.Net Security

Jim Fawcett

CSE686 – Internet Programming

Summer 2005

References

- Developmentor Slides from MSDNAA, Keith Brown
- Introduction to Evidence-based Security in .Net Framework, Brad Merrill, http://www.dcl.hpi.uni-potsdam.de/LV/Components04/VL7/05a_Security-detailed.pdf
- Securing, Deploying and Maintaining .Net Applications, Patrick Tisseghem, www.u2u.net

Agenda

- Threats
- Windows Role-Based Security
- Code Access Security

Basic Security Issues

- Confidentiality
  - Disclose information only to authorized users
- Integrity
  - Ensure that data is not modified without authorization
- Availability
  - Decide who has access to information and how to make access effective
- Authentication
  - Identify a user securely
- Authorization
  - Define a set of allowed actions for authorized users
- Non repudiation
  - Log users, their actions, and the objects used.

Security Models

- Windows and .Net
  - Role-based
    - Authenticate and authorize users, groups, and accounts (System, Local service, Network service)
    - Actions are authorized through permissions
  - Evidence-based or Code Access Security (CAS)
    - Code is elevated to the security status of a user.
    - Authorization is based on evidence
      - url, zone, publisher, strong name, custom assembly attributes
    - Actions are authorized through policies

Win Security Definitions

- Definitions for *people* and *groups* of people
  - SID – Security IDentifier
    - Data structure used to identify user or group.
  - Access Token
    - A data structure that holds a SID for a security principal, the SIDs for each group the principal belongs to, and a list of the principal's priviledges on the local computer.
  - Principal
    - An account holder that is assigned a SID for access to resources, e.g., user, group, service, or computer.

Win Security Definitions

- Definitions for *objects*
  - Files, directories, kernel objects
  - ACL – Access Control List
    - Set of permissions for a specific object or a set of the object's properties.
    - Discretionary (DACL) and System (SACL) are sub-groups.
  - Security Descriptor
    - A data structure holding information about a protected object, e.g., who can access, in what way, whether audited.

Win Security Definitions

- Combinations of *people* and *objects*
  - Security Context
    - Set of rules for a user's actions on a protected object
    - Combination of user's access token and object's security descriptor
  - Security Policy
    - Rules that define the allowable contexts and mandatory groups.

Role-Based Security

- Use role-based security in programs to control access to methods or properties at run-time.
- Host authenticates user and provides identity and role information to CLR.
  - Uses NTFS access control lists, IIS security settings.
- CLR makes that information available to code via APIs and permission demands.
- Can isolate security from code using attributes defined in System.Security or EnterpriseServices
  - System.Security is limited to Windows user groups
  - EnterpriseServices uses COM+ roles
    - Classes have to inherit from EnterpriseServices
- Which to choose?
  - If application has both managed and unmanaged use COM+.
  - If application is entirely managed then System.Security is appropriate.

Code Access Security

- Goals
  - End-user experience
    - Managed apps just run
    - Safe defaults, no run-time decisions needed
  - Administrator
    - All settings in one place and easy to customize
    - Simple policy model
    - Security administration tools
      - .Net configuration, CASPOL
  - Developer
    - Focus on application, security comes free
    - Easy to understand and extend when needed

Mobil Code

- Old Model
  - Obtained from a network, often via a web page.
  - Without CAS have either full trust or no trust.
  - User decides whether to run.
  - If run, code has all the user's priviledges.
  - Inproc COM component, when loaded, becomes part of the process.
  - Can't distinguish between library code and original application code.
- CAS model
  - Operation based on evidence.
  - Allowed actions can be defined at very detailed level.
  - Each assembly can have its own security context.

Evidence-Based Security

- Definitions
  - Permissions
    - Objects that represent specific authorized actions
    - Permission grant is an authorization for an action given to an assembly
    - Permission demand is a security check for specific grants
  - Policy
    - Set of permissions granted to an assembly
  - Evidence
    - Inputs to policy about code
  - All three can be extended using security APIs.

Standard Permissions

- Permissions for framework resources
  - Data, environment, file IO, Message Queue, reflection, sockets
  - Directory services, event log, web, performance counters, registry, UI
  - DNS, file dialog, isolated storage, printing, security system

Standard Permissions

- Identity permissions
  - Publisher, site, string name, url, zone
- User identity permission
  - Only non-code access permission in Framework.

Code Access Security

- Is evidence-based
- Most permissions are code access
  - Demanding permission performs a stack walk checking for grants of all callers
  - Two ways to make checks
    - Imperative – call a method
    - Declarative
      - Attributes in code
      - Attributes in configuration file
  - Get security by
    - Calling class libraries in Framework
    - Calling application code with checks

How it works

- Loader extracts evidence from assembly
- Evidence is input to policy
  - Each level, Enterprise, Machine, User, and AppDomain, are evaluated
    - For each level the union of grants for each matching code group is determined
    - Intersection of permissions from each of these levels are granted to the assembly
- Apply any assembly permission requests
- Result is the permissions granted to the assembly.

Stack Walk Modifiers

- Assertions
  - If code vouches for its callers then checks for permissions stop here.
- Gatekeeper classes
  - Managed wrappers for unmanaged resources
    - Demand permission to call unmanaged
    - Assert permission to call unmanaged
    - Make the call to unmanaged

Code Access Control

- Identity permissions can apply to code as well as users and groups
  - Based on evidence – signature, location, …
- Declarative checks made by JIT at compile-time.
- Imperative checks made by CLR at run-time.

Policy

- Process of determining what permissions to grant to code.
  - Per-assembly basis
- Policy levels
  - Enterprise
  - Machine
  - User
  - Application domain
- Each policy level is a collection of code groups
  - All code, internet zone, intranet zone, site, strong name (MS Office), publisher
- Permission grants are intersection of policy levels and union of collection of code groups.
  - Code gets only permissions common to Enterprise, Machine, user, AppDomain
  - Gets all permissions of all groups to which it belongs.

Default Policies

- Local Computer
  - Unrestricted
- Intranet
  - Limited read environment, UI, isolated storage, assertion, web access to same site, file read to same UNC directory
- Internet
  - Safe UI, isolated storage, web access to same site
- Restricted
  - No access, can't execute
- Strong name (Framework classes)
  - Unrestricted

Framework Support

- Classes used to represent evidence
  - Zone, Url, Site, ApplicationDirectory, StrongName, Publisher, Hash
- Classes used to represent permissions
  - DBDataPermission, PrintingPermission, SocketPermission, FileIOPermission, RegistryPermission, …

.Net Configuration Tool

Editing Permissions

All Standard Permissions

.NET Configuration 1.1

File   Action   View   Help

My Computer
- Assembly Cache
- Configured Assemblies
- Remoting Services
- Runtime Security Policy
  - Enterprise
    - Code Groups
    - Permission Sets
    - Policy Assemblies
  - Machine
    - Code Groups
    - Permission Sets
    - Policy Assemblies
  - User
    - Code Groups
      - All_Code
    - Permission Sets
      - FullTrust
      - SkipVerification
      - Execution
      - Nothing
      - LocalIntranet
      - Internet
      - Everything
    - Policy Assemblies
- Applications

Permission
- Environment Variables
- File Dialog
- File IO
- Isolated Storage File
- Reflection
- Registry
- Security
- User Interface
- DNS
- Printing
- Event Log
- Socket Access
- Web Access
- Performance Counter
- Directory Services
- Message Queue
- Service Controller
- OLE DB
- SQL Client

Can configure applications as well as users and machines.

Creating a User Code Group

Adding New Permissions

Specific Permissions

The Result

Evidence

- Evidence is input to policy
  - Strong name, publisher identity, location
- Evidence is extensible
  - Any object can become evidence
  - Only affects permission grants if some code group condition uses it
- Hosts
  - Machine, IIS, ASP.Net, SQL Server
- Fully trusted hosts specify implicitly trusted evidence.
- Semi-trusted hosts cannot provide evidence.
- Hosts can limit policy for AppDomains they create.

Requesting Permissions

- Assemblies can request permissions
    - Minimal, Optional, Refused
    - If policy does not grant everything in Minimal set, assembly will not load.
    - Assembly is granted:
      MaxAllowed $\cap$ (Minimal $\cup$ Optional) – Refused
    - Example:

      [assembly:UIPermissionAttribute
      (SecurityAction.RequestMinimum,
      Window=UIPermissionWindow.SafeSubWindows)
      ]

      [assembly:SecurityPermissionAttribute
      (SecurityAction.RequestRefused,
      UnmanagedCode=true)
      ]

Minimizing Security Flaws

- Safe code
  - Managed code verified for typesafety at runtime.
  - Eliminates:
    - Buffer overrun attacks
    - Reading private state or uninitialized memory
    - Access to arbitrary memory in process
    - Transfer execution to arbitrary location in process
- Developers can use Least Priviledge.
- Code access security blocks most luring attacks.
  - Stack walks prevent malicious code from using otherwise secure code obtained from naïve user.

Summary

- Managed code has both Role-based and Evidence-based (CAS) security applied.
    - Get a lot for free, simply by loggin in and running code that calls Framework Library.
    - You can add security features to your code as well.
- CAS is .Net model for mobile code.
- Evidence is discovered by loader
- Policy turns evidence into permissions
- Permissions determine what your code can and cannot do.