[2013 POWER OF XX_easy_reversing2]

먼저 프로그램을 실행시켜보면 다음과 같은 창이 나온다.

Plain Text를 넣어주고 SUBMIT 버튼을 누르면 아래 칸에 plain text를 암호화한 문자열이 출력된다.

문제와 함께 주어진 decrypt_message.txt에는 암호화 된 문자열이 들어있다. 이것을 복호화 하면 될 것 같다. 이제 ILSpy로 프로그램을 분석해 보자.

이전 문제와 비슷한 것 같아서 button부분을 먼저 살펴보았다.

```
this.button1.BackColor = Color.CornflowerBlue;
this.button1.FlatStyle = FlatStyle.Popup;
this.button1.Font = new Font("맑은 고딕", 9f, FontStyle.Bold, GraphicsUnit.Point, 129);
this.button1.ForeColor = Color.Black;
this.button1.Location = new Point(508, 69);
this.button1.Name = "button1";
this.button1.Size = new Size(75, 23);
this.button1.TabIndex = 10;
this.button1.Text = "SUBM1T";
this.button1.UseVisualStyleBackColor = false;
this.button1.Click += new EventHandler(this.button1_Click);
```

버튼이 하나밖에 없는 것으로 보아 button1에서 모든 암호화 과정이 이루어지는 것 같다.

button1.Click을 살펴보자.

```
private void button1_Click(object sender, EventArgs e)
{
    if (this.textBox2.Text.Length > this.d8jergu394r0nnsjd94jfs.Length - this.zsawsrf6g0i98t6vllp)
    {
        MessageBox.Show("Length Error", "ERROR_bb");
        return;
    }
    this.gettime();
    string text = this.h8r9gu4inheiprhgnncvjousdfgiuweg(this.textBox2.Text);
    this.textBox1.Text = text;
    this.lfkfidngigiwhiu3yr89igorg(text);
}
```

[난독화가 되어있으므로 함수나 변수의 이름은 앞 두 글자만 따오기로 하자.]

처음에 어떤 연산을 하고 오류 메세지를 출력한다. 메시지 내용으로 보아 input값의 길이를 검사하는 부분인 것 같다. 이 부분은 중요한 것 같아 보이지 않으니 넘어가고 이후를 살펴보자.

처음에 입력받은 문자열을 h8()함수에 넣고 리턴값을 textBox1에 써넣는다. textBox1이 암호화된 문자열을 출력하는 부분이고 h8()함수가 암호화 해주는 함수인 것 같다.

다음은 h8()함수의 내용이다.

```
public string h8r9gu4inheiprhgnncvjousdfgiuweg(string str)
{
    StringBuilder stringBuilder = new StringBuilder();
    string text = this.ldfogndkfvisgi490rjgdijgw434ref(Form1.ncfjgirerg430t34trdgdfs(str));
    string[] array = text.Split(new char[]
    {
        ' '
    });
    for (int i = 0; i < array.Length - 1; i++)
    {
        stringBuilder.Append(this.kfig9jepoingndkfvndjroger(array[i], i) + " ");
    }
    return stringBuilder.ToString();
}
```

입력받은 문자열을 nc()함수에 넣고 리턴값을 ld()함수에 넣은 후 kf()함수에서 어떤 연산을 한 뒤 리턴해 준다.

먼저 nc()함수를 살펴보자.

```
public static string ncfjgirerg430t34trdgdfs(string input)
{
    int length = input.Length;
    char[] array = new char[length];
    for (int i = 0; i < input.Length; i++)
    {
        array[i] = input[length - i - 1];
    }
    return new string(array);
}
```

입력받은 문자열을 뒤집어서 리턴을 해주는 함수이다.

다음은 ld()함수를 살펴보자.

```
public string ldfogndkfvisgi490rjgdijgw434ref(string a)
{
    StringBuilder stringBuilder = new StringBuilder();
    int num = this.zsawsrf6g0i98t6vllp;
    char[] array = a.ToCharArray();
    for (int i = 0; i < a.Length; i++)
    {
        stringBuilder.Append((int)(this.d8jergu394r0nnsjd94jfs[i + num] ^ array[i]) + " ");
    }
    return stringBuilder.ToString();
}
```

먼저 변수들을 살펴보자.

```
public int njgcgcxdxxx6r = DateTime.Today.DayOfYear;

public int zfgvjnkji8y6ug9u9i = DateTime.Now.Hour;

public int cljbyt798ygdre5 = DateTime.Now.Minute;

public int zsawsrf6g0i98t6vllp = DateTime.Today.Month;

public int qexyg8j9u8thuhg = DateTime.Today.Day;
```

현재 DayOfYear, 시간, 분, 일, 월을 받아 변수들에 저장한다.

```
public string d8jergu394r0nnsjd94jfs = "9pMaVs5DxiOPGe8JETXYmg3lbudro6Qk1WLKwyhfnS4Iv0ABtjUCc7RZz2NFHq";
```

d8에는 알 수 없는 문자열이 들어있다.

이제 ld()함수 내용을 보면 ld()함수는 d8[i+month]와 array[i]를 xor연산을 시키고 나온 결과를 문자열로 만들어 리턴해 주는 함수라는 것을 알 수 있다.

마지막으로는 kf()함수를 살펴보자

```
public int kfig9jepoingndkfvndjroger(string chr, int range)
{
    int num = int.Parse(chr);
    int num2 = range % 3;
    int num3 = 2;
    if (num2 == 0)
    {
        num += this.qexyg8j9u8thuhg * num3 + this.cljbyt798ygdre5 * num3 - this.zfgvjnkji8y6ug9u9i * 2;
    }
    else if (num2 == 1)
    {
        num += this.zsawsrf6g0i98t6vllp * 3 + this.cljbyt798ygdre5 * 2 - this.zfgvjnkji8y6ug9u9i * num2;
    }
    else if (num2 == 2)
    {
        num += this.njgcgcxdxxx6r - this.zsawsrf6g0i98t6vllp * (num2 * 5) - this.cljbyt798ygdre5 * num2 - this.zfgvjnkji8y6ug9u9i * (num3 + 4) - num2 * num3;
    }
    return num;
}
```

두 번째 인자값 % 3을 한 값을 기준으로 연산 방법이 나뉘는데 모두 위에서 살펴본 현재 시간, 분 등을 이용해서 연산을 하고 그 값을 첫 번째 인자값에 더해 리턴해 준다.

함수 분석은 끝났으니 복호화 코드를 짜보자.
암호화 루틴은 nc() -> ld() -> kf() 이므로, 복호화는 kf() -> ld() -> nc() 순으로 하면 된다.

kf()함수는 연산된 값을 더하는 방식으로 암호화를 하므로 복호화는 연산된 값을 빼주면 된다.
ld()함수는 xor연산이므로 다시 한번 xor연산을 해주면 복호화가 된다.
마지막으로 nc()함수는 문자열을 뒤집는 함수이므로 다시 한번 문자열을 뒤집어주면 된다.

이때 decrypt_message.txt에 암호화한 날짜와 시간이 나와 있으므로 이것을 사용해 복호화를 하도록 하자.

Chicken's Message :

```
auth_key = md5(flag); auth_key = auth_key.ToLower(); if(auth(auth_key) == true) { clear(); }
```

Egg's Message :

```
string flag = "W3_10vE_Ch1cKen_FoR3veEr";
```