Guide

TuxCare

Making Spring® Last Forever:

How to Skip the Code Rewrite and Safely Keep Using End-of-Life Spring® Versions



Navigating this Whitepaper

3 / Introduction

4 / When Spring® Hits End of Life

4 / Why Not Just Upgrade?

5 / Endless Lifecycle Support (ELS) for Spring®

7 / Gaining Spring® Security for the Long Haul

8 / About TuxCare





Introduction

Spring® has become synonymous with modern enterprise Java development. Its ecosystem – including Spring® Framework, Spring® Boot, Spring® Security, Spring® Cloud, and others – is foundational for building scalable, flexible, and cloud-native applications. The framework powers everything from internal line-of-business tools to globally distributed SaaS platforms and mission-critical services.

However, as with all software, Spring® versions follow a predictable lifecycle: active development, maintenance, and ultimately, end of life (EOL). Once EOL arrives, the original maintainer stops delivering security updates, bug fixes, and compatibility improvements. This isn't unique to Spring® – it's the open-source contract.

When the version of Spring® your applications depend on reaches the end of support, the risks don't simply fade – they increase each day. Without ongoing security patches, each passing day opens the door wider to cyber threats, compliance violations, and operational risks. Worse still, threat actors are fully aware of this dynamic. Vulnerabilities in popular end-of-life software are a prime hunting ground for ransomware operators, data thieves, and nation-state actors.

For organizations that use Spring® as part of their Java stack, EOL introduces a profound dilemma – continue using the out-of-support version or deal with the challenges of upgrading to a newer release.

But, for many organizations, upgrading is far from a simple fix. Whether it's deeply embedded technical debt, fragile legacy code, compatibility constraints with third-party systems, or the sheer expense of migration, countless teams find themselves locked into aging Spring® versions for longer than they'd like.

This guide aims to bridge the gap between the need for stability and the demands of security – showing how organizations can stay secure even when EOL becomes a reality.

When Spring® Reaches End of Life

For most organizations, the announcement of an EOL date feels distant and theoretical – until it isn't. When a critical dependency like Spring® Boot 2.7 or Spring® Framework 5 reaches EOL, the implications are concrete and immediate:

No More Security Patches

New CVEs won't be patched by the Spring® maintainers. If an exploit is published tomorrow, your systems remain exposed indefinitely.

No Bug Fixes or Maintenance

Operational bugs that previously received fixes are now your problem to solve internally.

Compatibility Risks Increase Over Time

As surrounding technologies – like Java versions, cloud providers, and libraries – evolve, your EOL Spring® application becomes increasingly brittle.

Mounting Compliance Pressure

Regulatory frameworks like PCI DSS, HIPAA, GDPR, and NIS2 require demonstrable security posture. Running unsupported software is an audit failure waiting to happen.

Spring® Framework 5.3 and Spring® Boot 2.7 officially went EOL in 2024. Despite that, millions of production workloads worldwide continue to depend on them. The stark reality is that, for many organizations, upgrading isn't feasible on EOL's timeline – and yet, neither is accepting the security risk of doing nothing.

Why Not Just Upgrade?

On the surface, upgrading seems like the logical solution – and it often is, eventually. But anyone with hands-on experience managing enterprise software knows the decision isn't that simple. There are very real barriers that prevent organizations from upgrading Spring® versions on demand:

1 Legacy Code and Architectural Constraints

Many Spring® applications were developed years ago – long before modern patterns like microservices, reactive programming, or cloud-native deployment became widespread. These systems are tightly coupled, often monolithic, and resistant to change. Adapting a legacy codebase to align with new Spring® versions can require a ground-up rewrite.

2 Dependency Lock-in

A Spring® upgrade rarely happens in isolation. Spring® upgrades often require simultaneous upgrades of:

- Java runtime versions (e.g., Java 8 Java 17 or 21)
- ORM tools like Hibernate
- Security frameworks like Spring® Security
- Middleware, messaging systems, and databases

This cascading dependency chain multiplies both complexity and risk.

3 Budget, Resources, and Time Constraints

Development teams are already stretched thin with feature development, maintenance, and incident response. Diverting resources to a massive upgrade project delays business goals, impacts velocity, and can require external contractors with niche skills – ballooning costs.

4 Operational and Stability Risks

Even after a successful code migration, the testing burden is immense. Functional testing, integration testing, performance regression, and load testing must be exhaustive. A single overlooked incompatibility could cause critical outages in production.

The Bottom Line

For many organizations, upgrading immediately simply isn't viable. But sitting idle with unpatched vulnerabilities is equally unacceptable. A third path is needed.

Endless Lifecycle Support (ELS) for Spring®

TuxCare's Endless Lifecycle Support (ELS) offers that third path – a proactive way to maintain security, stability, and compliance while remaining on your current Spring® version for as long as the business demands.

What is ELS for Spring®?

ELS delivers ongoing security patches for end-of-life versions Spring® Framework, Spring® Boot, and other Spring® Projects that are beyond their official support lifecycles. These are production-grade fixes for all risk levels of vulnerabilities, including critical and high-severity bugs – delivered without requiring any modification to your codebase.

Extended security patching for vulnerabilities in Spring® Framework, Spring® Boot, and other dependent Spring® projects for as long as you need it

14-day SLA
for High and Critical CVE patches

ELS FOR SPRING

Patches for Medium-severity CVEs in
60 days

Patches for Low-severity CVEs
90 days

Supported Spring® Projects

Endless Lifecycle Support for Spring® takes up the security support baton for the key Spring® ecosystem projects. You can request support for additional Spring® packages that are not currently on our list.

- Spring® Framework
- Spring® Boot
- Spring® Security
- Spring Data
- Spring GraphQL
- Spring® Batch
- Spring® Web Services
- Spring® AMQP
- Spring® Integration
- Spring® Hateoas
- Spring® LDAP
- Spring® Plugin
- Spring® Retry

Our Service Expands Your Protection by Also Patching Transitive Dependencies, Including these Key Projects:

- Google Guava
- Apache Santuario
- Apache Velocity
- Apache Kafka
- SnakeYAML
- Snappy Java
- Woodstox Core
- Jackson Databind
- Plexus Common Utilities
- Netty

What's Included in Endless Lifecycle Support?

Backed by decades of open-source expertise and a patching process refined to perfection, we keep the open-source stack that you run on secure, stable, and enterprise-ready – long after official support ends.

Because resilience isn't built once; it's maintained every day.

Rapid CVE Response

Quickly-deployed SLA-backed security patches for as long as you need

Preserved Compatibility

Extensively tested updates that work flawlessly with your existing code

Seamless Integration

Plug-and-play integration with your workflows and CI/CD pipelines

Secure Distribution

Signed, tamper-proof updates from TuxCare-hosted repositories you can trust

Trusted Support

24/7 expert assistance with CVE triage, patch request, and troubleshooting

Set Up in Just a Few Clicks

STEP 1

Get your ELS for Spring access credentials®

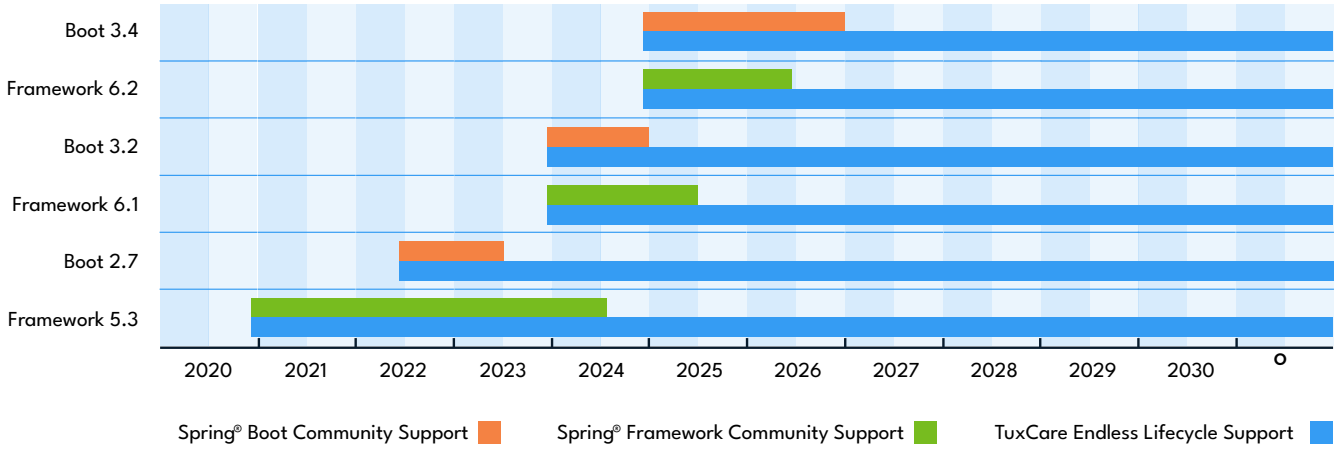
STEP 2

Adjust your Maven/Gradle configuration files

STEP 3

Switch to the TuxCare versions of your libraries

TuxCare extends your protection for as long as you need:



Gaining Spring® Security for the Long Haul

End of life doesn't have to mean end-of-security. Spring® may reach end-of-life, but your security posture shouldn't. Vulnerabilities don't disappear when upstream maintainers move on – in fact, the risks may even grow. But with the right strategy, you don't have to choose between stability and security.

TuxCare's Endless Lifecycle Support for Spring® enables you to navigate this challenge confidently – delivering continuous security updates for legacy Spring® applications while buying your teams the time they need to migrate on their terms.

With ELS for Spring,® your organization gains the ability to:

- **Close critical security gaps continuously – no matter how old the version**
- **Shift migration to your timeline – not the vendor's**
- **Extend the operational life of stable, revenue-generating applications**
- **Save DevOps resources by avoiding rushed replatforming or rewrites**
- **Maintain compliance with evolving regulations like PCI DSS, HIPAA, NIS2, and more**

When Spring® ends, your protection doesn't. When you decouple security from forced upgrades, you take back control of your roadmap.



About TuxCare

TuxCare is a global leader in open-source security, providing unmatched expertise in patching and security support for your enterprise systems. We deliver automated security patches to popular Linux distributions without reboots, long-term security updates for end-of-life products, next-generation vulnerability scanning, and enterprise-grade support for community-powered Linux systems – offering a comp ehensive security solution for all your infrastructure needs.

With over

220,000 patches – and counting

delivered to our users, TuxCare's solutions reduce vulnerability exposure, minimize downtime, eliminate patching-related disruptions, secure your open-source supply chain, and maintain system stability and compliance.

TuxCare protects the world's largest enterprises, government agencies, service providers, universities, and research institutions, safeguarding over 1.2 million workloads (and growing). Our mission is to drive continuous innovation through open-source technologies while minimizing the risk of cyber threats and serving as a trusted technology partner for innovative organizations across the globe.

(Spring® is a trademark of Broadcom Inc. and/or its subsidiaries.)

