0529 발표 스크립트

scp 부원
20학번 박준형

[1페이지]
안녕하십니까? abex crackme에 대해 발표하게 될 박준형입니다.

[2페이지]
우선 목차를 설명드리겠습니다. 첫 번째로 crackme의 정의, 두 번째로 기본적인 ollydbg 명령어에 대해서 설명드리고, 세 번째로 abex′ crack 실습, 마지막으로 Q&A 받고 마무리하겠습니다.

[3페이지]
(Part 1) 우선 Crackme란 무엇인지 설명드리겠습니다.

[4페이지]
crackme란 프로그램은 말 그대로 크랙 연습 목적으로 작성되어 공개된 프로그램입니다. 리버싱을 처음 시작할 때 간단한 crackme를 분석해 보면 실제로 디버거와 디스 엇셈 코드에 익숙해질 수 있습니다.

[5페이지]
(Part 2) 본격적인 실습 전 이번 예제에서 사용되는 기본적인 Ollydbg 명령어에 대해 설명드리겠습니다.

[6페이지]
본격적인 코드 분석 전에 코드에서 사용된 어셈블리 명령어를 설명 해드리겠습니다. PUSH는 스택에 값을 입력하는 명령어이고. CALL은 지정된 주소의 함수를 호출, INC는 Increase의 약자로 값을 1 증가, DEC는 Decrease의 약자로 값을 1 감소하라는 뜻입니다. JMP는 지정된 주소로 점프, CMP는 주어진 두 개의 피연산자 비교, JE는 조건 분기(Jump if equal)입니다.
[CMP: *SUB 명령어와 동일하나 operand 값이 변경되지 않고 EFLAGS 레지스터만 변경됨(두 opreand의 값이 동일하다면 SUB 결과는 0이고 ZF = 1로 세팅됨.) *ZF = 1 이면 점프
JE : 제로 플래그가 1인 경우에만 분기하는 조건 분기 명령어입니다. 즉, CMP 명령어로 비교한 결과가 동일하다면(제로 플래그가 1인 경우), JE 명령어에 의해 분기가 수행됩니다. 분기는 지정된 주소로 이동하여 프로그램의 실행 흐름

을 변경하는 것을 의미합니다.]

[7페이지]
다음은 동적분석에 해당하는 ollydbg로 간단한 예제 실습을 해보겠습니다.

[8페이지]
(Part3) 우선, 디버깅을 시작하기 전에 파일을 실행시켜서 어떤 프로그램인지 살펴봅니다. 첫 번째 팝업 창에서 리버스(사용자)가 무엇을 해야 할지 알려주고 있습니다. 즉, 이번 실습의 목적은 프로그램을 실행했을 때 HD(하드디스크)가 CD롬으로 인식되도록 변경하는 것입니다. 문제를 해결하는 방법은 다양하지만 저는 프로그램 코드를 변경해서 문제를 해결해 봤습니다.

[9페이지]
Ollydbg에서 파일을 열어서 디스어셈코드를 살펴보겠습니다.
0041044를 보면 저희가 원하는 메시지 박스 텍스트도 보입니다.(Ok, I really think that your HD isaCD-ROM!:p"

[10페이지]
EP 코드가 매우 짧은 걸 볼 수 있는데, 이를 통해서 abexcrackme 파일이 어셈블리 언어로 만들어진 실행 파일이라는 것을 알 수 있습니다. 개발 툴을 사용하지 않고 어셈블리 언어로 작성하면 어셈코드가 곧 디스어셈코드가 되기 때문입니다.
(개발 툴을 사용하면 자신이 작성한 소스코드 외에 컴파일러가 Stub Code를 추가시키기 때문에 디스어셈을 하면 복잡하게 보임)
*Stub code(스텁코드)는 프로그램의 소스 코드에서 자동으로 추가되는 작은 코드 조각을 의미합니다. 이러한 스텁코드는 일반적으로 컴파일러나 개발 도구에 의해 자동으로 생성되며, 주로 소스 코드의 다른 부분과의 인터페이스를 관리하거나, 외부 라이브러리나 함수와의 상호 작용을 도와줍니다.

[11페이지]
F8(StepOver)로 하나씩 넘기다 보면 아까 실행시켰을 때 봤던 처음 메시지 화면이 실행됩니다. 계속 F8 넘어가다 보면,,

[12페이지]

00401036주소에서 두 번째 메시지가 실행됩니다. 즉 이 두 개의 코드 사이를 조작하면 저희가 원하는 결과 메시지 창을 얻을 수 있다는 것을 알 수 있습니다.

[13페이지]
0041024 주소의 줄을 보면 CMP EAX, ESI라고 써져있습니다.
CMP EAX, ESI는 EAX와 ESI를 비교(CMP) 하여 같으면(제로 플래그= 1) 0040103D로 넘어가고, 다르면 그냥 밑 명령어 (JE SHORT 0040103D)를 실행시키라는 뜻입니다. JE는 이 연산자들의 값이 같으면(제로 플래그= 1이면) 점프하라고 하는 일종의 조건문입니다.

[14페이지]
GetDriveTypeA() 함수 실행이 끝난 상태에서 레지스터 정보를 살펴보면 EAX 레지스터에는 3이 저장되어 있고, ESI 레지스터에는 0이 저장되어 있습니다. 레지스터를 변경하는 명령어를 보면 ESI 레지스터는 모두 3만큼 증가시키고. EAX 레지스터는 2만큼 감소시킵니다. ESI 레지스터를 증가시키는 명령어 2개 (40101E 줄, 40101F 줄에 있는 INC ESI) 2개를 실행하지 않으면 EAX와 ESI 값이 동일해집니다.

[15페이지]
0040101D 줄 JMP 명령어를 아래 두 줄 INC ESI와 DEC EAX 가 실행되지 않게 수정하면 문제를 해결할 수 있습니다.

[16페이지]
INC ESI와 DEC EAX 명령어가 실행되지 않게 하려면 JMP SHORT 00401021 명령어를 JMP SHORT 00401023로 바꾸면 됩니다.

[17페이지]
코드 수정을 완료하고 F8(스텝 오버)를 눌러서 EAX 값과 ESI 값이 1이 되므로 제로 플래그가 둘 다 1이 되어서 저희가 원하는 메시지 창을 띄울 수 있습니다. 결과를 확인해 보면 "Ok, I really think that Your HD is a CD-ROM! :p"라는 메시지를 출력하는 창을 볼 수 있습니다.
여기서 주의할 점은 프로그램 실행 위치가 변경하고자 하는 코드 위에 있어야 한다는 것입니다. 그래야만 코드 변경 후 실행해서 동작을 확인할 수 있습니

다.

[18페이지]
Q&A 받겠습니다!