Securing Jakarta EE (Java EE) Application Servers:

An Executive Guide

Power Up Your Java Enterprise

User Guide

Contents

Guide Updated: **July 2025**

This executive guide looks into the important aspects of securing Jakarta EE (formerly Java EE) application servers. This guide provides a concise overview of critical security considerations for Jakarta EE application server runtimes, tailored for executives seeking a quick understanding of essential points.

Application Servers Security Vulnerabilities

Jakarta EE application servers and the applications they run are not immune to security vulnerabilities. These vulnerabilities are key in establishing effective security measures. Here's an overview of common vulnerabilities in popular application servers:

Vulnerability	Description	Mitigating Techniques
Insufficient Transport Layer Protection (ITLP)	Exposes sensitive data during transmission due to inadequate encryption or outdated security protocols. Attackers can intercept and steal sensitive information.	• Implement strong encryption protocols like TLS 1.3 • Ensure all communication channels are properly encrypted • Regularly update security certificates • Enforce HTTPS
SQL Injection (SQLi)	Allows attackers to inject malicious SQL code into database queries, granting unauthorized access to, modification, or deletion of data.	• Use parameterized queries or prepared statements • Sanitize user inputs • Use least privilege principles for database access • Implement input validation
Unpatched Libraries	Introduces security risks due to known vulnerabilities in outdated servers and libraries.	• Use supported application servers • Use vulnerability scanners for dependencies • Implement a patch management process
Cross-Site Scripting (XSS)	Enables attackers to inject malicious scripts into web pages, potentially stealing session data, hijacking sessions, or redirecting users.	• Sanitize user inputs and encode output • Use Content Security Policy (CSP) • Implement HTTP-only cookies

Vulnerability	Description	Mitigating Techniques
Denial of Service (DoS)	Aims to disrupt server availability by overloading it with requests, causing crashes or unresponsiveness.	• Implement rate limiting • Use a Web Application Firewall (WAF) • Use load balancing • Use production-optimized servers
Arbitrary File Read (Directory Traversal)	Allows attackers to access restricted files by manipulating file paths, leading to information disclosure.	• Implement strict input validation • Use chroot jails • Use least privilege principles • Avoid storing sensitive data in public directories
XML External Entity (XXE)	Allows processing of external entity references in XML documents, enabling server-side request forgery or file disclosure.	• Disable external entity processing • Use safe XML parsers • Validate and sanitize XML input • Configure XML parsers securely
Session Fixation	Attackers force users to use known session identifiers, enabling session hijacking.	• Generate new session IDs after authentication • Invalidate existing sessions on login • Use secure session management • Implement session timeout
Remote Code Execution (RCE)	Enables attackers to execute arbitrary code on the server through various vectors.	• Keep servers and dependencies updated • Implement strict input validation • Use security headers • Disable unnecessary features
Server-Side Request Forgery (SSRF)	Allows attackers to make requests from the server to internal or external systems.	• Validate and sanitize URLs • Use allowlists for external requests • Implement network segmentation • Monitor outbound connections

Vulnerability	Description	Mitigating Techniques
Insecure Deserialization	Enables attackers to execute arbitrary code by manipulating serialized objects.	• Implement integrity checks • Use secure deserialization libraries • Validate serialized data • Consider alternatives to serialization
Missing Function Level Access Control	Allows unauthorized access to server functions due to inadequate permission checks.	• Implement role-based access control • Verify permissions at the function level • Use secure defaults • Audit access controls regularly

Security Best Practices for Jakarta EE Application Servers

Implementing security best practices is important for building and deploying secure Jakarta EE applications. Here are some key recommendations:

Secure Coding Practices

- **Input Validation:** Validate and sanitize all user input to prevent injection attacks and other vulnerabilities.
- **Authentication and Authorization:** Use reliable authentication mechanisms, such as multi-factor authentication (MFA), and implement role-based access control (RBAC) to restrict access to sensitive resources.
- **Secure Session Management:** Use secure cookies, session timeouts, and regenerate session IDs upon login to protect user sessions.
- **Error Handling:** Implement proper error handling and avoid revealing sensitive information in error messages.
- **Secure Logging:** Log security events and anomalies, and use log analysis tools to identify suspicious activities.

Secure Deployment Practices

- **Keep Software Up-to-Date:** Choose supported and fully maintained application servers over unsupported and outdated free ones to ensure security and stability. Supported servers receive regular security patches and updates, protecting against known vulnerabilities. They also offer better performance, scalability, and support, reducing the risk of application failures and data breaches. Using unsupported software exposes applications to security risks and makes troubleshooting and maintenance difficult.
- **Secure Configuration:** Configure application servers with strong security settings, such as disabling unnecessary services and enabling secure protocols.
- **Network Security:** Use firewalls and network filters to control traffic and block malicious requests.
- **Secure Communication:** Use SSL/TLS protocols to encrypt communication between clients and servers.

Security Hardening Guides and Techniques

Security hardening involves implementing measures to strengthen the security of your Jakarta EE application servers. Here are some key techniques:

- **Disable Unnecessary Services:** Disable any services or features that are not required for your applications to reduce the attack surface.
- **Restrict Access:** Limit access to administrative interfaces and sensitive resources to authorized personnel only.
- **Enable Security Features:** Enable security features such as authentication, authorization, and encryption to protect your applications and data.
- **Regular Security Audits:** Conduct regular security audits and penetration testing to identify and address vulnerabilities.

Compliance with Security Standards

Jakarta EE application servers can be configured to comply with various security standards, such as PCI DSS and HIPAA. These standards provide guidelines for protecting sensitive data and ensuring the integrity of systems that handle sensitive information.

Choosing a Jakarta EE Application Server Based on Security

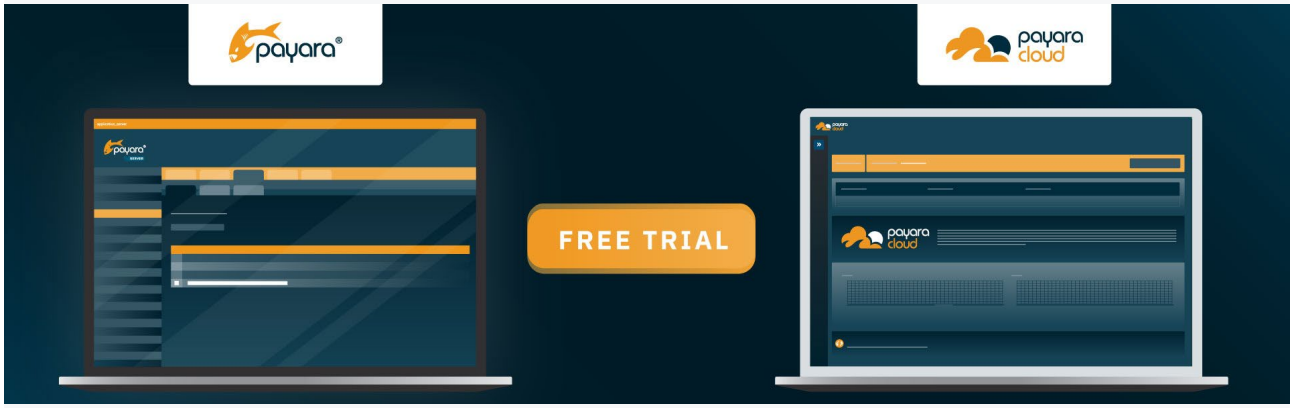
When selecting an application server, consider the following security factors:

- **Security Features:** Evaluate the built-in security features of the application server, such as authentication mechanisms, authorization frameworks, and encryption capabilities.
- **Compliance:** Ensure the application server supports the required security standards for your application.
- **Vulnerability Management:** Consider the application server's track record for addressing vulnerabilities and the availability of security updates.
- **Commercial Enterprise Support:** Opt for an application server with a reliable commercial support offering, including guaranteed service-level agreements (SLAs), dedicated support engineers, and timely security updates for production environments.

Conclusion

Securing Jakarta EE application servers is an ongoing process that requires a comprehensive approach. Understanding the potential vulnerabilities, implementing security best practices, and using security hardening techniques significantly enhance the security of application servers and sensitive data protection. For discussions on Payara Server Enterprise, the production supported Jakarta EE application server, reach out to Payara sales.

Interested in Payara? *Try Before You Buy*



PAYARA ENTERPRISE FREE TRIAL

PAYARA CLOUD FREE TRIAL

sales@payara.fish

UK: +44 800 538 5490
Intl: +1 888 239 8941

www.payara.fish

Payara Services Ltd 2025 All Rights Reserved. Registered in England and Wales; Registration Number 09998946
Registered Office: Malvern Hills Science Park, Geraldine Road, Malvern, United Kingdom, WR14 3SZ