

عنوان	توسعه نرم افزار برای سیستم عامل ویندوز موبایل، بخش ششم: امنیت دستگاه و نصب نرم افزار
عنوان اصلی	Windows Mobile App Development Part 6: Device Security and Application Deployment
کلمات کلیدی	C#, Win Mobile, XML, .NET, Dev
مؤلف	mstruys, dougturn
مرجع	http://www.codeproject.com
سطح	مبتدی
مترجم	مهدی عبداللهی http://m0911.wordpress.com
تاریخ انتشار	۱۱ فروردین ۱۳۸۹
تعداد صفحه	۱۳
مطالب مرتبط	توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش اول: ایجاد نخستین برنامه توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش دوم: شبیه ساز دستگاه و مدیریت شبیه ساز توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش سوم: توسعه ی برنامه با WinForm توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش چهارم: کنترل سفارشی و GPS توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش پنجم: مقدمه ای بر SQL Server CE توسعه ی نرم افزار برای سیستم عامل ویندوز موبایل، بخش هفتم: توسعه برای وب موبایل
فایل های ضمیمه	

توسعه ی نرم افزار ویندوز موبایل شباهت زیادی به توسعه ی نرم افزار در دستکتاب دارد به ویژه زمانی که یکی از دو زبان ویژوال بیسیک یا ویژوال سی شارپ دات نت را استفاده می کنید. شما همان ابزارهای توسعه ی برنامه های ویندوز دستکتاب را برای ویندوز موبایل هم استفاده می کنید لیکن تفاوت هایی نیز بین این دو محیط هست. بسته به شرکت سازنده یا اپراتور تلفن همراه که دستگاه ویندوز موبایل را عرضه می کنند، این دستگاه ها دارای تنظیمات امنیتی متنوعی هستند. از دیدگاه یک توسعه دهنده امنیت دستگاه تعیین کننده ی اجرا شدن یا نشدن یک برنامه روی آن و همچنین منابعی از سیستم است که برنامه به آن ها دسترسی دارد. این مسأله در چگونگی توزیع یا نصب نرم افزار به ویژه زمانی که با دستگاه (گوشی) های متنوعی از سازنده گان مختلف سر و کار دارید، اهمیت بیشتری پیدا می کند.

این مقاله اطلاعات مفیدی را در زمینه ی امنیت دستگاه های ویندوز موبایل، چگونگی آزمایش تنظیمات مختلف امنیتی با استفاده از ابزار مدیر امنیت (security manager) و ویژوال استودیو ۲۰۰۸ و در نهایت نحوه ی نصب درست برنامه ها روی دستگاه های ویندوز موبایل به شما ارائه می دهد.

امنیت در دستگاه های ویندوز موبایل

دستگاه های مختلف ویندوز موبایل بسته به شرکت سازنده شان تنظیمات امنیتی متنوعی دارند. یک دستگاه ممکن است به طور کامل نصب هر برنامه ای را بپذیرد و دستگاه دیگر ممکن است به هنگام اولین نصب یا اجرای یک برنامه از کاربر تأییدیه بگیرد. یا این که دستگاه به کلی در برابر نصب برنامه هایی که تأییدیه ی امنیتی از سازنده ی آن ندارد، مسدود باشد.

یک دلیل وجود تنظیمات امنیتی مختلف این است که اغلب دستگاه های پیشرفته ی ویندوز موبایل دارای امکانات تلفن همراه هستند و برای ارتباط در یک شبکه ای که در اختیار اپراتور تلفن همراه است، به کار می روند. لذا یکی از دلایل دستگاه ویندوز موبایل امنیت خود شبکه است. در واقع اپراتور های تلفن همراه می خواهند شبکه شان را در برابر نرم افزار هایی که به راحتی توسط کاربران نصب می شوند، محافظت کنند. به همین دلیل نمی توان هر برنامه ای را - حتی اگر مخصوص ویندوز موبایل نوشته شده باشد - روی یک دستگاه ویندوز موبایل نصب کرد.

بدین ترتیب برای تأمین امنیت مورد درخواست اپراتور های تلفن همراه، دستگاه های ویندوز موبایل دارای تنظیمات امنیتی مختلفی هستند. لیکن امنیت اعمال شده در یک دستگاه ویندوز موبایل صرفاً به دلیل اپراتور تلفن همراه یا شرکت استفاده کننده ی آن نیست، بلکه به نوع استفاده ی آن نیز بستگی دارد. امنیت دستگاه های ویندوز موبایل ضروری و به طور کامل داخل سیستم عامل تعبیه شده است. لایه های مختلف امنیتی با هم ترکیب شده اند تا در نهایت بر مبنای یک تنظیم امنیتی مشخص تعیین کنند که آیا برنامه ها می توانند روی دستگاه نصب و اجرا شوند یا نه و اگر پاسخ به این پرسش مثبت است، چگونه؟

مجوزهای نصب نرم افزار

بسته به تنظیمات امنیتی یک دستگاه خاص ویندوز موبایل برنامه ها ممکن است روی آن اجرا شوند یا این که به کلی اجرای آن ها ممنوع باشد. مجوز های زیر برای اجرای برنامه ها روی یک دستگاه ویندوز موبایل تعریف شده اند:

- ویژه (Privileged): نرم افزار هر کاری روی دستگاه می تواند انجام دهد، دسترسی کامل به سیستم فایل و رجیستری سیستم دارد و همچنین می تواند تأییدیه برای اجرای نرم افزار های دیگر روی دستگاه نصب کند.
- عادی (Normal): نرم افزار محدودیت هایی برای اجرا دارد. توابع Win32 API را نمی تواند فراخوانی کند، نمی تواند در بخش های محافظت شده ی رجیستری تغییر دهد یا روی فایل های سیستم بنویسد یا تأییدیه نصب کند.
- مسدود (Blocked): نرم افزار به هیچ وجه مجوز اجرا ندارد.

تأییدیه های نرم افزار

برای به دست آوردن مجوز اجرا برای نرم افزار تان می توانید برای آن نشان تأییدیه بگیرید. دستگاه های ویندوز موبایل دو نوع تأییدیه دارند. تأییدیه ی ویژه و تأییدیه ی عادی که مطابق با نوع تأییدیه سطح دسترسی آن هم تعیین می شود. از آن جا که یک گروه خاص (مانند شرکت سازنده یا اپراتور تلفن همراه) این تأییدیه ها را ارائه می دهند، شما باید با این مراکز برای دریافت تأییدیه ی نرم افزار تان در ارتباط باشید. اگر چه همانند یک ISV (فروشنده ی مستقل نرم افزار = Independent Software Vendor) ممکن است بخواهید که نرم افزار تان روی دستگاه های متنوعی کار کند. شرکت Mobile2Market چنین کاری را آغاز کرده است و هر تولید کننده ی نرم افزار می تواند محصولش را برای

دریافت تأییدیه به آن ها ارایه نماید. دریافت تأییدیه از Mobile2Market به این معنا است که شما یا سازمان شما به عنوان یک تولید و توزیع کننده ی نرم افزار به رسمیت شناخته می شوید.

سطوح دسترسی نرم افزار

سطوح دسترسی مختلف مشخص می کنند که یک نرم افزار بدون تأییدیه چه کارهایی می تواند روی دستگاه ویندوز موبایل انجام دهد. این سطوح دسترسی مختلف، لایه (tier) نامیده می شوند. ویندوز موبایل دو سطح دسترسی به شرح زیر دارد:

امنیت یک لایه (One-tier security): نرم افزار دارای تأییدیه با مجوز ویژه روی دستگاه اجرا می شود و دسترسی کامل به همه ی بخش های دستگاه دارد. نرم افزار بدون تأییدیه ممکن است بسته به تنظیمات امنیتی دستگاه بتواند روی آن اجرا شود. در این صورت با همان مجوز ویژه اجرا خواهد شد.

امنیت دو لایه (Two-tier security): نرم افزار دارای تأییدیه می تواند روی دستگاه اجرا شود و بسته به نوع تأییدیه اش می تواند با مجوز عادی یا ویژه اجرا شود. نرم افزار بدون تأییدیه بسته به تنظیمات امنیتی دستگاه ممکن است اجرا شود، لیکن در صورت اجرا با مجوز عادی اجرا خواهد شد.

تنظیمات امنیتی دستگاه

حال که تفاوت مجوز اجرای نرم افزار با نشان تأییدیه ی نرم افزار را متوجه شدید، یک پرسش مهم می ماند. آیا نرم افزار شما با وجود نداشتن تأییدیه می تواند روی یک دستگاه خاص ویندوز موبایل نصب و اجرا شود؟ چند تنظیم امنیتی استاندارد برای دستگاه های ویندوز موبایل تعریف شده است که برخی به شما این مجوز را خواهند داد که برنامه ی بدون تأییدیه را اجرا نماید:

- بدون امنیت (Security Off): همه ی نرم افزار ها صرف نظر از داشتن یا نداشتن تأییدیه می توانند با مجوز ویژه روی دستگاه اجرا شوند.
- تأیید یک لایه ای توسط کاربر (One-Tier Prompt): همه ی نرم افزار های دارای نشان تأییدیه ی امنیتی، با مجوز ویژه روی دستگاه اجرا می شوند. اجرای نرم افزار های بدون تأییدیه با تصمیم کاربر صورت می گیرد. اگر کاربر اجرای نرم افزار بدون تأییدیه را بپذیرد، نرم افزار با مجوز ویژه اجرا خواهد شد.
- تأیید دو لایه ای توسط کاربر (Two-Tier Prompt): همه ی نرم افزار های دارای تأییدیه روی دستگاه اجرا خواهند شد لیکن مجوز اجرای ویژه یا عادی بسته به نوع تأییدیه ی امنیتی تعیین می شود. نرم افزار های بدون تأییدیه، با اجازه ی کاربر می توانند اجرا شوند لیکن در صورت اجرا، فقط با مجوز عادی اجرا خواهند شد.
- تأییدیه شخص ثالث (Third-Party Signed): فقط نرم افزار های دارای تأییدیه ی امنیتی می توانند روی دستگاه اجرا شوند. مجوز اجرای ویژه یا عادی بسته به نوع تأییدیه ی نرم افزار خواهد بود. برای دریافت تأییدیه باید در یک سیستم توسعه ی نرم افزار مانند Mobile2Market ثبت نام کنید. اغلب دستگاه های ویندوز موبایل که توسط اپراتور های تلفن همراه خریداری و ارایه می شوند تأییدیه ی Mobile2Market را با خود دارند.
- مسدود (Locked): فقط نرم افزار های دارای تأییدیه ی امنیتی می توانند روی دستگاه اجرا شوند. مجوز اجرای ویژه یا عادی بر حسب نوع تأییدیه مشخص می شود.

مجوز اجرا و فایل های کتابخانه ی DLL

یک مورد دیگر هم هست که باید فکری به حال آن بکنیم و آن هم فایل های کتابخانه ی دینامیکی است که از این پس از همان اصطلاح آشنای DLL را استفاده می کنیم. همانند برنامه ها DLL ها هم می توانند بدون تأییدیه باشند یا این که تأییدیه با مجوز اجرای ویژه یا عادی داشته باشند. از آن جا که DLL ها از داخل برنامه ها استفاده می شوند یک سؤال جالب پیش می آید. اگر مثلاً یک برنامه با مجوز دسترسی ویژه یک DLL بدون تأییدیه را که آن هم از داخل خود یک تابع مورد اطمینان Win32 API را فرا می خواند، استفاده کند چه اتفاقی می افتد؟ اگر فایل DLL هم به همان سطح دسترسی برنامه ارتقا داده شود که این یک نقص امنیتی خواهد بود.

از این رو قوانین زیر برای برنامه ها ایجاد شده است تا مطابق آن به DLL ها دسترسی پیدا کنند:

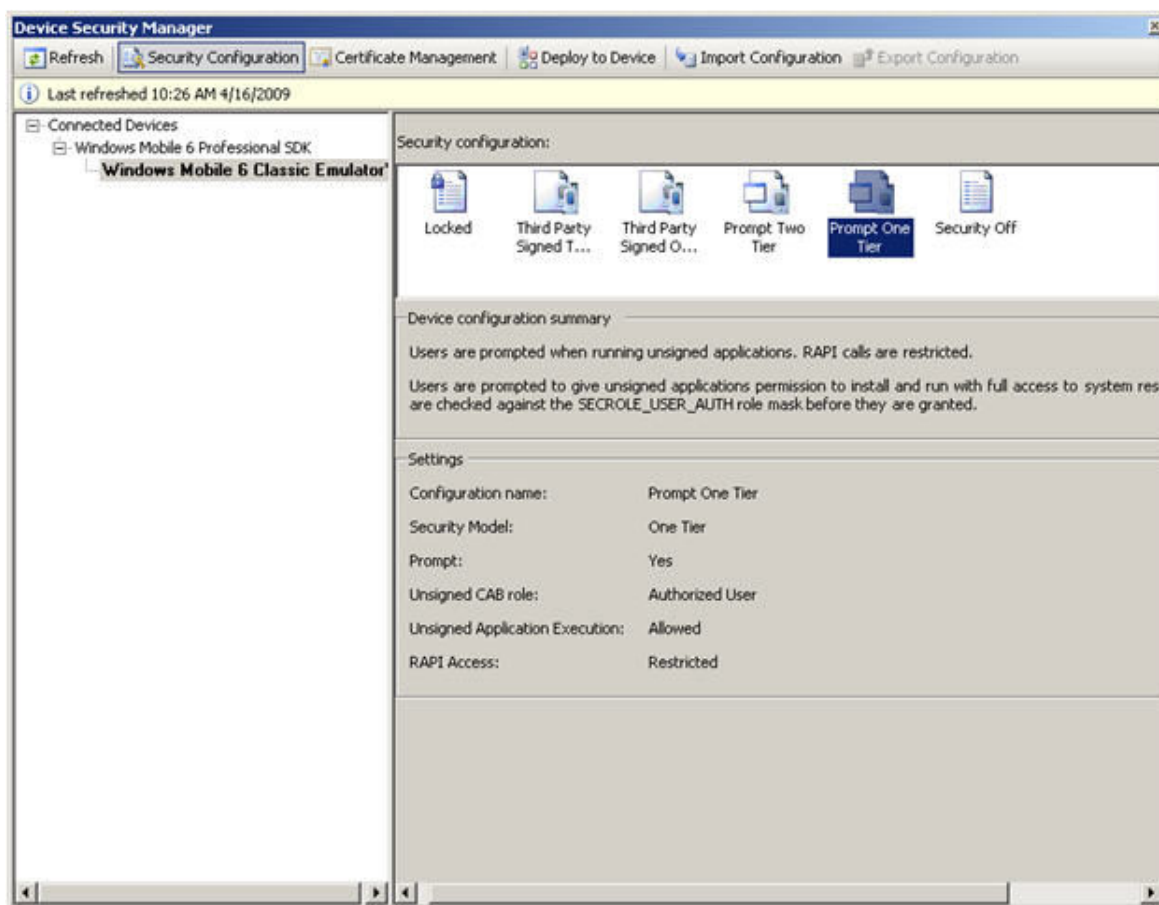
ترکیب برنامه و DLL	بدون امنیت	تأییدیه ی یک لایه ای	تأییدیه ی دو لایه ای
برنامه ی بدون مجوز			
DLL بدون مجوز	ویژه	ویژه با اجازه ی کاربر	عادی با اجازه ی کاربر
DLL با دسترسی عادی	ویژه	ویژه با اجازه ی کاربر	عادی با اجازه ی کاربر
DLL با دسترسی ویژه	ویژه	ویژه با اجازه ی کاربر	عادی با اجازه ی کاربر
برنامه ی با مجوز عادی			
DLL بدون مجوز	ویژه	ویژه	عادی با اجازه ی کاربر
DLL با دسترسی عادی	ویژه	ویژه	عادی
DLL با دسترسی ویژه	ویژه	ویژه	عادی
برنامه ی با مجوز ویژه			
DLL بدون مجوز	ویژه	ویژه	غیر مجاز
DLL با دسترسی عادی	ویژه	ویژه	غیر مجاز
DLL با دسترسی ویژه	ویژه	ویژه	ویژه

آزمایش برنامه در حالت های مختلف تنظیمات امنیتی

یکی از ابزار های ویژه استودیو ۲۰۰۸ «مدیر امنیت دستگاه» یا همان Device Security Manager است که به شما امکان می دهد که تنظیمات امنیتی فعال روی یک دستگاه ویندوز موبایل شبیه سازی شده را به دست آورید.

نکته: شما زمانی می توانید تنظیمات امنیتی را روی یک دستگاه فیزیکی ویندوز موبایل تغییر دهید که تنظیمات امنیتی فعلی آن، به شما اجازه ی این کار را بدهد. دقت کنید که اگر تنظیمات امنیتی یک دستگاه را به حالت مسدود (Locked) تغییر دهید، این حالت قابل بازگشت نیست. پس در این مورد احتیاط کنید. البته مدیر امنیت دستگاه در این حالت به شما هشدار خواهد داد. با استفاده از شبیه ساز دستگاه به راحتی می توانید تمامی حالت های امنیتی را آزمایش نمایید.

با استفاده از برنامه ی Device Security Manager می توانید هر تنظیم امنیتی را که از پیش تعریف شده است، فعال کنید یا تنظیم امنیتی جدید را وارد (import) برنامه کنید. به علاوه می توانید فهرست تمامی تأییدیه های امنیتی نصب شده روی دستگاه ویندوز موبایل یا شبیه ساز دستگاه را به دست آورید.



شکل ۱: برنامه ی Device Security Manager

تنظیمات امنیتی دستگاه ها می تواند داخل پرونده های XML تعریف شوند. مثال زیر را داخل ویژوال استودیو ۲۰۰۸ می توانید ببینید:

```
<wap-provisioningdoc />
  <characteristic type="SecurityPolicy" />
    <parm name="4102" value="1" />
    <parm name="4122" value="0" />
    <parm name="4123" value="0" />
  </characteristic />
</wap-provisioningdoc />
```

آزمایش سطوح مختلف دسترسی در تنظیمات امنیتی مختلف

حال که یاد گرفتید چطور تنظیمات امنیتی مختلف را اعمال کنید، چند آزمایش را انجام می دهیم. قطعه کد زیر به عنوان یک مثال ساده می تواند نقطه ی آغاز باشد. این قطعه کد از یک تابع تأیید شده ی Win32 API استفاده می کند. اگر شما این قطعه کد را در یک برنامه یا DLL قرار دهید و برای هر کدام تأییدیه (certificate) های مختلف بگیرید، یا حتی بدون تأییدیه اجرا کنید در نهایت خواهید توانست کارکرد برنامه را در تنظیمات مختلف امنیتی آزمایش کنید.

```
[DllImport("coredll.dll")]
public extern static void PowerOffSystem();
private void btnInstructions_Click(object sender, EventArgs e)
{
    PowerOffSystem();
}
```

<http://m0911.wordpress.com>

اگر برنامه مجاز به فراخوانی API های تأیید شده باشد، با اجرای آن می بینید که بلافاصله شبیه ساز دستگاه بسته می شود چون خاموش شده است و اگر برنامه مجاز به این فراخوانی نباشد هیچ اتفاقی نمی افتد و API نادیده گرفته می شود. اگر شما بخواهید در نواحی محافظت شده ی سیستم فایلی یا رجیستری دستگاه بنویسید با خطای دسترسی غیر مجاز (UnauthorizedAccessException) مواجه خواهید شد.

نصب برنامه

تنظیمات امنیتی دستگاه روی نصب برنامه ها هم اثر می گذارد. تا اینجای مقاله، فرض بر این بوده است که نصب نرم افزارها از داخل ویژوال استودیو ۲۰۰۸ نصب می شوند و تنظیمات امنیتی نیز از طریق Device Security Manager صورت می گیرد. صد البته هنگامی که در حال برنامه نویسی هستید این امکان، کار شما را خیلی آسان می کند، لیکن در نهایت شما برنامه نان را منتشر خواهید کرد و به همین دلیل باید تجربه مطلوبی در زمینه ی نصب برنامه برای کاربران نهایی (end users) داشته باشید و این یعنی برنامه ی شما به صورت یک فایل CAB یا MSI منتشر گردد. اگر برنامه تان به صورت فایل CAB منتشر شود، کاربر باید فایل را به داخل دستگاه ویندوز موبایل خودش کپی کند و توسط برنامه ی اکسپلورر آن را باز و اجرا نماید. اگر هم بخواهید فایل CAB را داخل یک فایل Microsoft Installer (یا همان MSI) قرار دهید کاربران می توانند از طریق ویندوز کامپیوتر دسکتاپ خودشان زمانی که دستگاه ویندوز موبایل شان از طریق برنامه ی ActiveSync در ویندوز ایکس پی یا برنامه ی Windows Mobile Device Center در ویندوز ویستا یا سون به کامپیوتر متصل است، برنامه را نصب کنند. اگر تنظیمات امنیتی دستگاه سختگیرانه باشد ممکن است اجازه ندهد که فایل CAB بدون امضا روی آن نصب شود. در واقع صرف تأییدیه داشتن برنامه تان کافی نیست بلکه باید برای فایل CAB هم امضای تأییدیه بگیرید. ابزار توسعه ی ویندوز موبایل ۶ یک برنامه ی خط فرمانی به نام cabsigntool.exe برای این کار دارد که از signtool.exe ی ویژوال استودیو ۲۰۰۸ استفاده می کند. برای راحتی کار بهتر است از خط فرمان ویژوال استودیو ۲۰۰۸ استفاده کنید که مجبور نباشید متغیر محیطی path را برای یافتن این برنامه مقدار دهی کنید. مثال زیر نحوه ی استفاده ی cabsigntool.exe را از طریق خط فرمان به شما نشان می دهد:

```
cabsigntool "C:\Users\UserVS2008\Documents\Visual Studio
2008\Projects\HOL6SampleApp\DVDsMobileCAB\Debug\DVDsMobileCAB.cab"
"C:\Users\UserVS2008\Documents\Visual Studio
2008\Projects\HOL6SampleApp\DVDsMobileCAB\Debug\DVDsMobileCAB.cab"
-f "C:\Program Files\Windows Mobile 6 SDK\Tools\Security\SDK Development
Certificates\SampleUnPrivDeveloper.pfx"
```

در این مثال فایل CAB با نام DVDsmobilecab.cab و تمام برنامه های اجرایی داخل آن، با یک تأییدیه ی آزمایشی - که داخل Windows Mobile 6 SDK است - امضا می شوند. برنامه ی cabsigntool.exe را در این مسیر هم می توانید پیدا کنید:

`\program files\Windows Mobile 6 SDK\Tools\Security`

با یک نگاه به دستورهای خط فرمانی بالا متوجه می شوید که با این همه دنگ و فنگ اگر قرار باشد هر بار موقع استفاده از cabsigntool.exe این پارامترها را وارد نمایید احتمال بروز خطای دستوری هست. پس بهتر آن است که در یک فایل دسته ای bat این دستورها را قرار دهید و هر بار آن را فراخوانی کنید.

به روز رسانی برنامه ها

بر خلاف برنامه های دسکتاپ که در ویژوال استودیو ۲۰۰۸ تولید می شوند در ویندوز موبایل امکان نصب به روش ClickOnce – که امکان به روز رسانی فایل های از قبل نصب شده را در صورت تغییر آن در نسخه ی جدید می دهد – وجود ندارد. لیکن شما می توانید کارکرد مشابه این را مثلا به وسیله ی وب سرویس ایجاد نمایید. در این مقاله روشی مشابه ClickOnce برای به روز رسانی برنامه های ویندوز موبایل خواهید دید. این مثال صرفا یک نقطه ی آغاز خواهد بود و به شما اطلاعات کافی خواهد داد که خودتان بتوانید به این روش برنامه های تان را به روز کنید. برای به روز رسانی خودکار برنامه های تان می توانید از وب سرویسی استفاده کنید که آخرین نگارش منتشر شده ی برنامه را اعلام و با نگارش برنامه ی مورد استفاده ی فعلی مقایسه کند. از آن جا که شما باید یک برنامه ی در حال اجرا را به روز رسانی کنید باید یک برنامه ی نصب کننده ی مجزا – همانند نصب کننده ای که از ابتدا برنامه تان را روی دستگاه نصب کرده است – بنویسید.

نصب یک برنامه ی جدید

برای نصب یک برنامه با قابلیت به روز رسانی خودکار باید برنامه تان را از طریق یک وب سایت ASP.NET نصب کنید. برای دانلود برنامه، کاربر باید سایت را برای پیدا کردن محل دانلود آن جستجو کند.

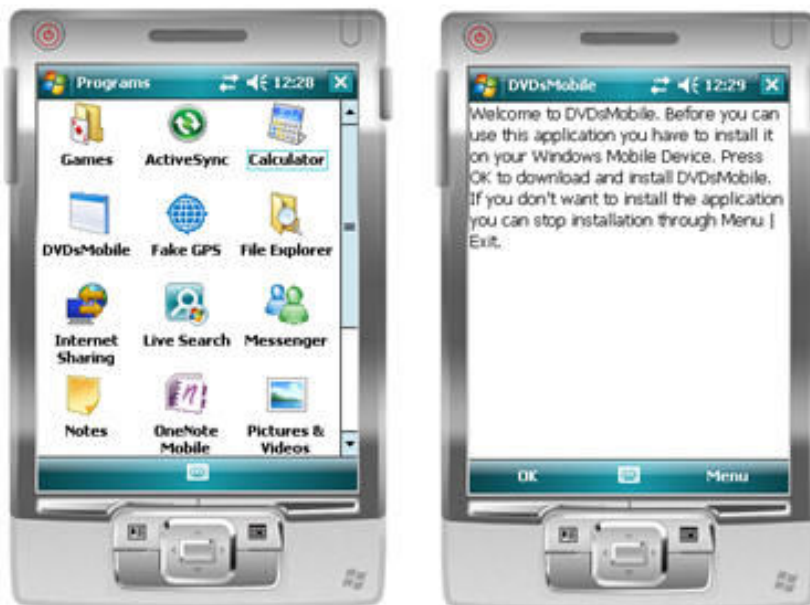


شکل ۲: وب سایت حاوی برنامه و دانلود به روز رسانی آن

در شکل ۲ شما یک وب سایت می بینید که یک برنامه ی جدید برای دانلود به صورت آماده دارد. با کلیک روی آن برنامه نصب نخواهد شد بلکه به جای آن یک برنامه ی نصب کننده روی دستگاه نصب می شود. برنامه ی نصب کننده در همان پوشه ای که برنامه ی اصلی در آن نصب خواهد شد قرار می گیرد. برنامه ی نصب کننده در عین حال یک میانبر هم نام با برنامه ی اصلی در پوشه ی Program Files ایجاد خواهد کرد.

<http://m0911.wordpress.com>

پس از دانلود برنامه ی به روز رسانی، کاربر می تواند در پوشه ی Programs روی آن کلیک و برنامه ی به روز رسانی را اجرا نماید. از آنجا که برنامه ی اصلی هنوز قابل دسترسی نیست برنامه ی به روز رسانی به جای آن اجرا می شود (شکل ۳)



شکل ۳: اولین اجرای برنامه

برنامه ی به روز رسانی از طریق پارامتر خط فرمان متوجه می شود که آیا توسط کاربر اجرا شده است یا از داخل برنامه ی اصلی. حالت دوم زمانی پیش می آید که به روز رسانی جدیدی داخل وب سایت منتشر شده باشد. کد زیر - که به عنوان بخشی از برنامه ی به روز رسانی داخل رویداد Form_Load اجرا می شود- نشان می دهد که چطور اجرای بار اول را از اجرای حالت به روز رسانی تشخیص دهیم.

```
private void MainForm_Load(object sender, EventArgs e)
{
    if (_args == null || _args.Length == 0 || _args[0].Length == 0)
    {
        _args = new string[1];
        _args[0] = initialURL;
        label1.Text = Properties.Resources.InstallString;
    }
    else
    {
        label1.Text = Properties.Resources.UpgradeString;
    }
}
```

زمانی که برنامه نصب می شود میانبر داخل پوشه ی Program Files را طوری تغییر می دهد که در اجرای بعدی که کاربر برنامه را از داخل پوشه ی Program Files اجرا می کند برنامه ی اصلی اجرا شود نه برنامه ی به روز رسانی.



شکل ۴: نصب برنامه و اجرای آن از طریق برنامه ی به روز رسانی (application updater)

پس از نصب برنامه توسط برنامه ی نصب کننده، برنامه ی اصلی به طور خودکار توسط نصب کننده اش اجرا می شود. این کار به محض پایان برنامه ی نصب توسط کاربر از طریق گزینه ی Menu | Exit انجام می شود. کد زیر نشان می دهد که چطور برنامه ی اصلی را از داخل برنامه ی نصب کننده اجرا کنیم. البته کد زیر همه ی آنچه شما به دنبالش هستید، نیست ولی می توانید به راحتی آن را دستکاری کنید و بسته به نیاز تان کم و زیادش کنید:

```
private void menuExit_Click(object sender, EventArgs e)
{
    if (label1.Text.Equals(Properties.Resources.InstallString) && ! appInstalled)
    {
        Close();
    }
    else
    {
        StartDVDsMobile();
    }
}

private void StartDVDsMobile()
{
    Process theApp = null;
    try
    {
        theApp = Process.Start(@"\Program Files\DVDsMobile\DVDsMobile.exe",
        Process.GetCurrentProcess().Id.ToString());
    }
    catch (Win32Exception exc)
    {
        if (exc.NativeErrorCode == ERROR_FILE_NOT_FOUND)
        {
```

```
        MessageBox.Show("DVDsMobile.exe not found.");
    }
    else if (exc.NativeErrorCode == ERROR_ACCESS_DENIED)
    {
        MessageBox.Show("No permissions to start DVDsMobile.exe.");
    }
}
}
```

در قطعه کد بالا مدیر رویداد `menuExit_Click` را می بینید که تعیین می کند آیا برنامه ی اصلی را اجرا نماید یا فقط از برنامه ی نصب کننده خارج شود. اگر نگارش جدیدتر برنامه نصب شده باشد، برنامه ی به روز رسانی، نگارش جدید را اجرا می کند و شناسه ی پردازش یا همان `process identification` خود را به داخل آن ارسال می کند که برنامه ی اصلی بتواند برنامه ی به روز رسانی را از حافظه خارج کند. با این که در ظاهر این کار ضروری به نظر نمی رسد اما از نظر این که توجه کاربر را به اجرای برنامه ی جدید پیش از اتمام برنامه ی فعلی جلب می کند مفید است.

هر بار که برنامه توسط کاربر اجرا می شود، می تواند یک وب سرویس از همان وب سایتی که برنامه ی نصب کننده را از آن دانلود کردیم، فراخوانی نماید. وب سرویس می تواند شماره ی آخرین نگارش تمامی اسمبلی ها مورد استفاده برنامه را اعلام نماید تا برنامه ی اصلی آن ها را با شماره ی نگارش های خودش مقایسه کند. اگر نگارش های جدید در وب سایت منتشر شده باشد برنامه یک اعلان به کاربر نشان می دهد که آیا می خواهد برنامه را به روز نماید یا این که نگارش فعلی را اجرا نماید. این کار باعث می شود که آغاز اجرای برنامه کمی طولانی تر شود ولی مزیت این را هم دارد که کاربر همیشه از آخرین نگارش استفاده می کند. برای ساده تر شدن کار، برنامه ی ما وصل بودن یا نبودن شبکه را بررسی نمی کند. البته در برنامه ی واقعی می توانید با استفاده از امکانات Stat & Notification Broker این امکان را فراهم نمایید.



شکل ۵: به روز رسانی خودکار در عمل

برای دانلود نگارش جدید برنامه، برنامه ی به روز رسانی از طریق یک کنترل مرورگر وب (Web Browser) مخفی وارد سایت می شود و برنامه را دانلود می کند. استفاده از کنترل مرورگر وب، کار دانلود برنامه را بدون نیاز به کار اضافی با وب سرویس امکان پذیر می نماید. کد مربوط به وب سایت را در این مقاله اشاره ای نکرده ام ولی شما می توانید یک روند کامل از این کار را در بخش هفتم همین سری مقاله ببینید. در کد زیر می بینید که چطور برنامه ی اصلی بررسی می کند که آیا نگارش جدیدترش منتشر شده است یا نه:

http://m0911.wordpress.com

```
private void MainFormNoTouch_Load(object sender, EventArgs e)
{
    this.Text = Properties.ResourcesNoTouch.MainFormNoTouchTitle;
    this.menuItemExit.Text = Properties.ResourcesNoTouch.MenuItemExitText;
    if (_processID != -1)
    {
        // We have just been updated, so kill our update process.
        Process.GetProcessById(_processID).CloseMainWindow();
    }
    else
    {
        DVDsMobileUpdateService.DVDsMobileUpdateService updateService =
            new DVDsMobileNoTouch.DVDsMobileUpdateService.DVDsMobileUpdateService();
        string[] latestDVDsMobileVersion = updateService.DVDsMobileVersion().Split(
            new char[] { '.' });
        string[] latestDVDsMobileVersionNoTouch =
            updateService.DVDsMobileVersionNoTouch().Split(
                new char[] { '.' });
        string[] currentDVDsMobileVersion = _callingAssembly.Split(
            new char[] { '.' });
        Version currentDVDsMobileVersionNoTouch =
            Assembly.GetExecutingAssembly().GetName().Version;
        bool update = false;
        update = (Convert.ToInt32(latestDVDsMobileVersion[0]) >
            Convert.ToInt32(currentDVDsMobileVersion[0])) ||
            (Convert.ToInt32(latestDVDsMobileVersion[1]) >
                Convert.ToInt32(currentDVDsMobileVersion[1])) ||
            (Convert.ToInt32(latestDVDsMobileVersionNoTouch[0]) >
                currentDVDsMobileVersionNoTouch.Major) ||
            (Convert.ToInt32(latestDVDsMobileVersionNoTouch[1]) >
                currentDVDsMobileVersionNoTouch.Minor);
        if (update)
        {
            if (MessageBox.Show(
                "A new version of DVDsUpdate is found! " +
                "Do you want to install the new version?",
                "New version available",
                MessageBoxButtons.YesNo,
                MessageBoxIcon.Question,
                MessageBoxDefaultButton.Button2) == DialogResult.Yes)
            {
                Process.Start(@"\Program Files\DVDsMobile\DVDsMobileAutoUpdater.exe",
                    @"http://192.168.2.102:5746/DownloadForm.aspx");
                Application.Exit();
            }
        }
    }
}
```

<http://m0911.wordpress.com>

در این قطعه کد مدیر رویداد Form_Load را برای بررسی این که نگارش جدیدتر منتشر شده است یا نه، استفاده می کنیم. بدین منظور دو متد مربوط به وب سرویس DVDsMocileUpdateService فرا خوانی می شوند. در عین حال برنامه شماره ی نگارش خودش را هم می خواند و اگر نگارش موجود در سایت جدیدتر باشد، یک پنجره ی پیغام برای کاربر نمایش داده می شود که از میان دو گزینه ی به روز رسانی و اجرای نگارش فعلی یکی انتخاب نماید. اگر کاربر گزینه ی به روز رسانی را انتخاب نماید، برنامه ی به روز رسانی اجرا می شود و برنامه ی اصلی، خودش را از حافظه خارج می کند. پس از نصب نگارش جدید روی دستگاه، برنامه ی به روز رسانی، نگارش جدید برنامه ی اصلی را اجرا می نماید.