عنوان: **نگهداری رشته ها (String) در حافظه به صورت Encrypt**
نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۳/۳۱ ۴:۹
آدرس: www.dotnettips.info
برچسب‌ها: Security, Tips

در همین سایت در بخش لینک‌های ارسالی ، لینکی توسط آقای امیر هاشم زاده به اشتراک گذاشته شده بود با عنوان "چرا هکرها نوع داده String را دوست دارند" ؛ مقاله ای بود در سایت CodeProject که در آن روش هایی که هکرها توسط آن می‌توانند اطلاعات حساس نرم افزار را که در قالب String در حافظه ذخیره شده اند را بررسی نمایند.
اصل مطلب را می‌توانید اینجا مطالعه کنید.

در دات نت فریم ورک کلاسی با عنوان SecureString وجود دارد که توسط آن می‌توان عبارات رشته ای که دارای اطلاعات حساس می‌باشند را به صورت رمز گذاری شده در حافظه ذخیره نمود.

نمونه ای از استفاده این تابع را در زیر مشاهده میکنید:

```
public class Example
{
    public static void Main()
    {
        .
        SecureString securePwd = new SecureString();
        ConsoleKeyInfo key;

        Console.Write("Enter password: ");
        do {
           key = Console.ReadKey(true);

           // بررسی میشود که کلید فشرده شده جزو حروف الفبا می‌باشد یا کلید دیگری است
           if (((int) key.Key) >= 65 && ((int) key.Key <= 90)) {
              // کاراکتر مربوط به کلید فشرده شده به انتهای متغیر سکوراسترینگ اضافه می‌شود
              securePwd.AppendChar(key.KeyChar);
              Console.Write("*");
           }
        // خروج از حلقه در صورت فشردن کلید اینتر
        } while (key.Key != ConsoleKey.Enter);
        Console.WriteLine();

        try
        {
            MessageBox.Show(securePwd);
        }
        catch (Win32Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
}
```

در کدهای بالا رمز عبور از کاربر دریافت شده و در متغیر securepwd که شئی از کلاس SecureString می‌باشد ذخیره می‌شود.پس از آن شئی SecureString عبارت مربوطه را به صورت رمز گذاری شده در حافظه ذخیره میکند.
در این روش ابتدا مقدار کلید فشرده شده در متغیر Key که از نوع ConsoleKeyInfo تعریف شده ذخیره می‌شود. بعد از آن مقدار آن بررسی شده و اگر جزو حروف الفبای انگلیسی بود به انتهای متغیر securepwd افزوده می‌شود. این کار با متد AppendChar انجام می‌شود. این عملیات تا فشرده شدن کلید Enter ادامه پیدا میکند.

نظرات خوانندگان

نویسنده: فرید
تاریخ: ۱۳۹۱/۰۳/۳۱ ۴:۵۳

سلام
مرسی از مطلب خوبتون
ولی یه سوال :

آیا کرکرها و کیلاگرها به سادگی می‌توانند به آنها پی ببرند؟

نویسنده: پرهام
تاریخ: ۱۳۹۱/۰۳/۳۱ ۸:۲۱

به کدوم قسمت پی ببرن، بخش تایپ شاید؟! یعنی قبل از اینکه به securePwd اضافه بشه. ولی وقتی اضافه شد و در حافظه قرا گرفت احتمالا یا نه یا خیلی سخت. بستگی به الگریتم رمزگذاری داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۸:۲۵

بحث key logger متفاوت است. این متد سبب نمی‌شود که کلید فشرده شده کاربر از دید یک لاگر مخصوص آن مخفی باقی بماند. این روش فقط اطلاعات رشته‌ای را حین استفاده در برنامه به صورت یک رشته ساده در حافظه قرار نمی‌دهد. بلافاصله پس از استفاده از آن رمزنگاری اطلاعات آن خودکار است.
جهت اطلاع password box در wpf به صورت پیش فرض از همین کلاس استفاده می‌کند.

نویسنده: محمد
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۰:۵

مرسی و خسته نباشید، مفید بود
مثلا اگه پسورد رو از دیتابیسی واکشی و با این مقدار مقایسه کنیم؛ برا مقایسه این رشته امن تابعی وجود داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۰:۲۶

- شما نباید کلمه عبور رو هش نشده در بانک اطلاعاتی ذخیره کنید. (یعنی واکشی کلمه عبور به صورت clear text کار اشتباهی است)
- برای مقایسه در اینجا بهتر است از یک loop و بررسی کاراکترها به نحوی که در اینجا بحث شده استفاده کنید. بعد هم حافظه رو تخریب کنید. بحث اصلی اینجا است که قرار است ردی در حافظه باقی نماند؛ آن هم به صورت رمزنگاری نشده.

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۳/۳۱ ۲۰:۵۳

البته این رو هم باید اضافه می‌کردید استفاده از این متد در حالت UnSafe(با سوئیچ UnSafe) قابل استفاده است به این دلیل که رشته‌ی رمزنگاری شده در حافظه‌ی مدیریت نشده(خارج از کنترل CLR)قرار می‌گیرد.بنابراین دسترسی به این بخش تقریبا غیرممکن است.
موفق باشید.

نویسنده: پوریا عالی نژاد

تاریخ: ۱۳۹۱/۰۴/۰۱ ۹:۵۵

بنده متوجه نشدم. وقتی هکر یا کرکر می‌تواند با Marshal.PtrToStringBSTR و چند تابع کمکی دیگر محتویات درون SecureString را کشف کند، آیا هدف ما فقط از رده خارج کردن تعدادی از کرکرهای آماتور و ابزارمحور هستند؟
و یا اصلا چه دلیلی دارد که پسورد را در حافظه نگه داریم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۰:۳۵

مشکلی که با string معمولی وجود دارد این است که به صورت معمولی و غیر رمزنگاری شده در حافظه قرار می‌گیرند و همچنین توسط برنامه هم قابل حذف از حافظه نیستند. اگر محتویات آن‌را تغییر دهید، یک وهله دیگر ایجاد شده و وهله قبلی هنوز در حافظه قرار داد و البته ... هر زمان که GC صلاح دید آن‌را پس از مدتی حذف می‌کند اما نه بلافاصله. اصطلاحا به این نوع اشیاء immutable هم گفته می‌شود.
SecureString اینبار mutable است و می‌توان مقدار آن‌را توسط برنامه واقعا تخریب کرد و منتظر GC نشد و تغییرات در آن، چندین کپی از آن‌را ایجاد نمی‌کند. SecureString توسط کدهای unmanaged تهیه شده و دسترسی به محتوای آن به این سادگی‌ها نیست. برای دسترسی به آن خارج از برنامه، یک شخص باید اشاره‌گر به (IntPtr unmanaged) این رشته رمزنگاری شده را بیابد. Marshal.PtrToStringBSTR فقط یک string pointer را دراختیار شما قرار می‌دهد.

ضمن اینکه امنیت بحثی نسبی است. اگر از خانه خارج می‌شوید بهتر است درب آن‌را قفل کنید. اما این به معنای نفوذناپذیری 100 درصد خانه شما نیست ولی این امر «نسبت به» خانه‌ای که درب آن کاملا باز است، خیلی بهتر است .

نویسنده: ابراهیم
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۵:۱۴

خیلی ممنون از مطلب خوبتون
سوالی داشتم : چطور میشه از رشته ای که در این نوع داده ذخیره شده استفاده کرد؟ منظورم اینه که چطور میشه در داخل برنامه فهمید که چه داده ای در این متغیر وجود داره

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۴/۰۴ ۲۳:۴۵

همانند یک متغیر از نوع String با آن رفتار می‌شود. برای مثال برای نمایش مقدار موجود در این متغیر می‌توان از دستور زیر استفاده کرد:

```
MessageBox.Show(securePwd);
```

نویسنده: ابراهیم
تاریخ: ۱۳۹۱/۰۴/۰۵ ۱۵:۶

وقتی این دستور رو اجرا میکنم پیغام زیر نمایش داده میشه

```
System.Security.SecureString
```

و محتوای رشته نمایش داده نمیشه