عنوان: **رمزنگاری کانکشن استرینگ در ASP.Net**
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۶ ۱۰:۱۴:۵۹
آدرس: www.dotnettips.info
برچسب‌ها: Security

ذخیره کردن رشته اتصالی به دیتابیس، به صورت یک رشته مشخص در کدهای برنامه، کاری است مزموم. زیرا پس از هر بار تغییر این مورد، نیاز خواهد بود تا تمامی سورس‌ها تغییر کنند و اگر از حالت web application استفاده کرده باشید، مجبور خواهید شد یکبار دیگر برنامه را کامپایل و دایرکتوری bin روی سرور را به روز کنید. به همین جهت، استاندارد برنامه‌های ASP.Net این است که این رشته اتصالی را در فایل web.config ذخیره کنیم تا با هر بار تغییر پارامترهای مختلف آن (مثلا تغییر نام سرور، یا تعویض ماهیانه پسوردها)، مجبور به کامپایل مجدد برنامه نشویم. شبیه به همین مورد در برنامه‌های PHP هم رایج است و عموما این مشخصات در فایل config.php و یا با اسامی شبیه به این صورت می‌گیرد.
در ASP.Net 1.x قسمت خاصی برای کانکشن استرینگ وجود نداشت اما از ASP.Net 2 به بعد ، قسمت ویژه‌ای مخصوص این کار در فایل web.config در نظر گرفته شده است.
خیلی هم خوب! اما این تجربه تلخ کاری را (که یکبار برای من رخ داد) هم همواره در نظر داشته باشید:
امکان خوانده شدن محتوای فایل کانفیگ، توسط همسایه شما در همان هاست اشتراکی که الان از آن دارید استفاده می‌کنید.
عموما هاست‌های اینترنتی اشتراکی هستند و نه dedicated و نه فقط مختص به شما. از یک سرور برای سرویس دهی به 100 ها سایت استفاده می‌شود. یکبار در یکی از سایت‌ها دیدم که فایل machine.config سرور را هم محض نمونه خوانده بودند چه برسد به فایل متنی کانفیگ شما! یا تصور کنید که وب سرور هک شود. عموما اس کیوال سرور بر روی سرور دیگری قرار دارد. به همین جهت رمزنگاری این رشته باز هم ضریب امنیت بیشتری را به همراه خواهد داشت.
به همین منظور رمزنگاری قسمت کانکشن استرینگ فایل وب کانفیگ الزامی است، چون آن‌هایی که به دنبال اطلاعاتی اینگونه هستند دقیقا می‌دانند باید به کجا مراجعه کنند.

راه حل‌ها:

الف) از وب کانفیگ برای این‌کار استفاده نکنید. یک فایل class library درست کنید (یک dll مجزا) و ارجاعی از این فایل را به پروژه خود اضافه کنید و از رشته اتصالی قرار گرفته در آن استفاده کنید. این فایل را هم می‌توان با روش‌های obfuscation محافظت کرد تا امنیت اطلاعات داخل آن‌را تا حد قابل قبولی بالا برد. همچنین می‌توان برای این فایل کتابخانه، امضای دیجیتال درنظر گرفت. زیرا امضای دیجیتال سبب می‌شود تا تغییر فایل dll رشته اتصالی، با یک کپی و paste معمولی قابل انجام نباشد (تمامی dll ها و اسمبلی‌های دیگری که ارجاعی از آن‌را در خود دارند باید یکبار دیگر هم کامپایل و به سرور منتقل شوند). این یک نوع اطمینان خاطر است اما در بلند مدت شاید تکرار اینکار خسته کننده باشد.

ب)استفاده از روش استاندارد رمزنگاری قسمت‌های مختلف کانکشن استرینگ فایل web.config
برای مشاهده نحوه انجام اینکار با برنامه نویسی به این مقاله مراجعه نمائید.
مزیت: نیازی به کد نویسی برای رمزگشایی و استفاده از آن نیست و اینکار به صورت خودکار توسط ASP.Net انجام می‌شود.
ایراد:فایل حاصل قابل انتقال نیست. چون رمزنگاری بر اساس کلیدهای منحصربفرد سرور شما ایجاد می‌شوند، این فایل از یک سرور به سرور دیگر قابل انتقال و استفاده نخواهد بود. یعنی اگر بر روی کامپیوتر برنامه نویسی شما این‌کار صورت گرفت، برنامه در سرور کار نخواهد کرد. البته شاید ایراد آنچنانی نباشد و فقط باید یکبار دیگر روی هاست نیز این کار را تکرار کرد. اما باید درنظر داشت که همسایه محترم شما نیز می‌تواند بر روی همان هاست به سادگی فایل شما را رمزگشایی کند! بنابراین نباید اصلا به این روش در هاست‌های اشتراکی دل خوش کرد.

ج)بکارگیری روش‌های غیراستاندارد رمزنگاری
منظور از غیراستاندارد، حالت‌های دیگر استاندارد رمزنگاری و رمزگشایی نسبت به روش استاندارد ارائه شده توسط مایکروسافت است (که همه از آن مطلع هستند). به شخصه از این روش در هاست‌ها استفاده می‌کنم. (مثلا ، البته با کمی تغییر و پیچ و تاب بیشتر)

الگوریتم‌های رمزنگاری و رمزگشایی در یک فایل dll به برنامه اضافه می‌شوند (بنابراین این فایل قرار نیست تغییر کند). رشته رمزنگاری شده در فایل web.config قرار می‌گیرد. بدیهی است در هر بار اتصال به دیتابیس این رشته باید رمزگشایی شود اما سربار آن بسیار کم است و اصلا مشهود نیست. در هر حال این هزینه‌ای است که باید پرداخت شود. بدست آوردن ساده کانکشن استرینگ یعنی امکان پاک کردن سریع کل اطلاعات شما.

د)اگر سرور dedicated است حتما از روش windows authentication استفاده کنید
برای مثال یک سرور dedicated مخصوص کار ویژه‌ای تهیه کرده اید یا در شبکه اینترانت یک شرکت برنامه شما نصب شده است.
روش اعتبار سنجی از نوع ویندوزی برای اتصال به اس کیوال سرور نسبت به حالت sql server authentication امن تر است، زیرا نیازی نیست تا در وب کانفیگ نام کاربری یا پسوردی را مشخص نمائید و همچنین در این حالت پسوردها در شبکه منتقل نمی‌شوند (در حالت sql server authentication اینطور نیست). اما عموما در هاست‌های اشتراکی برای ساده تر کردن کار ، از این روش استفاده نمی‌کنند.
بنابراین در اینجا حتی اگر شخصی به رشته اتصالی شما دسترسی پیدا کند، کار خاصی را نمی‌تواند انجام دهد چون هیچگونه نام کاربری یا پسوردی در آن لحاظ نشده است.
در این روش به صورت پیش فرض از اکانت ASP.Net استفاده می‌شود. یعنی تمام برنامه‌ها محدود به یک اکانت خواهند شد.
برای تغییر این مورد دو کار را می‌توان انجام داد : استفاده از impersonation یا مطالعه قسمت بعد (ه)
توصیه: از روش impersonation به دلیل اینکه باید نام کاربری و کلمه عبور را باز هم به صورت واضحی ذکر نمود اجتناب کنید.

ه)ایجاد application pool مجزا به ازای هر برنامه ASP.Net در ویندوزهای سرور
Application pool که برای اولین بار در ویندوز سرور 2003 معرفی شده جهت ایزوله کردن برنامه‌های ASP.Net بکار برده می‌شود. به این صورت می‌شود برای هر pool یک اکانت ویندوزی مجزا تعریف کرد . حال می‌توان به این اکانت در اس کیوال سرور دسترسی داد. به این صورت برنامه‌های مختلف تحت یک اکانت واحد (یوزر asp.net) کار نکرده (می‌توانند هم کار کنند، اما امکان تعریف identity جدید برای کاربر آن در IIS وجود دارد) و ضریب امنیتی بالاتری را تجربه خواهید کرد (در تکمیل روش (د))

نظرات خوانندگان

نویسنده: نیما
تاریخ: ۱۳۸۷/۱۰/۱۶ ۲۳:۱۰:۰۰

سلام آقای نصیری.خوبین؟من چجوری میتونم مراتب تشکرم روبه شما اعلام کنم.فقط میتونم دعا کنم خدا کار شما رو همیشه راه بندازه.ان شا الله.
من برای انکریپشن از این کتابخونه استفاده میکنم:
http://www.codeproject.com/KB/security/SimpleEncryption.aspx
من برای ویندوز یه فایل ایکس ام ال برای پیکر بندی ایجاد میکنم(شبیه app.config) که مواردی که برای برنمم لازم هست رو با الگوریتم TripleDES و با یه رشته عجیب غریب که توش کاراکترهای عجیب غریب داره کد میکنم.الان موردی که هست اینه که من برای پروژه DAL ام باید این فایل رو باز کنم رشته کانکشن استرینگ مورد نظر رو دیکد کنم و بعد ازش استفاده کنم.آیا این قیمتیه که باید برای امنیت بدیم؟راه بهتری وجود داره؟یعنی میشه یه بار این رشته رو دیکد کرد و در حافظه نگه داشت؟اما خوب برای این کار باید همیشه یه نمونه از کلاس DAL وجود داشته باشه.
برای نت هم آیا به صرفه هست یه پروایدر دیگه برای دیکد بنویسیم که از غیر قابل برگشت بودن اون مظمئن بشیم؟
ممنون از لطف و محبت شما
همیشه موفق باشید
نیما

نویسنده: حسین
تاریخ: ۱۳۸۷/۱۰/۱۶ ۲۳:۳۰:۰۰

مشترک فیدت شدیم

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۷ ۰۰:۰۹:۰۰

@ نیما
- در وب برای اینکه این کاراکترهای عجیب و غریب مشکل ساز نشوند یکبار دیگر هم اطلاعات رمزنگاری شده را از فیلتر base64 encoding عبور می‌دهند. به این صورت مشکلی برای نگهداری آن‌ها در فایل‌ها وجود نخواهد داشت.
- نگهداری اطلاعات حساس در حافظه به صورت plain کار اشتباهی است چون دامپ حافظه ویندوز و یا تمام سیستم عامل‌های دیگر کار ساده‌ای است. برنامه‌های زیادی هستند که پروسس‌های ویندوز را لیست می‌کنند و به شما اجازه می‌دهند حافظه آن‌ها را دامپ کنید (به راحتی چند کلیک). استخراج اطلاعات حساس هم از یک فایل دامپ تر و تازه زیاد مشکل نیست.
- سرعت الگوریتم‌های رمزنگاری واقعا بالا است. پیاده سازی‌های خیلی خوبی هم دارند. بنابراین زیاد نگران این سربار نباشید. چون در حد یک رشته ساده و امثال آن اصلا سربار به حساب نمی‌آیند و بسیار سریع عمل می‌کنند.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۰۷/۱۷ ۱۱:۲۶

خیلی ممنون مقاله خیلی مفیدی هست
لطفا در صورت امکان یه لینک دیگه برای روش (ج) که خودتون ازش استفاده می‌کنید معرفی کنید که این روش رو با مثال توضیح داده باشه. لینکی که معرفی کردید غیر قابل دسترس هست. البته این به خاطر این هست که این مطلب سال 87 نوشته شده.
به نظرتون امکان داره این فایل dll که الگوریتم‌های رمزگزاری و رمزگشایی داخلش نوشته می‌شن به دست مهاجم بیفته؟ یعنی dllهای دایرکتوری bin پروژه به دست مهاجم بیفته؟
خیلی ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۷ ۱۲:۲۰

- این مطلب هست (البته این خیلی ساده است؛ ولی ایده‌اش مهم است).
- بله (در صورت دسترسی به سرور و یا وجود باگ LFI). البته روش‌های obfuscation می‌تونه در این حالت مفید باشه.