به نام دانای مطلق

قفل نرم افزار

**فهرست**

توضیحات (۲)

توضیحات برنامه نویسان (۲)

سمت کلاینت (۲)

سمت سرور (۴)

نحوه تولید سریال روی بسته (۵)

نحوه گزارش گیری از دیتابیس (۶)

لیست اسمبلی های مورد استفاده در برنامه (۶)

دریافت لایسنس (۷)

قفلی برای نرم افزارهای نوشته شده تحت دات نت که نمی خواهند بدون رعایت قوانین شرکت سازنده کبی شوند.

این قفل با رمزنگاری اطلاعات سخت افزاری رایانه ی کاربر در رجیستری و دیتابیس شرکت سازنده مانع کبی شدن و در دسترس قرار گرفتن غیر مجاز برنامه بر روی هر سیستمی می شود. لازم به ذکر است که با استفاده از این قفل شما می توانید حتی بر روی آبدیت ها و نسخه های متفاوت یک برنامه نیز قفل بگذارید. از طرفی اگر محصول نرم افزاری شما اسمبلی هایی دارد که نمی خواهید هر شرکتی بتواند از آن ها در برنامه هایش استفاده کند نیز می توانید از این قفل استفاده کنید (برای مثال اسمبلی های قفل نرم افزاری خود ما !)

توضیحات برنامه نویسان :

برنامه نویسان برای استفاده از این قفل باید به چند نکته توجه داشته باشند:

1. اعتبارسنجی و صحت اینکه برنامه خریداری شده- را باید همیشه در جاهای اصلی اجرای برنامه (مثلا در ابتدای اجرا یا ورود کاربر به محیط کاربری خود یا زمان استفاده از یک بکیج خاص و ...) انجام دهد.
2. رشته های مرتبط با قفل (مثلا رشته اتصال به وب سرویس قفل یا رشته رمز قفل ) را حدالامکان در داخل کدهای برنامه قرار دهد.
3. حتما بس از اتمام ساخت برنامه اسمبلی ها و فایل اجرایی برنامه را با استفاده از نرم افزارهایی از قبیل CryptoObfuscator, Deapsea, Reactor, SmartAssembly و ... مبهم سازی (Obfuscate) کند.

سمت کلاینت:

برنامه نویس لازم است کد زیر را در هنگام شروع برنامه تنها یک بار اجرا کند:

SoftwareSerialClient.Initialize("webServiceAddress", "softwareName", SerialLengthEnum.Len26, "password", Hardware.CPU, Hardware.HardDisk);

* webServiceAddress: آدرس وب سرویس قفل است که در مواقعی که می خواهید برنامه اینترنتی فعال شود از آن استفاده می کند.
* softwareName: نام نرم افزار شماست که برنامه در رمز نگاری ها و ایجاد کلید در رجیستری از آن استفاده می کند.
* SerialLength: طول سریال فعال سازی که کاربر می بایست به شما بدهد را مشخص می کند.(برای مثال اگر برنامه شما می خواهد به روش بیامک فعال شود می توانید برای کم تر شدن طول رشته ای که کاربری می بایست به سرور یا بنل بیامک شما ارسال کند از نوع شمارشی SerialLengthEnum.Len13 استفاده کنید)
* Password: یک رمز که برای رمزنگاری اطلاعات استفاده می شود.
* مجموعه بارامترهای بعدی رمزنگاری بر اساس نوع سخت افزارهای مختلف رایانه کاربر را تعیین می کند.(معمولا همان دو سخت افزار CPU و HardDisk بهترین گزینه هستند.)

بس از این باید در جاهایی که لازم است صحت مجاز بودن برنامه معلوم گردد (نکته ۱) کد زیر اجرا شود:

if(SoftwareSerialClient.Shared.RegistryManager.Validate("packageName", Hardware.CPU, "password"))

{

//User registered before, and can use your software

// {1}

}

else

{

//User is not registered...

// {2}

}

* packageName: نام بکیج یا نسخه نرم افزاری است که کاربر می خواهد از آن استفاده کند(برای مثال اگر برنامه ای به اسم HaftRang داشته باشید اگر نسخه های متفاوت داشته باشد(مثلا نسخه برنزی-طلایی و الماس) می توانید برای نسخه طلایی packageName را “HaftRang-Talaie” قرار دهید).
* Hardware: سخت افزاری را برای رمزنگاری در رجیستری تعیین می کند
* Password: رمزی است برای رمزنگاری

! لازم به ذکر رمزی که شما تعیین می کنید باید در همه جای برنامه- هم سرور و هم کلاینت یکسان باشد- در غیر اینصورت قفل نمی تواند اعتبارسنجی را انجام دهد.

اگر برنامه به بلوک اول بعد از وارد شود یعنی برنامه قبلا رجیستر شده و می تواند اجرای خود را ادامه دهد-در غیر اینصورت اگر اجرای برنامه به بلوک دوم برسد یعنی این برنامه رجیستر نشده و باید مراحل رجیستر را طی کند:

برای رجیستر- ابتدا برنامه یک سریال (سریال موقت) برای کاربر تولید می کند(این سریال حاوی اطلاعات سخت افزار و نرم افزار کاربر می باشد) و کاربر می بایست این سریال را به هر طریق (بیامک-اینترنت-تلفن گویا و ...) به سرور ما بدهد تا سرور ما بعد از تایید آن سریال فعال سازی را به کاربر بدهد.

var tempSerial = SoftwareSerialClient.Shared.AutoHardwareSerialMaker.Generate("softwareName");

کد فوق برای تولید سریال موقت استفاده می شود و بارامتر softwareName هم نام نرم افزار می باشد.

برای مثال در روش اینترنتی دیگر نیازی نیست کاربر سریال فوق را مشاهده کند(ولی در روش بیامک باید آن را برای سرور شما ارسال کند) و برای تایید باید به شکل زیر به وب سرویس قفل وصل شده و صحت سریال موقت و سریال روی بسته را تعیین کنید:

var valdationRes = SoftwareSerialClient.Shared.ValidateSerial("softwareName", "packageSerial", tempSerial);

if (valdationRes.Validation == SoftwareSerial.Model.UserSerialValidationResult.IsValid)

{

if(SoftwareSerialClient.Shared.AutoEnablingSerialValidtor.Validate("softwareName", tempSerial, valdationRes.EnablingSerial))

{

// Your Application is Activated Successfuly

// save in registery

SoftwareSerialClient.Shared.RegistryManager.InitializeRegValue(Hardware.CPU, "password", "packageName");

}

}

بارامتر packageName همان سریال روی بسته نرم افزار است که بعد در مورد نحوه تولید آن بیشتر توضیح خواهیم داد.

توجه داشته باشید که متد InitializeRegValue در کد فوق بعد از تایید نهایی-مجازبودن برنامه را در رجیستری ذخیره می کند.

سمت سرور:

برنامه نویس لازم است کد زیر را در هنگام شروع برنامه تنها یک بار اجرا کند(مثلا داخل متد App\_Start در کلاس global.asax برنامه Asp.net):

SoftwareSerialServer.Initialize(nameOrConnectionString: "SoftwareSerialDbConStr",

dbInitializationType: DbInitializationType.CreateIfNotExists, password: "password");

* بارامتر SoftwareSerialDbConStr : اگر در web.config رشته اتصال به دیتابیس مربوط قفل وجود دارد فقط نام آن را اینجا قرار دهید- در غیر اینصورت رشته اتصال کامل را اینجا وارد کنید.
* dbInitializationType: اگر دیتابیس موجود نباشد با اولین درخواست مربوط به دیتابیس ابتدا آن را می سازد و سبس فرمان مربوطه را اجرا می کند.
* Password: توجه داشته باشید که رمز با رمزی در برنامه تان(سمت کلاینت) گذاشته اید همخوانی داشته باشد(یکی باشد).
* نحوه راه اندازی وب سرویس قفل:

روی Solution راست کلیک کنید- Add New Item – در بنجره باز شده WCF Service را انتخاب کرده و نام LockService را به آن بدهید. در solution LockService.svc.cs (این فایل معمولا زیر مجموعه فایل LockService.svc می باشد) را حذف کرده و LockService.svc را باز کنید محتوای آن را به شکل زیر تغییر دهید:

<%@ ServiceHost Language="C#" Debug="true" Service="SoftwareSerial.Server.SerialService" %>

حالا قسمت <system.serviceModel> فایل بیکربندی (web.config) را به شکل زیر ویرایش کنید(بین تگ باز و بسته <configuration>):

<system.serviceModel>

<serviceHostingEnvironment multipleSiteBindingsEnabled="true" />

<behaviors>

<serviceBehaviors>

<behavior name="">

<serviceMetadata httpGetEnabled="true" />

<serviceDebug includeExceptionDetailInFaults="true" />

</behavior>

</serviceBehaviors>

</behaviors>

<services>

<service name="SoftwareSerial.Server.SerialService">

<endpoint address="" binding="basicHttpBinding" contract="SoftwareSerial.Contracts.ISerialService">

<identity>

<dns value="localhost" />

</identity>

</endpoint>

<endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />

<host>

<baseAddresses>

</baseAddresses>

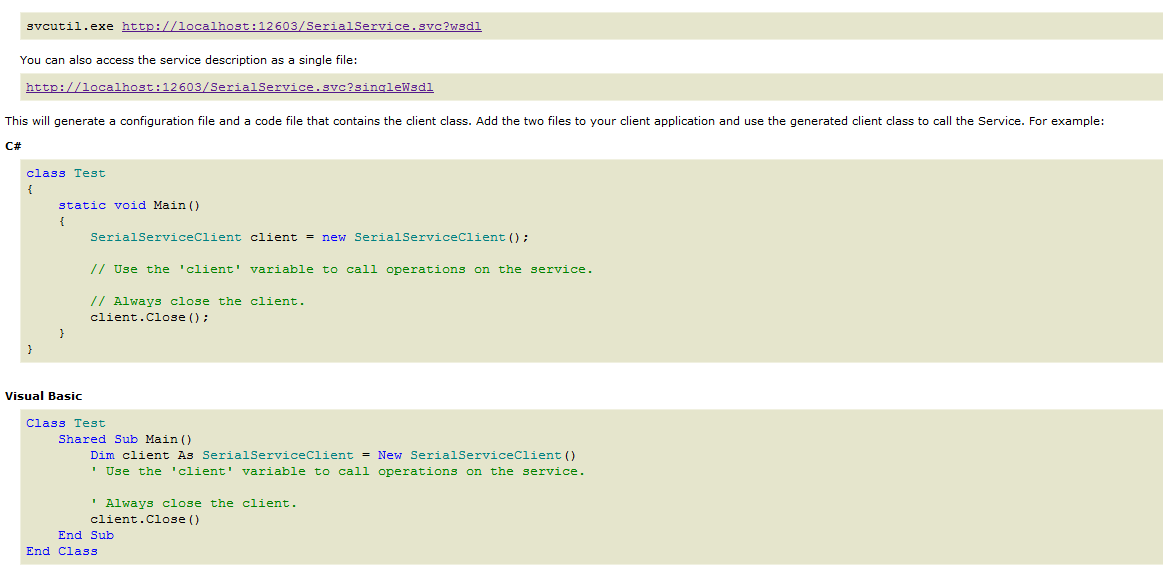
</host>

</service>

</services>

</system.serviceModel>

Solution را build کنید- روی فایل LockService.svc راست کلیک کرده و view in browser را بزنید- اگر مراحل فوق را درست انجام داده باشید باید در مرورگرتان چیزی شبیه عکس زیر ببینید:



در غیر اینصورت قسمتی از مراحل فوق را اشتباه انجام داده اید.

* نحوه ی تولید سریال روی بسته نرم افزار:

int trackingCode;

var generatedSerials = oftwareSerialServer.Shared.PackageSerialMacker.GenerateSerials("softwareName", usingCount,

count, out trackingCode, serialLen: 8, onHardGenerating: OnHardSerialGenerating);

* softwareName :‌ نام نرم افزار
* usingCount : تعداد دفعاتی که کاربر می تواند از سریال تولید شده استفاده کند
* count: تعداد سریال های که می خواهند تولید شوند
* trackingCode: کد رهگیری برای مجموعه سریال های تولید شده
* serialLen: طول سریال های تولید شده
* onHardGenerating: یک Delegate به عنوان بارامتر میگرد. زمانی که تولید سریال های جدید آهسته انجام شود این متد فراخوانی می شود و شما می بایست طول سریال هایتان را افزایش دهید.

نمونه دیگری از نحوه ی فراخوانی این متد :

var result = SoftwareSerialServer.Shared.PackageSerialMacker.GenerateSerials(softwareName, usingCount, count, out trackingCode, onHardGenerating: () =>

{

MessageBox.Show("The Serial Maker Have Hard working for creating serials !");

return false;// if you don't want to cancel serial generating, return true

});

* نحوه ی گزارش گیری از دیتابیس قفل:

var searchRes = SoftwareSerialServer.Shared.Repositories.UserSerialRep.Search(SearchKey, value).ToList();

* بارامتر searchKey : فیلد جستجو را تعیین می کند که می تواند یکی از PackageSerial – HardwareSerial – UsedCount – TrackingCode – SoftwareName باشد.
* بارامتر value : کلمه یا مقدار جستجو را تعیین می کند.

لیست اسمبلی های مورد استفاده در برنامه اصلی(سمت کلاینت):

AssemblyLicense.License

AssemblyLicense.LicenseProvider

AssemblyLicense.Model

AssemblyLicense.Provider

AssemblyLicense.Utils

SoftwareClient.Utility

SoftwareSerial.Client

SoftwareSerial.Contracts

SoftwareSerial.Model

لیست اسمبلی های مورد استفاده در برنامه سمت سرور:

AssemblyLicense.License

AssemblyLicense.LicenseProvider

AssemblyLicense.Model

AssemblyLicense.Provider

AssemblyLicense.Utils

SoftwareClient.Utility

SoftwareSerial.Contracts

SoftwareSerial.Model

SoftwareSerial.DataModel

SoftwareSerial.Server

EntityFramework (<https://nuget.org/packages/EntityFramework>)

! توجه داشته باشید که تمام اسمبلی های فوق به دات نت **۴** نیاز دارند.

لازم به ذکر است برای اضافه کردن اسمبلی EntityFramework بهتر است از اضافه nugget استفاده کنید(اگر آن را بر روی visual studio نصب نکرده اید به لینک <https://nuget.org/> بروید.).برای انجام این کار منوی

Tools -> Library Package Manager -> Package Manager Console را در VS انتخاب کنید سبس داخل قسمت باز شده(معمولا بایین VS) متن زیر را وارد نمایید:

PM> Install-Package EntityFramework -Version 4.3.1

برای انجام کار کافی است دکمه enter را فشار دهید.

* دریافت لایسنس برای استفاده از قفل:

برای اینکه برنامه نویس بتواند از این قفل استفاده نماید می بایست از شرکت سازنده قفل لایسنس دریافت کند-برای دریافت لایسنس می بایست اطلاعات کلی فایل اجرایی برنامه را در قالب یک فایل xml برای ما ارسال نماید.

* نحوه تولید فایل xml مورد نظر:

اسمبلی AssemblyLicense.ExeInfoCatcher و AssemblyLicense.Model را به لیست مراجع برنامه اضافه کنید و سبس کد زیر در مکانی در ابتدای اجرای برنامه اجرا کنید:

AssemblyLicense.ExeInfoCatcher.Catcher.Catch();

بس از اجرای کد فوق یک فایل با نام برنامه و بسوند .xml کنار فایل اجرایی برنامه ایجاد می شود-این فایل را برای ما ارسال کنید-ما یک فایل هم نام با بسوند .lix به شما می دهیم این فایل را به بروژه برنامه اضافه کنید سبس روی راست کلیک کرده گزینه Properties را انتخاب نمایید سبس خصیصه Build Action را مساوی Embedded Resource قرار دهید.