1786
The Principles of Securing a WebSphere Application Server

Stephen Pipes
IBM Hursley Park Labs, United Kingdom
Nashville 2002

Agenda

- Security fundamentals
- Authentication
- Digital signatures and certificates
- Cryptography
- Authorisation
- WebSphere security model
- Global system security
- Securing an application
- Security differences between versions 3.5 and 4.0

What is security?

Security is relevant in a variety of scenarios.

Two main areas are

- Physical security
  - prevents access to hardware, rooms, buildings
  - protects communication channels, wired and wireless
- Logical security
  - protects communication when connected to untrusted networks
  - defends applications in order to secure access to resources

Security policies are organisation definitions that focus on security-related issues.

Seven-step guide to security

- Authentication/Identification validates communication
- Access control prevents unauthorised use of resource
- Privacy ensures that information is not disclosed to unauthorised clients
- Integrity confirms the correctness of the information
- Accountability/non-repudiation ensures that all actions may be traced to the originator
- Administration are methods in which the security olicies are implemented
- Monitoring ensures that the system has met its security objectives

Security fundamentals

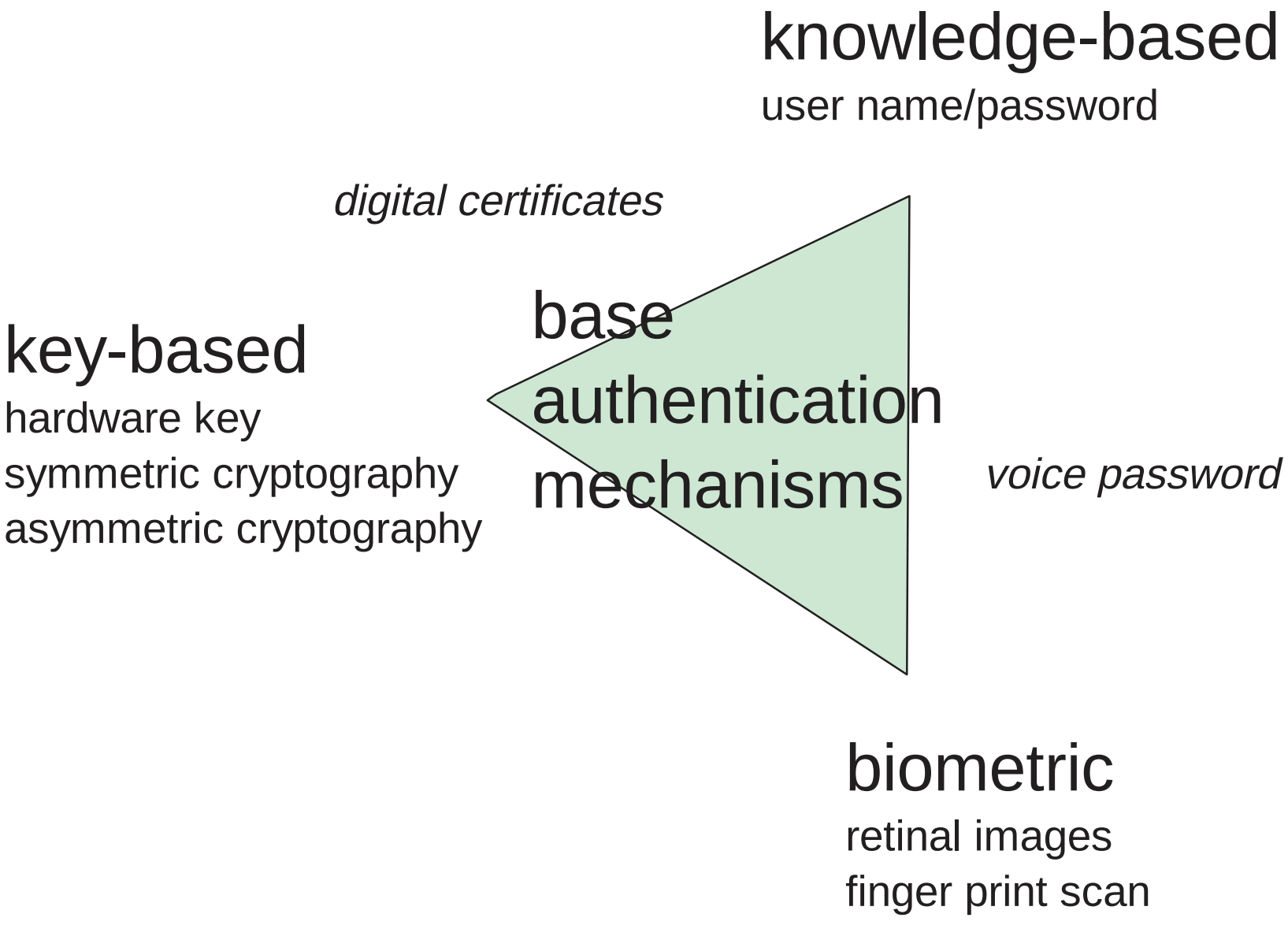
WebSphere supports three types of security service

- Authentication
- Authorisation
- Delegation

What is authentication?

Authentication is the process of establishing if a client is valid in a given context.
The authentication process involves gathering unique information about the client.

Some types of authentication



Some types of authentication

- User name and password pairs is most common method of gathering client credentials
- Physical keys are objects that prove identity of the holder
- Biometrics compare the physical characteristics of a client against a stored representation
- Digital certificates are based on asymmetric cryptography

User name and password

- Probably the simplest form of authentication
- Software support readily available
- Easily copied and distributed (by owner and eavesdoppers alike)
- Sent in plain-text over network
- Policy required to enforce "secure" passwords

Digital signatures

- Created by applying private key to some information
- Only an owner of a private key may "sign" the information
- Signatory can be acknowledged by applying appropriate public key (identity of person not guaranteed)

Digital certificates

- Based on asymmetric cryptography (covered later)
- Issued by Certification Authorities who will vouch for a client's identity
- The certificate is digitally signed by the CA and issued to the owner of the key pair
- The client may present this certificate during authentication, thus automating this process
- Certificate can be safely sent over an unsecured network, unlike user name and password

Symmetric cryptography

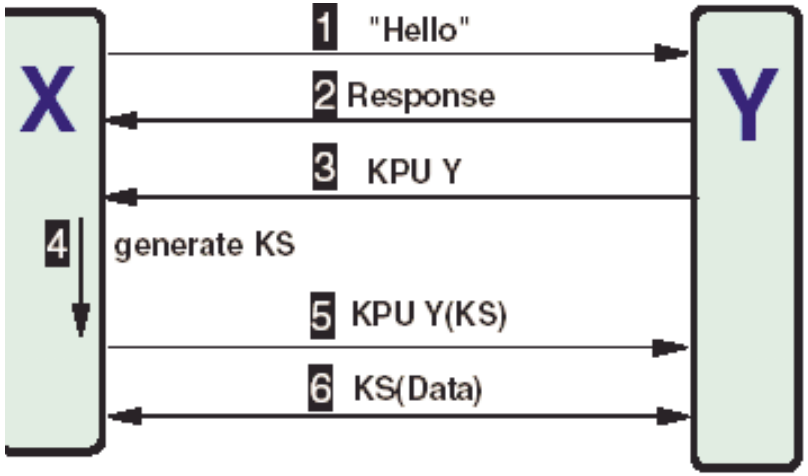
- Uses a shared key that is held by all recipients
- Often referred to as secret-key cryptography
- Information is encrypted and decrypted with same key
- One-time pads are a good source of random data
- If the key is compromised, then communication must be halted until a new key has been generated
- Key distribution is, potentially, a big problem
- Faster than asymmetric cryptography
- Popular algorithms are DES, 3-DES...

Asymmetric cryptography

- Uses a key pair that contains a public key and a private key
- Often referred to as public-key cryptography
- Information encrypted with the public key may only be decrypted using the private key and vice versa
- Public key is common knowledge; private key is known only by owner of key pair
- If private key is compromised, then the public key must be revoked

Public-key crypto in action: SSL

ecure Sockets Layer (SSL) uses a public-key algorithm
Web browsers to establish a secret key.
hy should SSL establish a secret key?



1. Communication request from browse
2. Response from web server
3. Server sends public key Y
4. Browser generates session key KS
5. KS encrypted with Y and sent to serv
6. Both browser and server use sessior key to communicate

This is not enough to authenticate the server.

Public-key crypto in action: SSL

We need a digital certificate to authenticate...

. Communication request from browser
2. Response from web server
3. Server sends *certificate*
3a. Browser authenticates certificate
4. If certificate is authentic, browser generates session
key KS
5. KS encrypted with Y and sent to server
6. Both browser and server use session
key to communicate

ow can the web browser be sure of server's identity?

What is authorisation?

Authorisation is the process of verifying that a client has access to a requested resource.

Two basic forms of authorisation:

- Access control lists
- Capability lists

Access Control lists

Maps resources to roles and defines their access rights.

resources	bank teller role	manager role
getBalance method	yes	yes
setBalance method	no	yes

Capability lists

Maps roles to resources and defines their access rights.

roles	getBalance method	setBalance method
bank teller role	yes	no
manager role	yes	yes

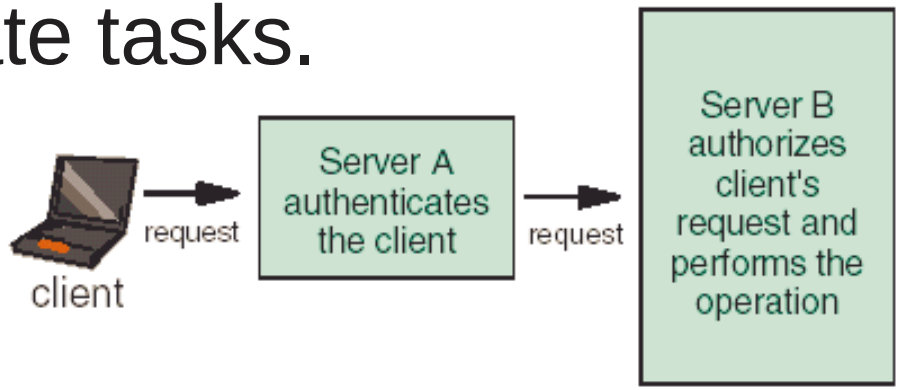
The same information but indexed differently.

Improves access depending on lookup criteria.

Typically, the Application Server will make use of Access Control lists due to the nature of the requests.

Delegation

- Delegation allows an intermediary to perform a task initiated by a client under an identity according to the delegation policy.
- Can delegate as client id, system id or specified id.
- Not defined in EJB 1.1 specification; WebSphere provides extensions for delegation.
- Fully addressed, however, in EJB 2.0.
- Web resources cannot delegate tasks.



Hashing

Used to protect information in an insecure environment.

Hash algorithm takes an arbitrary-length input and generates a fixed-length output.
Algorithm is such that output cannot be reverse-engineered.
Popular algorithms are MD5 and SHA1.

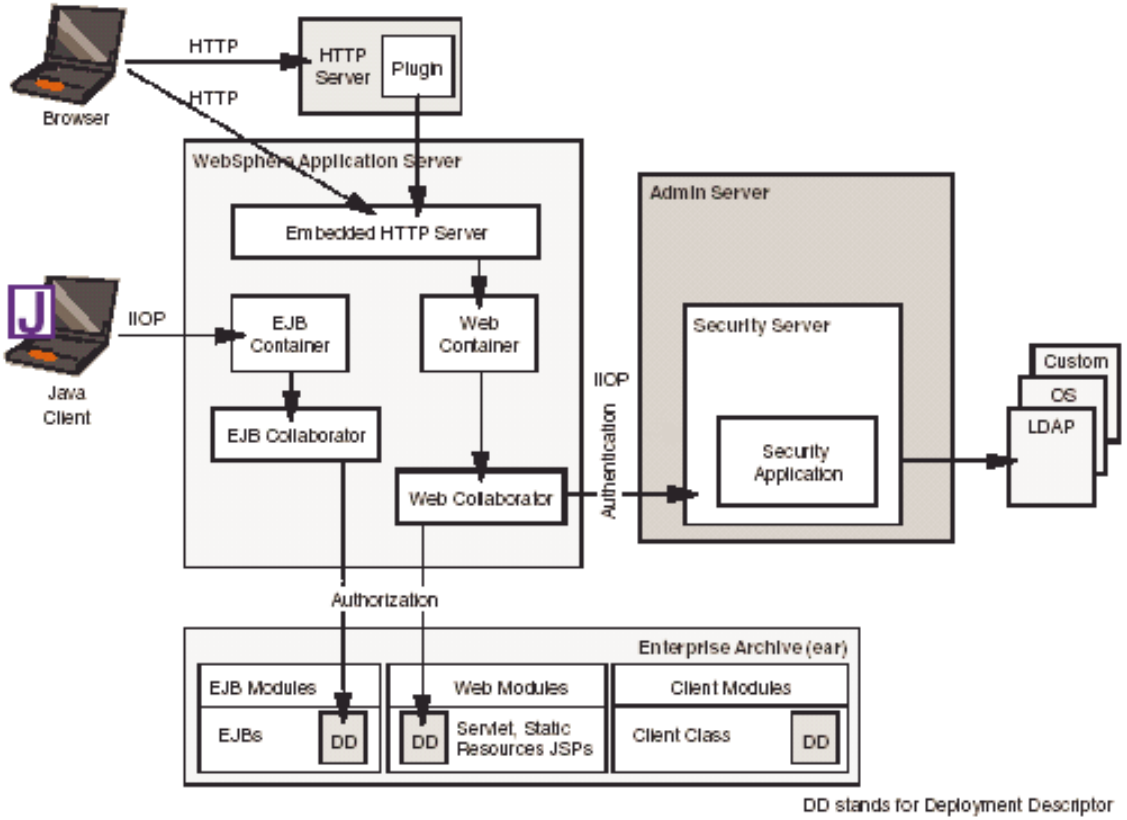
Can be used to protect private data stored on a filesyste and to provide message digests.

WebSphere security model

The security components of the App Server are:

- Security server
- Security collaborators
  - Web collaborator
  - EJB collaborator
- Security policies
- Secure Association Service (SAS)
- Secure Sockets Layer (SSL)

WebSphere security model



Security server

Security server resides in the Admin Server and provide
authentication services for Web and EJB containers.

Authentication may involve communication with an LDA
server, the local OS registry or a local repository.

Security collaborators

Security collaborators reside in each application server and enforce the security constraints defined in the deployment descriptors of Web and EJB components.

Web collaborator authenticates and authorises request.

EJB collaborator authorises but does not authenticate - relies on SAS to authenticate.

Security policies

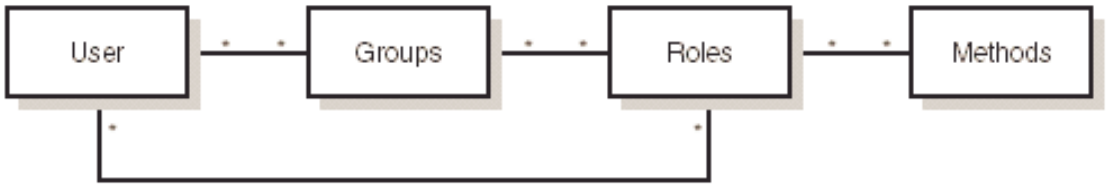
ecurity attributes are specified in deployment escriptors.

ttributes are

- Role and method permissions
- Run-as mode or delegation policy
- Login configuration and challenge type
- Data protection settings

2EE uses the concept of security roles to encapsulate e grouping of method permissions.

ne role and method permission mappings are



Secure Association Service (SAS)

ecure Association Service is a service provided by the
RB.

JB's require SAS in order to authenticate clients.

lenty of detail in Info Center.

AS effectively intercepts requests on the IIOP client an
athers credentials.

hese credentials are attached to the request and sent t
ıe server.

AS intercepts requests on the server and extracts
redentials, performs authentication and forwards to EJB
ontainer.

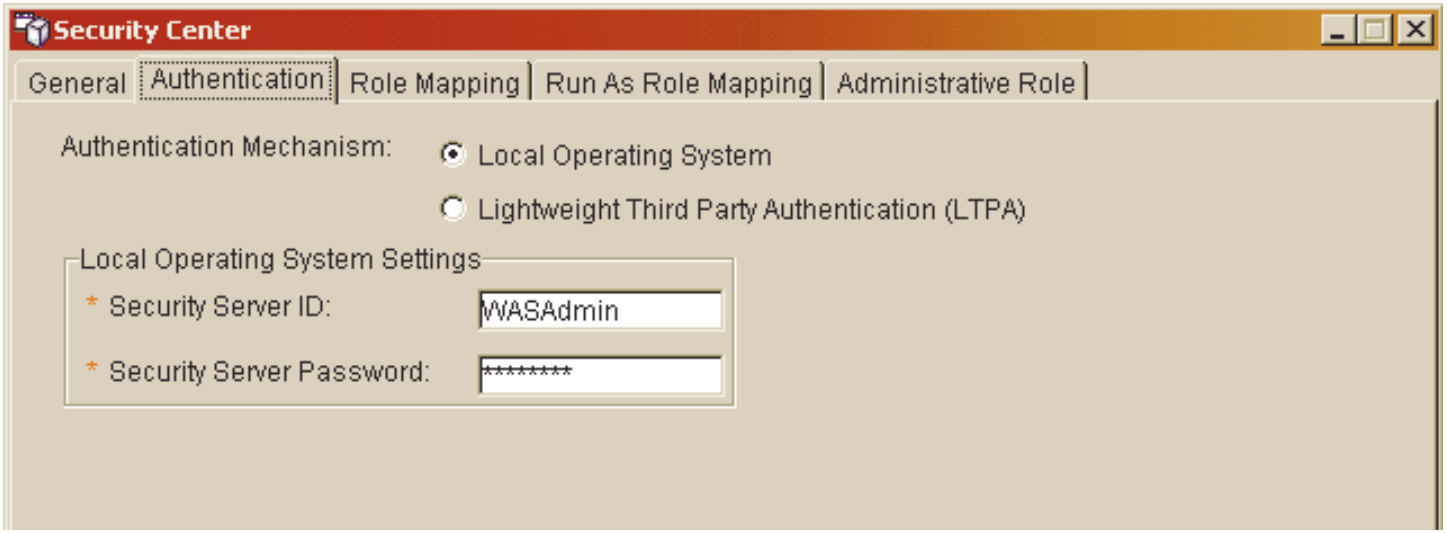
Global system security

Global security applies to all applications
Must be enabled for security services to be available
Define the type of registry to be used
Define SSL configuration
Admin can set a Security server user name and password
Define administrative role
Accessible from Admin client

Securing the Admin Server

We can protect access to the Admin server by requestin
ser credentials before granting access.

User name and password are entered into the Security
Center settings, accessible from the Admin client.



The user must exist in the selected registry.

Securing the Admin Server

The changes will take effect after the Admin server has been restarted.

The administrator will be presented with a dialog box when attempting to connect to the Admin server.

Securing an application

Useful reference for storage locations of application elements.

Application level:

Security information	Storage location	Tool
Define business roles (whole application)	application.xml	Application Assembly Tool (AAT)
Web resources	WAR Deployment Descriptor (DD) web.xml	AAT
EJB resources	EJB DD ejb-jar.xml	AAT

Securing an application

Web resource level:

Security information	Storage location	Tool
Define Web Component Authentication	WAR DD web.xml	AAT
Define security constraints and assign to roles	WAR DD web.xml	AAT
Optionally, define role reference	WAR DD web.xml	AAT | Admin Client (AC)
Secure static resources delivered by the Web server	N/A	N/A

Securing an application

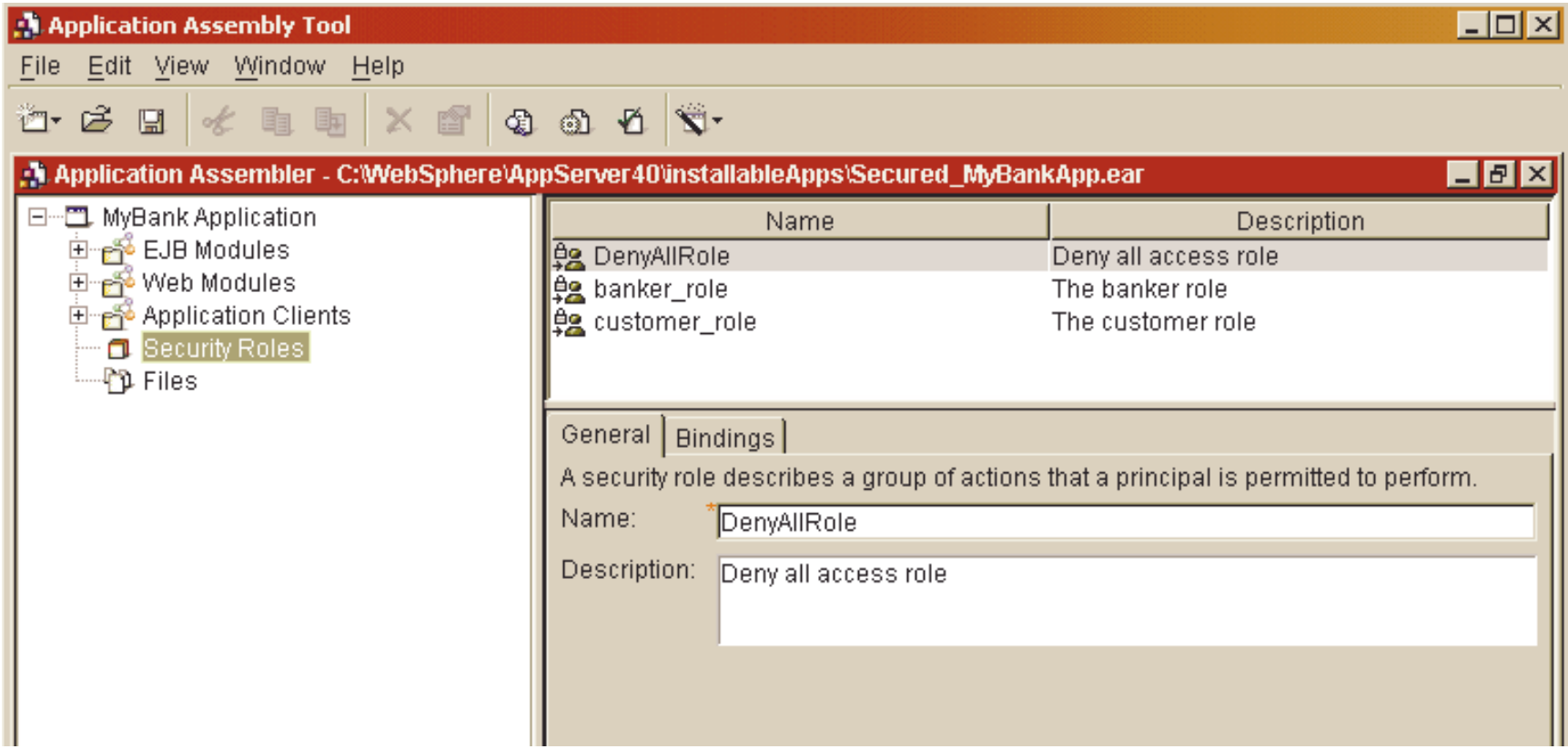
EJB resource level:

Security information	Storage location	Tool
For each EJB, assign each method to one or more roles	EJB DD ejb-jar.xml	AAT
Optionally, set up Security Role references	EJB DD ejb-jar.xml	AAT | AC
Configure the delegation policy - Run As	EJB DD ibm-ejb-jar-ext.xmi	AAT
Run-As mapping	Repository DB	AC

Associate principles	EAR DD ibm-application-bnd.xmi: Repository DB	AAT | AC
Configure global security authentication	Repository DB	AC

Securing an application

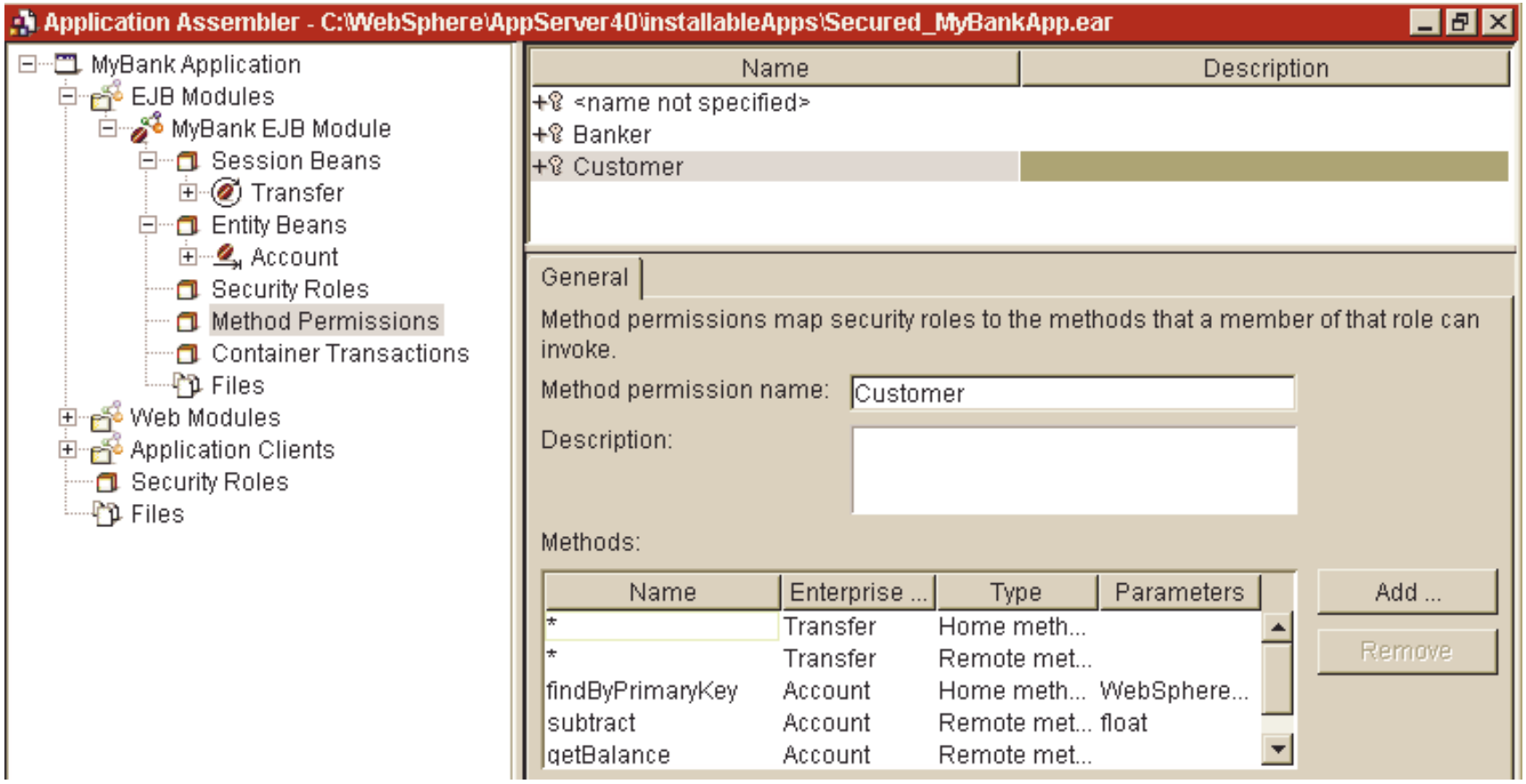
- A simple example.
- Firstly, add users and groups to the local OS registry
- Use the AAT to create security roles



Securing an application

Secure EJB's.

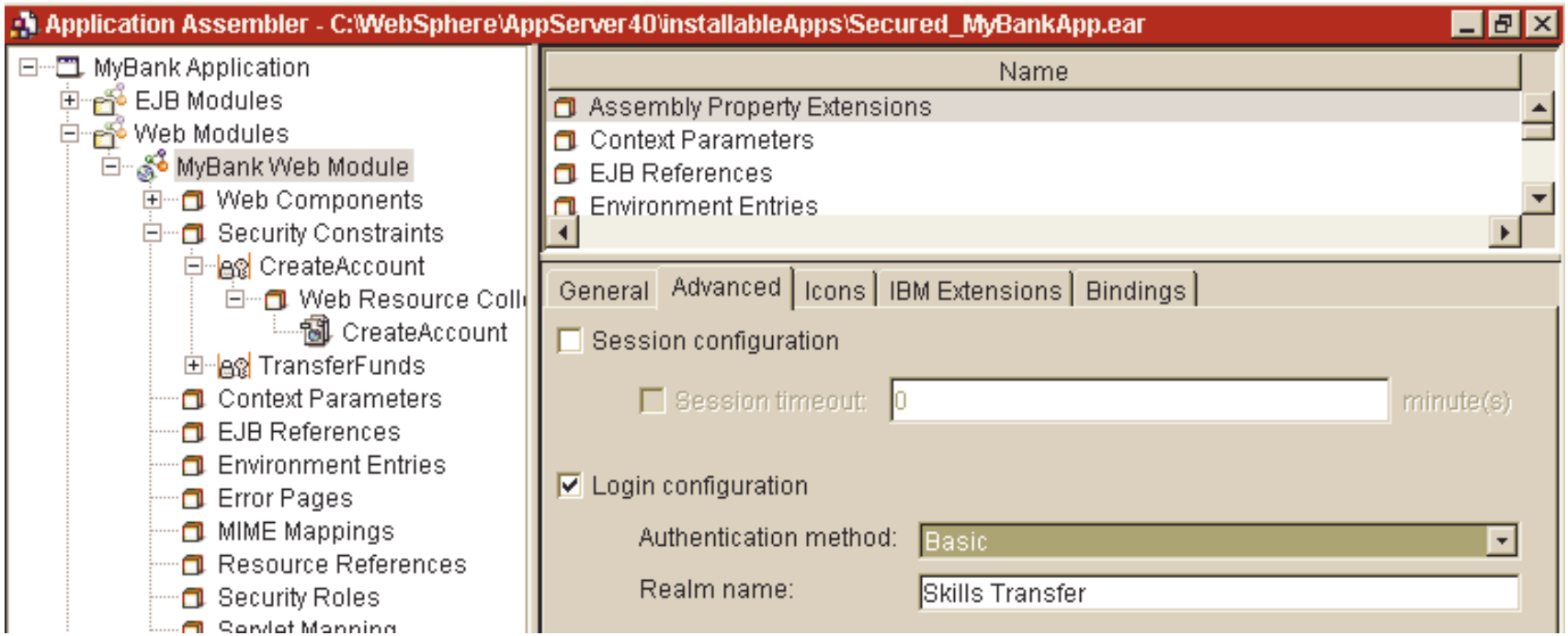
- Create a method permission and add roles
- Select relevant methods to apply permissions



Securing an application

Secure Web applications.

- Set login authentication method



- Define a web constraint to protect web content
- Add a resource collection to the constraint

Securing an application

VAS security must now be enabled (from the Security Center)

- Select registry and enter user credentials
- Restart Admin server

Finally, install secured application.

- Ensure that access is denied to all unprotected methods
- Map groups and users to roles (if not already done in AAT)

Access application and enter credentials when required.

Security differences between 4.0 and 3.5

Version 4	Version 3.x
When global security is enabled, only the resources of the administrative application are protected. All other resources are unprotected.	When global security is enabled, enterprise beans are protected by default.
WebSphere no longer secures or protects URIs, for example, HTML files and CGI scripts, that are served by an external Web server, for example, Apache or IHS. WebSphere secures or protects only URIs served by WebSphere.	WebSphere can protect URIs served by an external Web server.
Deployment descriptors are provided in XML. The web.xml, ejb-jar.xml, and application.xml deployment-descriptor files are used to declare security contraints. Security constraints include the identification of the methods belonging to roles, the login configuration or challenge mechanism, whether HTTPS is required.	Most of application-specific security attributes are defined by using the administrative console during the application's deployment phase.
The login configuration and challenge type apply to individual Web applications, not to individual enterprise applications.	The challenge type applies to an entire enterprise application.

Security differences between 4.0 and 3.5

Version 4	Version 3.x
The local operating-system user registry now supports J2EE form-based login configuration. This means that AEs can now supports the form-based login configuration.	AbstractLoginServlet, CustomLoginServlet, and SSOAuthenticator are features used to create custom or form based login mechanisms for web applications.
Passwords are encoded with a simple masking algorithm in various ASCII WebSphere configuration files to deter casual observation.	Passwords are in plain text.

What have we learnt?

- Seven-step guide to security principles
- Symmetric crypto provides secret key
- Asymmetric crypto provides public/private keypair - basis of SSL
- Digital certificates used to identify sender and/or recipient
- Access control lists and capability lists used for authorisation
- WAS security model - collaborators
- How to secure the Admin server and an application

What haven't we covered?

- Authenticaton mechanisms: basic, digest, client and form-based
- SSL configuration between nodes
- Creation of keys using *iKeyMan*
- Client-side certificates
- Single-sign on mechanisms
- Web trust association (delegate authentication to a trusted reverse proxy)
- LDAP configuration
- Programmatic security