Understanding the Business Risks of Using JBoss EAP 7 Application Server in Production Environments

Power Up Your Java Enterprise

User Guide

Contents

Guide Updated: **September 2025**

Understanding the Business Risks of Using JBoss EAP 7 Application Server in Production Environments

Legacy Java EE server runtime environments force your enterprise to depend on outdated and unsupported software. This can open a dangerous portal to security vulnerabilities that malicious hackers can exploit, exposing the entire application infrastructure.

Before a runtime becomes completely unsupported, it first enters a reduced-support phase, as is now the case with JBoss EAP 7 in Extended Lifecycle Support – Phase 1 (ELS-1), your enterprise is suddenly depending on a platform with limited updates, minimal fixes and no ongoing innovation.

Thus, while the application server is not yet completely unsupported, relying on such technology for key applications creates a widening gap between your production environment and current security best practices. Unpatched vulnerabilities, outdated libraries and unsupported configurations can all become open invitations for malicious actors. One exploited weakness could compromise your authentication and authorization systems, potentially exposing high-privilege administrator accounts — and from there, databases, services, and sensitive data.

It's not a question of if attackers will try. It's a matter of when they succeed if the underlying runtime is left to age without full support.

In this guide, we explore the specific business risks of running JBoss EAP 7 in ELS-1 and outline practical steps to safeguard your applications by migrating to a fully supported, runtime platform that offers extended support to Java EE.

What Happens When an Application Server Enters Extended Life Support

An application server "runtime" is the essential environment your DevOps team uses to run enterprise applications. Without it, your software can't be delivered to customers or internal users.

But when that runtime moves into **Extended Life Support**, as JBoss EAP 7 has, you're no longer getting the same level of security patches, bug fixes, compatibility updates as well as technical assistance as during full support. While ELS-1 may provide limited critical fixes, the cadence is slower, feature development and software enhancement have stopped. Also, since the Java EE platform transitioned to **Jakarta EE** at the Eclipse Foundation, the compatibility gap with newer Jakarta EE standards begins to widen.

As production systems relying on JBoss EAP are no longer aligned with the most current, secure and compatible specifications, they are more vulnerable to key risks. These are real, not theoretical, and have already manifested.

One of the most notable examples was the **SamSam ransomware** campaign. This directly targeted systems using older, vulnerable versions of JBoss Application Server (AS), now known as WildFly. Attackers leveraged the JexBoss tool to scan for vulnerabilities, deploy webshells and encrypt data server-side. This ransomware disrupted K-12 schools and hospitals across the U.S. If your organization is still operating on a runtime in ELS-1, you could be exposing your business to face identical issues in the future.

More precisely, production environments and their business-critical applications left vulnerable in this way are leaving the door open for:

- An imminent breach of the runtime environment, application, services and private company as well as customers' data
- Undesirable and substantial business risks – both financial and legal
- A situation where you have nowhere to turn to get support to fix vulnerabilities quickly, since the runtime you use is only partially supported

What's more, in sectors with strict requirements, it would be difficult to be compliant with regulations and compliance standards in such a setup.

The Hidden Costs of Staying on JBoss EAP 7 ELS-1 in Production

The costs of running JBoss EAP 7 in ELS-1 extend well beyond the immediately visible security headaches. At first glance the trade-off may look purely operational a temporary respite compounds into measurable, ongoing expense across engineering, operations, legal as well as the bottom line.

Limited vendor support amplifies both risk and cost. When a runtime is in ELS-1, v**endor engagement narrows**: security fixes are focused, feature work ceases and qualifying for an emergency patch becomes a higher bar. That means escalation paths are longer and some classes of problems are pushed back to the development team.

In particular, to ensure business continuity, **security management overhead** becomes more burdensome and more expensive. Without a steady stream of vendor patches, vulnerability management moves from "apply vendor patch" to "identify CVE, determine exploitability, develop and test mitigation, backport fixes or deploy compensating controls."

As a result, teams spend increasing proportions of their sprint capacity addressing runtime issues. As such, **developer time drain** is the most obvious and most persistent hidden cost.

What Could Happen if a Malicious Hacker Gained Access to Your JBoss 7 Application Servers and Their Infrastructure?

By exploiting an unpatched security vulnerability in a runtime stuck in reduced support, attackers could:

- Gain access to databases that contain user credentials, such as usernames and passwords – including those belonging to administrators with wide-ranging access from their respective logins
- Find all types of data:
  - International banking as well as other financial lodgments and transactions at scale
  - Country/state-wide passports, government IDs and citizen records
  - Airline schedules, company travel bookings and itineraries
  - Shipping and logistics schedules and manifests
  - Hospital groups and trusts' treatment targets, outcomes and reporting
  - State-wide communications/utilities capability, disaster planning, access procedures and performance
- Inject malicious code into your website application to delete, steal, replace, publish or sell some of that data
- Redirect your application's genuine users to illegitimate websites to 'phish' or 'pharm' their personal data
- Flood your application with requests that ultimately result in a DDOS (denial of service) to your users (a 'crash' where your application goes offline)
- Delete, steal or replace parts of your application code
- Cause unplanned outages or downtime

Business Implications Following a Breach

The list above looks only at the software and data implications – severe though they are. But there are wider business implications for large enterprises relying on JBoss EAP 7 servers, especially if these are managing thousands of user records, as is the case for financial institutions, government agencies or utility companies. Let's examine each in turn.

Financial Losses

In 2025, the global average cost of a data breach is estimated to be USD 4.44 million and, in most situations, recovery takes longer than 100 days. When security incidents involve shadow AI, the cost is typically USD 200,000 higher. Even more, 32% of breaches result in fines and, in half of these cases, these are above USD 100,000.

Government Penalties

Data protection regulations worldwide carry substantial fines for breaches. Even more, these penalties do not stop with your organization and can hold security leaders, such as CIOs and CISOs, personally accountable. Running an application server in ELS-1 risks complicating compliance with standards that require active maintenance, patch cadence or support SLA. In effect, it could be interpreted as failing to take “reasonable measures” to protect data.

Insurance Pressures

Cyber insurers increasingly demand evidence of proactive patching, secure configurations and built-in application security before offering coverage or lower premiums. Many will not cover the time and resources required to investigate, patch and recover from a breach, including recovering any lost data, retraining for your team, any ransom demands, some legal costs, nor the unpredictable loss of reputation and sales.

Reputational Damage

Customers, partners and insurers pay attention to your adherence to security standards. such as PCI-DSS, HIPAA, ISO 27001 and OWASP guidelines. Falling short while competitors maintain compliance can harm your standing in the market. The public fallout from SamSam victims shows how difficult it is to rebuild trust.

Litigation

Some affected users will leave, while others will seek legal redress, potentially triggering cascading lawsuits. Organizations impacted by SamSam also faced legal scrutiny for failing to maintain current, secure infrastructure.

Business Implications Beyond Security Risks

The risks of running JBoss EAP 7 in ELS-1 extend far beyond the security domain and can impact operations in multiple ways.

Limited Portability & Vendor Lock-In

Remaining on JBoss EAP 7 enhances vendor lock-in. As the wider enterprise Java ecosystem moves forward, your application will be able to use fewer third-party tools, libraries, platforms and other technologies. The more you remain on a shrinking foundation, the harder and more disruptive it becomes to move to a modern/different runtime.

The competitive implications are equally serious. On a frozen platform, user experience improvements are harder to implement. Besides, customer expectations can outpace what your infrastructure can provide. Over time, the technology gap can become a market perception gap, with your brand seen as lagging behind in both capability and innovation.

Modernization Delays & Technical Debt

Modernization delays create a growing reservoir of technical debt that becomes more expensive to address over time. Each month spent on a partially supported runtime adds another layer of out-dated libraries, stale APIs and brittle integrations. When the eventual migration does come, it will require more resources, longer testing cycles and more extensive code rewrites. In the meantime, engineering teams find themselves devoting increasing hours to firefighting compatibility problems, retrofitting manual patches and maintaining fragile configurations rather than delivering innovation.

Operational Inefficiencies

There is also the operational drag of day-to-day inefficiency. Without the safety net of timely vendor fixes, every incident takes longer to diagnose and resolve. Patch deployment windows stretch as teams navigate an ever-growing list of potential compatibility issues. Infrastructure becomes more fragile, with small changes in one service triggering unexpected issues in others. In addition, the reduced releases for JBoss EAP 7 make it harder to plan updates to your application with confidence.

Compliance teams, too, can feel the strain, and may feel the need to develop more comprehensive documentation or additional testing to prove compensating controls are in place.

In short, staying on JBoss EAP 7 in ELS-1 may seem expedient in the near term, but the hidden costs compound quickly across productivity, vendor dependence, security operations, compliance and business continuity.

How to Minimize the Risks Posed by End-of-Life Runtimes

So, how do you deal with Java EE 8 applications that rely on JBoss 7? Here are some ways to protect your systems and minimize the risks.

Select a Compatible, Fully Supported Application Server with High Portability

When your application server enters a limited-support phase like ELS-1, the vendor's priorities have already shifted toward newer products. This isn't necessarily bad. In fact, it's part of the natural technology lifecycle. However, it can leave development teams exposed if there's no clear plan for how and when to modernize.

By contrast, some vendors continue to offer full support for Java EE 8-compatible runtimes for much longer. Payara Platform Enterprise 5, for example, remains in full support well beyond JBoss EAP 7's ELS-1 phase. For development teams, the move to a such solution offers a strategic advantage. It lets organizations reduce any risks associated with runtime obsolescence while giving them the ability to modernize at their own pace, aligning with internal readiness rather than being forced into a rushed migration dictated by a vendor's end-of-support timeline.

Regular Updates, Patches and Fixes Beyond Full Support

Choosing a runtime that is still in Full Support is ideal, as you'll get the latest features, performance enhancements and broad compatibility with modern tooling. But it's equally important to think ahead and evaluate what happens after that phase ends.

While some vendors sharply reduce coverage in extended support stages, others take a more comprehensive approach, ensuring that even post–Full Support, your platform remains secure, stable and operationally viable.

For example, Payara Platform Enterprise's Lifetime Support, which follows its Full and Maintenance Support cycles, continues to provide critical bug fixes, security patches and active technical assistance. This means teams can rely on predictable, high-quality support well into the future, avoiding the risk of running production workloads on an unpatched, end-of-life runtime.

Select a Vendor with Robust & Responsive Support SLAs

In case of vulnerability management and data breaches, robustness and responsiveness are key. Thus, selecting an application server vendor that offers solid service level agreements and quick response time is key to minimizing the impact of potential issues. In effect, the speed and quality of vendor support directly impact your team's productivity and issue resolution.

While JBoss EAP offers a 1-hour guaranteed response time for critical support cases, Payara's median response time is just 0.4 hours. In addition, it offers Enterprise customers a Diagnostic Tool to streamline issue resolution that is unique in the market. This means most Payara customers experience faster initial engagement from expert engineers, enabling quicker troubleshooting and less downtime.

Educate Your Team

One of the most overlooked risks in enterprise environments is simple unawareness. Many teams don't realize that their runtime is about to enter or already is in a reduced-support phase, and don't necessarily know what it means in practical terms for their activities.

Start by communicating to all stakeholders what lifecycle stage your middleware is in as well as what that means for day-to-day operations and long-term planning. Map your applications that depend on the affected runtime, classify them by business criticality, and assign migration or modernization paths. Build this into your project portfolio so that resource allocation and budget planning are aligned with the end-of-support deadline.

Finally, avoid treating lifecycle awareness as a one-off exercise. Make it part of your operational culture.

Do You Want to Confidently Reassure Your Customers Their Data is Secure?

The strongest and most effective tactic you can adopt to underpin the long-term security of your business-critical applications from attacks is to deploy a fully supported Java EE 8-compatible application server with regular updates and patches, such as Payara Platform Enterprise 5. While JBoss EAP 7 is not yet completely unsupported, its ELS-1 status means the protections around your production environment are already weakening.

If you continue running on ELS-1, each month widens the gap between your systems and current security expectations, increasing the likelihood of an incident like SamSam that impacts revenue, reputation and compliance. By opting for a straightforward migration from JBoss EAP 7 to Payara Platform Enterprise 5, you can enhance the support you receive, reduce costs while reducing the resources needed to modernize at your own pace in the future.

If you want to find out how to deploy your application right into production with a secure and developer-oriented Java EE 8 application server, Try Payara Server Enterprise for free today.

sales@payara.fish

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

www.payara.fish