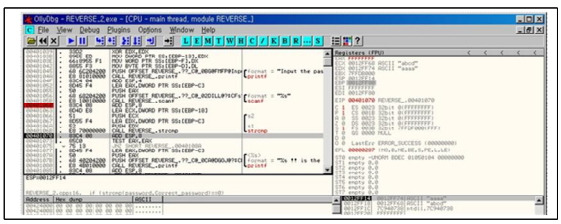
* EBP 기준으로 아스키로 스택의 내용 바뀌 내용 확인



* 'EBP-C' 문자열과 'EBP-18' 문자열 비교, 패스워드 적절성 여부 검사

* 'aaaa'는 테스트용으로 넣은 문자열,'abcd'가 맞는 문자열

* 다시 실행하여'abcd' 입력



【학습정리】

1. 리버스 엔지니어링 공격은 로직을 파악하여 해당 로직을 이용할 수 있고 특정 로직을 우회하도록 프로그램을 변경함으로써 이루어진다.

2. 리버스 엔지니어링에 대한 대응책으로는 패킹(Packing), 안티 디버깅, 타이밍 체크, 쓰레기 코드(Garbage Code) 넣기와 코드 치환(Permutation)이 있다.