عنوان: **تبدیل فایل‌های pfx به snk**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱۱:۴۳
آدرس: www.dotnettips.info
گروه‌ها: Security, Certificate, PKCS, X509, Cryptography

مرسوم است و توصیه شده است که جهت ارائه کتابخانه‌های دات نتی خود از امضای دیجیتال استفاده کنید. VS.NET برای این منظور در برگه signing خواص یک پروژه، چنین امکانی را به صورت توکار ارائه می‌دهد.
حال اگر بخواهیم همین پروژه را به صورت سورس باز ارائه دهیم، استفاده کنندگان نهایی به مشکل برخواهند خورد؛ زیرا فایل pfx حاصل، توسط کلمه عبور محافظت می‌شود و در سایر سیستم‌ها بدون درنظر گرفتن این ملاحظات قابل استفاده نخواهد بود.
معادل فایل‌های pfx، فایلهایی هستند با پسوند snk که تنها تفاوت مهم آن‌ها با فایل‌های pfx، عدم محافظت توسط کلمه عبور است و ... برای کارهای خصوصا سورس باز انتخاب مناسبی به شمار می‌روند. اگر دقت کنید، اکثر پروژه‌های سورس باز دات نتی موجود در وب (مانند NHibernate، لوسین، iTextSharp و غیره) از فایل‌های snk برای اضافه کردن امضای دیجیتال به کتابخانه نهایی تولیدی استفاده می‌کنند و نه فایل‌های pfx محافظت شده.
در اینجا اگر فایل pfx ایی دارید و می‌خواهید معادل snk آن‌را تولید کنید، قطعه کد زیر چنین امکانی را مهیا می‌سازد:

```
using System.IO;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;

namespace PfxToSnk
{
    class Program
    {
        /// <summary>
        /// Converts .pfx file to .snk file.
        /// </summary>
        /// <param name="pfxData">.pfx file data.</param>
        /// <param name="pfxPassword">.pfx file password.</param>
        /// <returns>.snk file data.</returns>
        public static byte[] Pfx2Snk(byte[] pfxData, string pfxPassword)
        {
            var cert = new X509Certificate2(pfxData, pfxPassword, X509KeyStorageFlags.Exportable);
            var privateKey = (RSACryptoServiceProvider)cert.PrivateKey;
            return privateKey.ExportCspBlob(true);
        }

        static void Main(string[] args)
        {
            var pfxFileData = File.ReadAllBytes(@"D:\Key.pfx");
            var snkFileData = Pfx2Snk(pfxFileData, "my-pass");
            File.WriteAllBytes(@"D:\Key.snk", snkFileData);
        }
    }
}
```

نظرات خوانندگان

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۷/۳۰ ۱۱:۹

آقای نصیری ممنون از لطف شما.
ممکنه بیشتر درمورد این امضا و نحوه‌ی کاربرد اون صحبت کنید؟(مثلا بنده یک کتابخانه‌ی آزمایشی را با استفاده از امضای موجود در بخش Signing امضا نموده و فایل pfx مربوطه را ساختم.اما اسمبلی مربوطه به سادگی در سایر پروژه‌ها قابل استفاده و حتی قابل مشاهده است(از طریق metadata)).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۳۰ ۱۱:۱۲

بله. این امضای دیجیتال، فقط به این معنا است که کار تولید شده متعلق به شما می‌باشد. هیچ نوع محدودیت دیگری را اعمال نمی‌کند.
+ وجود آن اندکی patch کردن برنامه‌ها رو مشکل می‌کنه. خصوصا در مورد برنامه‌های WPF و سیلورلایت.

نویسنده: سام ناصری
تاریخ: ۱۳۹۱/۱۲/۱۶ ۶:۲

مطلب خوبی بود وحید جان. ممنونم.
البته من بعد از اینکه مطلب شما رو خوندم و متوجه شدم که دو نوع فایل pfx و snk هست که با اون میشه sign کرد اندکی تو اینترنت گشتم و متوجه یک نکته شدم که گفتم بد نیست اینجا مطرح کنم.
هر چند مطلب شما درباره تبدیل فایل pfx به snk است اما متنی که نوشتید این موضوع را القا میکند که نمیشود به سادگی این فایل رو ساخت.
به هر روی، میتوان فایل snk را از طریق فایل زبانه signing در خواص پروژه ساخت. برای این کار کافیست که گزینه Protect my key file with a password را آنتیک کرد و در این حالت به جای اینکه فایل pfx ساخته شود فایل snk ساخته میشود.
مطلب دیگر اینکه من پروژه‌های متن باز دیگری را دیده ام که الان حضور ذهن ندارم بگم(احتمالاً یکیشون RavenDB بود) که از طریق خواص پروژه ویژوال استودیو کار signing را انجام نمیدهند یعنی در آنجا گزینه sign کردن را انتخاب نکرده اند. چون فایل snk را اگر منتشر کنیم همه میتونند با اون اسمبلی‌ها را sign کنند و معنای strong name بودن اسمبلی به طور کلی میره زیر سوال. در عوض از یک customized build استفاده میکنند که فقط توسط خودشون(مالکان پروژه) قابل فراخوانی است و توسط اون اسمبلی‌های release را میسازند. البته در اینباره باید بیشتر بررسی کنم و شاید دقیقاً ماجرا 100 درصد به این شکل که گفتم نیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۶ ۹:۳۱

- علت اینکه این مطلب رو نوشتم مربوط به زمانی بود که پروژه‌ای از قبل موجود بود با فایل pfx آن و قصد داشتم معادل محافظت نشده فایل pfx آن‌را تولید کنم.
- در مورد تولید فایل‌های pfx و snk یک مطلب نسبتا جامع در سایت داریم .
- به نظر من زمانیکه یک پروژه سورس باز است، امضا کردن اسمبلی‌های آن آنچنان مفهومی ندارد چون دسترسی به سورس و حتی ارائه آن بر اساس اطمینان به جامعه مصرف کننده صورت می‌گیرد. خیلی خیلی کم هستند موارد سوء استفاده از اسمبلی‌های امضاء شده به این صورت. مگر اینکه بحث پروژه کرنل لینوکس با تعداد مصرف کننده بالا و اهمیت امنیتی آن مطرح باشد که نیاز به امضای فایل‌های باینری آن وجود داشته باشد.