Secure Smart Client Deployment with ClickOnce

Brian Noyes
IDesign Inc. (www.idesign.net)
brian.noyes@idesign.net

About Brian

- Principal Software Architect, IDesign Inc. (www.idesign.net)
- Microsoft MVP in ASP.NET
- Writing
  - MSDN Magazine, CoDe Magazine, The Server Side .NET, asp.netPRO, Visual Studio Magazine, .NET Developers Journal
  - Data Binding in Windows Forms 2.0, Addison-Wesley, expected release Fall 2005
- Speaking
  - Microsoft TechEd US and Malaysia, Visual Studio Connections, VSLive!, DevEssentials, INETA Speakers Bureau, MSDN Webcasts
- Participates in Microsoft Design Reviews
- Email: brian.noyes@idesign.net
- Blog: http://www.softinsight.com/bnoyes

Agenda

- Quick Intro to ClickOnce
- Code Access Security at a Glance
- ClickOnce Security Protections
- Elevating privilege through ClickOnce
- Configuring Trusted Publishers
- Working with Application Trust

Quick Intro to ClickOnce

- Smart Client deployment technology
  - Automatic deployment
  - Automatic / on-demand update
- Trustworthy deployment mechanism
  - Prevent harm to other apps or data on machine
    - At deployment
    - At runtime
- Best of two worlds:
  - Rich client user experience
  - Ease of maintenance and deployment of web applications

How ClickOnce Works

- Build Client App
- Publish it to deployment server
  - Visual Studio wizard
- Provide link to users
- User clicks on link
  - App downloads and caches on user machine
  - App executes in secure sandbox
  - App updates based on update policy if new version available

Demo

Simple ClickOnce Deployment

Code Access Security

- Component oriented security
- Based on identity or origin of code, not user
- Complex permissions schema
- Managed programmatically and administratively
- Restricts operations or access to resources based on code evidence

Demo

Configuring Code Access Security

ClickOnce Deployment

- Application executable and supporting files downloaded and cached on client
- Placed under user profile
  - Isolated by user
  - Isolated by application
  - Isolated by version
- Cannot perform any custom install steps
  - Use bootstrapper for pre-requisites
- Manifests are signed
  - Provides authentication and tamper detection

ClickOnce Deployment

- Two deployment modes
  - Launched
    - Like No-Touch Deployment in .NET 1.X
    - Must be connected every time app runs
    - Files cache on machine
    - No other artifacts added to machine
  - Installed
    - Supports disconnected operation
    - Files cache on machine
    - Start menu shortcut added
    - Add/Remove Programs Control Panel item added

ClickOnce Execution

- Protected by Code Access Security
- Permissions Determined By:
  - Address of deployment manifest
  - Application manifest permission specs
  - User prompting or trusted publishers
  - Other CAS policies on the machine that match
- ClickOnce creates Application Trust

User Prompting Permissions

- Configure security settings for application
- Publish application
- User machine does not have user prompting turned off (default)
- User is prompted at app launch

Demo

User Prompting ClickOnce Permissions

Trusted Publisher Permissions

- Prevent user prompting
- Enterprise environment
  - Admins can configure desktop one time on all client machines
- Install trusted publisher certificate
  - Certmgr management console or command line
  - Trusted Publishers store
  - Signing authority must be in Trusted Root
- ClickOnce application manifests signed by trusted publisher get requested permissions

Demo

Trusted Publishers Permissions

Application Trust

- Machine and User Policies
- One trust per app per version
- Created when app launches
  - User prompt accepted
  - Trusted publisher match
- Elevates app privilege to requested permissions in manifest
  - Applies to all assemblies that are part of manifest

App-Domain Evidence

- App Domain Evidence
  - Satisfied by all assemblies loaded into AppDomain of the launched application
  - Allows use of dynamically loaded assemblies that are not part of manifest
  - Used by host application (AppLaunch.exe) to set up requested security permissions

User Prompting

- Registry Key
  - HKLM\Software\Microsoft\.NETFramework\Security\TrustManager\PromptingLevel
  - String values:
    - Internet
    - LocalIntranet
    - MyComputer
    - TrustedSites
    - UntrustedSites
  - Values:
    - Enabled
    - Disabled
    - AuthenticodeRequired

User Prompting Reg Keys

- Meaning:
  - Enabled – Can prompt regardless of who signed the manifest
  - Disabled – If it needs to prompt, can't run
  - AuthenticodeRequired – Can only prompt if signed by a trusted root certificate
- Defaults:
  - "Internet" – "AuthenticodeRequired"
  - "LocalIntranet" – "Enabled"
  - "MyComputer" – "Enabled"
  - "TrustedSites" – "Enabled"
  - "UntrustedSites" – "Disabled"

Demo

Application Trust and Prompting Configuration

Administrative ClickOnce Security Management - MAGE

- MAnifest GEnerator
- Admin editing of manifests through GUI
- Signing of manifests

Demo

MAGE

Traditional CAS Effects

- Theory:
  - Can still configure custom code groups
    - Strong name evidence
  - Resulting permissions are union of all matching code groups
    - Includes ClickOnce Application Trust
    - Includes Built-in code groups
    - Includes custom code groups
- Beta 2:
  - Only Application Trust Permissions are granted

Summary

- ClickOnce provides robust, secure mechanism for auto-deployment and update of smart client apps
- Prefer trusted publishers for enterprise environments
- Understand CAS affects on execution

- Email: brian.noyes@idesign.net
- Blog: http://www.softinsight.com/bnoyes