

Using GSS API For Securing Web Services

Jan Alexander

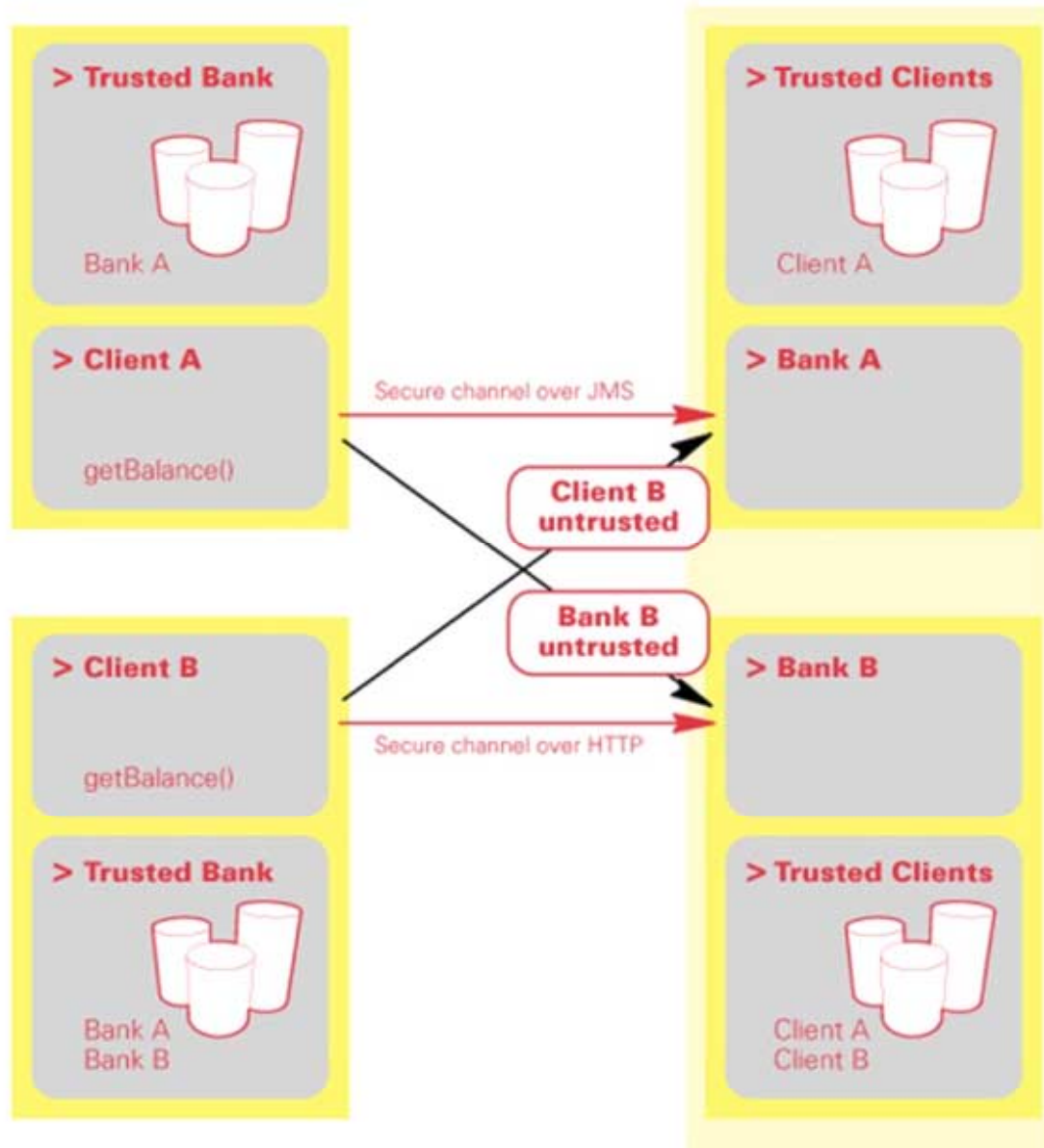
WASP Server Chief Architect

Systinet Corp.

- Web Services and Security Issues
- Security Mechanisms Comparison
- Our Design
- Conclusion



- We want to be security mechanism independent
- We want to be interoperable with existing Web-based authentication systems
- We want to provide Identity delegation, end-to-end security, mutual authentication
- We want to be transport independent
- We want to be able to identify individual services running on one server



- Basic Authentication mechanisms
- SSL/TLS
- GSS API

- SAML

- HTTP Basic Authentication
 - Transport dependent
 - Widely supported by Application Servers
- SOAP Basic and Digest Authentication
 - Based on SOAP Headers
 - Transport independent
 - Only an IETF Draft yet
 - Not widely supported
- For simple use and backwards compatibility only

- RFC 2246
- asymmetric cryptography
- X.509 Certificates
- TCP sockets dependent
- only one identity per host:port pair
- no support for Principal Identity delegation

- RFC 2743, RFC 2853 -Java bindings
- security mechanism independent
 - SPKM
 - asymmetric cryptography
 - X.509 certificates
 - Kerberos
 - conventional cryptography
 - centralized identity database
- transport independent
- support for Principal Identity delegation
- multiple identities per transport endpoint
- support for end-to-end and session message integrity and confidentiality

- Not an Authentication mechanism
- Only description how to express the authentication, authorization and attribute related information
- If accepted, it will allow interoperability between different authentication mechanisms
- Must be complemented by “real” authentication mechanism (like Kerberos, etc.)
- SAML Tokens are not protected against eavesdropping and replay attacks
 - must be solved on transport layer for now
 - should be solved by XML Encryption in the future

- Security Framework
- GSS-API Integration
- GSS/SPKM Implementation
- GSS/Kerberos Implementation
- Other Security Providers

- Use of GSS-API as a main Security Provider
- Use SSL for compatibility with other SOAP clients
- Use Basic Authentication as a simple-to-use simple-to-setup authentication mechanism
- Provide integration with SAML

- Security Provider Independent API
 - Principal Authenticator
 - Credentials
 - Security Provider Abstraction
 - Received Credentials
- Integrated into WASP Server Runtime
- Integrated with Java Authentication and Authorization Service (JAAS)
 - JAAS Subject can be obtained from both Credentials and Received Credentials
- Support for Principal Identity Delegation

- GSS API uses Tokens abstraction
- Application is responsible of transmission of these tokens
- Integration is based on WASP Server Interceptors
 - working on transport independent level
 - allow access to the RAW transport connection
 - transmitted data can be changed by interceptor
- GSS API Credential held in the Web Service context provided by WASP runtime
- Initiating context is associated with Web Service endpoint
- GSS API is responsible for matching GSS Context with incoming token

- Simple Public Key Mechanism - RFC 2025
- Based on JCE implementation
- Uses X.509 certificates
- Supports existing PKIs using Java Certification Path API - JSR 55
 - PKCS.10 Certificate Requests
 - Certificate validation using CRLs, OCSP
 - Support for CRLs distribution points
 - Integration with LDAP

- Kerberos v5 authentication protocol - RFC 1510, RFC 1964
- Compatible with MIT Kerberos
 - support for MIT Kerberos CCache and Keytab files
- Compatible with MS Windows 2000 Active Directory
- Supported both Client and Server sides
- Support for Principal Identity Delegation

- **SSL based Security Provider**
 - for compatibility with existing HTTP clients
- **HTTP Basic Authentication Security Provider**
- **SOAP Basic and Digest Authentication**
 - <http://www.zolera.com/resources/opensource/i-d/soap-auth.html>
 - using SOAP headers
- **Single Sign-on Security Provider**
 - SAML Authentication Assertions
 - Multiple Authentication methods supported
 - Kerberos
 - X.509 Certificates (a.k.a. SSL)
 - Basic Authentication (username, password)

- GSS-API mechanisms for “real” Web Services security in single security technology domain
- SSL and HTTP Basic Authentication for backwards compatibility only
- SOAP Basic and Digest Authentication for simple (toy-like) authentication

- SAML for interoperability between security technology domains
- Strong XML based security mechanisms have yet to be invented and it will take long time before they will be ready for production environment

Questions ?
