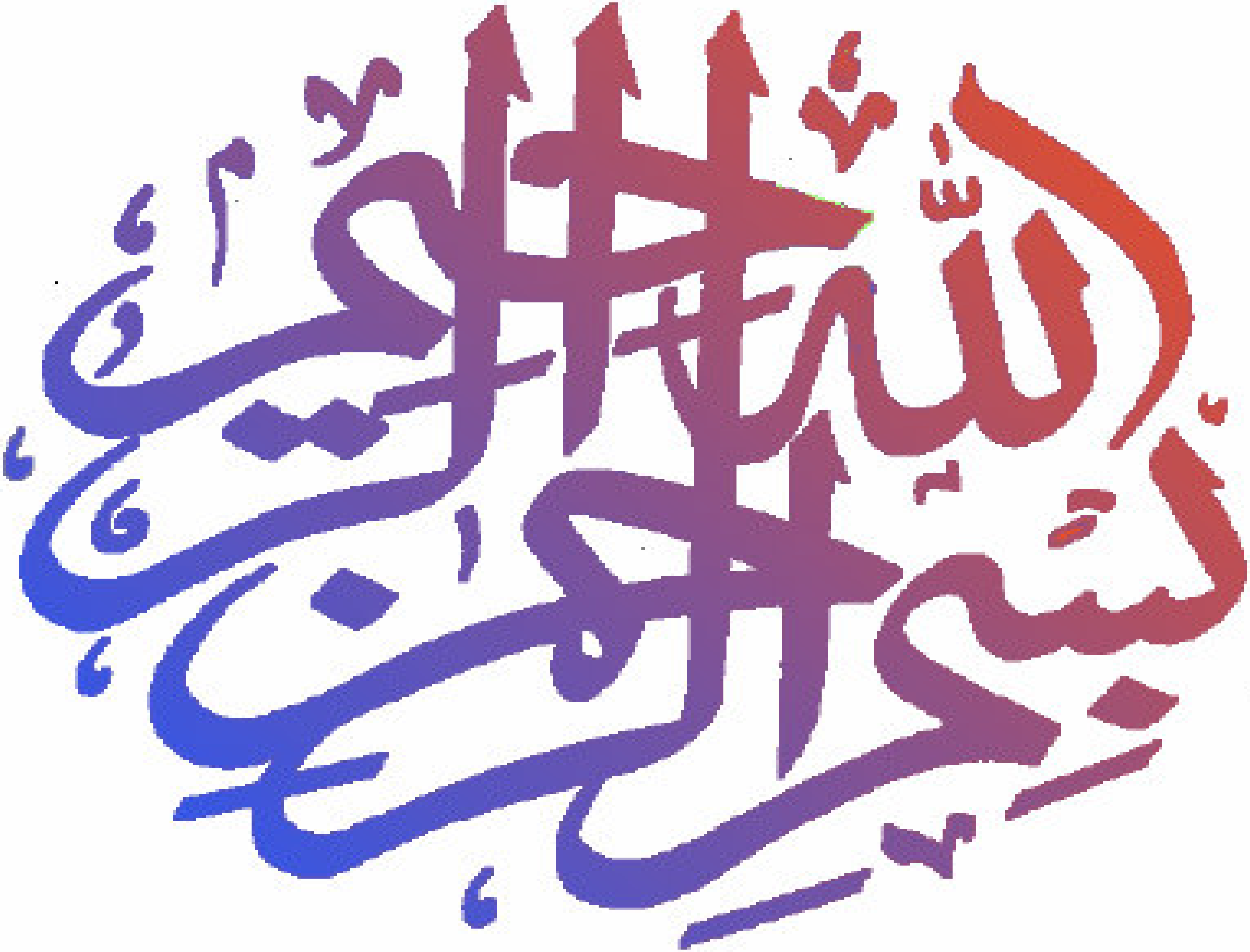


بسم الله الرحمن الرحيم

.Net WiZard

Developing “Secure” Enterprise Scale Applications [using .Net Framework]

Speaker:

Hammad Rajjoub.

- **Technical Lead – Meezan Bank Limited**
- **Group Leader, Chairman UG Relations Committee and Member Speakers Bureau INETA Pakistan.**
- **MCP.Net.**
- **Author.**
- **(URL: http://dotnetwizards.blogspot.com**
- **Mail: hammad.rajjoub@ineta.org)**

Agenda:

- Why Write Secure Code ?
- What is Secure Product ?
- Introduction to .NET Security.
- .NET Security in Action!
- Securing Enterprise Scale Applications.
- Future: Trustworthy Computing.
- Resources.
- Q/A.

Why write secure code ?

“I am a programmer, security isn't my concern… talk to my network admin/security expert!”

- Programmers are responsible for the code they write.
- Need to develop secure system.
- Secure System is not just secure code or code that implements security features rather it’s a code that is designed, developed and tested to withstand attacks.

What is secure Product?

A Secure Product is

" A product that protects the confidentiality, integrity and availability of the customers' information, and the integrity and availability of processing resources, under control of the system's owner or administrator"

A Security Vulnerability is

"A flaw in a product that makes it infeasible – even when using the product properly- to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, assuming un granted trusts.

-Source: Microsoft.com

Introduction to .Net Security:

- .NET framework includes a large variety of security features commensurate with the breadth of the framework it self. Following are the few important parts of it:
    - Role Based Security.
    - Code Access Security.
    - Evidence Based Security.

Role Based Security

“*Role-based security allows you to programmatically control what actions users are permitted to perform.*”

Role based security is modeled around Real-World roles. This security strategy has its basis in roles that we as employees, manager, auditors play in life.

The best way to work with Role Based Security is to follow “Principle of Least Privileges”.

Role Based Security contd.

- Implementing Role Based Security:
  - To implement Role-Based Security in .NET, we need to know about two important things:
    - Authentication &
    - Authorization.
- Authentication:
  - *“Authentication verifies you are who you say you are.”*
- Authorization:
  - *“Authorization verifies you are permitted to perform a specific activity.”*

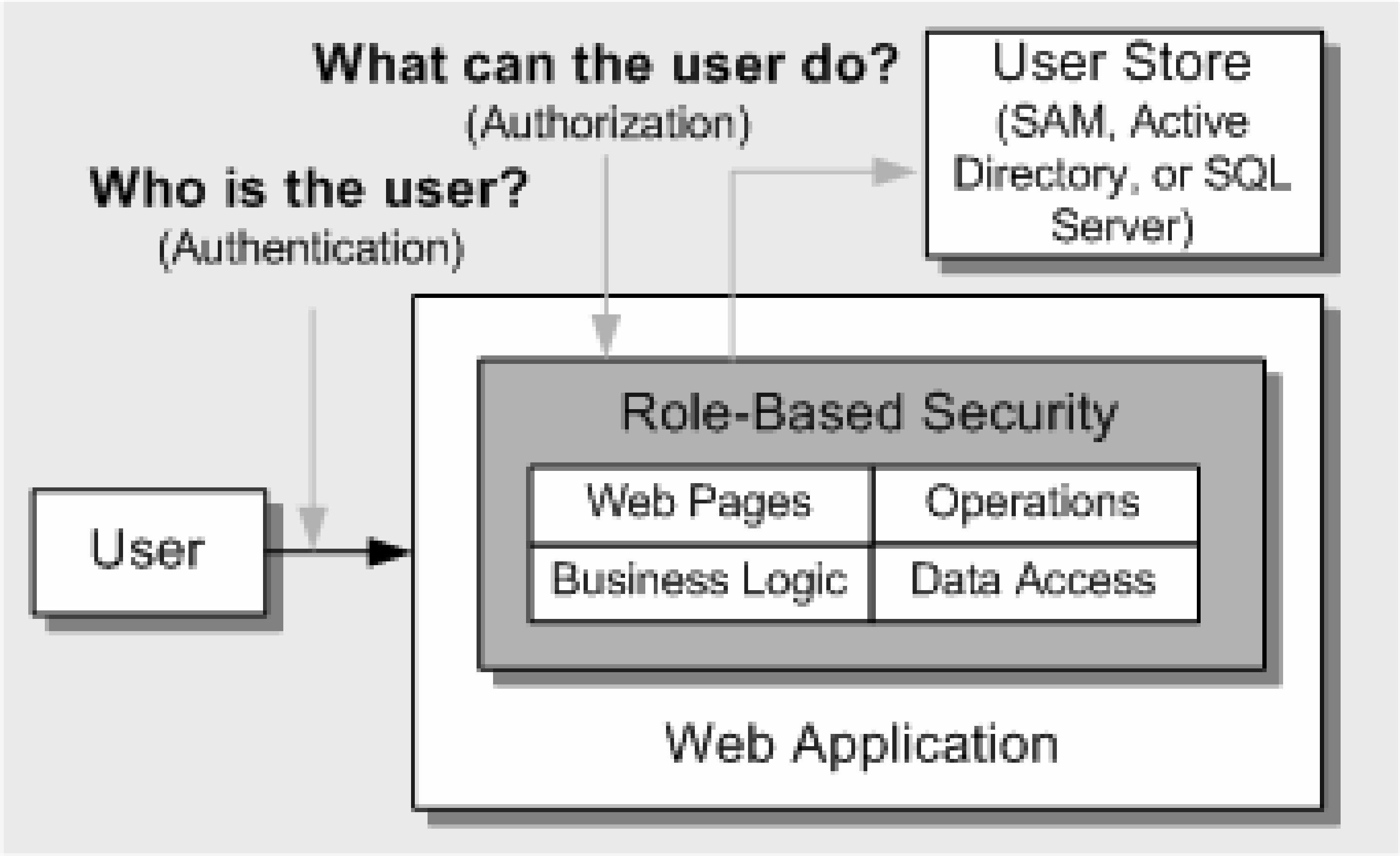
Role Based Security contd.

- Implementing Authentication and Authorization:
  - □ Permissions and Principals:
    - "Permission" is a code that represents set of operations that can be secured for specified resources. Permissions are used by both, your application code and .NET Runtime, in following ways:
      - □ Code Requests the permission it needs in order to run.
      - □ .Net Runtime policy grants permission to code in order for it to run.
      - □ Code demands that calling code has a permission.
      - □ Code overrides the security stac assert/deny/permit-only.

Role Based Security contd.

- “Principal” represents the context for the use on whose behalf the code is running, this includes users Identity (implemented using IIdentity intrface).In other words all authenticated users are represented by a an object implementing IPrinciple interface. It encapsulates users identity and role and can be used to validate users identity against a “Principal Permission” object.

Role Based Security contd.



.Net WiZard

DEMO

- Sample code implementing Authorization and Authentication.

Authentication and Authorization Summary

- Role based authentication and authorization enables you to create a single application composed of one or more dlls and exes that allows users to perform tasks exactly in the manner that you want them to perform!

C.A.S (Code Access Security)

- Code Access Security or CAS, as it is affectionately called, is designed to protect applications and components in shared environments from the following risks:
  - □ Inadvertently or intentionally damaging or destroying data.
  - □ Crippling the computer on which the code is available running by consuming all the available resources, event known as DOS (Denial Of Service) Attack.

Code Access Security contd.

- Allowing calling code or attackers to intentionally or un intentionally elevate their privileges to perform actions such as viewing sensitive data etc, also known as “**Luring Attack”. E.g. Online Chart Example**

Code Access Security contd.

- How actions are considered safe or unsafe?

  - How is .NET Runtime smart enough to decide which action to be performed by code in consideration is safe or un safe ?

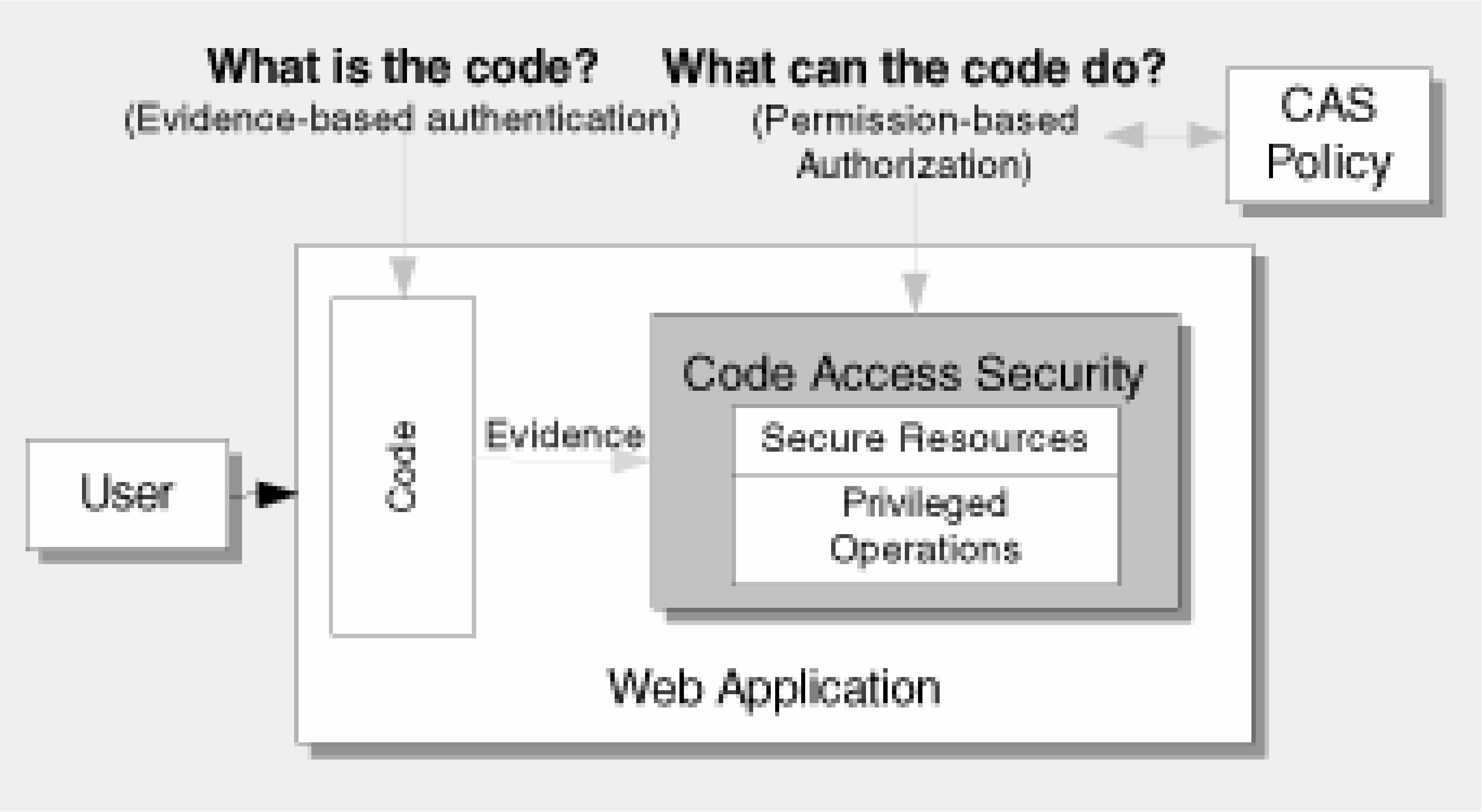
Code Access Security contd.

- Notion of Trust and Zones:
  - CAS assigns your application or component permissions like IO Operations, UI, and Network permissions as the basis of determining what safe and unsafe operations your code is allowed to perform.
  - The collective set of permission assigned to your application or component is based on the level of “Trust” against your application or component. (e.g. High level of trust, medium level of trust etc.)

Code Access Security contd.

- What prevents *Harmful* code from executing?
  - □ Demand, Check and Execute Principal.
  - □ CAS Security is ON by default.

Code Access Security contd.



.Net WiZard

Evidence Based Security

- CAS uses *evidences* to identify what should be allowed to executed and what should be prohibited, thus behaving smartly.
- Evidence can be any thing known fact about the code: any valid digital signature, url, or site or zone the code comes from.

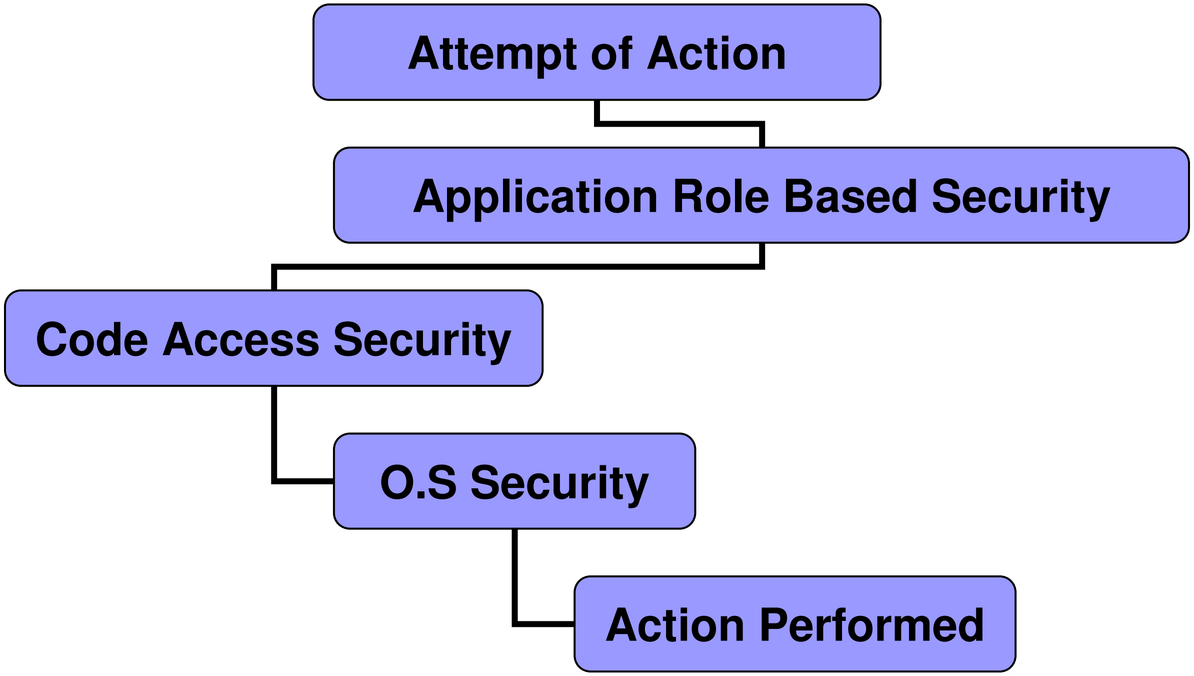
Evidence Based Security contd.

- This enables a number of possibilities not formerly available to applications:
  - mobile code can be downloaded from unsecured sources and executed safely with restrictions
  - ISP server hosts can run different site applications together in-process safely, increasing performance
  - server applications can be extended with user-written code that runs constrained to not interfere with overall server operation
  - programmable applications can safely run macro script associated with user documents

CAS vs. Application Role Based Security

- In Application Role Based Security you (as a programmer) choose what to allow and what not to allow. The choice is with you and you are responsible for your decision, you rule the application security domain.
- With CAS, however .NET Runtime decides what should be allowed and what should be disallowed. The decision lies with framework to make the right the choices.
- What am I safe with: Application Role Based Security or Code Access Security? (give example here of “Air Trip”)

Security Stack (ed) !



.NET Security in Action.

- **Defying “The Luring Attack”:**
  - □ **Discuss Example.**
  - □ **.NET uses Stack Walk!**

Trustworthy Computing

- Trustworthy Computing means helping to ensure a safe and reliable computing experience that is both expected and taken for granted.
- The goals set for Trustworthy Computing are designed to deliver the level of trust and responsibility that people expect from the computing industry: Security, Privacy, Reliability, and Business Integrity.

Trustworthy Computing contd.

- Security
  - □ Amid increasingly frequent and sophisticated network attacks, users expect their systems to remain resilient, and for system and data confidentiality, integrity, and availability to be maintained.

Trustworthy Computing contd.

- Privacy
  - □ People are increasingly using computers to manage information important to their everyday lives. They expect and demand control over access to and use of their personal information.

Trustworthy Computing contd.

- Reliability
  - As computers become increasingly central to how people work and live, it becomes increasingly essential that they perform as expected. Users look for a consistently trouble-free computing experience.

Trustworthy Computing contd.

- Business Integrity
  - □ People's perception of technology reflects their perception of the technology industry. Belief in technology is stronger when the industry is responsive, responsible, and respectful.

Resources:

- Writing Secure Code (Haward and LeBlanc) by Microsoft Press.
- Security for Visual Basic.Net (Robinson Bond) by Microsoft Press.
- http://msdn.microsoft.com/security/