

Write-up: HackTheBox - Reversing - TearOrdear

SEBASTIÁN SEPÚLVEDA @PIBLACK

18 de mayo de 2019

Información: Categoría Reversing, puntuación 20.

Descripción: Find the username and password and put them in the flag in the format: HTBusername:password.

Warning: It can produce false positives.

WRITEUP

Lo primero que realizamos al partir es un análisis estático [1](#) del programa `TearOrDear.exe` que está comprimido en el archivo .zip. Vemos que está escrito en .NET y es de 32 bits, por lo que ocupamos `dnSpy_32bits` para revisar el archivo binario.

```
root@retro:~/Descargas/teardear# file TearORDear.exe
TearORDear.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
```

Figura 1. análisis estático con file

Lo primero que podemos hacer es ver cómo funciona el programa, así que lo corremos con dnSpy (Ver Figura 2)

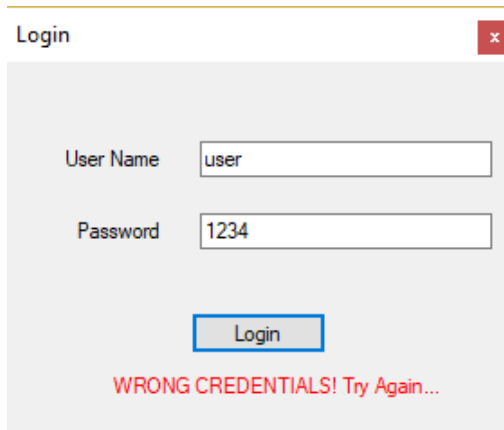


Figura 2. Login fallida en TearOrDear.exe

Al abrir dnSpy, comenzamos revisando todos los archivos de TearOrDear, sin mucho éxito en la mayoría, pues no hay información relevante, hasta que en la pestaña `button1_Click` (Ver Figura 3), nos encontramos con `if(this.username == this.o && this.check1(s))`. Por cómo está escrito el código inferimos que es el chequeo que hace el programa para revisar si los parámetros del password son correctos o incorrectos (Ver Figura 4)

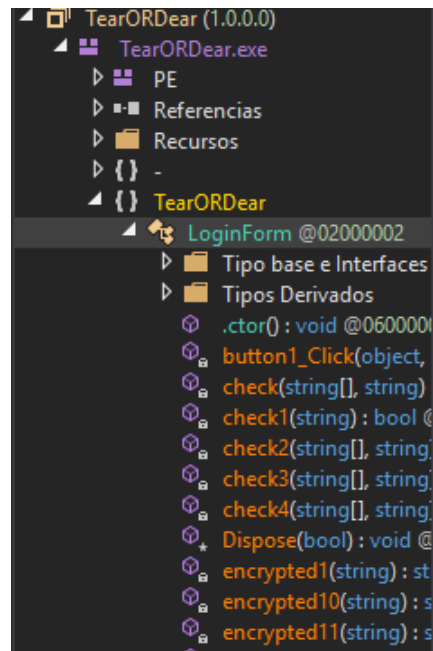


Figura 3. Ubicación de textttbutton1_Click

```

288
289 // Token: 0x0600000B RID: 11 RVA: 0x000288C File Offset: 0x0000A8C
290 private void button1_Click(object sender, EventArgs e)
291 {
292     this.label_Result.Text = "";
293     this.kapa(sender, e);
294     this.pep = 0;
295     this.aa = this.Multiply(this.encrypted1(this.textBox_user.Text).Substring(0, 5), -1);
296     this.aa = this.aa.Remove(this.aa.Length - 1);
297     string s = this.Multiply(this.oura, -9);
298     if (this.username == this.o && this.check1(s))
299     {
300         MessageBox.Show("Correct!");
301         return;
302     }
303     this.label_Result.Text = "WRONG CREDENTIALS! Try Again...";
304 }
305

```

Figura 4. Contenido función textttbutton1_Click

Para que la condición sea verdadera, el *username*, en mi caso 1234, debe tener el mismo valor que la variable *o* y además la función **check1** debe devolver verdadero.

Al buscar el valor de *o* observamos que vale **roiw!@#** (Ver Figura 4)

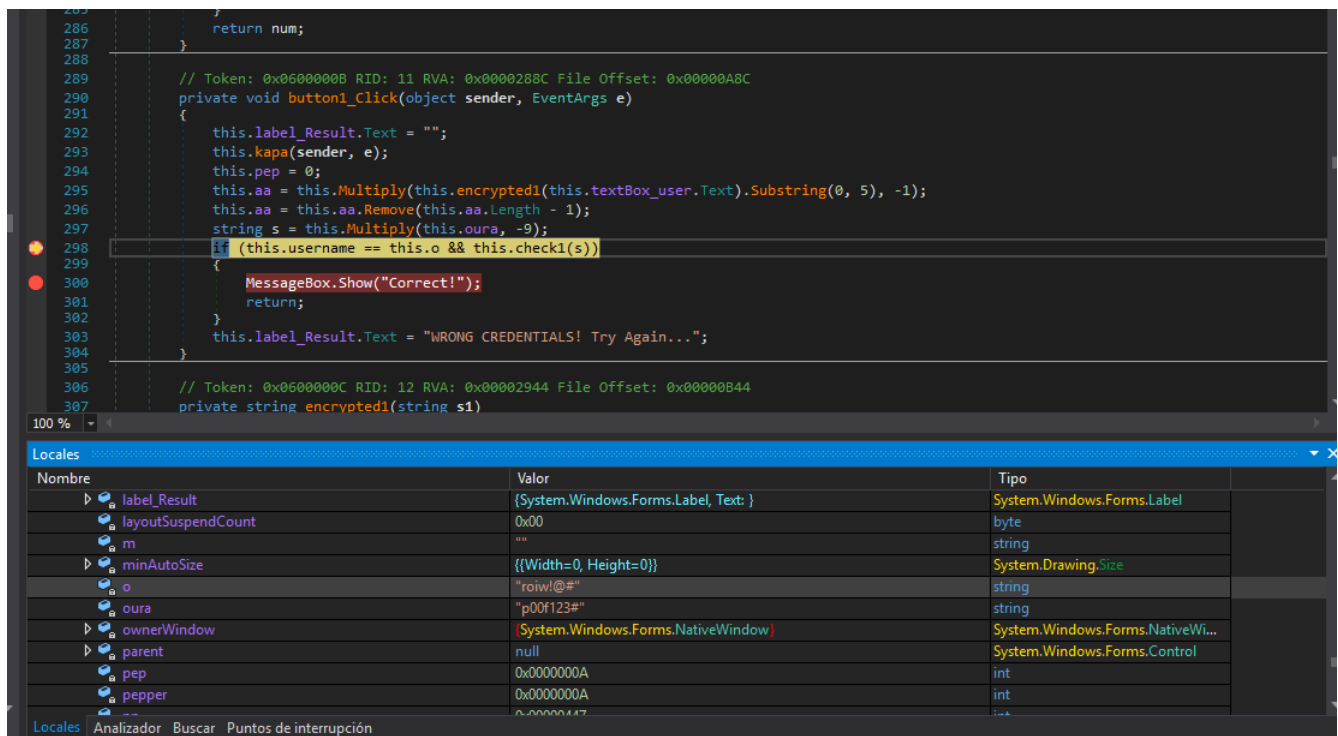


Figura 5. Resultado al generar una ruptura en la línea 298 del código, password

Luego de realizar lo anterior, revisamos la función **check** que devuelve verdadero si lo que hemos introducido en **textBox_user**, en nuestro caso *user* es igual a la variable **aa** y también si el **array[0]** es igual a **array[22]** que siempre es verdad pues ambos son "q".

```
// Token: 0x06000009 RID: 9 RVA: 0x00002620 File Offset: 0x00000820
private bool check(string[] s1, string s2)
{
    string[] array = new string[]
    {
        "q", array[0]
        "w",
        "e",
        "r",
        "t",
        "y",
        "u",
        "i",
        "o",
        "p",
        "a",
        "s",
        "d",
        "f",
        "g",
        "h",
        "j",
        "k",
        "l",
        "z",
        "x",
        "c",
        "q", array[22]
        "b",
        "n",
        "m"
    };
    array[3] + array[8] + array[7] + array[(int)Math.Sqrt(2.0)];
    return this.textBox_user.Text == this.aa && array[0] == array[22];
}
```

Figura 6. Función check con return

Al revisar el valor de **aa** (Ver Figura 7) vemos que el usuario debería ser igual a **piph**

locales			
Nombre	Valor	Tipo	
Width	0x0000012C	int	
WindowExStyle	0x00010180	int	
WindowState	Normal	System.Windows.Forms.FormWin...	
WindowStyle	0x16C80000	int	
WindowTarget	(System.Windows.Forms.Control.ControlNativeWindow)	System.Windows.Forms.IWindowT...	
WindowText	"Login"	string	
aa	"piph"	string	
activeControl	{System.Windows.Forms.TextBox, Text: hello}	System.Windows.Forms.Control (S...	
autoScaleBaseSize	{{Width=0, Height=0}}	System.Drawing.Size	
autoScaleDimensions	{{Width=6, Height=13}}	System.Drawing.SizeF	
autoScaleMode	Font	System.Windows.Forms.AutoScaleMode...	
locales Analizador Buscar Puntos de interrupción			

Figura 7. Resultado al generar una ruptura en la linea 298 del código, user

Comprobamos que estamos en lo cierto (Ver Figura 8)

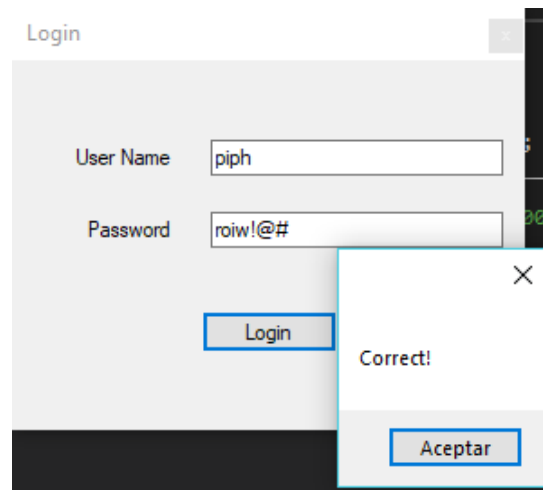


Figura 8. User y Password correctos

Como el flag es HTBusername:password, obtenemos el resultado.

Flag: HTB{piph:roiw!@#}