Securing Apache Tomcat

Dennis Jacob
Dennis Jacob, CISSP

- Senior Consultant @ Leading Payment Industry organization
- Part of Middleware Engineering Group
- Interests
  - Application Server Technologies
  - Cloud Native Technologies
  - Web application security

https://www.linkedin.com/in/dennis-jacob
https://github.com/dennisjacob

This session and the opinions expressed are made on my personal capacity, and do not reflect the official policy or position of the company I work for.
Agenda

Introduction to Securing Tomcat
Key Security Considerations for Tomcat
Securing Configurations in Tomcat
Transport Layer Security
Secure Authentication/Authorization
Secure Request Processing
Secure Session Management
Secure Logging and Auditing
Vulnerability Management
Introduction to Securing Tomcat

Why security is important for Tomcat?

• Critical component in the web application stack
• Building trusted application services
• Preventing unauthorized access to application and data
• Protection against web attacks
• Maintain availability of web applications
• Compliance and regulatory requirements
• Incident Response Preparedness
Key Security Considerations for Tomcat

- Application Security
- Access Controls
- Scanning and Remediation
- Monitoring and Incident Response
- Secure Build
- Securing Configuration
Secure Configurations in Tomcat

- Release Integrity
  - Package File checksums and PGP Signature Verification
  - Using custom targets in ANT for building Tomcat from source
- Removal of default packaged applications
- Modification of default configurations and properties
- Securing Manager Applications

- Separate CATALINA_HOME, CATALINA_BASE and logs
- Implement proper file/directory ownership and permissions
- Enabling SELinux

- Remove default passwords
- Securing sensitive passwords in Secure Vaults or custom Encryption utilities
- Modify the default advertised server information
Transport Layer Security

• Securing Connectors
  • Removal of unused connectors
  • Certificate configurations
• Transport Layer Security
  • TLS Protocols (TLS 1.2 / TLS 1.3)
  • Cipher suites (Perfect Forward Secrecy Cipher Suites)
    • Implementing perfect forward secrecy
• Securing JMX or any exposed end points
• Enabling 2-way TLS for API Endpoints
Secure Authentication/Authorization

- Default Credentials provided with Host Managed in Tomcat package
- Basic Authentication Vs Digest Authentication
- Form Based Authentication
- Single Sign-On
- Realms and External Authentication
- Custom Valves
  - mTLS Authentication and certificate pinning
Secure Request Processing in Tomcat

- **Tomcat Filters**
  - CORS Filter
  - CSRF Prevention Filter
  - HTTP Header Security Filter
  - Remote Address Filter
  - Remote CIDR Filter
  - Remote Host Filter
  - Remote IP Filter

- **Tomcat Valves**
  - Remote Address Valve
  - Remote Host Valve
  - Remote CIDR valve
  - Error Report Valve
  - Custom mTLS Valve
  - Rate Limiter Valves
Secure Session Management in Tomcat

- HTTP-Only and Secure session cookies
- Session Timeout
- Custom Session Listeners
  - Session Validation
  - Proper Session invalidation
- Session Monitoring and Auditing
Secure Logging and Auditing in Tomcat

- Importance of a “balanced” logging
  - What is essential to be logged?
  - Performance implications
- Securing Log and archived Locations.
  - Access restricted
  - Tamper proofing
- Implementing Log Rotation and Retention.
- Masking and Redaction of sensitive information.
- Custom Log Valves to feed to stream processing systems.
  - Kafka | Flink | Elastic Search
- Log Integrity and Tampering Detection.
- Integration of Logs with SIEM solutions.
Vulnerability Management

- Patch Management.
- Tomcat Release updates.
- Application Security Scans (SAST and DAST)
- Vulnerability scans and Software Composition Analysis (SCA).
- Configuration Management
  - Version controlled
  - Securely stored
MBean Attributes for configuration management

- CIS Benchmarking (https://www.cisecurity.org/cis-benchmarks)
- Monitoring large number of Tomcat instance configurations
- Capturing MBean Attribute values for properties picked up by Tomcat

```java
// Create an instance of MBeanServer
MBeanServer mBeanServer = ManagementFactory.getPlatformMBeanServer();

// Define the object name for Tomcat's MBean
ObjectName objectName = new ObjectName("Catalina:type=Server");

// Get the MBean attribute value
Object attributeValue = mBeanServer.getAttribute(objectName, "serverInfo");

// Print the attribute value
System.out.println("Tomcat Server Info: " + attributeValue);
```
Thank You

Dennis Jacob

https://www.linkedin.com/in/dennis-jacob

https://github.com/dennisjacob