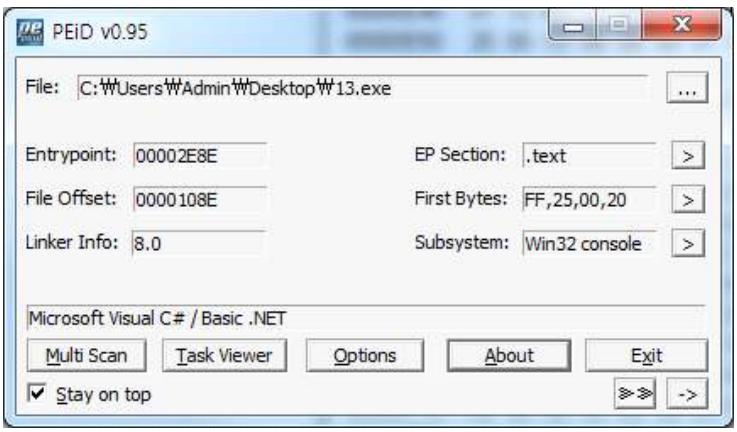
Challenges : Basic 13

Author : Basse 2002

Korean :
정답은 무엇인가

English :
Find the answer

엄청 간소하다 -_-; 뭐지??
일단 순서대로 절차를 밟자
PEID



C#, 베이직, 닷넷 음음...

PEVIEW

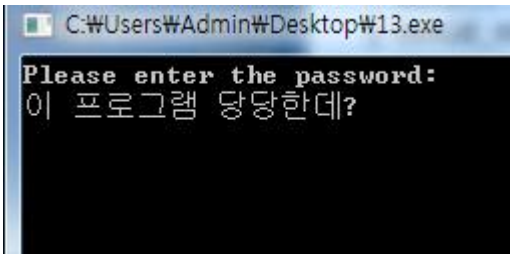
```
E.!..7P.l.e.a.s.
e. .e.n.t.e.r. .
t.h.e. .p.a.s.s.
w.o.r.d.:. ..5W.
e.l.l. .D.o.n.e.
!. .Y.o.u. .c.r.
a.c.k.e.d. .i.t.
!..)B.a.d. .L.u.
c.k.!. .T.r.y. .
a.g.a.i.n.!.....
```

영어 문구가 보인다 음음...

혹시 답?? 인가 싶어 넣어 봤지만 아니다...

```
    ''
..........Micros
oft..  ...Copyrig
ht  ..  Microsoft
2008..)..$d78e10
b0-28c4-4f52-858
7-c18c8b7a10bf..
....1.0.0.0.....
................
```

실행 화면 (당당하게 답을 요구한다 짜아식)



특이한 놈이다 올리디 올렸는데 이상한 주소로 가버린다 -_- (시스템 영역이라고 합니다)
노가다를 시전해야 겠다.

```
lity.......1B.n.
C.x.G.i.N.4.a.J.
D.E.+.q.U.e.2.y.
l.m.8.Q.=.=...^.
F.7.9.e.j.k.5.6.
$....D.H.j.4.7.
&.*.).$.h...M.D.
5..!&.!...$.%.^.
&.*.(.).C.v.H.g.
E.!..7P.l.e.a.s.
e. .e.n.t.e.r. .
t.h.e. .p.a.s.s.
w.o.r.d.:. ..5W.
e.l.l. .D.o.n.e.
!. .Y.o.u. .c.r.
a.c.k.e.d. .i.t.
```

요기 뭔가 있을꺼 같기도 한데...

그레고리안 달력을 발견했다 신세계다 -ㅅ-;;; 러시아도 찾았다....(미치기 시작한다...)

```
UNICODE "ss"
UNICODE "(Russia)"
ASCII "2"
UNICODE "Gregorian Calendar ["
UNICODE "risian"
```

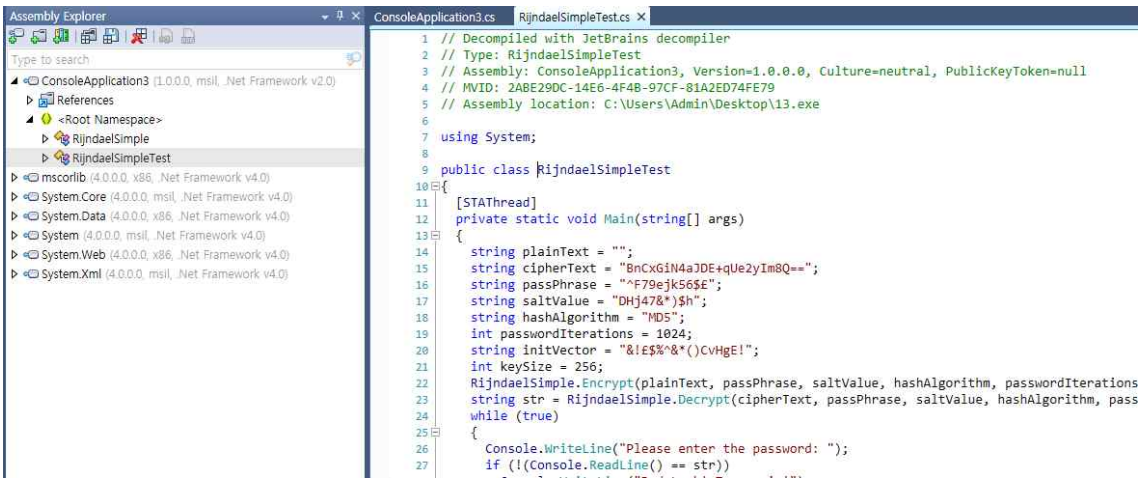
```
75B9B941   8945 10          MOV DWORD PTR SS:[EBP+10],EAX
75B9B944   85C0             TEST EAX,EAX
75B9B946   0F84 0CF90100    JE 75BBB258                          75BBB258
75B9B94C   E9 EDF80100      JMP 75BBB23E                         75BBB23E
75B9B951   8945 0C          MOV DWORD PTR SS:[EBP+C],EAX
75B9B954   F6C3 08          TEST BL,8
75B9B957   0F85 A1F80100    JNZ 75BBB1FE                         75BBB1FE
75B9B95D   E9 A4F80100      JMP 75BBB206                         75BBB206
75B9B962   FF75 10          PUSH DWORD PTR SS:[EBP+10]
75B9B965   8D45 E8          LEA EAX,DWORD PTR SS:[EBP-18]
75B9B968   50               PUSH EAX
75B9B969   FFD7             CALL EDI
75B9B96B   8BF3             MOV ESI,EBX
75B9B96D   83E6 42          AND ESI,42
75B9B970   74 09            JE SHORT 75B9B97B                    75B9B97B
75B9B972   F6C3 20          TEST BL,20
75B9B975   0F85 32090000    JNZ 75B9C2AD                         75B9C2AD
75B9B97B   8D45 FC          LEA EAX,DWORD PTR SS:[EBP-4]
75B9B97E   50               PUSH EAX
75B9B97F   53               PUSH EBX
75B9B980   8D45 E8          LEA EAX,DWORD PTR SS:[EBP-18]
75B9B983   50               PUSH EAX
75B9B984   8D45 F0          LEA EAX,DWORD PTR SS:[EBP-10]
75B9B987   50               PUSH EAX
75B9B988   E8 48100000      CALL 75B9C9D5                        75B9C9D5
75B9B98D   8BF8             MOV EDI,EAX
75B9B98F   85FF             TEST EDI,EDI
EAX=00414414, (UNICODE "C:\Users\Admin\Desktop;C:\Windows\system32;C:\Windows\system;C:\Windows;.;C:\Windows\system32;C:\Win")
Stack SS:[0031F6DC]=00000000
```

모르겠다... -ㅅ-;; 풀이를 참조해야 될꺼 같다... 윽....
풀이를 보면 리버싱이 안된다는데 나는 되는데? 뭐시여?

디컴파일러를 받아야 한다고 한다.. http://www.jetbrains.com/decompiler/

설치 과정 생략하고 옵션 들어가서 글자크기 좀 바꾸고 라인넘버 나오게 설정했습니다.

암호푸는 코드랑 암호거는 코드 그리고 메인 코드가 나옵니다



메인코드 입니다

그리고 References 밑에 SimpleTest 부분을 클릭해 줍니다.

아 제가 코딩은 참 허접한데 자바가 갑자기 생각나네요 -_-;;
아는 단어 끄적여 보면 (확실하지 않습니다)
plainText - 평문
cipherText - 암호화된 문장
passPhrase - 복호화 같습니다.
hashAlgorithm - 해쉬 알고리즘은 MD5를 쓰네요
keySize 는 256 그 외는 대충 아실껍니다.

만약 소스를 수정해서 해결하는 거라면 밑에 소스를 참조하고 바꾸면 해결이 된다고 합니다...

// 암호화 거는 소스
using System;
using System.Collections.Generic;
using System.Linq;

```
using System.Text;
using System.IO;
using System.Security.Cryptography;

public class RijndaelSimple
{
    // Methods
    public RijndaelSimple()
    {
    }
    public static string Decrypt(string cipherText, string passPhrase, string saltValue, string hashAlgorithm, int passwordIterations, string initVector, int keySize)
    {
        byte[] bytes = Encoding.ASCII.GetBytes(initVector);
        byte[] rgbSalt = Encoding.ASCII.GetBytes(saltValue);
        byte[] buffer = Convert.FromBase64String(cipherText);
        byte[] rgbKey = new PasswordDeriveBytes(passPhrase, rgbSalt, hashAlgorithm, passwordIterations).GetBytes(keySize / 8);
        RijndaelManaged managed = new RijndaelManaged();
        managed.Mode = CipherMode.CBC;
        ICryptoTransform transform = managed.CreateDecryptor(rgbKey, bytes);
        MemoryStream stream = new MemoryStream(buffer);
        CryptoStream stream2 = new CryptoStream(stream, transform, CryptoStreamMode.Read);
        byte[] buffer5 = new byte[buffer.Length];
        int count = stream2.Read(buffer5, 0, buffer5.Length);
        stream.Close();
        stream2.Close();
        return Encoding.UTF8.GetString(buffer5, 0, count);
    }

// 암호문 푸는 소스
 public static string Encrypt(string plainText, string passPhrase, string saltValue, string hashAlgorithm, int passwordIterations, string initVector, int keySize)
    {
        byte[] bytes = Encoding.ASCII.GetBytes(initVector);
        byte[] rgbSalt = Encoding.ASCII.GetBytes(saltValue);
        byte[] buffer = Encoding.UTF8.GetBytes(plainText);
        byte[] rgbKey = new PasswordDeriveBytes(passPhrase, rgbSalt, hashAlgorithm, passwordIterations).GetBytes(keySize / 8);
        RijndaelManaged managed = new RijndaelManaged();
        managed.Mode = CipherMode.CBC;
        ICryptoTransform transform = managed.CreateEncryptor(rgbKey, bytes);
        MemoryStream stream = new MemoryStream();
```

```
        CryptoStream stream2 = new CryptoStream(stream, transform,
CryptoStreamMode.Write);
        stream2.Write(buffer, 0, buffer.Length);
        stream2.FlushFinalBlock();
        byte[] inArray = stream.ToArray();
        stream.Close();
        stream2.Close();
        return Convert.ToBase64String(inArray);
    }
}
```

그리고 메인 소스

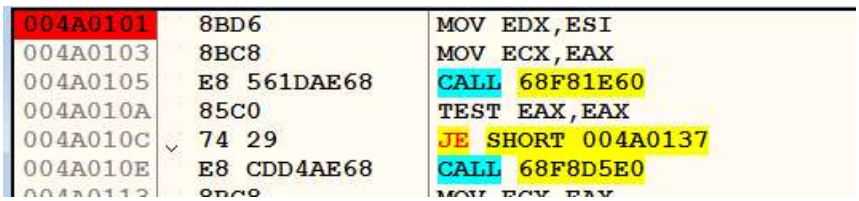
```
public class RijndaelSimpleTest
{
    // Methods
    public RijndaelSimpleTest()
    {
    }
    [STAThread]
    private static void Main(string[] args)
    {
        string plainText = "";
        string cipherText = "BnCxGiN4aJDE+qUe2yIm8Q==";
        string passPhrase = "^F79ejk56$\x00a3";
        string saltValue = "DHj47&*)$h";
        string hashAlgorithm = "MD5";
        int passwordIterations = 0x400;
        string initVector = "&!\x00a3$%^&*()CvHgE!";
        int keySize = 0x100;
        RijndaelSimple.Encrypt(plainText, passPhrase, saltValue, hashAlgorithm,
passwordIterations, initVector, keySize);
        plainText = RijndaelSimple.Decrypt(cipherText, passPhrase, saltValue, hashAlgorithm,
passwordIterations, initVector, keySize);
    Label_0056:
        Console.WriteLine("Please enter the password: ");
        if (Console.ReadLine() == plainText)
        {

            Console.WriteLine("Well Done! You cracked it!");

            Console.ReadLine();
        }
        else
```

```
        {
            Console.WriteLine(plainText);
            goto Label_0056;
        }
    }
}
```

우아 대단하다... -_-;;;
솔직히 소스는 추가 된 부분만 눈으로 확인하고 돌려보지는 않았습니다....

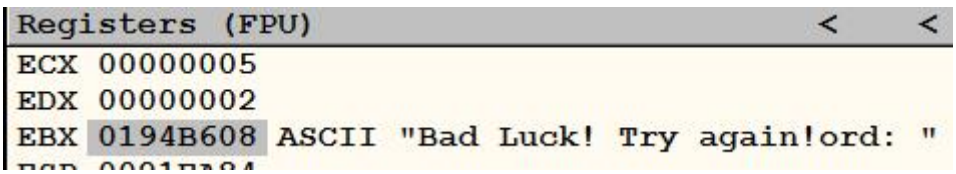
그러면 방법은 다시 올리디버거!!! (괜찮아 잘 될 거야~~) 젠장
어떻게든 리버싱으로 풀어보고 싶었습니다....
여기서 부터는 제 맘대로 풀었습니다.

계속 진행해다가 보시면 이 코드 부분이 나오는데 이 부분에서 레지스터안에 들어있는 값이
답일 꺼라고 예측하고 답을 계속 넣었지만 실패했습니다.....
그리고 이 코드 부분이 답이 틀렸다고 문구가 나오기 바로 직전의 함수 콜 부분입니다...
입력된 답이랑 비교하는 부분이 겠죠...

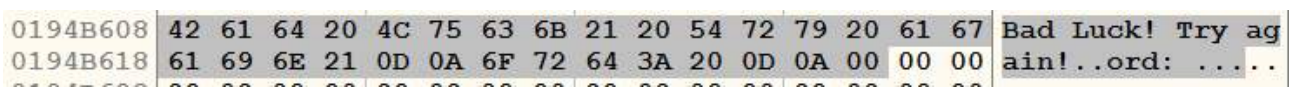


(많은 부분이 생략 되었지만 주소가 계속 왔다 갔다 합니다. BP걸어서 계속 접근 하시길 바랍니다)

계속 틀리다가 혹시 이거 답이 레지스터안에 들어간 것이 아닌게 아닐까라는 생각이 문득 들었습니다.
그래서 스택이랑 덤프를 미친 듯이 뒤졌구요



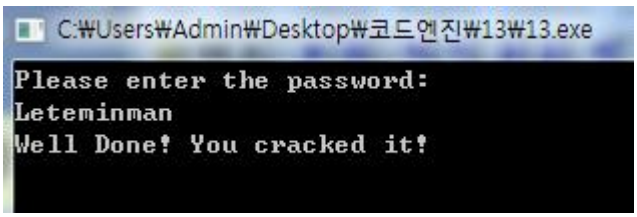
조금 더 진행하다가 보면
아스키 Bad Luck 부분에 덤프걸고 따라가 봅니다
혹시 그 주위에 답이 맞았다는 글이 있을지 모르겠다는 생각을 해봤습니다.



```
0194ABA8  0A 00 00 00 4C 00 65 00 74 00 65 00 6D 00 69 00  ....L.e.t.e.m.i.
0194ABB8  6E 00 6D 00 61 00 6E 00 00 00 00 00 00 00 00 00  n.m.a.n.........
```

덤프 영역에서 계속 위로 올라 가다보면 너무 깔끔하게 단어가 따로 들어간 영역을 찾을 수 있습니다
혹시 이건가 하고 실행시켜봤는데 -_-;; 두근두근....

와 대박...
이 맛에 리버싱 하는구나 생각했습니다 -_-b



보충 설명

종류	설명
Software Breakpoint	**소프트웨어로 Break point설정,** ollydbg, windbg -> 디버깅프로그램 통해서 지원하는 기능, F2 키를 눌러서 표시 break point 수행 위치의 **1byte를 기계어 코드 CC (interupt신호)로 변경후 해당 하는 break pointlist라는 곳에 번지를 표시** ∘ (세상의 모든 cc가 break point는 아니기때문에) ∘ CC로 변경시에는 **원래 코드는 백업** 시킨다 . ▪ **CC만나면 break point list확인후 없으면 원래 CC코드에 백업한 코드를 원상복귀** 시킨다. **장점 : 갯수의 제한이 없다.** **단점 : 정확도가 약간 떨어진다.** **코드를 실행할때 이외에는 break point 걸수 없다.**
Hardware Breakpoint	**코드 또는 데이터 주소값** 한번에 4개 이상 설정할수 없다. **디버그 레지스터를 쓴다 : DR0~DR7** DR0 ~ DR3 : 브레이크 포인트를 설정할 곳의 주소 지정해준다. DR4 ~ DR5 : 사용하지 않는다 DR6 : 디버그 상태 레지스터 DR7 : 디버그 컨트롤 레지스터 **장점 : 뛰어난 정확도** **쉽게 우회하기가 어려움** **속도가 빠르다.** **코드나 데이터 모드 break 포인트 걸수 있다..ex) 카드 패 읽기 게임 에서 카드 패 읽을때 break point 걸어 놓는다.** **메모리에 데이터 쓸때 접근할때 코드를 실행할 때 모두 break point 를 걸수있다.** **단점 : 한번에 4개밖에 걸수 없다.**