

# Security in the .NET Framework

MSImpact TechTalk

January 24, 2003

compiled by

Nimit Sawhney (nimits@cmu.edu)

1

## Contents

- Section 1: Overview
- Section 2: Core Concepts
- Section 3: Permissions
- Section 4: Security Administration
- Section 5: Cryptography Support
- Summary

Microsoft

2

## ■ Section 1: Overview

- Looking back ...
- .NET security core concepts

Microsoft

3

## ■ Object based security models

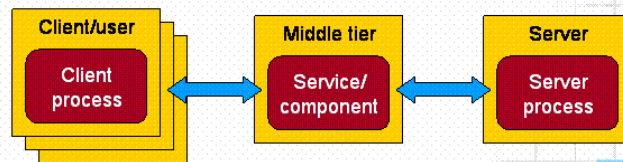
- Securing distributed applications
  - User identification and authentication
  - Data integrity and privacy
  - User authorization
  - Security must be administrable
- User wants to access securable objects
  - Access token and security descriptors

Microsoft

4

## ■ Looking back ...

- Security mechanisms
  - Different solutions for different issues
  - Samples:
    - Identification and authentication: Operating system account
    - Authorization: Active Directory – a security database
    - Encryption: HTTPS (HTTP over SSL)
- DCOM, CORBA, and TPMs



Microsoft

5

## ■ What's wrong with that?

- Trust all or nothing at all
- TPMs are difficult to administer
- „Luring attacks“

Microsoft

6

## Section 2: Core Concepts

- Kinds of Security
- Permissions, Policies, and Roles
- Common Language Runtime
  - Code Groups
  - Stack Walking

Microsoft

7

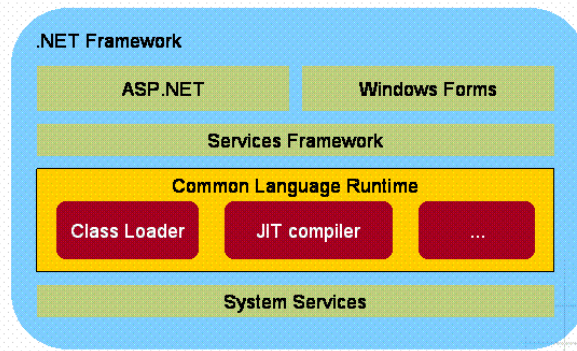
## Kinds of Security

- Code access security
  - Protection against malicious mobile code
- Role-based security
  - Principals
  - User authorization
- Security model is based on permissions
- Heavily based on Common Language Runtime

Microsoft

8

## Common Language Runtime



Microsoft

9

## Application Domain Host

- Host sets up Application Domain and loads assembly
  - Trusted host and evidence
- Different hosts
  - Shell
  - Browser
  - Server
  - Custom-designed

Microsoft

10

## Evidence

- Information about the code
  - Who published the Code
  - Where did the Code come from
- Samples of types of evidence
  - Signature
    - Publisher of the code
    - Strong name
  - URL and Site of origin

Microsoft

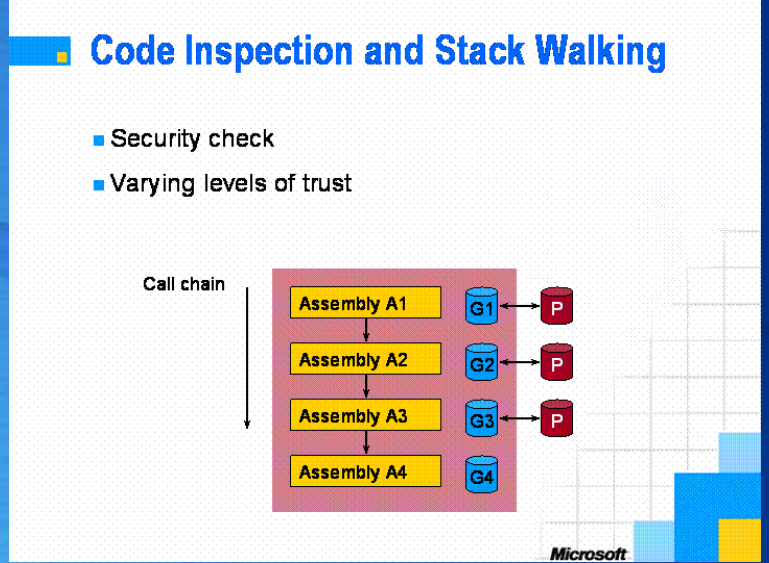
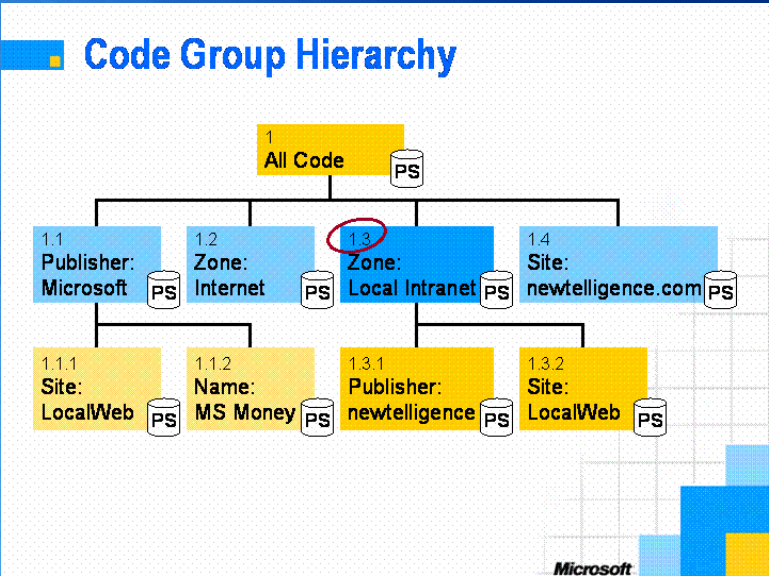
11

## Permissions, Policies, and Roles

- Permissions
  - Access code to restricted areas
  - Objects to control restrictions on managed code
- Security policy
  - Rules, that the runtime must follow to check permissions
- Roles and the principal
  - Named set of users
  - Principals

Microsoft

12



## ■ Security Namespace

- **System.Security.Policy**
  - Classes to deal with permissions
- **System.Security.Permissions**
  - Classes to control access to operations and resources
- **System.Security.Principal**
  - Object acts on behalf of the caller
- **System.Security.Cryptography**
  - Cryptographic services

Microsoft

15

## ■ Declarative Security

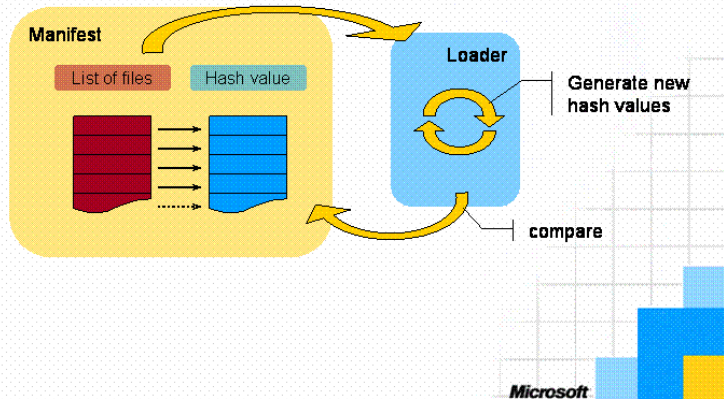
- ... with attributes
- Specifying security at assembly, class or member level
  - Security at lower level overrides higher level
- Syntax
  - SecurityAttribute class
  - SecurityAction enumeration
- C# sample: security demands

```
using System.Security.Permissions;
[FileIOPermissionAttribute(
    SecurityAction.Demand)]
```

Microsoft

16

## Security and the Manifest



17

## Imperative Security

- ... with explicit code
- Create a permission object and call its methods
- Scope of protection is the method
- Permission-based judgements made at run time
- Sample: security demands

```
using System.Security.Permissions;  
FileIOPermission myPerm =  
    new FileIOPermission(...);  
myPerm.Demand();
```

Microsoft

18

## Section 3: Permissions

- Permissions
  - Different kinds of permissions
- Using permissions
- Managing permissions

Microsoft



19

## Kinds of Permissions

- Permission and permission set
  - XML representation of permissions
- Code access permissions
  - Protect resources and operations
- Identity permissions
  - Characteristics of an assembly's identity
- Role-based permissions
  - Discover a user's role or identity
- Custom permissions
  - Design and implement your own permissions

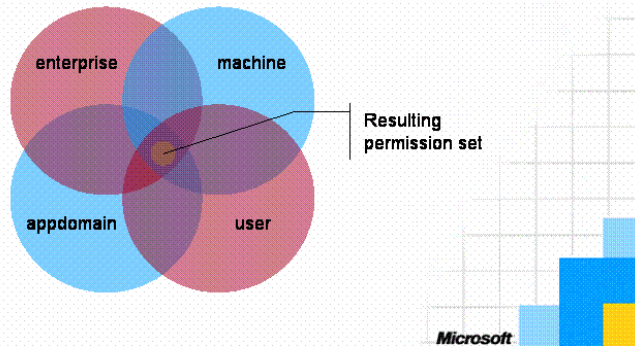
Microsoft



20

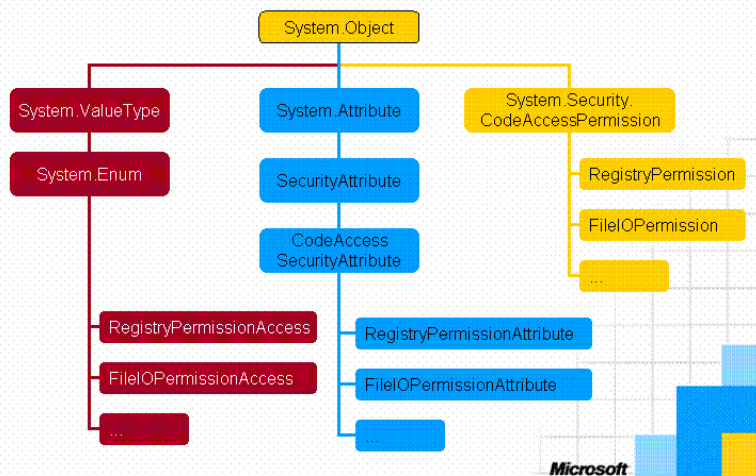
## Managing Permissions: Policies

- Policy levels
  - Enterprise, machine, user, application domain



21

## Permission Namespace



22

## ■ Requesting Permissions

- Provide security related information to the runtime
  - Used to check permissions
- Place attributes in your code
  - Compiler stores the request in the metadata
- Don't ask for more than you need ...
  - Minimum
  - Optional
  - Refused
- Code cannot assign rights to itself

Microsoft

23

## ■ Demanding Permissions

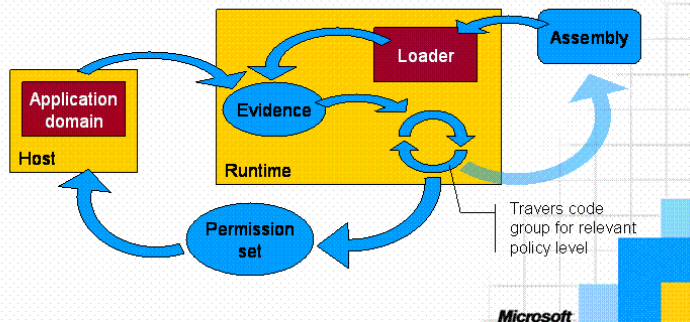
- Enforce restrictions on calling code
  - Ask the runtime to check permissions
- Secure either methods or complete code blocks
  - Declaratively or imperatively
- Guidelines
  - Check identity when giving additional access
  - To restrict object creation secure its constructor

Microsoft

24

## ■ Granting Permissions

- The runtime grants permissions
  - To application domains and assemblies
  - Based on identity, requested permissions, and trust



25

## ■ Overriding Code Access Permissions

- Override the outcome of the stack walk security check
- Assert
  - Specify permissions that should not be checked
  - Security hole
- Deny
  - Explicitly deny permissions
  - If one caller in call chain fails, all will fail
- PermitOnly
  - Specify a certain resource that can be accessed

Microsoft

26

## Code Access Permissions 1/3

- Protect Resources and operations
  - SecurityPermission class
  - SocketPermission class
  - WebPermission class
  - PrintingPermissions
- User Interface Access
  - UIPermission class
  - Secure windows prevent spoofing
    - Prevent code to steal from clipboard

Microsoft

27

## Code Access Permissions 2/3

- Access and modify environment, registry, and metadata
  - EnvironmentPermission
  - RegistryPermission
  - ReflectionPermission
  - DNSPermission
  - EventLogPermission
  - ServiceControllerPermission
- Protect files and directories
  - FileIOPermission
  - FileDialogPermission

Microsoft

28

## Code Access Permissions 3/3

- Protect Data
  - DirectoryServicesPermission
  - IsolatedStoragePermission
  - IsolatedStorageFilePermission
  - OleDbPermission
  - SqlClientPermission
  - MessageQueuePermission
  - PerformanceCounterPermission

Microsoft

29

## Identity Permissions

- Identity of an assembly
- Relevant classes
  - PublisherIdentityPermission
  - SiteIdentityPermission
  - StrongNameIdentityPermission
  - ZoneIdentityPermission
  - URLIdentityPermission

Microsoft

30

## ■ Role-based Permissions

- Principals
  - Generic: unauthenticated users and roles
  - Windows: Windows users/accounts
  - Custom: principals defined by application
- PrincipalPermission Class
  - Perform checks against active principal
- Authentication and authorization

Microsoft

31

## ■ Custom Permissions

- System.Security.Permissions namespace
- Consider thoroughly – overlapping and redundancy
- Code access permissions
  - Design
    - Which resource is to be protected?
    - How's the granulation of access?
  - Implement
    - IPermission interface
  - Demand
    - Update the policy

Microsoft

32

## Type Safe Code and Trust

- No memory access to the „neighbour's" private fields
  - Isolated assemblies
- Compiler checks if code is type-safe
  - Not all language compilers can generate type-safe code
- JIT compiler verifies type-safety
  - If code is not type-safe the code is not trustworthy
  - Not type-safe code may call unmanaged code
    - And perform malicious operations

Microsoft

33

## Wrapping Unmanaged Code

- Calling unmanaged code is risky
  - Direct calls into unmanaged code can bypass security
- Use managed wrapper classes
  - Enforce security restrictions
  - Such classes are different from CCW and RCW
- Secure class libraries
  - Security demands
  - Check each call to resources exposed by the library
  - „Code access security does not eliminate the possibility of human error in writing code"

Microsoft

34

## ■ Security Tools

- Managing certificates
  - Cert2spc.exe, Certmgr.exe, or Makecert.exe
- Managing assemblies
  - Sn.exe
    - Shared Name utility
  - GACUtil.exe
    - Global Assembly Cache utility
- PermView.exe
  - View permissions requested by an assembly

Microsoft

35

## ■ Managing Permissions and Policies

- Code Access Security Policy Commandline Utility
  - Caspol.exe
- Configure machine and user policy
  - Adding, modifying, and deleting
    - Code groups
    - Permissions and permission sets
- Samples:
  - `caspol -list`
  - `caspol -machine -addfulltrust myPerm.exe`
  - `caspol -machine -ag 1.1 -zone Internet execution`

Microsoft

36

## ■ mscorcfg.msc

- Graphical User Interface
  - Microsoft Management Console Snap-In
- Manage Security Policies
  - Modify code groups and permission (sets)
  - On enterprise, machine, and user level

Microsoft

37

## ■ Sample

- Creating named permission sets
  - Create an XML representation
    - Permission set = permission + name + description
  - Associate permission set and code group(s)
    - Modifying security policy
  - Built-in named permission sets
    - Nothing, Internet, Everything, ...
  - Custom permissions

Microsoft

38

## Section 5: Cryptography Support

- Hashing
- Encryption
- Digital signatures

Microsoft

39

## Summary

- Powerful security system
  - Flexible
  - Administrable
- Fine-grained control on security
  - A number of classes and security tools
  - Different security solutions
- Rich set of cryptography services

Microsoft

40

## Some useful links:

- [http://www.gotdotnet.com/team/clr/about\\_security.aspx](http://www.gotdotnet.com/team/clr/about_security.aspx)
- <http://www.foundstone.com/pdf/dotnet-security-framework.pdf>
- <http://msdn.microsoft.com/library/en-us/dnnetsec/html/netframeseccover.asp>

Cheers 😊