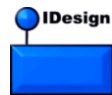




WCF Security Fundamentals

Michele Leroux Bustamante
Chief Architect, IDesign
Microsoft Regional Director

©2004 IDesign Inc. All rights reserved

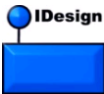


About Michele Leroux Bustamante

- Chief Architect, IDesign Inc., www.idesign.net
- Microsoft Regional Director
- MVP – Connected Systems
- IASA - Board Member
- Frequently published author
- International Speaker, INETA
- Program Advisor, UCSD Extension
- Track Chair, SD Expo
- www.dasblonde.net, www.thatindigogirl.com



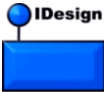
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Agenda

- Core security concepts
- Overview of WCF security settings
- Security mode and protection level
- Intranet settings
- Impersonation
- Business partner and cross-machine settings
- Negotiation and secure sessions
- Internet settings
- Controlling authentication and authorization
- IDesign's declarative security model

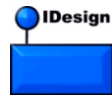
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Core Security Concepts

- **Mutual Authentication** – a means for sender and receiver to identity one another
- **Authorization** – determining the rights of the authenticated party
- **Confidentiality** – ensuring that only the intended recipient can view message content
- **Integrity** – ensuring that message content has not been altered by malicious parties
- **Reliability** – preventing replay and DoS

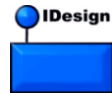
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Authentication/Authorization

- Implies passing appropriate credentials to identify callers
 - Windows tokens, certificates, username and password
- Services must also be identified
 - Windows tokens, certificates
- Authorization against the appropriate credential store
 - Gather roles, permissions, access rights

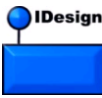
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Transfer Security

- Protecting messages while transferred from point to point
 - Across network nodes
 - Between applications
 - Across interoperable boundaries
- Encryption and digital signatures facilitate
- Transport security is on the wire
- Message security can traverse network nodes

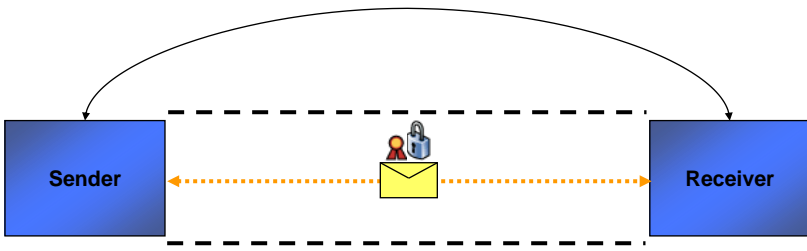
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



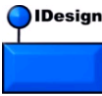
Transport Level Security

- SSL, TLS, IPsec
- Point-to-point
- Applies to entire message

Trust Relationship



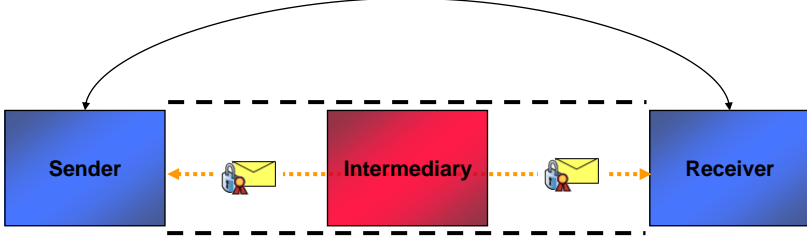
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



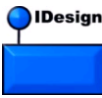
Message Level Security

- Web services security (WS*)
- Secure to ultimate message receiver
 - Through intermediaries (XML firewalls, proxies, etc.)
- Secure message parts

Trust Relationship



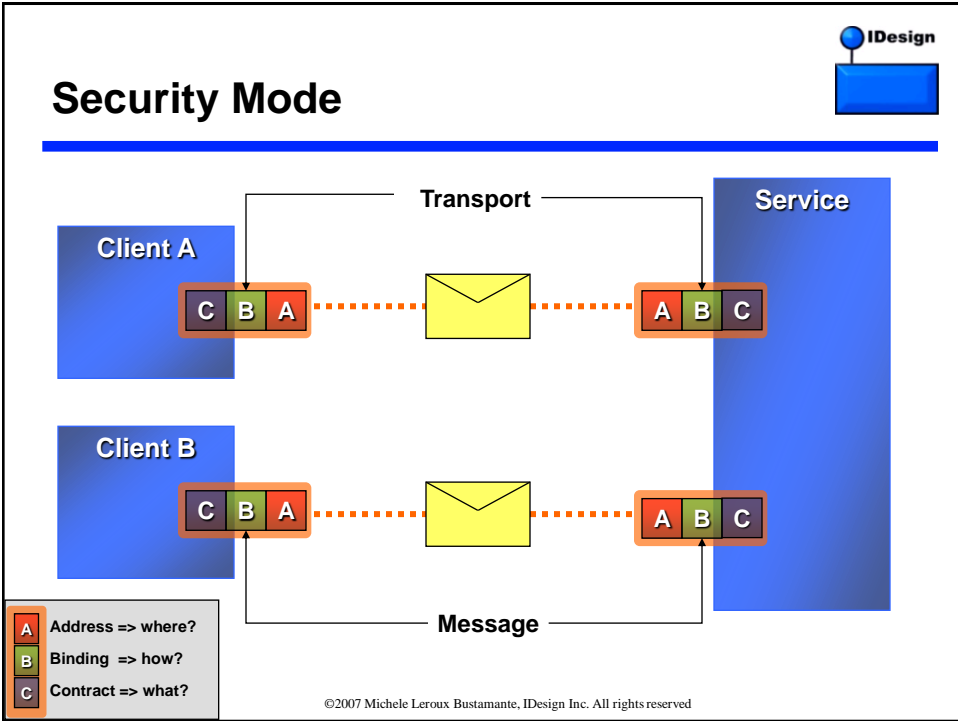
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

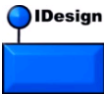


WCF Security Settings

- Security mode
- Protection level
- Client and service credentials
- Impersonation
- Credential negotiation
- Secure sessions
- Authentication and authorization behaviors

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



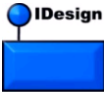


Security Mode

- Transport Security

```
<netTcpBinding>  
  <binding name="netTcpTransportSecurity">  
    <security mode="Transport">  
      <transport  
clientCredentialType="Windows" />  
    </security>  
  </binding>  
</netTcpBinding>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

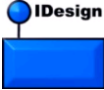


Security Mode

- Message Security

```
<wsHttpBinding>  
  <binding name="wsHttpMessageSecurity">  
    <security mode="Message">  
      <message  
clientCredentialType="UserName" />  
    </security>  
  </binding>  
</wsHttpBinding>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

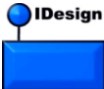


Security Mode

- Mixed Security

```
<basicHttpBinding>  
  <binding name="basicHttp">  
    <security  
mode="TransportWithMessageCredential">  
      <transport />  
      <message  
clientCredentialType="UserName"/>  
    </security>  
  </binding>  
</basicHttpBinding>
```

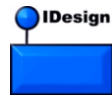
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Protection Level

- By default all messages are signed and encrypted
 - For secure bindings
- Can throttle message protection
- Can require minimum level of protection
- Protection level options are: None, Sign, EncryptAndSign

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

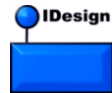


Protection Level

- Throttling protection levels on the wire
 - For transport protection, set binding properties for TCP, named pipes and MSMQ
 - Cannot throttle SSL settings

```
<netTcpBinding>
  <binding name="signOnly">
    <security>
      <transport protectionLevel="Sign"/>
    </security>
  </binding>
</netTcpBinding>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

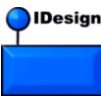


Protection Level

- Throttling message protection levels
 - For message protection, set contract properties
 - At service contract, operation contract, message contract

```
[ServiceContract (Namespace=
"http://www.thatindigogirl.com/samples/2006/06")]
public interface IHelloIndigoService
{
  [OperationContract (ProtectionLevel=ProtectionLevel.Sign)]
  string HelloIndigo (string inputString);
}
```

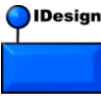
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



DEMO

- Controlling protection level

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Client Credentials

- Credential options:
 - Windows
 - Username and password
 - X.509 certificates
 - Issued SAML tokens (including CardSpace claims) or custom tokens
- Selections vary for binding configurations
- Provide proxy with credentials

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Service Credentials

- Credential options:
 - Windows
 - X.509 certificates
- When clients use Windows credentials, so does the service
- When clients use non-Windows credentials, service must provide a certificate
 - Supplied by transport (SSL) or by associated service behavior

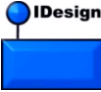
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Intranet Scenario

- Windows credentials for mutual authentication
- Authentication and authorization use default Windows membership and role providers
- Messages encrypted and signed by the transport layer
- The service implements role-based permission demands on protected operations
- The service usually rejects impersonation in favor of trusted subsystem model

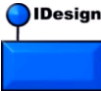
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



DEMO

- Intranet scenario
 - Windows credentials for mutual authentication and authorization
 - Transport security

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Impersonation

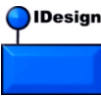
- Service can control impersonation level
 - **OperationBehaviorAttribute**
 - **ImpersonationLevel** setting: NotAllowed, Allowed, Required

```
[OperationBehavior (Impersonation=ImpersonationOption.NotAllowed) ]  
public string HelloIndigo (string inputString)
```

- Can control for all service operations
 - **ServiceAuthorization** behavior

```
<serviceAuthorization impersonateCallerForAllOperations="false"/>
```

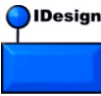
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



DEMO

- Impersonation

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Impersonation

- Clients can control impersonation level
 - **TokenImpersonationLevel**
 - None, Anonymous, Identification, Impersonation, Delegation

```
proxy.ClientCredentials.Windows.AllowedImpersonationLevel =  
TokenImpersonationLevel.Identification;
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

Business Partners and Cross-Machine Calls



- Service certificate identifies the service and protects messages during transfer
- Certificates uniquely identify partners or calling applications across machines
- Certificates authenticated using default certificate authentication mechanism
- Certificates authorized using peer trust and online revocation
 - Place public keys in the TrustedPeople store

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

Service Certificate



- Authenticates service to client and protects messages
 - Required for non-windows credentials
- Supply **ServiceCredentials** behavior

```
<behaviors>
  <serviceBehaviors>
    <behavior name="serviceBehavior" >
      <serviceCredentials>
        <serviceCertificate findValue="RPKey"
storeLocation="LocalMachine" storeName="My"
x509FindType="FindBySubjectName" />
      </serviceCredentials>
    </behavior>
  </serviceBehaviors>
</behaviors>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Client Certificate

- Authenticate partners or applications
- Supply **ClientCredentials** behavior

```
<behaviors>
  <endpointBehaviors>
    <behavior name="clientBehavior">
      <clientCredentials>
        <clientCertificate findValue="SubjectKey"
storeLocation="CurrentUser" storeName="My"
x509FindType="FindBySubjectName"/>
      </clientCredentials>
    </behavior>
  </endpointBehaviors>
</behaviors>
```


©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Negotiation

- Service credentials can be negotiated
 - Windows credentials rely on SPNego
 - Non-Windows credentials rely on TLSNego
- Requires a service certificate
 - Transport protection uses SSL
 - Message level uses WS-Trust as tunnel
- Negotiation removes the need to provision certificates to clients ahead of time

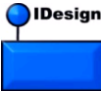
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



DEMO

- Mutual certificate authentication
- Negotiation

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

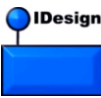


Negotiation

- Negotiation is not interoperable (yet)
 - Can disable negotiation for most HTTP bindings
- If disabled:
 - For Windows credentials a Kerberos domain must be present
 - For non-Windows credentials, clients must know the service certificate

```
<security mode="Message">  
  <message clientCredentialType="UserName"  
    negotiateServiceCredential="false" />  
</security>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

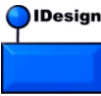


Secure Sessions

- Reduce the overhead of one-off key exchange and validation
- Security context token (SCT) generated for authentication and message protection
 - Enabled by default for most HTTP bindings
- Can disable for single call exchange
 - Client provides credentials for each call, service authenticates and authorizes each call

```
<security mode="Message">  
  <message clientCredentialType="UserName"  
    establishSecurityContext="false" />  
</security>
```

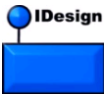
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Internet Scenario

- An SSL connection is used for transport security and service authentication
- A service certificate is supplied for message security and service authentication
- Username and password credentials are used for client authentication
- Authentication and authorization use the built-in ASP.NET membership and provider model
 - Possibly customized tables

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

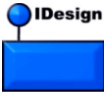


Authentication

- Can control authentication settings for each credential type
 - Settings in **ServiceCredentials** behavior
 - Control membership provider, certificate authentication modes, anonymous users, etc.

```
<serviceCredentials>
  <windowsAuthentication allowAnonymousLogons="false"
includeWindowsGroups="true" />
  <userNameAuthentication
userNamePasswordValidationMode="MembershipProvider"/>
  <clientCertificate>
    <authentication certificateValidationMode="ChainTrust"
revocationMode="Online"/>
  </clientCertificate>
</serviceCredentials>
```

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

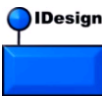


Authorization

- Can configure authorization with behaviors
 - **ServiceAuthorization** behavior
 - **PrincipalPermissionMode** setting
 - ▲ None, UseWindowsGroups, UseAspNetRoles, Custom
- Controls the type of security principal
 - **WindowsPrincipal**, **RoleProviderPrincipal**
 - Used for role-based security checks

System.Threading.Thread.CurrentPrincipal.Identity.Name

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved

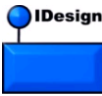


Authorization

- Can use ASP.NET roles provider

```
<serviceAuthorization  
principalPermissionMode="UseAspNetRoles"/>
```

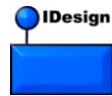
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



DEMO

- Internet scenario
 - UsernameToken credentials
 - Service certificate authentication
 - Provider model authentication and authorization

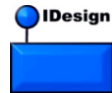
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Summary

- WCF provides granular control over security through bindings and behaviors
- Common scenarios include intranet, business partner exchanges and Internet applications
- The environment is highly extensible
 - Credential types, authentication, authorization models, etc.
- WCF also supports rich federated and claims-based security models

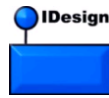
©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Resources

- www.idesign.net
 - Code library, coding standards, sample architecture report, IDesign Method™ documentation
- WCF Master Classes
 - IDesign Public Classes
 - On-site classes also available

©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved



Resources

- Learning WCF
 - Michele Leroux Bustamante, O'Reilly 2007 (book blog: www.thatindigogirl.com)
- Programming WCF Services
 - Juval Lowy, O'Reilly 2007
- My Blog
 - <http://www.dasblonde.net>
- IDesign Downloads
 - <http://www.idesign.net>



©2007 Michele Leroux Bustamante, IDesign Inc. All rights reserved