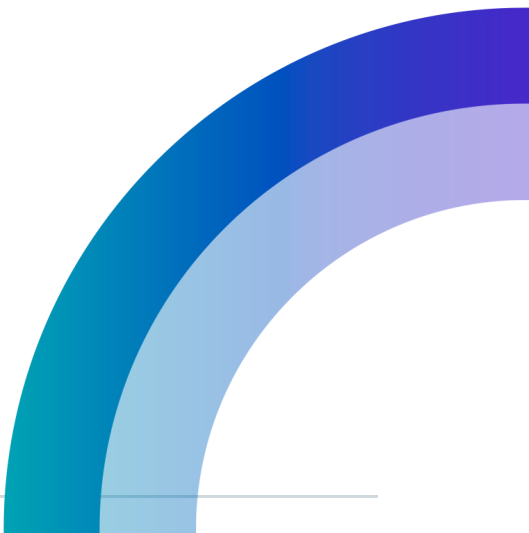
Análise das versões do TLS para o Java (VERSÃO PÚBLICA)



Sumário

1.TLS

TLS é o acrônimo de *Transport Layer Security* (Segurança em Camada de Transporte), um protocolo publicado em 1999 na RFC 2246. O objetivo principal do protocolo TLS é fornecer privacidade e integridade de dados entre dois aplicativos em comunicação. O protocolo é composto por duas camadas: o protocolo de registro TLS e o protocolo de handshake TLS. No nível mais baixo, sobreposto a algum protocolo de transporte confiável (por exemplo, TCP[TCP]), está o protocolo de registro TLS. A Figura 1 mostra como o TLS é implementado na camada de Sessão do modelo Open systems interconnection (OSI)[1].

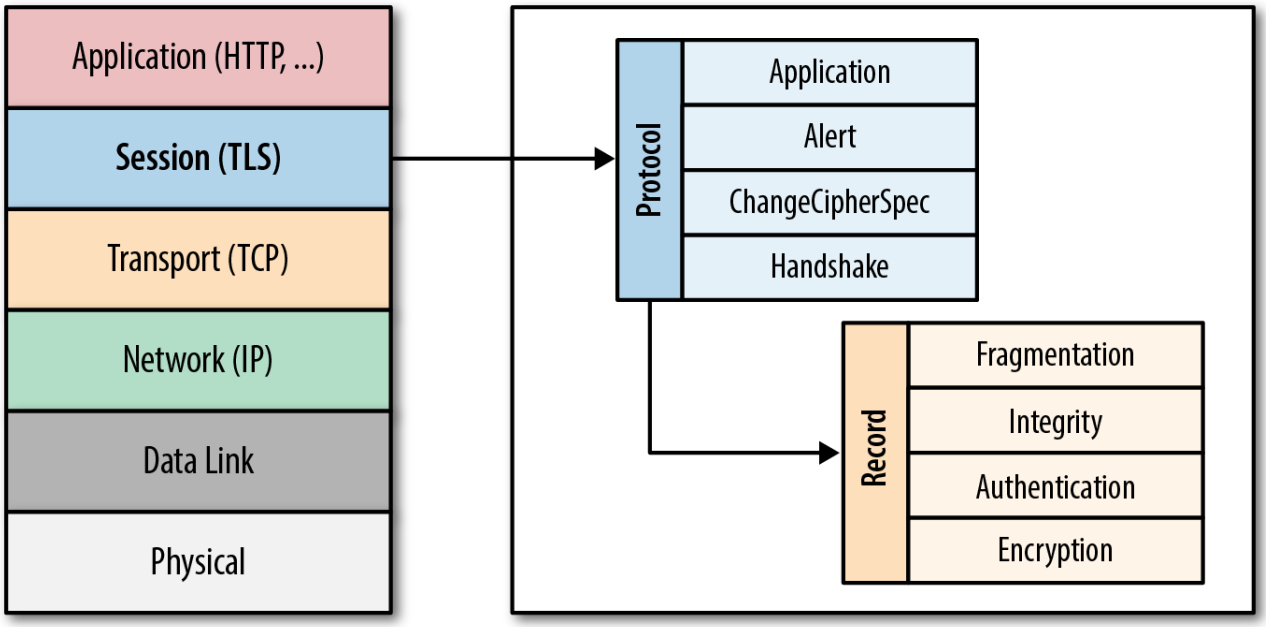


Figura 1: Arquitetura do protocolo Transport Layer Security (TLS)

1.1. Versões do TLS

O protocolo tem mais de uma versão, e assim, mais de uma implementação disponível em diversas linguagens de programação. O uso da versão mais recente é recomendável porque ela implementa o estado da arte da segurança em comunicação.

Tabela 1: Versões do TLS

Versão	RFC	Definida em
1.0	2246	1999
1.1	4346	2006
1.2	5246	2008
1.3	8446	2018

1 A especificação OSI está disponível em: https://www.iso.org/ics/35.100/x/

Há uma implementação de TLS em código livre e aberto, o OpenSSL. No lugar de implementar diretamente o TLS, as linguagens de programação podem usar o OpenSSL. A versão estável mais recente do OpenSSL é a série 3.2 com suporte até 23/11/2025. Também está disponível a série 3.1 com suporte até 14/03/2025, e a série 3.0 que é uma versão de suporte de longo prazo (LTS) com suporte até 07/09/2026. Todas as versões mais antigas versões (incluindo 1.1.1, 1.1.0, 1.0.2, 1.0.0 e 0.9.8) agora estão sem suporte e não devem ser usadas. Os usuários dessas versões mais antigas são incentivados a atualizar para 3.2 ou 3.0 o mais rápido possível. O sítio do OpenSSL contém a lista de vulnerabilidades da implementação e quais são as versões que as corrigem. A Tabela 2 mostra em quais versões de OpenSSL foi adicionado suporte às versões de TLS.

Tabela 2: Versões do OpenSSL e suporte a TLS

OpenSSL	Lançamento	Adicionou suporte a TLS
0.9.0	Não houve	1.0
0.9.1	Não houve	1.1
1.0.1	14/03/2012	1.2
3.0	07/09/2021	1.3

SSL é o acrônimo de *Secure Sockets Layer*, um protocolo publicado na versão 2.0 por uma equipe da Netscape liderada pelo criptógrafo egípcio Taher Elgamal, em 1995. O protocolo TLS foi lançado em 1999 como sucessor do SSL 3.0. O TLS é o sucessor direto do SSL, e todas as versões do SSL agora estão obsoletas. No entanto, é comum encontrar o termo *SSL* descrevendo uma conexão TLS (como no caso de funções e argumentos da linguagem Java). Na maioria dos casos, os termos *SSL* e *SSL/TLS* referem-se ao protocolo TLS e aos certificados TLS. Assim, o OpenSSL deveria se chamar OpenTLS, pois já nasceu após a obsolescência do SSL.

1.2. Versões do Java

A versão mais recente do Java é 25 (LTS), com suporte premier até setembro de 2030 e suporte estendido até setembro de 2023. É vital que a versão de Java em produção tenha pelo menos o suporte estendido. Recomenda-se, entretanto, a atualização para a versão LTS mais recente com suporte premier para evitar possíveis explorações de vulnerabilidades pela demora na migração de versões. Na Tabela 3 mostramos qual a versão mínima de TLS suportada por cada versão de Java.

Tabela 3: Versões de Java e versões do TLS

Java	Suporte	SSL/TLS default	Outras versões suportadas
6	Não suportada	TLS 1.0	TLS 1.1 (update 111 e superior), SSLv3.0[2]
7	Não suportada	TLS 1.0	TLS 1.2, TLS 1.1, SSLv3.0
8	Suporte estendido até dezembro de 2030 (não tem mais *updates*)	TLS 1.2	TLS 1.1, TLS 1.0, SSLv3.0
11	Suporte estendido até janeiro de 2032	TLS 1.3	TLS 1.2, TLS 1.1, TLS 1.0, SSLv3.0

2 O suporte para SSLv3 foi desabilitado pelos patch releases de janeiro de 2015.

Tabela 4: Roadmap de suporte do Java SE da Oracle[3]

Release	Lançamento	Suporte premier até	Suporte estendido até
8 (LTS)	Março de 2014	Março de 2022	Dezembro de 2030
9 - 10 (non-LTS)	Setembro de 2017 – Março de 2018	Março de 2018 – Setembro de 2018	Não disponível
11 (LTS)	Setembro 2018	Setembro de 2023	Janeiro de 2032
12 - 16 (non-LTS)	Março de 2019 – Março de 2021	Setembro de 2019 – Setembro de 2021	Não disponível
17 (LTS)	Setembro de 2021	Setembro de 2026	Setembro de 2029
18 - 20 (non-LTS)	Março 2022 – Março 2023	Setembro de 2022 – Setembro de 2023	Não disponível
21 (LTS)	Setembro de 2023	Setembro de 2028	Setembro de 2031
22 (non-LTS)	Março de 2024	Setembro de 2024	Não disponível
23 (non-LTS)	Setembro de 2024	Março de 2025	Não disponível
24 (non-LTS)	Março 2025	Setembro de 2025	Não disponível
25 (LTS)	Setembro de 2025	Setembro de 2030	Setembro de 2033

Há duas propriedades que uma aplicação cliente Java pode usar para especificar a versão de handshake SSL/TLS: jdk.tls.client.protocols e https.protocols (IBM, 2021).

Embora seja possível habilitar suporte para os protocolos SSL 3.0 e TLS 1.0 (ORACLE, 2024b), isso não é recomendável.

A documentação do Elastic Search (ELASTIC, 2024) contém uma orientação concisa mas precisa sobre a habilitação de versões adicionais de SSL/TLS na JDK.

O uso de TLS 1.3 com Java 17 está documentado em Oracle (2024d).

3 Oracle (2024c)

2.Análise

Conforme relatamos no capítulo anterior, todas as versões de Java superiores a 11 suportam TLS 1.3 (ORACLE, 2024a). A recomendação é que todas as aplicações Java em produção no Serpro utilizem preferencialmente a versão LTS mais recente com suporte premier ativo. No mínimo, deve ser usada uma versão com suporte estendido. O uso de versões de Java sem suporte estendido é um risco de segurança para a empresa e para seus clientes.

Considerando que o suporte premier da versão 11 de Java terminou em setembro de 2023, devem ser tomadas providências para migrar as aplicações pelo menos para a 17, que tem suporte premier até setembro de 2026.

A existência de versões de Java anteriores à 11 é uma situação temerosa, com gravidade inversamente proporcional ao número da versão. Se existem aplicações em produção nessa situação, elas devem ser migradas com máxima urgência. Caso não seja possível uma migração direta para a versão 17 por causa do acúmulo de incompatibilidades, a migração pode ser planejada para ser feita mais de uma etapa (por exemplo, migrando para a versão 11 e depois para a versão 17).

3.Conclusão

Para um uso funcional e seguro de TLS com Java, é recomendável o uso mínimo da versão 17, considerando que o suporte premier dessa versão dura até setembro de 2026.

A recomendação para versões de Java inferiores a 8 é apenas uma: migração imediata para a versão LTS com suporte estendido ativo mais recente possível.

Este documento foi finalizado em 7 de agosto de 2024.

Referências

IBM. **How do I change the default SSL/TLS protocol my Java™ application will use?** IBM Support. 6 ago. 2021. Disponível em: <https://www.ibm.com/support/pages/how-do-i-change-default-ssltls-protocol-my-java%E2%84%A2-application-will-use>. Acesso em: 7 ago. 2024.

ELASTIC. **Supported SSL/TLS versions by JDK version**. Manually configure security. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/jdk-tls-versions.html>. Acesso em: 7 ago. 2024.

JACOBS, Bill. **Transport Level Security (TLS) and Java**. A-Team Chronicles. 9 mai. 2016. Disponível em: <https://www.ateam-oracle.com/post/transport-level-security-tls-and-java>. Acesso em: 7 ago. 2024.

ORACLE. **JDK 11 Release Notes**. Java SE. Disponível em: <https://www.oracle.com/java/technologies/javase/11-relnote-issues.html>. Acesso em: 7 ago. 2024a.

ORACLE. **Steps to enable the SSL 3.0 and TLS 1.0 protocols**. Tools and Frameworks Migration Guide. Disponível em: <https://docs.oracle.com/cd/E66320_01/tools.11-2/EndecaCommerceMigrate/html/cifm_ssl_protocols.xmlsection_D8DEBBF578894C7CA3B607B40E9298F0.html>. Acesso em: 7 ago. 2024b.

ORACLE. **Oracle Java SE Support Roadmap**. 19 mar. 2024c. Disponível em: <https://www.oracle.com/java/technologies/java-se-support-roadmap.html>. Acesso em: 7 ago. 2024.

ORACLE. **Transport Layer Security (TLS) Protocol Overview**. Disponível em: <https://docs.oracle.com/en/java/javase/17/security/transport-layer-security-tls-protocol-overview.html#GUID-C6554A00-CF26-4661-991D-EA1B9EA6CCBE>. Acesso em: 7 ago. 2024d.