

در [سلسله مقالات](#) قبلی ما فصل اول از بخش اول را به پایان بردیم و مبحث آشنایی با CLR و نحوه اجرای برنامه را یاد گرفتیم. در این سلسله مقالات که مربوط به فصل دوم از بخش اول است، در مورد نحوه ساخت و توزیع برنامه صحبت می‌کنیم.

در طی این سال‌ها ویندوز به ناپایداری و پیچیدگی متهم شده است. صرف نظر از این که ویندوز شایستگی این اتهامات را دارد یا خیر، این اتهامات نتیجه‌ی چند عامل است:

اول از همه برنامه‌ها از dll هایی استفاده می‌کنند که بسیاری از آن‌ها نوشته‌ی برنامه نویسانشان نیست و توسط توسعه دهندگان دیگر ارائه شده‌اند و توسعه دهندگان مربوطه نمی‌توانند صد در صد مطمئن شوند که افراد دیگر، به چه نحوی از dll آن‌ها استفاده می‌کنند و در عمل ممکن هست باعث دردهای زیادی شود که البته این نوع مشکلات عموماً از قبل خودشان را نشان نمی‌دهند، چرا که توسط سازنده‌ی برنامه تست و دیباگ شده‌اند.

موقعی کاربرها بیشتر دچار دردر می‌گردند که برنامه‌های خودشان را به روز می‌کنند و عموماً شرکت‌ها در آپدیت‌ها، فایل‌های جدید زیادی را روی سیستم کاربر منتقل می‌کنند که ممکن هست سازگاری با فایل‌های قبلی موجود نداشته باشند و از آنجا که همیشه تست این مورد برای توسعه دهنده امکان ندارد، به مشکلاتی بر می‌خورند و نمی‌توانند صد در صد مطمئن باشند که تغییرات جدید باعث تأثیر ناخوشایند نمی‌شود.

مطمئن هستم شما بسیاری از این مشکلات را دیده‌اید که کاربری یک برنامه را نصب می‌کند و شما متوجه می‌شوید که یک برنامه‌ی از قبل نصب شده به خاطر آن دچار مشکل می‌شود و این مورد به [DLL hell](#) مشهور هست. این مورد باعث ایجاد ترس و لرز برای کاربر شده تا با دقت بیشتری به نصب برنامه‌ها بپردازد.

دومین مورد مربوط به نصب برنامه‌ها است که متهم به پیچیدگی است. امروزه هر برنامه‌ای که روی سیستم نصب می‌شود، بر همه جای سیستم تأثیر می‌گذارد. یک برنامه را نصب می‌کنید و به هر دایرکتوری تعدادی فایل کپی می‌شود. تنظیمات رجیستری را آپدیت می‌کند، یک آیکن روی دسکتاپ و یکی هم start menu یا مترو را اضافه می‌کند. به این معنی که یک نصب کننده به عنوان یک موجودیت واحد شناخته نمی‌شود. شما نمی‌تونید راحت از یک برنامه بکاپ بگیرید. باید فایل‌های مختلفش را جمع آوری کنید و تنظیمات رجیستری را ذخیره کنید. عدم امکان انتقال یک برنامه به یک سیستم دیگر هم وجود دارد که باید مجدد برنامه را نصب کنید و نکته‌ی نهایی، حذف برنامه که گاهی اوقات حذف کامل نیست و به شکل نامنظم و کثیفی اثراتش را به جا می‌گذارد.

سومین مورد امنیت هست. موقعی که کاربر برنامه‌ای را نصب می‌کند انواع فایل‌ها از شرکت و تولید کننده‌های مختلف روی سیستم نصب می‌شوند. گاهی اوقات برنامه‌ها بعضی از فایل هایشان را از روی اینترنت دریافت می‌کنند و کاربر اصلاً متوجه موضوع نمی‌شود و این فایل‌ها می‌توانند هر کاری از حذف فایل از روی سیستم گرفته تا ارسال ایمیل را انجام بدهند که این موارد باعث وحشت کاربرها از نصب یک برنامه‌ی جدید می‌شود که این مورد را با قرار دادن یک سیستم امنیت داخلی با اجازه و عدم اجازه کاربر می‌شود تا حدی رفع کرد.

دات نت فریمورک هم این معضل را به طور عادی در زمینه‌ی DLL hell دارد که در فصل آتی حل آن بررسی خواهد شد. ولی برخلاف COM، نوع‌های موجود در دات نت نیازی به ذخیره تنظیمات در رجیستری ندارند؛ ولی متأسفانه لینک‌های میانبر هنوز وجود دارند. در زمینه امنیت دات نت شامل یک مدل امنیتی به نام [Code Access security](#) می‌باشد؛ از آنجا که امنیت ویندوز بر اساس هویت کاربر تأمین می‌شود. [code access security](#) به برنامه‌های میزبان مثل sql server اجازه می‌دهد که مجوز مربوطه را خودشان بدهند تا بدین صورت بر اعمال کامپوننت‌های بار شده نظارت داشته باشند که البته این مجوزها در حد معمولی و اندک هست. ولی اگر برنامه خود میزبان که به طور محلی روی سیستم نصب می‌شوند، باشد دسترسی کامل به مجوزها را دارد. پس بدین صورت کاربر این اجازه را دارد که بر آن چیزی که روی سیستم نصب یا اجرا می‌شود، نظارت داشته باشد تا کنترل سیستم به طور کامل در اختیار او باشد.

در قسمت بعدی با نحوه توزیع برنامه آشنا خواهیم شد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۴/۰۵/۲۵ ۲۰:۴۱

دات نت فریمورک هم یک معضل بزرگ در زمینه‌ی `DLL hell` دارد که برای حل مشکل در پخش کردن فایل‌ها در جای جای هارد دیسک راه درازی در پیش است.

GAC به همین منظور تدارک دیده شد. در GAC می‌توان چندین نگارش یک DLL دات نتی را ذخیره کرد، بدون اینکه برنامه‌های مختلف دات نتی با مشکل نصب یا ارتقاء مواجه شوند.