عنوان: **چگونه تشخیص دهیم اسمبلی دات نت ما وصله شده است؟**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۰۶ ۱۲:۲۹:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: Security

یکی از روش‌هایی که برای بررسی یکپارچگی فایل‌ها مورد استفاده قرار می‌گیرد و عموما در دنیای سخت افزار و firmware های نوشته شده برای آن‌ها مرسوم است، قرار دادن CRC32 فایل در قسمتی از فایل و بررسی آن حین Boot سیستم است. اگر CRC32 جدید با CRC32 اصلی یکسان نباشد به این معنا است که فایل در حال اجرا پیش تر دستکاری شده است.
اما در دات نت فریم ورک روش متداول اینکار چیست؟ برای این منظور اضافه کردن امضای دیجیتال به فایل و اسمبلی نهایی تولیدی (فایل exe یا dll تولیدی) توصیه می‌شود (مراجعه به قسمت خواص پروژه و افزودن امضای دیجیتال جدید فقط با چند کلیک، +).
این مورد خوب است (با توجه به اینکه از الگوریتم‌های RSA و SHA1 استفاده می‌کند)، لازم است، اما کافی نیست زیرا ابزارهای حذف آن وجود دارند. به عبارتی برای وصله کردن این فایل‌ها فقط کافی است این امضای دیجیتال حذف شود و زمانی هم که نباشد، بررسی خاصی در مورد یکپارچگی فایل صورت نخواهد گرفت.
اما اگر باز هم نگران patch یا وصله شدن اسمبلی دات نت خود هستید این مورد افزودن امضای دیجیتال را حتما انجام دهید.
مهم‌ترین خاصیت آن این است که یک سری تابع native در دات نت فریم ورک برای بررسی نبود آن وجود دارند (+):

```
[DllImport("mscoree.dll", CharSet=CharSet.Unicode)]
public static extern bool StrongNameSignatureVerificationEx(string wszFilePath, bool
fForceVerification, ref bool pfWasVerified);
```

wszFilePath مسیر فایلی است که باید بررسی شود.
fForceVerification آیا متغیر pfWasVerified نیز مقدار دهی گردد؟
خروجی تابع مشخص می‌سازد که آیا strong name موجود و معتبر است یا خیر؟

و مثالی از استفاده‌ی آن (که بهتر است در یک تایمر نیم ساعت پس از اجرای برنامه رخ دهد):

```
using System;
using System.Reflection;
using System.Runtime.InteropServices;

namespace SigCheck
{
   public class Validation
   {
       [DllImport("mscoree.dll", CharSet = CharSet.Unicode)]
       public static extern bool StrongNameSignatureVerificationEx(
           string wszFilePath, bool fForceVerification, ref bool pfWasVerified);

       public static void SigCheck()
       {
           var assembly = Assembly.GetExecutingAssembly();
           bool pfWasVerified = false;
           if (!StrongNameSignatureVerificationEx(assembly.Location, true, ref pfWasVerified))
           {
               //خاتمه برنامه در صورت عدم وجود امضای دیجیتال معتبر
               throw new Exception();
           }
       }
   }

   class Program
   {
       static void Main(string[] args)
       {
           Validation.SigCheck();
       }
   }
```

```
}
```

خوب، شاید پس از حذف و وصله شدن اسمبلی، مجددا strong name به آن اضافه شود! ، آن وقت چه باید کرد؟
زمانیکه به اسمبلی خود امضای دیجیتال اضافه می‌کنید، هش رمزنگاری شده فایل با الگوریتم RSA ، به همراه public key مورد نیاز در اسمبلی ذخیره می‌شوند. از آنجائیکه private key الگوریتم RSA را منتشر نکرده‌اید، شکستن الگوریتم RSA کار ساده‌ای نیست، مگر اینکه جفت کلید خودشان را تولید کنند و public key جدید را در فایل نهایی قرار دهند. بدیهی است این public key جدید با کلید عمومی ما که متناظر است با کلید خصوصی منتشر نشده‌ی اصلی، تطابق نخواهد داشد. برای آشنایی با تابعی که این بررسی را انجام می‌دهد به مقاله ذکر شده رجوع کنید:

Checking For A Valid Strong Name Signature