MESSAGING SECURITY USING GLASSFISH AND OPEN MESSAGE QUEUE

OWASP AppSec USA 2011 Conference (@appsecusa / hashtag: #appsecusa)
Srini Penchikala (@sринip)
09.23.11
GOALS AND SCOPE

- **Goals:**
  - Messaging security architecture & design considerations and best practices
  - How to use NoSQL Databases for some messaging security use cases

- **Is Not:**
  - Security vulnerabilities talk

- **Is:**
  - Focus on application security in messaging:
    - Authentication & authorization
    - Message encryption
    - Logging and monitoring
  - Code examples on messaging security aspects (Java based)

- **Target Audience:**
  - Architects, Application Developers, and Security Ops

- **Format:**
  - 45 min presentation + 5 min Q&A
  - Demo Application (Java)
ABOUT THE SPEAKER

- Security Architect
- Certified Scrum Master
- Author, Editor (InfoQ)
- IASA Austin Chapter Leader
- Detroit Java User Group Leader (past)
- Future: Role of DSL in Architecture Enforcement, NoSQL Security Tools and Frameworks
Before We Start

- How many are currently using messaging in their applications?
- How many are currently working in security architecture or development?
- Any regulatory compliance requirements (Federal, State, Local, or Finance related)?
BACKGROUND

- Financial services
- J2EE security model
- Agile software development
- Regulatory compliance and its impact on IT
- Software architecture
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
AGENDA
- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
MESSAGING ARCHITECTURE

- Asynchronous Communication
- Message-Oriented Middleware (MOM) pattern
- Java Message Service (JMS)
- Architecture Components:
  - Broker
  - Connection
  - Destination (Queue or Topic)
  - Transactions
  - Message
- Enterprise Integration Patterns*

*Source: http://www.eaipatterns.com/toc.html
MESSAGING SECURITY CONSIDERATIONS

- Current state: Not enough focus on the middleware
- Authentication
- Role Based Access Control
- Message Encryption
- Transport Layer Security
- Message Persistence
- Secure Logging and Auditing
- Secure Message Monitoring
  - Standards based administration (JMX)
- Availability
  - Broker clustering
  - Automatic reconnect
  - Connection event notification
TECHNOLOGIES

- **GlassFish (v3.1)**
  - Java EE 6 compliant application server
  - Open source and commercial versions

- **Open Message Queue (v4.5)**
  - Implements JMS standard
  - Stand-alone service or deployed in an application server
  - Standard JMS Provider in GlassFish Server
  - Supports messaging security at various levels
  - Broker clusters (conventional and enhanced clusters)

- **Spring Integration Framework**
  - Secure message channel (role based access)
SAMPLE DEMO APPLICATION

Technologies:
- GlassFish 3.1 Application Server
- Open MessageQueue
- Neo4J (authentication and authorization)
- MongoDB (persistence and logging)
- Spring Data Framework
- Spring Integration
- Aspect-oriented Programming

Tools:
- JDK 1.6
- Eclipse
SAMPLE APPLICATION ARCHITECTURE

- Client
- SOAP Client
- OpenMQ Server
- JMX Client
- AuthN/ AuthZ
- Neo4J DB
- Mongo DB
- Msg Persistence in DB
- Logging

Mongo DB
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
AUTHENTICATION

- Authentication
  - Flat file repository
  - LDAP authentication
  - JAAS-based authentication
- Per broker repository
- NoSQL database for storing user profiles
  - Graph databases (Neo4J)
JAAS AND MESSAGE QUEUE

Source: OpenMessage Queue 4.5 Administration Guide
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
AUTHORIZATION

- Users and groups
- Pre-defined groups
  - admin
  - user
  - anonymous
- Access control events
  - Connecting to a broker
  - Creating a message producer or consumer
  - Auto-creating or browsing a queue destination
- Configuration based
  - accesscontrol.properties
AUTHENTICATION AND AUTHORIZATION

Source: OpenMessage Queue 4.5 Administration Guide
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
MESSAGING SECURITY

- Broker Level (Connection)
- Destination Level
- Message Level
ENCRYPTION

- Encrypted client-broker communication
- SSL based connection service
- Connection service types
  - ssljmsservice (TCP/IP)
  - httpsjms (HTTPS Tunnel Servlet with HTTP protocol)
  - ssladmin (TCP/IP)
  - Cluster connection service
  - JMX connector (RMI over TCP)
**Encryption Best Practices**

- **Symmetric Key Algorithms:**
  - AES with minimum 128 bit key length

- **Hash Functions:**
  - SHA-256
  - Always use a salt value (salted SHA, SSHA) especially for passwords to defend against rainbow table attacks

- **Asymmetric or Public Key Algorithms:**
  - rDSA with 1024 bit minimum key length

- **Data Integrity:**
  - Data Integrity/Data Signature or Message Authentication Code – HMAC (hash function-based message authentication code)
  - Use any underlying hashing algorithm since HMACs are substantially less affected by the potential for collisions that the related hashing functions alone

- **Secure Network Communication:**
  - SSLv3 or TLS to ensure the encrypted transmission of data between systems

- **Security Standards Java API:**
  - OWASP’s ESAPI libraries
MESSAGE PERSISTENCE

- Configurable persistence
  - File or JDBC-based data store
- Securing a JDBC persistence store
- NoSQL DB for persistence (MongoDB)
Persistence Options

Source: OpenMessage Queue 4.5 Administration Guide
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
SECURITY LOGGING AND AUDITING

- Event management (SIEM) and forensics
- Critical for regulatory compliance (SOX, HIPAA, PCI, FISMA)
- Log events
  - Startup, shutdown, restart, and removal of broker
  - User authentication and authorization
  - Reset of a persistent store
  - Creation, purge, and destruction of destinations
  - Administrative destruction of durable subscribers
- Message Queue audit logging
  - JAAS based audit logging
- Performance:
  - MongoDB Logger
SECURITY LOGGING CONSIDERATIONS

- What data needs to be logged for security analytics purposes?
- What should be the log format for business v. security logs?
- Do we need to store the security logs in a different file (a new log4j appender) so only authorized users (admin) will have access to it?
- How would the logs work with SIEM tool (if applicable)?
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
MONITORING

- Standards:
  - JMX
  - Remote JMX
- Secure JMX
- Tools:
  - JConsole/VisualVM
Monitoring Messaging Components

- Messaging components
  - Broker
  - Connections
  - Destinations
  - Producers
  - Consumers
  - Messages

- Monitoring Tools:
  - VisualVM
  - Open MQ Administration Console
  - Any standards based monitoring tool (e.g. JMX for Java)
MONTORING SERVICES

Source: OpenMessage Queue 4.5 Administration Guide
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
BEST PRACTICES

- Separation of messaging logic from application & business logic
- Messaging code should be agnostic to MQ container
- Declarative v. Programmatic message security
- Emerging trends in messaging architecture
  - Light-weight and Agile message brokers
  - Embedded brokers (for unit testing)
  - Built-in security, monitoring and clustering
- “Build in” security logging and monitoring capabilities into the product as a feature
- Involve Dev & Ops from early phases of project lifecycle
AGENDA

- Messaging and Security
- Authentication
- Authorization
- Message Security and Encryption
- Logging
- Monitoring
- Best Practices
- Conclusions
CONCLUSIONS

- Messaging security is critical
  - Message at rest, in transit and in use
- Messaging security considerations
  - Authentication, Role Based Access, Encryption
- Logging and Monitoring
- Messaging security support in Open MQ container
- Role of NoSQL DBs in messaging security use cases
- “One Size Fits All” fits nothing
RESOURCES

- Messaging Patterns
- Java Message Service (JMS)
- Open Message Queue
- Java Management Extensions (JMX) Technology
- Open Message Queue 4.5 Administration Guide
- Glassfish Samples
- GlassFish Server Open Source Edition 3.1 Security Guide
- GlassFish Server Open Source Edition 3.1 Administration Guide
THANK YOU

- Thank you
- Contact Information
  - http://www.infoq.com/author/Srini-Penchikala
  - srinipenchikala@gmail.com
  - @srinip
- Questions?
BONUS SLIDES
Implement messaging security using Aspects

Architecture:
- Separate security event logic from application and business logic

Tools & Technologies:
- ActiveMQ
- MongoDB
- Esper
- AspectJ and SpringAOP

Demo
INTEGRATING SOAP AND MESSAGE QUEUE