

Publicação Especial 800-218 do NIST por

---

# Estrutura de Desenvolvimento de Software Seguro (SSDF) Versão 1.1:

*Recomendações para mitigar  
o risco de vulnerabilidades de software*

---

Murugiah Souppaya  
Karen Scarfone  
Donna Dodson

Esta publicação está disponível gratuitamente no site:  
<https://doi.org/10.6028/NIST.SP.800-218.pdf>

**NIST**  
National Institute of  
Standards and Technology  
U.S. Department of Commerce

**Publicação Especial 800-218 do NIST por**

# **Estrutura de Desenvolvimento de Software Seguro (SSDF) Versão 1.1:**

*Recomendações para mitigar  
o risco de vulnerabilidades de software*

Murugiah Souppaya  
*Divisão de Segurança de Computadores  
Laboratório de Tecnologia da Informação*

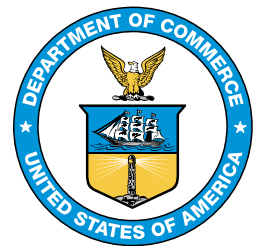
Karen Scarfone  
*Scarfone Cybersecurity  
Clifton, VA*

Donna Dodson\*

*\* Ex-funcionária do NIST; todo o trabalho para esta publicação foi feito enquanto estava no NIST.*

Esta publicação está disponível gratuitamente no site:  
<https://doi.org/10.6028/NIST.SP.800-218.por>

Fevereiro de 2022



Departamento de Comércio dos EUA  
*Gina M. Raimondo, Secretária*

Instituto Nacional de Padrões e Tecnologia (NIST)  
*James K. Olthoff, Desempenhando as funções e os deveres não exclusivos do Subsecretário de Comércio  
para o NIST e Diretor, Instituto Nacional de Padrões e Tecnologia (NIST)*

## Autoridade

Esta publicação foi desenvolvida pelo NIST de acordo com suas responsabilidades estatutárias sob a Lei Federal de Modernização da Segurança da Informação (FISMA) de 2014, 44 U.S.C. § 3551 *et seq.*, Direito Público (D.P.) 113-283. O NIST é responsável pelo desenvolvimento de padrões e diretrizes de segurança da informação, incluindo requisitos mínimos para sistemas de informação federais, mas esses padrões e diretrizes não se aplicam aos sistemas de segurança nacional sem a aprovação expressa dos funcionários federais apropriados que exercem autoridade política sobre esses sistemas. Esta diretriz está de acordo com as exigências da Circular A-130 do Escritório de Gestão e Orçamento (OMB).

Nada nesta publicação deve ser considerado como contraditório em relação aos padrões e diretrizes tornados obrigatórios e vinculantes para os órgãos federais pelo Secretário de Comércio sob autoridade estatutária. Essas diretrizes também não devem ser interpretadas como uma alteração ou substituição das autoridades existentes do Secretário de Comércio, do Diretor do OMB ou de qualquer outro funcionário federal. Esta publicação pode ser usada por organizações não governamentais em caráter voluntário e não está sujeita a direitos autorais nos Estados Unidos. A atribuição, no entanto, seria apreciada pelo NIST.

Publicação Especial 800-218 do Instituto Nacional de Padrões e Tecnologia por.

Public. Espec. Instit. Nac. Pad. Tecnol. 800-218 por, 36 páginas (Fevereiro de 2022)  
CODEN: NSPUE2

Esta publicação está disponível gratuitamente em:  
<https://doi.org/10.6028/NIST.SP.800-218.por>

Determinadas entidades comerciais, equipamentos ou materiais podem ser identificados neste documento a fim de descrever adequadamente um procedimento ou conceito experimental. Essa identificação não tem a intenção de implicar recomendação ou endosso pelo NIST, nem que as entidades, materiais ou equipamentos sejam necessariamente os melhores disponíveis para a finalidade.

Pode haver referências nesta publicação a outras publicações atualmente em desenvolvimento pelo NIST, de acordo com suas responsabilidades estatutárias atribuídas. As informações contidas nesta publicação, incluindo conceitos e metodologias, podem ser usadas por órgãos federais mesmo antes da conclusão dessas publicações complementares. Assim, até que cada publicação seja concluída, os requisitos, as diretrizes e os procedimentos atuais, quando existentes, permanecem em vigor. Para fins de planejamento e transição, os órgãos federais podem querer acompanhar de perto o desenvolvimento dessas novas publicações do NIST.

As organizações são incentivadas a revisar todos os rascunhos de publicações durante os períodos de comentários públicos e fornecer feedback ao NIST. Muitas publicações de segurança cibernética do NIST, além das mencionadas acima, estão disponíveis em <https://csrc.nist.gov/publications>.

**Envie comentários sobre esta publicação para: [ssdf@nist.gov](mailto:ssdf@nist.gov)**

Instituto Nacional de Padrões e Tecnologia  
Aos cuidados de: Computer Security Division, Information Technology Laboratory  
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Todos os comentários estão sujeitos à liberação de acordo com a Lei de Liberdade de Informação (FOIA).

Traduzido para o NIST pela TaikaTranslations LLC sob o contrato {133ND23PNB770271}. Tradução oficial do governo dos EUA. Todos os direitos reservados, Secretaria de Comércio dos EUA.

Translated for NIST by TaikaTranslations LLC under contract {133ND23PNB770271}. Official U.S. Government Translation. All rights reserved, US Secretary of Commerce.

## **Relatórios sobre tecnologia de sistemas de computador**

O ITL (Information Technology Laboratory, Laboratório de Tecnologia da Informação) do NIST (National Institute of Standards and Technology, Instituto Nacional de Padrões e Tecnologia) promove a economia e o bem-estar público dos EUA, fornecendo liderança técnica para a infraestrutura de medição e padrões do país. O ITL desenvolve testes, métodos de teste, dados de referência, implementações de prova de conceito e análises técnicas para promover o desenvolvimento e o uso produtivo da tecnologia da informação. As responsabilidades do ITL incluem o desenvolvimento de padrões e diretrizes gerenciais, administrativos, técnicos e físicos para a segurança econômica e a privacidade de informações não relacionadas à segurança nacional em sistemas de informações federais. A série Publicação Especial 800 relata os esforços de pesquisa, diretrizes e extensão do ITL em segurança de sistemas de informação e suas atividades de colaboração com organizações da indústria, governamentais e acadêmicas.

### **Resumo**

Poucos modelos de ciclo de vida de desenvolvimento de software (SDLC) abordam explicitamente a segurança de software em detalhes, portanto, as práticas de desenvolvimento de software seguro geralmente precisam ser adicionadas a cada modelo de SDLC para garantir que o software que está sendo desenvolvido seja bem protegido. Este documento recomenda a Estrutura de Desenvolvimento de Software Seguro (SSDF) - um conjunto básico de práticas de desenvolvimento de software seguro de alto nível que pode ser integrado a cada implementação do SDLC. Seguir essas práticas deve ajudar os produtores de software a reduzir o número de vulnerabilidades nos softwares lançados, reduzir o possível impacto da exploração de vulnerabilidades não detectadas ou não corrigidas e tratar as causas básicas das vulnerabilidades para evitar futuras recorrências. Como a estrutura fornece um vocabulário comum para o desenvolvimento seguro de software, os adquirentes de software também podem usá-la para promover a comunicação com os fornecedores nos processos de aquisição e em outras atividades de gerenciamento.

### **Palavras-chave**

desenvolvimento seguro de software; estrutura de desenvolvimento seguro de software (SSDF); práticas de desenvolvimento seguro de software; aquisição de software; desenvolvimento de software; ciclo de vida de desenvolvimento de software (SDLC); segurança de software.

### **Informações sobre marcas registradas**

Todas as marcas registradas ou marcas comerciais pertencem às suas respectivas organizações.

## Agradecimentos

Os autores agradecem a todas as organizações e indivíduos que forneceram informações para essa atualização da SSDF. Em resposta à Seção 4 da Ordem Executiva (EO) 14028 sobre "[Melhorar a segurança cibernética da nação](#)", o NIST realizou um [workshop em junho de 2021](#) e recebeu [mais de 150 documentos de posicionamento](#), muitos dos quais sugeriam práticas de desenvolvimento de software seguro, tarefas, exemplos de implementações e referências a serem consideradas nesta atualização da SSDF. Os autores agradecem todas essas sugestões, bem como a contribuição daqueles que falaram ou participaram do workshop e compartilharam suas ideias durante ou após o workshop.

Além disso, os autores agradecem os comentários públicos enviados por dezenas de organizações e indivíduos e gostariam de agradecer o feedback particularmente útil da Amazon Web Services, Apiiro, Blackberry, BSA | The Software Alliance, Enterprise Cloud Coalition, General Services Administration (GSA), Google, IBM, Medical Imaging & Technology Alliance (MITA), Microsoft, Oracle, Software Assurance Forum for Excellence in Code (SAFECode), Synopsys, a Marinha dos Estados Unidos, Xoomworks e Robert Grupe. Representantes da Siemens Energy e da Synopsys contribuíram com mapeamentos para novas referências.

Os autores agradecem a todos os colegas do NIST pelo apoio durante toda a atualização da SSDF, especialmente Curt Barker, Paul Black, Jon Boyens, Jim Foti, Barbara Guttman, Mat Heyman, Nicole Keller, Matt Scholl, Adam Sedgewick, Kevin Stine e Isabel Van Wyk.

Os autores também gostariam de agradecer a todas as pessoas e organizações que fizeram comentários sobre os rascunhos da versão original da SSDF, incluindo o Administrative Offices of the U.S. Courts, The Aerospace Corporation, BSA | The Software Alliance, Capitis Solutions, Consortium Information & Software Quality (CISQ), HackerOne, Honeycomb Secure Systems, iNovex, Ishpi Information Technologies, o Information Security and Privacy Advisory Board (ISPAB), Juniper Networks, Microsoft, MITA, Naval Sea Systems Command (NAVSEA), NIST, Northrop Grumman, o Escritório do Subsecretário de Defesa para Pesquisa e Engenharia, Red Hat, SAFECode, e o Instituto de Engenharia de Software (SEI).

## Público

Há dois públicos principais para este documento. O primeiro são os produtores de software (por exemplo, fornecedores de produtos comerciais prontos para uso [COTS], desenvolvedores de software governamentais prontos para uso [GOTS], desenvolvedores de software personalizado, equipes internas de desenvolvimento), independentemente do tamanho, do setor ou do nível de maturidade. O segundo são os adquirentes de software, tanto órgãos federais quanto outras organizações. Não se espera que os leitores deste documento sejam especialistas em desenvolvimento seguro de software para entendê-lo, mas esse conhecimento é necessário para implementar suas práticas recomendadas.

Os funcionários das seguintes categorias de força de trabalho e áreas de especialidade da National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework [SP800181] provavelmente acharão esta publicação de interesse:

- Provisionamento seguro (SP): Gerenciamento de riscos (RSK), desenvolvimento de software (DEV), planejamento de requisitos de sistemas (SRP), teste e avaliação (TST), desenvolvimento de sistemas (SYS)
- Operar e manter (OM): Análise de sistemas (ANA)
- Supervisionar e governar (OV): Treinamento, educação e conscientização (TEA); gerenciamento de segurança cibernética (MGT); liderança cibernética executiva (EXL); gerenciamento de programas/projetos (PMA) e aquisição
- Proteger e defender (PR): Resposta a incidentes (CIR), avaliação e gerenciamento de vulnerabilidades (VAM)
- Analisar (AN): Análise de ameaças (TWA), análise de exploração (EXP)

### **Nota para os leitores**

Recomendamos que você forneça feedback sobre a SSDF a qualquer momento, especialmente quando implementar a SSDF em sua própria organização e em seus esforços de desenvolvimento de software. As contribuições de diversos produtores de software serão particularmente úteis para o refinamento e a revisão do SSDF. A publicação será atualizada periodicamente para refletir suas contribuições e feedback.

Se você pertence a uma organização de desenvolvimento de padrões ou a outra organização que produziu um conjunto de práticas seguras e gostaria de mapear seu padrão ou orientação de desenvolvimento seguro de software para a SSDF, entre em contato conosco pelo [e-mail ssdf@nist.gov](mailto:ssdf@nist.gov). Gostaríamos de apresentá-lo ao [National Online Informative References Program \(OLIR\)](#) para que você possa enviar seu mapeamento para aumentar o conjunto existente de referências informativas.

### **Aviso de divulgação de patente**

*AVISO: O Laboratório de Tecnologia da Informação (ITL) solicitou que os detentores de reivindicações de patentes cujo uso possa ser necessário para a conformidade com as orientações ou requisitos desta publicação divulguem essas reivindicações de patentes ao ITL. No entanto, os detentores de patentes não são obrigados a responder às solicitações de patentes do ITL e o ITL não realizou uma pesquisa de patentes para identificar quais patentes, se houver, podem se aplicar a esta publicação.*

*Até a data da publicação e após a(s) solicitação(ões) para a identificação de reivindicações de patentes cujo uso pode ser necessário para a conformidade com as orientações ou requisitos desta publicação, nenhuma reivindicação de patente foi identificada ao ITL.*

*O ITL não faz nenhuma declaração nem deixa implícito que as licenças não são necessárias para evitar a violação de patentes no uso desta publicação.*

## Resumo Executivo

Este documento descreve um conjunto de práticas fundamentais e sólidas para o desenvolvimento seguro de software, denominado Estrutura de Desenvolvimento Seguro de Software (Secure Software Development Framework, SSDF). As organizações devem integrar a SSDF em suas práticas de desenvolvimento de software existentes, expressar seus requisitos de desenvolvimento de software seguro para fornecedores terceirizados usando as convenções da SSDF e adquirir software que atenda às práticas descritas na SSDF. O uso da SSDF ajuda as organizações a atenderem às seguintes recomendações de desenvolvimento seguro de software:

- As organizações devem garantir que seu pessoal, seus processos e sua tecnologia estejam preparados para realizar o desenvolvimento seguro de software.
- As organizações devem proteger todos os componentes de seu software contra adulteração e acesso não autorizado.
- As organizações devem produzir software bem protegido com o mínimo de vulnerabilidades de segurança em suas versões.
- As organizações devem identificar as vulnerabilidades residuais em suas versões de software e reagir adequadamente para resolver essas vulnerabilidades e evitar que outras semelhantes ocorram no futuro.

A SSDF não prescreve como implementar cada prática. O foco está nos resultados das práticas, e não nas ferramentas, técnicas e mecanismos para realizá-las. Isso significa que a SSDF pode ser usada por organizações de qualquer setor ou comunidade, independentemente do tamanho ou da sofisticação da segurança cibernética. Ela também pode ser usada para qualquer tipo de desenvolvimento de software, independentemente da tecnologia, da plataforma, da linguagem de programação ou do ambiente operacional.

A SSDF define apenas um subconjunto de alto nível do que as organizações podem precisar realizar. Portanto, as organizações devem consultar as referências e outros recursos para obter informações adicionais sobre a implementação das práticas. Nem todas as práticas são aplicáveis a todos os casos de uso; as organizações devem adotar uma abordagem baseada em riscos para determinar quais práticas são relevantes, apropriadas e eficazes para mitigar as ameaças às suas práticas de desenvolvimento de software.

As organizações podem comunicar como estão abordando as cláusulas da Seção 4 da Ordem Executiva do Presidente (EO) sobre "[Melhoria da Segurança Cibernética da Nação \(14028\)](#)" fazendo referência às práticas e tarefas da SSDF descritas no Apêndice A.

## Tabela de Conteúdos

<b>Resumo Executivo .....</b>	<b>vi</b>
<b>1 Introdução .....</b>	<b>1</b>
<b>2 A Estrutura de Desenvolvimento de Software Seguro .....</b>	<b>5</b>
<b>Referências .....</b>	<b>22</b>
<b>Apêndice A—A SSDF e a Ordem Executiva 14028.....</b>	<b>26</b>
<b>Apêndice B—Acrônimos .....</b>	<b>27</b>
<b>Apêndice C—Registro de Alterações .....</b>	<b>29</b>

## Lista de tabelas

Tabela 1: A Estrutura de Desenvolvimento de Software Seguro (SSDF) Versão 1.1 .....	7
Tabela 2: Práticas da SSDF Correspondentes às Cláusulas da EO 14028 .....	26

## 1 Introdução

Um *ciclo de vida de desenvolvimento de software (SDLC)*<sup>1</sup> é uma metodologia formal ou informal para projetar, criar e manter software (inclusive código incorporado ao hardware). Há muitos modelos de SDLCs, incluindo cascata, espiral, ágil e, em especial, ágil combinado com práticas de desenvolvimento de software e operações de TI (DevOps). Poucos modelos de SDLC abordam explicitamente a segurança de software em detalhes, de modo que as práticas de desenvolvimento de software seguro geralmente precisam ser adicionadas e integradas a cada modelo de SDLC. Independentemente do modelo de SDLC utilizado, as práticas de desenvolvimento seguro de software devem ser integradas a ele por três motivos: reduzir o número de vulnerabilidades no software lançado, reduzir o impacto potencial da exploração de vulnerabilidades não detectadas ou não abordadas e abordar as causas básicas das vulnerabilidades para evitar recorrências. As vulnerabilidades incluem não apenas bugs causados por falhas de codificação, mas também pontos fracos causados por definições de configuração de segurança, suposições de confiança incorretas e análises de risco desatualizadas. [\[IR7864\]](#)

A maioria dos aspectos de segurança pode ser abordada várias vezes em um SDLC, mas, em geral, quanto mais cedo a segurança for abordada no SDLC, menor será o esforço e o custo necessários para atingir o mesmo nível de segurança. Esse princípio, conhecido como *shifting left*, é extremamente importante, independentemente do modelo de SDLC. O *shifting left* minimiza qualquer dívida técnica que exigiria a correção de falhas de segurança iniciais no final do desenvolvimento ou depois que o software estiver em produção. O *shifting left* também pode resultar em um software com maior segurança e resiliência.

Há muitos documentos existentes sobre práticas de desenvolvimento seguro de software, incluindo os listados na seção [Referências](#). Este documento não introduz novas práticas nem define nova terminologia. Em vez disso, ele descreve um conjunto de práticas de alto nível com base em padrões estabelecidos, orientações e documentos de práticas de desenvolvimento seguro de software. Essas práticas, coletivamente chamadas de Estrutura de Desenvolvimento de Software Seguro (SSDF), têm como objetivo ajudar os públicos-alvo a atingirem os objetivos de desenvolvimento de software seguro. Muitas das práticas envolvem diretamente o próprio software, enquanto outras o envolvem indiretamente (por exemplo, proteger o ambiente de desenvolvimento).

Trabalhos futuros podem expandir esta publicação e, possivelmente, abranger tópicos como a forma como a SSDF pode se aplicar e variar para determinadas metodologias de desenvolvimento de software e práticas associadas, como DevOps, como uma organização pode fazer a transição de suas práticas atuais de desenvolvimento de software para também incorporar as práticas da SSDF e como a SSDF poderia ser aplicada no contexto do software de código aberto. O trabalho futuro provavelmente assumirá a forma de casos de uso para que os insights sejam mais prontamente aplicáveis a tipos específicos de ambientes de desenvolvimento, e provavelmente incluirá a colaboração com a comunidade de código aberto e outros grupos e

---

<sup>1</sup> Observe que o SDLC também é amplamente usado para "ciclo de vida de desenvolvimento de sistemas". Todo uso de "SDLC" neste documento se refere a software, não a sistemas.

organizações.

Este documento identifica as práticas de desenvolvimento seguro de software, mas não prescreve como implementá-las. O foco está nos resultados das práticas a serem implementadas, e não nas ferramentas, técnicas e mecanismos usados para isso. As vantagens de especificar as práticas em um nível elevado incluem as seguintes:

- Pode ser usado por organizações de qualquer setor ou comunidade, independentemente do tamanho ou da sofisticação da segurança cibernética
- Pode ser aplicado ao software desenvolvido para dar suporte à tecnologia da informação (TI), aos sistemas de controle industrial (ICS), aos sistemas ciberfísicos (CPS) ou à Internet das Coisas (IoT)
- Pode ser integrado a qualquer fluxo de trabalho de desenvolvimento de software existente e cadeia de ferramentas automatizadas; não deve afetar negativamente as organizações que já têm práticas robustas de desenvolvimento de software seguro em vigor
- Torna as práticas amplamente aplicáveis, não específicas a determinadas tecnologias, plataformas, linguagens de programação, modelos de SDLC, ambientes de desenvolvimento, ambientes operacionais, ferramentas etc.
- Pode ajudar uma organização a documentar suas práticas atuais de desenvolvimento seguro de software e a definir suas práticas futuras como parte de seu processo de melhoria contínua
- Pode ajudar uma organização que atualmente usa um modelo clássico de desenvolvimento de software a fazer a transição de suas práticas de desenvolvimento de software seguro para uso com um modelo moderno de desenvolvimento de software (por exemplo, ágil, DevOps)
- Pode ajudar as organizações que estão adquirindo e usando software a entender as práticas de desenvolvimento de software seguro empregadas por seus fornecedores

Este documento fornece uma linguagem comum para descrever as práticas fundamentais de desenvolvimento seguro de software. Essa abordagem é semelhante à adotada pela *Estrutura para Melhoria da Segurança Cibernética da Infraestrutura Crítica*, também conhecida como Estrutura de Segurança Cibernética do NIST [NISTCSF].<sup>2</sup> Não é necessário ter experiência em desenvolvimento seguro de software para entender as práticas. A linguagem comum ajuda a facilitar a comunicação sobre práticas de software seguro entre os participantes internos e externos da organização, como, por exemplo:

- Proprietários de empresas, desenvolvedores de software, gerentes e líderes de projetos, profissionais de segurança cibernética e engenheiros de operações e de plataforma de

---

<sup>2</sup> As práticas da SSDF podem ajudar a dar suporte às funções, categorias e subcategorias da estrutura de segurança cibernética do NIST, mas as práticas da SSDF não são mapeadas para elas e, normalmente, são de responsabilidade de diferentes partes. Os desenvolvedores podem adotar práticas da SSDF, e os resultados de seu trabalho podem ajudar as organizações com sua segurança operacional em apoio à estrutura de segurança cibernética.

uma organização que precisam se comunicar claramente entre si sobre o desenvolvimento seguro de software

- Adquirentes de software, incluindo órgãos federais e outras organizações, que desejam definir as características necessárias ou desejadas para o software em seus processos de aquisição, a fim de obter software de maior qualidade (especialmente com menos vulnerabilidades de segurança significativas)<sup>3</sup>
- Produtores de software (por exemplo, fornecedores de produtos comerciais prontos para uso [COTS], desenvolvedores de software governamentais prontos para uso [GOTS], desenvolvedores de software que trabalham dentro ou em nome de organizações adquirentes de software) que desejam integrar práticas de desenvolvimento de software seguro em seus SDLCs, expressar suas práticas de software seguro para seus clientes ou definir requisitos para seus fornecedores

As práticas deste documento não se baseiam no pressuposto de que todas as organizações têm os mesmos objetivos e prioridades de segurança. Em vez disso, as recomendações refletem o fato de que cada produtor de software pode ter premissas de segurança exclusivas, e cada adquirente de software pode ter necessidades e requisitos de segurança exclusivos. Embora o objetivo seja que cada produtor de software siga todas as práticas aplicáveis, a expectativa é que o grau de implementação de cada prática e a formalidade da implementação variem de acordo com as premissas de segurança do produtor. As práticas oferecem flexibilidade para os implementadores, mas também são claras para evitar deixar muita margem para interpretação.

Embora a maioria dessas práticas seja relevante para qualquer esforço de desenvolvimento de software, algumas não são. Por exemplo, se o desenvolvimento de um determinado software não envolver o uso de um compilador, não haverá necessidade de seguir uma prática de configuração do compilador para melhorar a segurança do executável. Algumas práticas são fundamentais, enquanto outras são mais avançadas e dependem de certas práticas fundamentais já implementadas. Além disso, as práticas não são igualmente importantes para todos os casos.

Fatores como risco, custo, viabilidade e aplicabilidade devem ser considerados ao decidir quais práticas adotar e quanto tempo e recursos dedicar a cada prática.<sup>4</sup> A automatização também é um fator importante a ser considerado, especialmente para a implementação de práticas em escala. As práticas, tarefas e exemplos de implementação representam um ponto de partida a ser considerado; eles devem ser alterados e personalizados, e não são priorizados. Qualquer frequência declarada para a realização de práticas é apenas ilustrativa. A intenção do SSDF não é criar uma lista de verificação a ser seguida, mas fornecer uma base para o planejamento e a implementação de uma abordagem baseada em riscos para a adoção de práticas seguras de

---

<sup>3</sup> Trabalhos futuros podem fornecer orientações mais práticas para os adquirentes de software sobre como eles podem aproveitar a SSDF em casos de uso específicos.

<sup>4</sup> As organizações que buscam orientação sobre como iniciar o desenvolvimento seguro de software podem consultar muitas referências disponíveis publicamente, como "SDL That Won't Break the Bank", de Steve Lipner, da SAFECODE(<https://i.blackhat.com/us-18/Thu-August-9/us-18-Lipner-SDL-For-The-Rest-Of-Us.pdf>), "Application Software Security and the CIS Controls": A Reference Paper", de Steve Lipner e Stacy Simpson, da SAFECODE(<https://safecode.org/resource-publications/cis-controls/>), e "Simplified Implementation of the Microsoft SDL", da Microsoft(<https://www.microsoft.com/en-us/download/details.aspx?id=12379>).

desenvolvimento de software.

A responsabilidade pela implementação das práticas pode ser distribuída entre diferentes organizações com base no fornecimento do software e dos serviços (por exemplo, infraestrutura como serviço, software como serviço, plataforma como serviço, contêiner como serviço, sem servidor). Nessas situações, provavelmente segue um modelo de responsabilidade compartilhada que envolve os provedores de plataforma/serviço e a organização do locatário que está consumindo essas plataformas/serviços. A organização locatária deve estabelecer um acordo com os provedores que especifique qual parte é responsável por cada prática e tarefa e como cada provedor atestará sua conformidade com o acordo.

## 2 A estrutura de desenvolvimento de software seguro

Este documento define a versão 1.1 da Estrutura de Desenvolvimento de Software Seguro (SSDF) com práticas recomendadas fundamentais, sólidas e seguras baseadas em documentos de práticas de desenvolvimento de software seguro estabelecidos. As práticas estão organizadas em quatro grupos:

1. **Preparar a organização (PO):** As organizações devem garantir que seu pessoal, seus processos e sua tecnologia estejam preparados para realizar o desenvolvimento seguro de software em nível organizacional. Muitas organizações descobrirão que algumas práticas de PO também são aplicáveis a subconjuntos de seu desenvolvimento de software, como grupos ou projetos de desenvolvimento individuais.
2. **Proteger o software (PS):** As organizações devem proteger todos os componentes de seu software contra adulteração e acesso não autorizado.
3. **Produzir software bem protegido (PW):** As organizações devem produzir software bem protegido com o mínimo de vulnerabilidades de segurança em suas versões.
4. **Responder a vulnerabilidades (RV):** As organizações devem identificar as vulnerabilidades residuais em suas versões de software e reagir adequadamente para resolver essas vulnerabilidades e evitar que outras semelhantes ocorram no futuro.

Cada definição de prática inclui os seguintes elementos:

- **Prática:** O nome da prática e um identificador exclusivo, seguidos de uma breve explicação sobre o que é a prática e por que ela é benéfica
- **Tarefas:** Uma ou mais ações que podem ser necessárias para realizar uma prática
- **Exemplos de implementação ilustrativos:** Um ou mais exemplos ilustrativos de tipos de ferramentas, processos ou outros métodos que poderiam ser usados para ajudar a implementar uma tarefa. Nenhum exemplo ou combinação de exemplos é necessário, e os exemplos indicados não são as únicas opções viáveis. Alguns exemplos podem não ser aplicáveis a determinadas organizações e situações.
- **Referências:** Apontamentos para um ou mais documentos de práticas de desenvolvimento seguro estabelecidos e seus mapeamentos para uma tarefa específica. Nem todas as referências se aplicam a todas as instâncias de desenvolvimento de software.

A Tabela 1 define as práticas. Elas são apenas um **subconjunto** do que uma organização pode precisar fazer. As informações da tabela são limitadas em termos de espaço; muito mais informações sobre cada prática podem ser encontradas nas referências. Observe que a ordem das práticas, tarefas e exemplos de implementação ilustrativos na tabela não tem a intenção de implicar a sequência de implementação ou a importância relativa de qualquer prática, tarefa ou exemplo.

A tabela usa termos como "dados sensíveis", "pessoa qualificada" e "bem protegido", que não são definidos nesta publicação. As organizações que adotam a SSDF devem definir esses termos

no contexto de seus próprios ambientes e casos de uso. O mesmo se aplica aos nomes dos ambientes, como "desenvolvimento", "construção", "preparação", "integração", "teste", "produção" e "distribuição", que variam muito entre organizações e projetos. A enumeração de seus ambientes é necessária para protegê-los adequadamente e, principalmente, para evitar a movimentação lateral de invasores de um ambiente para outro.

Tabela 1: A Estrutura de Desenvolvimento de Software Seguro (SSDF) Versão 1.1

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
<b>Preparar a organização (PO)</b>			
<p><b>Definir requisitos de segurança para o desenvolvimento de software (PO.1):</b> Garantir que os requisitos de segurança para o desenvolvimento de software sejam conhecidos o tempo todo, para que possam ser levados em conta em todo o SDLC e para que a duplicação de esforços seja minimizada, pois as informações sobre os requisitos podem ser coletadas uma vez e compartilhadas. Isso inclui requisitos de fontes internas (por exemplo, políticas da organização, objetivos comerciais e estratégia de gerenciamento de riscos) e fontes externas (por exemplo, leis e regulamentações aplicáveis).</p>	<p><b>PO.1.1:</b> Identificar e documentar todos os requisitos de segurança para as infraestruturas e os processos de desenvolvimento de software da organização e manter os requisitos ao longo do tempo.</p>	<p><b>Exemplo 1:</b> Definir políticas para proteger as infraestruturas de desenvolvimento de software e seus componentes, incluindo endpoints de desenvolvimento, durante todo o SDLC e manter essa segurança.</p> <p><b>Exemplo 2:</b> Definir políticas para proteger os processos de desenvolvimento de software em todo o SDLC e manter essa segurança, inclusive para componentes de software de código aberto e outros componentes de software de terceiros utilizados pelo software que está sendo desenvolvido.</p> <p><b>Exemplo 3:</b> Revisar e atualizar os requisitos de segurança pelo menos uma vez por ano, ou antes, se houver novos requisitos de fontes internas ou externas, ou se tiver ocorrido um incidente significativo de segurança direcionado à infraestrutura de desenvolvimento de software.</p> <p><b>Exemplo 4:</b> Instruir as pessoas afetadas sobre as mudanças previstas nos requisitos.</p>	<p><b>BSAFSS:</b> SM.3, DE.1, IA.1, IA.2  <b>BSIMM:</b> CP1.1, CP1.3, SR1.1, SR2.2, SE1.2, SE2.6  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> SM-7, SM-9  <b>NISTCSF:</b> ID.GV-3  <b>OWASPASVS:</b> 1.1.1  <b>OWASPMASVS:</b> 1.10  <b>OWASPSAMM:</b> PC1-A, PC1-B, PC2-A  <b>PCISSLC:</b> 2.1, 2.2  <b>SCFPSSD:</b> Planejamento da implementação e implantação de práticas de desenvolvimento seguro  <b>SP80053:</b> SA-1, SA-8, SA-15, SR-3  <b>SP800160:</b> 3.1.2, 3.2.1, 3.2.2, 3.3.1, 3.4.2, 3.4.3  <b>SP800161:</b> SA-1, SA-8, SA-15, SR-3  <b>SP800181:</b> T0414; K0003, K0039, K0044, K0157, K0168, K0177, K0211, K0260, K0261, K0262, K0524; S0010, S0357, S0368; A0033, A0123, A0151</p>
	<p><b>PO.1.2:</b> Identificar e documentar todos os requisitos de segurança do software desenvolvido pela organização para atender e manter os requisitos ao longo do tempo.</p>	<p><b>Exemplo 1:</b> Definir políticas que especifiquem a arquitetura de software baseada em riscos e os requisitos de projeto, como tornar o código modular para facilitar a reutilização e as atualizações do código; isolar os componentes de segurança de outros componentes durante a execução; evitar comandos e configurações não documentados; e fornecer recursos que ajudarão os adquirentes de software na implantação, operação e manutenção seguras do software.</p> <p><b>Exemplo 2:</b> Definir políticas que especifiquem os requisitos de segurança para o software da organização e verificar a conformidade nos principais pontos do SDLC (por exemplo, classes de falhas de software verificadas por portas, respostas a vulnerabilidades descobertas no software lançado).</p> <p><b>Exemplo 3:</b> Analisar o risco das stacks de tecnologia aplicáveis (por exemplo, linguagens, ambientes, modelos de implantação) e recomendar ou exigir o uso de stacks que reduzam o risco em comparação com outras.</p> <p><b>Exemplo 4:</b> Definir políticas que especifiquem o que precisa ser arquivado para cada versão de software (por exemplo, código, arquivos de pacotes, bibliotecas de terceiros, documentação, inventário de dados) e por quanto tempo precisa ser retido com base no modelo SDLC, no fim da vida útil do software e em outros fatores.</p> <p><b>Exemplo 5:</b> Certificar-se de que as políticas abrangem todo o ciclo de vida do software, incluindo a notificação aos usuários sobre o fim iminente do suporte ao software e a data do fim da vida útil do software.</p> <p><b>Exemplo 6:</b> Revisar todos os requisitos de segurança pelo menos uma vez por ano, ou antes, se houver novos requisitos de fontes internas ou externas, se uma vulnerabilidade significativa for descoberta em um software lançado ou se ocorrer um incidente significativo de segurança direcionado a um software desenvolvido pela organização.</p> <p><b>Exemplo 7:</b> Estabelecer e seguir processos para lidar com solicitações de exceção de requisitos, incluindo revisões periódicas de todas as exceções aprovadas.</p>	<p><b>BSAFSS:</b> SC.1-1, SC.2, PD.1-1, PD.1-2, PD.1-3, PD.2-2, SI, PA, CS, AA, LO, EE  <b>BSIMM:</b> SM1.1, SM1.4, SM2.2, CP1.1, CP1.2, CP1.3, CP2.1, CP2.3, AM1.2, SFD1.1, SFD2.1, SFD3.2, SR1.1, SR1.3, SR2.2, SR3.3, SR3.4  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> SR-3, SR-4, SR-5, SD-4  <b>ISO27034:</b> 7.3.2  <b>MSSDL:</b> 2, 5  <b>NISTCSF:</b> ID.GV-3  <b>OWASPMASVS:</b> 1.12  <b>OWASPSAMM:</b> PC1-A, PC1-B, PC2-A, PC3-A, SR1-A, SR1-B, SR2-B, SA1-B, IR1-A  <b>PCISSLC:</b> 2.1, 2.2, 2.3, 3.3  <b>SCFPSSD:</b> Estabelecer padrões e convenções de codificação  <b>SP80053:</b> SA-8, SA-8(3), SA-15, SR-3  <b>SP800160:</b> 3.1.2, 3.2.1, 3.3.1  <b>SP800161:</b> SA-8, SA-15, SR-3  <b>SP800181:</b> T0414; K0003, K0039, K0044, K0157, K0168, K0177, K0211, K0260, K0261, K0262, K0524; S0010, S0357, S0368; A0033, A0123, A0151</p>
	<p><b>PO.1.3:</b> Comunicar os requisitos a todos os terceiros que fornecerão componentes de software comercial à organização para reutilização pelo próprio software da organização. [Anteriormente PW.3.1]</p>	<p><b>Exemplo 1:</b> Definir um conjunto básico de requisitos de segurança para componentes de software e incluí-lo em documentos de aquisição, contratos de software e outros acordos com terceiros.</p> <p><b>Exemplo 2:</b> Definir critérios relacionados à segurança para selecionar o software; os critérios podem incluir o programa de divulgação de vulnerabilidades</p>	<p><b>BSAFSS:</b> SM.1, SM.2, SM.2-1, SM.2-4  <b>BSIMM:</b> CP2.4, CP3.2, SR2.5, SR3.2  <b>EO14028:</b> 4e(vi), 4e(ix)  <b>IDASOAR:</b> 19, 21  <b>IEC62443:</b> SM-9, SM-10</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
		<p>do terceiro e os recursos de resposta a incidentes de segurança do produto ou a adesão do terceiro às práticas definidas pela organização.</p> <p><b>Exemplo 3:</b> Exigir que terceiros atestem que seu software está em conformidade com os requisitos de segurança da organização.</p> <p><b>Exemplo 4:</b> Exigir que terceiros forneçam dados de proveniência<sup>5</sup> e mecanismos de verificação de integridade para todos os componentes de seu software.</p> <p><b>Exemplo 5:</b> Estabelecer e seguir processos para lidar com o risco quando houver requisitos de segurança que os componentes de software de terceiros a serem adquiridos não atendam; isso deve incluir revisões periódicas de todas as exceções aprovadas aos requisitos.</p>	<p><b>MSSDL:</b> 7  <b>NISTCSF:</b> ID.SC-3  <b>OWASPSAMM:</b> SR3-A  <b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 8  <b>SCFPSSD:</b> Gerenciar o risco de segurança inerente ao uso de componentes de terceiros  <b>SCSIC:</b> Controles de integridade do suprimento do fornecedor  <b>SP80053:</b> SA-4, SA-9, SA-10, SA-10(1), SA-15, SR-3, SR-4, SR-5  <b>SP800160:</b> 3.1.1, 3.1.2  <b>SP800161:</b> SA-4, SA-9, SA-9(1), SA-9(3), SA-10, SA-10(1), SA-15, SR-3, SR-4, SR-5  <b>SP800181:</b> T0203, T0415; K0039; S0374; A0056, A0161</p>
<p><b>Implementar funções e responsabilidades (PO.2):</b> Garantir que todas as pessoas dentro e fora da organização envolvidas no SDLC estejam preparadas para desempenhar suas funções e responsabilidades relacionadas ao SDLC durante todo o SDLC.</p>	<p><b>PO.2.1:</b> Criar novas funções e alterar as responsabilidades das funções existentes, conforme necessário, para abranger todas as partes do SDLC. Revisar e manter periodicamente as funções e responsabilidades definidas, atualizando-as conforme necessário.</p>	<p><b>Exemplo 1:</b> Definir funções e responsabilidades relacionadas ao SDLC para todos os membros da equipe de desenvolvimento de software.</p> <p><b>Exemplo 2:</b> Integrar as funções de segurança à equipe de desenvolvimento de software.</p> <p><b>Exemplo 3:</b> Definir funções e responsabilidades para a equipe de segurança cibernética, defensores da segurança, gerentes e líderes de projeto, gerência sênior, desenvolvedores de software, testadores de software, líderes e equipe de garantia de software, proprietários de produtos, engenheiros de operações e de plataforma e outros envolvidos no SDLC.</p> <p><b>Exemplo 4:</b> Realizar uma revisão anual de todas as funções e responsabilidades.</p> <p><b>Exemplo 5:</b> Instruir os indivíduos afetados sobre as mudanças iminentes nas funções e responsabilidades e confirmar que eles entendem as mudanças e concordam em segui-las.</p> <p><b>Exemplo 6:</b> Implementar e usar ferramentas e processos para promover a comunicação e o envolvimento entre indivíduos com funções e responsabilidades relacionadas ao SDLC, como a criação de canais de mensagens para discussões em equipe.</p> <p><b>Exemplo 7:</b> Designar um grupo de indivíduos ou uma equipe como proprietário do código para cada projeto.</p>	<p><b>BSAFSS:</b> PD.2-1, PD.2-2  <b>BSIMM:</b> SM1.1, SM2.3, SM2.7, CR1.7  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> SM-2, SM-13  <b>NISTCSF:</b> ID.AM-6, ID.GV-2  <b>PCISL:</b> 1.2  <b>SCSIC:</b> Controles de integridade do desenvolvimento de software do fornecedor  <b>SP80053:</b> SA-3  <b>SP800160:</b> 3.2.1, 3.2.4, 3.3.1  <b>SP800161:</b> SA-3  <b>SP800181:</b> K0233</p>
	<p><b>PO.2.2:</b> Fornecer treinamento baseado em funções para todo o pessoal com responsabilidades que contribuam para o desenvolvimento seguro. Analisar periodicamente a proficiência do pessoal e o treinamento baseado em funções e atualizar o treinamento conforme necessário.</p>	<p><b>Exemplo 1:</b> Documentar os resultados desejados do treinamento para cada função.</p> <p><b>Exemplo 2:</b> Definir o tipo de treinamento ou currículo necessário para alcançar o resultado desejado para cada função.</p> <p><b>Exemplo 3:</b> Criar um plano de treinamento para cada função.</p> <p><b>Exemplo 4:</b> Adquirir ou criar treinamento para cada função; o treinamento adquirido pode precisar ser personalizado para a organização.</p> <p><b>Exemplo 5:</b> Medir o desempenho dos resultados para identificar as áreas em que as mudanças no treinamento podem ser benéficas.</p>	<p><b>BSAFSS:</b> PD.2-2  <b>BSIMM:</b> T1.1, T1.7, T1.8, T2.5, T2.8, T2.9, T3.1, T3.2, T3.4  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> SM-4  <b>MSSDL:</b> 1  <b>NISTCSF:</b> PR.AT  <b>OWASPSAMM:</b> EG1-A, EG2-A  <b>PCISL:</b> 1,3  <b>SCAGILE:</b> Tarefas de segurança operacional 14, 15; Tarefas que exigem a ajuda de especialistas em segurança 1  <b>SCFPSSD:</b> Planejamento da implementação e implantação de práticas de desenvolvimento seguro  <b>SCSIC:</b> Controles de integridade do desenvolvimento de software do fornecedor  <b>SP80053:</b> SA-8  <b>SP800160:</b> 3.2.4, 3.2.6  <b>SP800161:</b> SA-8  <b>SP800181:</b> OV-TEA-001, OV-TEA-002; T0030, T0073, T0320; K0204, K0208, K0220, K0226, K0243, K0245, K0252; S0100, S0101; A0004, A0057</p>

<sup>5</sup> [Proveniência](#) é "a cronologia da origem, desenvolvimento, propriedade, localização e alterações em um sistema ou componente do sistema e dados associados. Também pode incluir o pessoal e os processos usados para interagir ou fazer modificações no sistema, no componente ou nos dados associados" [SP80053].

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
	<p><b>PO.2.3:</b> Obter o compromisso da alta gerência ou do funcionário responsável pela autorização para garantir o desenvolvimento seguro e transmitir esse compromisso a todos com funções e responsabilidades relacionadas ao desenvolvimento.</p>	<p><b>Exemplo 1:</b> Designar um único líder ou equipe de liderança para ser responsável por todo o processo de desenvolvimento de software seguro, incluindo a responsabilidade de liberar o software para produção e delegar responsabilidades conforme apropriado.</p> <p><b>Exemplo 2:</b> Aumentar a conscientização dos funcionários responsáveis pela autorização sobre os riscos do desenvolvimento de software sem integrar a segurança em todo o ciclo de vida do desenvolvimento e sobre a redução de riscos proporcionada pelas práticas de desenvolvimento seguro.</p> <p><b>Exemplo 3:</b> Auxiliar a gerência superior a incorporar o suporte seguro ao desenvolvimento em suas comunicações com o pessoal com funções e responsabilidades relacionadas ao desenvolvimento.</p> <p><b>Exemplo 4:</b> Instruir todo o pessoal com funções e responsabilidades relacionadas ao desenvolvimento sobre o compromisso da alta gerência com o desenvolvimento seguro e a importância do desenvolvimento seguro para a organização.</p>	<p><b>BSIMM:</b> SM1.3, SM2.7, CP2.5  <b>EO14028:</b> 4e(ix)  <b>NISTCSF:</b> ID.RM-1, ID.SC-1  <b>OWASPSAMM:</b> SM1.A  <b>PCISL:</b> 1.1  <b>SP800181:</b> T0001, T0004</p>
<p><b>Implementar cadeias de ferramentas de suporte (PO.3):</b> Usar a automação para reduzir o esforço humano e melhorar a precisão, a reprodutibilidade, a usabilidade e a abrangência das práticas de segurança em todo o SDLC, além de fornecer uma maneira de documentar e demonstrar o uso dessas práticas. As cadeias de ferramentas e as ferramentas podem ser usadas em diferentes níveis da organização, como em toda a organização ou em um projeto específico, e podem abordar uma parte específica do SDLC, como um pipeline de construção.</p>	<p><b>PO.3.1:</b> Especificar quais ferramentas ou tipos de ferramentas precisam ou deveriam ser incluídos em cada cadeia de ferramentas para mitigar os riscos identificados, bem como a forma como os componentes da cadeia de ferramentas devem ser integrados uns aos outros.</p>	<p><b>Exemplo 1:</b> Definir categorias de cadeias de ferramentas e especificar as ferramentas obrigatórias ou os tipos de ferramentas a serem usados em cada categoria.</p> <p><b>Exemplo 2:</b> Identificar ferramentas de segurança a serem integradas à cadeia de ferramentas do desenvolvedor.</p> <p><b>Exemplo 3:</b> Definir quais informações devem ser transmitidas entre as ferramentas e quais formatos de dados devem ser usados.</p> <p><b>Exemplo 4:</b> Avaliar os recursos de assinatura das ferramentas para criar registros/logs imutáveis para auditabilidade dentro da cadeia de ferramentas.</p> <p><b>Exemplo 5:</b> Usar tecnologia automatizada para o gerenciamento e a orquestração da cadeia de ferramentas.</p>	<p><b>BSIMM:</b> CR1.4, ST1.4, ST2.5, SE2.7  <b>CNCFSSCP:</b> Proteção de materiais - verificação; Proteção de pipelines de compilação - verificação, Automação, Autenticação/acesso seguros; Proteção de artefatos - verificação;  <b>EO14028:</b> 4e(iii), 4e(ix)  <b>MSSDL:</b> 8  <b>OWASPSAMM:</b> IR2-B, ST2-B  <b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 9  <b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor  <b>SP80053:</b> SA-15  <b>SP800161:</b> SA-15  <b>SP800181:</b> K0013, K0178</p>
	<p><b>PO.3.2:</b> Seguir as práticas de segurança recomendadas para implantar, operar e manter ferramentas e cadeias de ferramentas.</p>	<p><b>Exemplo 1:</b> Avaliar, selecionar e adquirir ferramentas e avaliar a segurança de cada ferramenta.</p> <p><b>Exemplo 2:</b> Integrar ferramentas com outras ferramentas e processos e fluxos de trabalho de desenvolvimento de software existentes.</p> <p><b>Exemplo 3:</b> Usar a configuração baseada em código para cadeias de ferramentas (por exemplo, pipelines como código, cadeias de ferramentas como código).</p> <p><b>Exemplo 4:</b> Implementar as tecnologias e os processos necessários para compilações reproduzíveis.</p> <p><b>Exemplo 5:</b> Atualizar, fazer upgrade ou substituir as ferramentas conforme necessário para solucionar as vulnerabilidades das ferramentas ou adicionar novos recursos às ferramentas.</p> <p><b>Exemplo 6:</b> Monitorar continuamente as ferramentas e os registros de ferramentas para detectar possíveis problemas operacionais e de segurança, inclusive violações de políticas e comportamentos anômalos.</p> <p><b>Exemplo 7:</b> Verificar regularmente a integridade e a procedência de cada ferramenta para identificar possíveis problemas.</p> <p><b>Exemplo 8:</b> Consultar <a href="#">PW.6</a> sobre compilador, interpretador e ferramentas de compilação.</p> <p><b>Exemplo 9:</b> Consultar <a href="#">PO.5</a> sobre a implementação e manutenção de ambientes seguros.</p>	<p><b>BSAFSS:</b> DE.2  <b>BSIMM:</b> SR1.1, SR1.3, SR3.4  <b>CNCFSSCP:</b> Proteção de pipelines de compilação - verificação, Automação, Ambientes controlados, Autenticação/acesso seguros; Proteção de artefatos - verificação, Automação, Ambientes controlados, Criptografia; Proteção de implantações - verificação, Automação  <b>EO14028:</b> 4e(i)(F), 4e(ii), 4e(iii), 4e(v), 4e(vi), 4e(ix)  <b>IEC62443:</b> SM-7  <b>IR8397:</b> 2.2  <b>OWASPASVS:</b> 1.14.3, 1.14.4, 14.1, 14.2  <b>OWASPMASVS:</b> 7.9  <b>OWASPSCVS:</b> 3, 5  <b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 9  <b>SCFPSSD:</b> Use as versões atuais do compilador e da cadeia de ferramentas e opções seguras do compilador  <b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor  <b>SP80053:</b> SA-15  <b>SP800161:</b> SA-15  <b>SP800181:</b> K0013, K0178</p>
	<p><b>PO.3.3:</b> Configure ferramentas para gerar artefatos<sup>6</sup> de seu suporte a práticas seguras de desenvolvimento</p>	<p><b>Exemplo 1:</b> Usar as ferramentas existentes (por exemplo, rastreamento de fluxo de trabalho, rastreamento de problemas, mapeamento de fluxo de valor) para</p>	<p><b>BSAFSS:</b> PD.1-5  <b>BSIMM:</b> SM1.4, SM3.4, SR1.3</p>

<sup>6</sup> Um artefato é "uma peça de evidência" [adaptado de [IR7692](#)]. Evidência é "base para crença ou descrença; dados nos quais se baseia a prova ou se estabelece a verdade ou a falsidade" [[SP800160](#)]. Os artefatos fornecem registros de práticas seguras de desenvolvimento de software.

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
	de software, conforme definido pela organização.	<p>criar uma trilha de auditoria das ações relacionadas ao desenvolvimento seguro que são realizadas para fins de melhoria contínua.</p> <p><b>Exemplo 2:</b> Determinar a frequência com que as informações coletadas devem ser auditadas e implementar os processos necessários.</p> <p><b>Exemplo 3:</b> Estabelecer e aplicar políticas de segurança e retenção para dados de artefatos.</p> <p><b>Exemplo 4:</b> Atribuir a responsabilidade de criar quaisquer artefatos necessários que as ferramentas não possam gerar.</p>	<p><b>CNCFSSCP:</b> Proteção de pipelines de compilação - verificação, Automação, Ambientes controlados; Proteção de artefatos - verificação</p> <p><b>EO14028:</b> 4e(i)(F), 4e(ii), 4e(v), 4e(ix)</p> <p><b>IEC62443:</b> SM-12, SI-2</p> <p><b>MSSDL:</b> 8</p> <p><b>OWASPSAMM:</b> PC3-B</p> <p><b>OWASPSCVS:</b> 3.13, 3.14</p> <p><b>PCISL:</b> 2.5</p> <p><b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 9</p> <p><b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor</p> <p><b>SP80053:</b> SA-15</p> <p><b>SP800161:</b> SA-15</p> <p><b>SP800181:</b> K0013; T0024</p>
<p><b>Definir e usar critérios para verificações de segurança de software (PO.4):</b> Ajudar a garantir que o software resultante do SDLC atenda às expectativas da organização, definindo e usando critérios para verificar a segurança do software durante o desenvolvimento.</p>	<p><b>PO.4.1:</b> Definir critérios para verificações de segurança de software e acompanhe todo o SDLC.</p>	<p><b>Exemplo 1:</b> Assegurar-se de que os critérios indiquem adequadamente a eficácia com que o risco de segurança está sendo gerenciado.</p> <p><b>Exemplo 2:</b> Definir indicadores-chave de desempenho (KPIs), indicadores-chave de risco (KRIs), pontuações de gravidade de vulnerabilidade e outras medidas de segurança de software.</p> <p><b>Exemplo 3:</b> Adicionar critérios de segurança de software às verificações existentes (por exemplo, a Definição de Feito nas metodologias ágeis do SDLC).</p> <p><b>Exemplo 4:</b> Analisar os artefatos gerados como parte do sistema de fluxo de trabalho de desenvolvimento de software para determinar se eles atendem aos critérios.</p> <p><b>Exemplo 5:</b> Registrar as aprovações, rejeições e solicitações de exceção de verificações de segurança como parte do fluxo de trabalho e do sistema de rastreamento.</p> <p><b>Exemplo 6:</b> Analisar os dados coletados no contexto dos sucessos e fracassos de segurança de cada projeto de desenvolvimento e usar os resultados para aprimorar o SDLC.</p>	<p><b>BSAFSS:</b> TV.2-1, TV.5-1</p> <p><b>BSIMM:</b> SM1.4, SM2.1, SM2.2, SM2.6, SM3.3, CP2.2</p> <p><b>EO14028:</b> 4e(iv), 4e(v), 4e(ix)</p> <p><b>IEC62443:</b> SI-1, SI-2, SVV-3</p> <p><b>ISO27034:</b> 7.3.5</p> <p><b>MSSDL:</b> 3</p> <p><b>OWASPSAMM:</b> PC3-A, DR3-B, IR3-B, ST3-B</p> <p><b>PCISL:</b> 3.3</p> <p><b>SP80053:</b> SA-15, SA-15(1)</p> <p><b>SP800160:</b> 3.2.1, 3.2.5, 3.3.1</p> <p><b>SP800161:</b> SA-15, SA-15(1)</p> <p><b>SP800181:</b> K0153, K0165</p>
	<p><b>PO.4.2:</b> Implementar processos, mecanismos etc. para coletar e proteger as informações necessárias em apoio aos critérios.</p>	<p><b>Exemplo 1:</b> Usar a cadeia de ferramentas para coletar automaticamente informações que embasam a tomada de decisões de segurança.</p> <p><b>Exemplo 2:</b> Implementar ferramentas adicionais, se necessário, para apoiar a geração e a coleta de informações que apoiem os critérios.</p> <p><b>Exemplo 3:</b> Automatizar os processos de tomada de decisão utilizando os critérios e revisar periodicamente esses processos.</p> <p><b>Exemplo 4:</b> Permitir apenas que o pessoal autorizado acesse as informações coletadas e impedir qualquer alteração ou exclusão das informações.</p>	<p><b>BSAFSS:</b> PD.1-4, PD.1-5</p> <p><b>BSIMM:</b> SM1.4, SM2.1, SM2.2, SM3.4</p> <p><b>EO14028:</b> 4e(iv), 4e(v), 4e(ix)</p> <p><b>IEC62443:</b> SI-1, SVV-1, SVV-2, SVV-3, SVV-4</p> <p><b>OWASPSAMM:</b> PC3-B</p> <p><b>PCISL:</b> 2.5</p> <p><b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor</p> <p><b>SP80053:</b> SA-15, SA-15(1), SA-15(11)</p> <p><b>SP800160:</b> 3.2.5, 3.3.7</p> <p><b>SP800161:</b> SA-15, SA-15(1), SA-15(11)</p> <p><b>SP800181:</b> T0349; K0153</p>
<p><b>Implementar e manter ambientes seguros para o desenvolvimento de software (PO.5):</b> Garantir que todos os componentes dos ambientes de desenvolvimento de software sejam fortemente protegidos contra ameaças internas e externas para evitar o comprometimento dos ambientes ou do software que está sendo desenvolvido ou mantido neles. Exemplos de ambientes para desenvolvimento de software incluem ambientes de desenvolvimento, compilação, teste e distribuição.</p>	<p><b>PO.5.1:</b> Separar e proteger cada ambiente envolvido no desenvolvimento de software.</p>	<p><b>Exemplo 1:</b> Usar autenticação multifatorial baseada em risco e acesso condicional para cada ambiente.</p> <p><b>Exemplo 2:</b> Usar a segmentação da rede e os controles de acesso para separar os ambientes entre si e dos ambientes de produção, e para separar os componentes entre si em cada ambiente que não seja de produção, a fim de reduzir as superfícies de ataque e o movimento lateral dos invasores e o escalonamento de privilégios/acesso.</p> <p><b>Exemplo 3:</b> Impor a autenticação e restringir rigidamente as conexões que entram e saem de cada ambiente de desenvolvimento de software, incluindo a minimização do acesso à Internet apenas para o que for necessário.</p> <p><b>Exemplo 4:</b> Minimizar o acesso humano direto aos sistemas de cadeia de ferramentas, como serviços de construção. Monitorar e auditar continuamente</p>	<p><b>BSAFSS:</b> DE.1, IA.1, IA.2</p> <p><b>CNCFSSCP:</b> Proteção de pipelines de compilação - ambientes controlados</p> <p><b>EO14028:</b> 4e(i)(A), 4e(i)(B), 4e(i)(C), 4e(i)(D), 4e(i)(F), 4e(ii), 4e(iii), 4e(v), 4e(vi), 4e(ix)</p> <p><b>IEC62443:</b> SM-7</p> <p><b>NISTCSF:</b> PR.AC-5, PR.DS-7</p> <p><b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 11</p> <p><b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor</p> <p><b>SP80053:</b> SA-3(1), SA-8, SA-15</p> <p><b>SP800161:</b> SA-3, SA-8, SA-15</p> <p><b>SP800181:</b> OM-NET-001, SP-SYS-001; T0019, T0023, T0144, T0160, T0262, T0438, T0484, T0485, T0553; K0001, K0005, K0007, K0033, K0049, K0056, K0061, K0071,</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
		<p>todas as tentativas de acesso e todo o uso de acesso privilegiado.</p> <p><b>Exemplo 5:</b> Minimizar o uso de software e serviços de ambientes de produção em ambientes que não são de produção.</p> <p><b>Exemplo 6:</b> Registrar, monitorar e auditar regularmente as relações de confiança para autorização e acesso entre os ambientes e entre os componentes de cada ambiente.</p> <p><b>Exemplo 7:</b> Registrar e monitorar continuamente as operações e os alertas em todos os componentes do ambiente de desenvolvimento para detectar, responder e recuperar-se de tentativas e incidentes cibernéticos reais.</p> <p><b>Exemplo 8:</b> Configurar os controles de segurança e outras ferramentas envolvidas na separação e proteção dos ambientes para gerar artefatos para suas atividades.</p> <p><b>Exemplo 9:</b> Monitorar continuamente todos os softwares implantados em cada ambiente em busca de novas vulnerabilidades e responder a elas de forma adequada, seguindo uma abordagem baseada em riscos.</p> <p><b>Exemplo 10:</b> Configurar e implementar medidas para proteger as infraestruturas de hospedagem dos ambientes seguindo uma arquitetura de confiança zero<sup>7</sup>.</p>	<p>K0104, K0112, K0179, K0326, K0487; S0007, S0084, S0121; A0048</p>
	<p><b>PO.5.2:</b> Proteger e fortalecer os endpoints de desenvolvimento (ou seja, endpoints para designers de software, desenvolvedores, testadores, construtores etc.) para executar tarefas relacionadas ao desenvolvimento usando uma abordagem baseada em riscos.</p>	<p><b>Exemplo 1:</b> Configurar cada endpoint de desenvolvimento com base em guias de fortalecimento aprovados, listas de verificação, etc.; por exemplo, ativar a criptografia compatível com FIPS de todos os dados sensíveis em repouso e em trânsito.</p> <p><b>Exemplo 2:</b> Configurar cada endpoint de desenvolvimento e os recursos de desenvolvimento para fornecer a menor funcionalidade necessária aos usuários e serviços e para aplicar o princípio do menor privilégio.</p> <p><b>Exemplo 3:</b> Monitorar continuamente a postura de segurança de todos os endpoints de desenvolvimento, incluindo o monitoramento e a auditoria de todo o uso de acesso privilegiado.</p> <p><b>Exemplo 4:</b> Configurar os controles de segurança e outras ferramentas envolvidas na proteção e no fortalecimento dos endpoints de desenvolvimento para gerar artefatos para suas atividades.</p> <p><b>Exemplo 5:</b> Exigir autenticação multifatorial para todo o acesso a endpoints de desenvolvimento e recursos de desenvolvimento.</p> <p><b>Exemplo 6:</b> Fornecer endpoints de desenvolvimento dedicados em redes que não sejam de produção para realizar todas as tarefas relacionadas ao desenvolvimento. Fornecer endpoints separados nas redes de produção para todas as outras tarefas.</p> <p><b>Exemplo 7:</b> Configurar cada endpoint de desenvolvimento seguindo uma arquitetura de confiança zero.</p>	<p><b>BSAFSS:</b> DE.1-1, IA.1, IA.2  <b>EO14028:</b> 4e(i)(C), 4e(i)(E), 4e(i)(F), 4e(ii), 4e(iii), 4e(v), 4e(vi), 4e(ix)  <b>IEC62443:</b> SM-7  <b>NISTCSF:</b> PR.AC-4, PR.AC-7, PR.IP-1, PR.IP-3, PR.IP-12, PR.PT-1, PR.PT-3, DE.CM  <b>SCAGILE:</b> Tarefas que exigem a ajuda de especialistas em segurança 11  <b>SCSIC:</b> Controles de integridade da entrega de software do fornecedor  <b>SP80053:</b> SA-15  <b>SP800161:</b> SA-15  <b>SP800181:</b> OM-ADM-001, SP-SYS-001; T0484, T0485, T0489, T0553; K0005, K0007, K0077, K0088, K0130, K0167, K0205, K0275; S0076, S0097, S0121, S0158; A0155</p>
<b>Proteger o software (PS)</b>			
<p><b>Proteger todas as formas de código contra acesso não autorizado e adulteração (PS.1):</b> Ajudar a evitar alterações não autorizadas no código, tanto inadvertidas quanto intencionais, que poderiam contornar ou negar as características de segurança pretendidas do software. Para códigos que não se destinam a ser acessíveis ao público, isso ajuda a evitar o roubo do software e pode tornar mais difícil ou demorado para os invasores encontrarem vulnerabilidades no software.</p>	<p><b>PS.1.1:</b> Armazenar todas as formas de código, inclusive código-fonte, código executável e configuração como código, com base no princípio do menor privilégio, de modo que somente o pessoal, as ferramentas, os serviços etc. autorizados tenham acesso.</p>	<p><b>Exemplo 1:</b> Armazenar todo o código-fonte e a configuração como código em um repositório de código e restringir o acesso a ele com base na natureza do código. Por exemplo, o código-fonte aberto destinado ao acesso público pode precisar ter sua integridade e disponibilidade protegidas; outro código também pode precisar ter sua confidencialidade protegida.</p> <p><b>Exemplo 2:</b> Usar os recursos de controle de versão do repositório para rastrear todas as alterações feitas no código com responsabilidade pela conta individual.</p> <p><b>Exemplo 3:</b> Usar a assinatura do commit para repositórios de código.</p> <p><b>Exemplo 4:</b> Fazer com que o proprietário do código revise e aprove todas as alterações feitas no código por outras pessoas.</p> <p><b>Exemplo 5:</b> Usar a assinatura de código<sup>8</sup> para ajudar a proteger a integridade</p>	<p><b>BSAFSS:</b> IA.1, IA.2, SM.4-1, DE.1-2  <b>BSIMM:</b> SE2.4  <b>CNCFSSCP:</b> Proteção do Código-Fonte—Verificação, Automação, Ambientes Controlados, Autenticação Segura; Proteção de Materiais—Automação  <b>EO14028:</b> 4e(iii), 4e(iv), 4e(ix)  <b>IDASOAR:</b> Folha de dados 25  <b>IEC62443:</b> SM-6, SM-7, SM-8  <b>NISTCSF:</b> PR.AC-4, PR.DS-6, PR.IP-3  <b>OWASPASVS:</b> 1.10, 10.3.2  <b>OWASPMASVS:</b> 7.1</p>

<sup>7</sup> Consulte o NIST SP 800-207, *Zero Trust Architecture*, para obter informações adicionais (<https://doi.org/10.6028/NIST.SP.800-207>).

<sup>8</sup> Para obter mais informações sobre assinatura de código, consultar o White Paper de segurança cibernética do NIST, *Security Considerations for Code Signing* (<https://doi.org/10.6028/NIST.CSWP.01262018>).

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
		<p>dos executáveis.</p> <p><b>Exemplo 6:</b> Usar criptografia (por exemplo, hashes criptográficos) para ajudar a proteger a integridade do arquivo.</p>	<p><b>OWASPSAMM:</b> OE3-B  <b>PCISSLC:</b> 5.1, 6.1  <b>SCSIC:</b> Controles de Integridade de Entrega de Software do Fornecedor, Controles de Integridade de Desenvolvimento de Software do Fornecedor  <b>SP80053:</b> SA-10  <b>SP800161:</b> SA-8, SA-10</p>
<p><b>Fornecer um Mecanismo para Verificar a Integridade da Versão do Software (PS.2):</b> Ajudar os adquirentes de software a garantir que o software adquirido seja legítimo e não tenha sido adulterado.</p>	<p><b>PS.2.1:</b> Disponibilizar informações sobre a verificação da integridade do software para os adquirentes de software.</p>	<p><b>Exemplo 1:</b> Publicar hashes criptográficos para arquivos de liberação em um site bem protegido.</p> <p><b>Exemplo 2:</b> Usar uma autoridade de certificação estabelecida para assinatura de código, de modo que os sistemas operacionais dos consumidores ou outras ferramentas e serviços possam confirmar a validade das assinaturas antes do uso.</p> <p><b>Exemplo 3:</b> Revisar periodicamente os processos de assinatura de código, incluindo renovação, rotação, revogação e proteção de certificados.</p>	<p><b>BSAFSS:</b> SM.4, SM.5, SM.6  <b>BSIMM:</b> SE2.4  <b>CNCFSSCP:</b> Proteção de Implementações—Verificação  <b>EO14028:</b> 4e(iii), 4e(ix), 4e(x)  <b>IEC62443:</b> SM-6, SM-8, SUM-4  <b>NISTCSF:</b> PR.DS-6  <b>NISTLABEL:</b> 2.2.2.4  <b>OWASPSAMM:</b> OE3-B  <b>OWASPSCVS:</b> 4  <b>PCISSLC:</b> 6.1, 6.2  <b>SCSIC:</b> Controles de Integridade da Entrega de Software do Fornecedor  <b>SP80053:</b> SA-8  <b>SP800161:</b> SA-8  <b>SP800181:</b> K0178</p>
<p><b>Arquivar e Proteger Cada Versão de Software (PS.3):</b> Preservar as versões do software para ajudar a identificar, analisar e eliminar as vulnerabilidades descobertas no software após o lançamento.</p>	<p><b>PS.3.1:</b> Arquivar com segurança os arquivos e dados de suporte necessários (por exemplo, informações de verificação de integridade, dados de procedência) a serem retidos para cada versão do software.</p>	<p><b>Exemplo 1:</b> Armazenar os arquivos da versão, as imagens associadas etc. em repositórios de acordo com a política estabelecida pela organização. Permitir o acesso somente de leitura a eles pelo pessoal necessário e nenhum acesso por qualquer outra pessoa.</p> <p><b>Exemplo 2:</b> Armazenar e proteger as informações de verificação da integridade da versão e os dados de procedência, por exemplo, mantendo-os em um local separado dos arquivos da versão ou assinando os dados.</p>	<p><b>BSAFSS:</b> PD.1-5, DE.1-2, IA.2  <b>CNCFSSCP:</b> Proteção de Artefatos—Automação, Ambientes Controlados, Criptografia; Proteção de implantações—Verificação  <b>EO14028:</b> 4e(iii), 4e(vi), 4e(ix), 4e(x)  <b>IDASOAR:</b> 25  <b>IEC62443:</b> SM-6, SM-7  <b>NISTCSF:</b> PR.IP-4  <b>OWASPSCVS:</b> 1, 3.18, 3.19, 6.3  <b>PCISSLC:</b> 5.2, 6.1, 6.2  <b>SCSIC:</b> Controles de Integridade da Entrega de Software do Fornecedor  <b>SP80053:</b> SA-10, SA-15, SA-15(11), SR-4  <b>SP800161:</b> SA-8, SA-10, SA-15(11), SR-4</p>
	<p><b>PS.3.2:</b> Coletar, proteger, manter e compartilhar dados de procedência de todos os componentes de cada versão de software (por exemplo, em uma lista de materiais de software [SBOM]).</p>	<p><b>Exemplo 1:</b> Disponibilizar os dados de procedência para os adquirentes de software de acordo com as políticas da organização, de preferência usando formatos baseados em padrões.</p> <p><b>Exemplo 2:</b> Disponibilizar os dados de procedência para as equipes de operações e resposta da organização para ajudá-las a reduzir as vulnerabilidades do software.</p> <p><b>Exemplo 3:</b> Proteger a integridade dos dados de procedência e fornecer uma maneira de os destinatários verificarem a integridade dos dados de procedência.</p> <p><b>Exemplo 4:</b> Atualizar os dados de procedência sempre que qualquer componente do software for atualizado.</p>	<p><b>BSAFSS:</b> SM.2  <b>BSIMM:</b> SE3.6  <b>CNCFSSCP:</b> Proteção de Materiais—Verificação, Automação  <b>EO14028:</b> 4e(vi), 4e(vii), 4e(ix), 4e(x)  <b>NTIASBOM:</b> Todos  <b>OWASPSCVS:</b> 1.4, 2  <b>SCSIC:</b> Controles de Integridade da Entrega de Software do Fornecedor  <b>SCTPC:</b> MAINTAIN3  <b>SP80053:</b> SA-8, SR-3, SR-4  <b>SP800161:</b> SA-8, SR-3, SR-4</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
<b>Produzir Software Bem Protegido (PW)</b>			
<p><b>Projetar Software para Atender aos Requisitos de Segurança e Atenuar os Riscos de Segurança (PW.1):</b> Identificar e avaliar os requisitos de segurança do software; determinar os riscos de segurança que o software provavelmente enfrentará durante a operação e como o projeto e a arquitetura do software devem atenuar esses riscos; e justificar os casos em que a análise baseada em riscos indica que os requisitos de segurança devem ser relaxados ou dispensados. A abordagem dos requisitos e riscos de segurança durante o projeto do software (seguro por projeto) é fundamental para melhorar a segurança do software e ajudar a aumentar a eficiência do desenvolvimento.</p>	<p><b>PW.1.1:</b> Usar formas de modelagem de risco – como modelagem de ameaças, modelagem de ataques ou mapeamento da superfície de ataque – para ajudar a avaliar o risco de segurança do software.</p>	<p><b>Exemplo 1:</b> Treinar a equipe de desenvolvimento (especialmente os campeões de segurança) ou colaborar com um especialista em modelagem de riscos para criar modelos e analisar como usar uma abordagem baseada em riscos para comunicar os riscos e determinar como tratá-los, incluindo a implementação de atenuações.</p> <p><b>Exemplo 2:</b> Realizar avaliações mais rigorosas para áreas de alto risco, como a proteção de dados confidenciais e a proteção da identificação, autenticação e controle de acesso, incluindo o gerenciamento de credenciais.</p> <p><b>Exemplo 3:</b> Analisar os relatórios de vulnerabilidade e as estatísticas de softwares anteriores para informar a avaliação de risco de segurança.</p> <p><b>Exemplo 4:</b> Usar métodos de classificação de dados para identificar e caracterizar cada tipo de dados com os quais o software irá interagir.</p>	<p><b>BSAFSS:</b> SC.1  <b>BSIMM:</b> AM1.2, AM1.3, AM1.5, AM2.1, AM2.2, AM2.5, AM2.6, AM2.7, SFD2.2, AA1.1, AA1.2, AA1.3, AA2.1  <b>EO14028:</b> 4e(ix)  <b>IDASOAR:</b> 1  <b>IEC62443:</b> SM-4, SR-1, SR-2, SD-1  <b>IR8397:</b> 2,1  <b>ISO27034:</b> 7.3.3  <b>MSSDL:</b> 4  <b>NISTCSF:</b> ID.RA  <b>OWASPASVS:</b> 1.1.2, 1.2, 1.4, 1.6, 1.8, 1.9, 1.11, 2, 3, 4, 6, 8, 9, 11, 12, 13  <b>OWASPMASVS:</b> 1.6, 1.8, 2, 3, 4, 5, 6  <b>OWASPSAMM:</b> TA1-A, TA1-B, TA3-B, DR1-A  <b>PCISSLC:</b> 3.2, 3,3  <b>SCAGILE:</b> Tarefas que Exigem a Ajuda de Especialistas em Segurança 3  <b>SCFPSSD:</b> Modelagem de Ameaças  <b>SCTTM:</b> Guia completo  <b>SP80053:</b> SA-8, SA-11(2), SA-11(6), SA-15(5)  <b>SP800160:</b> 3.3.4, 3.4.5  <b>SP800161:</b> SA-8, SA-11(2), SA-11(6), SA-15(5)  <b>SP800181:</b> T0038, T0062; K0005, K0009, K0038, K0039, K0070, K0080, K0119, K0147, K0149, K0151, K0152, K0160, K0161, K0162, K0165, K0297, K0310, K0344, K0362, K0487, K0624; S0006, S0009, S0022, S0078, S0171, S0229, S0248; A0092, A0093, A0107</p>
	<p><b>PW.1.2:</b> Rastrear e manter os requisitos de segurança, os riscos e as decisões de projeto do software.</p>	<p><b>Exemplo 1:</b> Registrar a resposta a cada risco, incluindo como as mitigações devem ser alcançadas e quais são as justificativas para quaisquer exceções aprovadas aos requisitos de segurança. Adicionar quaisquer atenuações aos requisitos de segurança do software.</p> <p><b>Exemplo 2:</b> Manter registros de decisões de projeto, respostas a riscos e exceções aprovadas que possam ser usadas para fins de auditoria e manutenção durante o restante do ciclo de vida do software.</p> <p><b>Exemplo 3:</b> Reavaliar periodicamente todas as exceções aprovadas para os requisitos de segurança e implementar as alterações necessárias.</p>	<p><b>BSAFSS:</b> SC.1-1, PD.1-1  <b>BSIMM:</b> SFD3.1, SFD3.3, AA2.2, AA3.2  <b>EO14028:</b> 4e(v), 4e(ix)  <b>IEC62443:</b> SD-1  <b>ISO27034:</b> 7.3.3  <b>MSSDL:</b> 4  <b>NISTLABEL:</b> 2.2.2.2  <b>OWASPASVS:</b> 1.1.3, 1.1.4  <b>OWASPMASVS:</b> 1.3, 1,6  <b>OWASPSAMM:</b> DR1-B  <b>PCISSLC:</b> 3.2, 3,3  <b>SP80053:</b> SA-8, SA-10, SA-17  <b>SP800161:</b> SA-8, SA-17  <b>SP800181:</b> T0256; K0005, K0038, K0039, K0147, K0149, K0160, K0161, K0162, K0165, K0344, K0362, K0487; S0006, S0009, S0078, S0171, S0229, S0248; A0092, A0107</p>
	<p><b>PW.1.3:</b> Quando apropriado, criar suporte para o uso de recursos e serviços de segurança padronizados (por exemplo, permitindo que o software se integre aos sistemas existentes de gerenciamento de registros, gerenciamento de identidade, controle de acesso e gerenciamento de vulnerabilidades) em vez de criar implementações proprietárias de recursos e serviços de segurança. [Anteriormente PW.4.3]</p>	<p><b>Exemplo 1:</b> Manter um ou mais repositórios de software de módulos para dar suporte a recursos e serviços de segurança padronizados.</p> <p><b>Exemplo 2:</b> Determinar configurações seguras para módulos de suporte a recursos e serviços de segurança padronizados e disponibilizar essas configurações (por exemplo, como configuração como código) para que os desenvolvedores possam usá-las prontamente.</p> <p><b>Exemplo 3:</b> Definir critérios para os recursos e serviços de segurança que devem ser suportados pelo software a ser desenvolvido.</p>	<p><b>BSAFSS:</b> SI.2-1, SI.2-2, LO.1  <b>BSIMM:</b> SFD1.1, SFD2.1, SFD3.2, SR1.1, SR3.4  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> SD-1, SD-4  <b>MSSDL:</b> 5  <b>OWASPASVS:</b> 1.1.6  <b>OWASPSAMM:</b> SA2-A  <b>SCFPSSD:</b> Padronizar o Gerenciamento de Identidade e Acesso; Estabelecer Requisitos de Registro e Práticas de Auditoria</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
<p><b>Revisar o Projeto do Software para Verificar a Conformidade com os Requisitos de Segurança e as Informações de Risco (PW.2):</b> Ajudar a garantir que o software atenderá aos requisitos de segurança e abordará de forma satisfatória as informações de risco identificadas.</p>	<p><b>PW.2.1:</b> Ter 1) uma pessoa (ou pessoas) qualificada(s) que não esteja(m) envolvida(s) com o projeto e/ou 2) processos automatizados instanciados na cadeia de ferramentas revise(m) o projeto do software para confirmar e reforçar que ele atende a todos os requisitos de segurança e aborda satisfatoriamente as informações de risco identificadas.</p>	<p><b>Exemplo 1:</b> Revisar o design do software para confirmar que ele atende aos requisitos de segurança aplicáveis.  <b>Exemplo 2:</b> Revisar os modelos de risco criados durante o projeto do software para determinar se eles parecem identificar adequadamente os riscos.  <b>Exemplo 3:</b> Revisar o projeto do software para confirmar que ele aborda de forma satisfatória os riscos identificados pelos modelos de risco.  <b>Exemplo 4:</b> Fazer com que o projetista do software corrija as falhas para atender aos requisitos.  <b>Exemplo 5:</b> Alterar o projeto e/ou a estratégia de resposta a riscos se os requisitos de segurança não puderem ser atendidos.  <b>Exemplo 6:</b> Registrar os resultados das revisões de projeto para que sirvam como artefatos (por exemplo, na especificação do software, no sistema de rastreamento de problemas, no modelo de ameaças).</p>	<p><b>BSAFSS:</b> TV.3  <b>BSIMM:</b> AA1.1, AA1.2, AA1.3, AA2.1, AA3.1  <b>EO14028:</b> 4e(iv), 4e(v), 4e(ix)  <b>IEC62443:</b> SM-2, SR-2, SR-5, SD-3, SD-4, SI-2  <b>ISO27034:</b> 7.3.3  <b>OWASPASVS:</b> 1.1.5  <b>OWASPSAMM:</b> DR1-A, DR1-B  <b>PCISL:</b> 3.2  <b>SP800181:</b> T0328; K0038, K0039, K0070, K0080, K0119, K0152, K0153, K0161, K0165, K0172, K0297; S0006, S0009, S0022, S0036, S0141, S0171</p>
<p><b>Verificar se o Software de Terceiros está em Conformidade com os Requisitos de Segurança (PW.3):</b> Movido para PW.4</p>	<p><b>PW.3.1:</b> Movido para PO.1.3  <b>PW.3.2:</b> Movido para PW.4.4</p>		
<p><b>Reutilizar Software Existente e Bem Protegido Quando for Viável, em vez de Duplicar a Funcionalidade (PW.4):</b> Reduzir os custos de desenvolvimento de software, agilizar o desenvolvimento de software e diminuir a probabilidade de introduzir vulnerabilidades de segurança adicionais no software, reutilizando módulos e serviços de software que já tiveram sua postura de segurança verificada. Isso é particularmente importante para o software que implementa a funcionalidade de segurança, como módulos e protocolos criptográficos.</p>	<p><b>PW.4.1:</b> Adquirir e manter componentes de software bem protegidos (por exemplo, bibliotecas de software, módulos, middleware, estruturas) de desenvolvedores comerciais, de código aberto e de terceiros para uso no software da organização.</p>	<p><b>Exemplo 1:</b> Revisar e avaliar componentes de software de terceiros no contexto de seu uso esperado. Se um componente for usado de uma maneira substancialmente diferente no futuro, fazer a revisão e a avaliação novamente com esse novo contexto em mente.  <b>Exemplo 2:</b> Determinar configurações seguras para componentes de software e disponibilizá-las (por exemplo, como configuração como código) para que os desenvolvedores possam usar prontamente as configurações.  <b>Exemplo 3:</b> Obter informações de proveniência (por exemplo, SBOM, análise de composição de fontes, análise de composição de software binário) para cada componente de software e analisar essas informações para avaliar melhor o risco que o componente pode apresentar.  <b>Exemplo 4:</b> Estabelecer um ou mais repositórios de software para hospedar componentes de código aberto sancionados e aprovados.  <b>Exemplo 5:</b> Manter uma lista de componentes de software comerciais aprovados pela organização e versões de componentes, juntamente com seus dados de procedência.  <b>Exemplo 6:</b> Designar quais componentes devem ser incluídos no software a ser desenvolvido.  <b>Exemplo 7:</b> Implementar processos para atualizar os componentes de software implantados para versões mais recentes e reter as versões mais antigas dos componentes de software até que todas as transições dessas versões tenham sido concluídas com êxito.  <b>Exemplo 8:</b> Se a integridade ou a procedência dos binários adquiridos não puder ser confirmada, criar binários a partir do código-fonte após verificar a integridade e a procedência do código-fonte.</p>	<p><b>BSAFSS:</b> SM.2  <b>BSIMM:</b> SFD2.1, SFD3.2, SR2.4, SR3.1, SE3.6  <b>CNCFSSCP:</b> Proteção de Materiais—Verificação  <b>EO14028:</b> 4e(iii), 4e(vi), 4e(ix), 4e(x)  <b>IDASOAR:</b> 19  <b>IEC62443:</b> SM-9, SM-10  <b>MSSDL:</b> 6  <b>NISTCSF:</b> ID.SC-2  <b>OWASPASVS:</b> 1.1.6  <b>OWASPSAMM:</b> SA1-A  <b>OWASPSCVS:</b> 4  <b>SCSIC:</b> Controles de Integridade do Suprimento do Fornecedor  <b>SCTPC:</b> MANUTENÇÃO  <b>SP80053:</b> SA-4, SA-5, SA-8(3), SA-10(6), SR-3, SR-4  <b>SP800161:</b> SA-4, SA-5, SA-8(3), SA-10(6), SR-3, SR-4  <b>SP800181:</b> K0039</p>
	<p><b>PW.4.2:</b> Criar e manter componentes de software bem protegidos internamente, seguindo os processos do SDLC, para atender às necessidades internas comuns de desenvolvimento de software que não podem ser melhor atendidas por componentes de software de terceiros.</p>	<p><b>Exemplo 1:</b> Seguir as práticas de segurança estabelecidas pela organização para o desenvolvimento seguro de software ao criar e manter os componentes.  <b>Exemplo 2:</b> Determinar configurações seguras para componentes de software e disponibilizá-las (por exemplo, como configuração como código) para que os desenvolvedores possam usar prontamente as configurações.  <b>Exemplo 3:</b> Manter um ou mais repositórios de software para esses componentes.  <b>Exemplo 4:</b> Designar quais componentes devem ser incluídos no software a ser desenvolvido.  <b>Exemplo 5:</b> Implementar processos para atualizar componentes de software implantados para versões mais recentes e manter versões mais antigas de componentes de software até que todas as transições dessas versões tenham sido concluídas com êxito.</p>	<p><b>BSIMM:</b> SFD1.1, SFD2.1, SFD3.2, SR1.1  <b>EO14028:</b> 4e(ix)  <b>IDASOAR:</b> 19  <b>OWASPASVS:</b> 1.1.6  <b>SCTPC:</b> MANUTENÇÃO  <b>SP80053:</b> SA-8(3)  <b>SP800161:</b> SA-8(3)  <b>SP800181:</b> SP-DEV-001</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
	<p><b>PW.4.3:</b> <i>Movido para PW.1.3</i></p> <p><b>PW.4.4:</b> Verificar se os componentes de software comercial, de código aberto e todos os outros componentes de software de terceiros adquiridos estão em conformidade com os requisitos, conforme definido pela organização, ao longo de seus ciclos de vida.</p>	<p><b>Exemplo 1:</b> Verificar regularmente se há vulnerabilidades conhecidas publicamente nos módulos e serviços de software que ainda não foram corrigidas pelos fornecedores.</p> <p><b>Exemplo 2:</b> Incorporar à cadeia de ferramentas a detecção automática de vulnerabilidades conhecidas em componentes de software.</p> <p><b>Exemplo 3:</b> Usar os resultados existentes de serviços comerciais para verificar os módulos e serviços de software.</p> <p><b>Exemplo 4:</b> Certificar-se de que cada componente de software ainda seja mantido ativamente e não tenha atingido o fim da vida útil; isso deve incluir novas vulnerabilidades encontradas no software que está sendo corrigido.</p> <p><b>Exemplo 5:</b> Determinar um plano de ação para cada componente de software que não está mais sendo mantido ou que não estará disponível em um futuro próximo.</p> <p><b>Exemplo 6:</b> Confirmar a integridade dos componentes de software por meio de assinaturas digitais ou outros mecanismos.</p> <p><b>Exemplo 7:</b> Revisar, analisar e/ou testar o código. Consulte <a href="#">PW.7</a> e <a href="#">PW.8</a>.</p>	<p><b>BSAFSS:</b> SC.3-1, SM.2-1, SM.2-2, SM.2-3, TV.2, TV.3</p> <p><b>BSIMM:</b> CP3.2, SR2.4, SR3.1, SR3.2, SE2.4, SE3.6</p> <p><b>CNCFSSCP:</b> Proteção de Materiais—Verificação, Automação</p> <p><b>EO14028:</b> 4e(iii), 4e(iv), 4e(vi), 4e(ix), 4e(x)</p> <p><b>IDASOAR:</b> 21</p> <p><b>IEC62443:</b> SI-1, SM-9, SM-10, DM-1</p> <p><b>IR8397:</b> 2.11</p> <p><b>MSSDL:</b> 7</p> <p><b>NISTCSF:</b> ID.SC-4, PR.DS-6</p> <p><b>NISTLABEL:</b> 2.2.2.2</p> <p><b>OWASPASVS:</b> 10, 14.2</p> <p><b>OWASPMASVS:</b> 7.5</p> <p><b>OWASPSAMM:</b> TA3-A, SR3-B</p> <p><b>OWASPSCVS:</b> 4, 5, 6</p> <p><b>PCISSLC:</b> 3.2, 3.4, 4.1</p> <p><b>SCAGILE:</b> Tarefas que Exigem a Ajuda de Especialistas em Segurança 8</p> <p><b>SCFPSSD:</b> Gerenciar o Risco de Segurança Inerente ao Uso de Componentes de Terceiros</p> <p><b>SCSIC:</b> Controles de Integridade de Suprimento de Fornecedores, Revisões por Pares e Testes de Segurança</p> <p><b>SCTPC:</b> MANTER, AVALIAR</p> <p><b>SP80053:</b> SA-9, SR-3, SR-4, SR-4(3), SR-4(4)</p> <p><b>SP800160:</b> 3.1.2, 3.3.8</p> <p><b>SP800161:</b> SA-4, SA-8, SA-9, SA-9(3), SR-3, SR-4, SR-4(3), SR-4(4)</p> <p><b>SP800181:</b> SP-DEV-002; K0153, K0266; S0298</p>
<p><b>Criar Código-Fonte Seguindo as Práticas de Codificação Segura (PW.5):</b> Diminuir o número de vulnerabilidades de segurança no software e reduzir os custos, minimizando as vulnerabilidades introduzidas durante a criação do código-fonte que atendem ou excedem os critérios de gravidade de vulnerabilidade definidos pela organização.</p>	<p><b>PW.4.5:</b> <i>Movido para PW.4.1 e PW.4.4</i></p> <p><b>PW.5.1:</b> Seguir todas as práticas de codificação segura adequadas às linguagens e ao ambiente de desenvolvimento para atender aos requisitos da organização.</p>	<p><b>Exemplo 1:</b> Validar todas as entradas e validar e codificar adequadamente todas as saídas.</p> <p><b>Exemplo 2:</b> Evitar usar funções e chamadas inseguras.</p> <p><b>Exemplo 3:</b> Detectar erros e tratá-los com elegância.</p> <p><b>Exemplo 4:</b> Fornecer recursos de registro e rastreamento.</p> <p><b>Exemplo 5:</b> Usar ambientes de desenvolvimento com recursos automatizados que incentivem ou exijam o uso de práticas de codificação seguras com treinamento just-in-time no local.</p> <p><b>Exemplo 6:</b> Seguir os procedimentos para garantir manualmente a conformidade com as práticas de codificação segura quando os métodos automatizados forem insuficientes ou não estiverem disponíveis.</p> <p><b>Exemplo 7:</b> Usar ferramentas (por exemplo, linters, formatadores) para padronizar o estilo e a formatação do código-fonte.</p> <p><b>Exemplo 8:</b> Verificar se há outras vulnerabilidades que são comuns às linguagens e ao ambiente de desenvolvimento.</p> <p><b>Exemplo 9:</b> Fazer com que o desenvolvedor revise seu próprio código legível por humanos para complementar (e não substituir) a revisão de código realizada por outras pessoas ou ferramentas. Consulte <a href="#">PW.7</a>.</p>	<p><b>BSAFSS:</b> SC.2, SC.3, LO.1, EE.1</p> <p><b>BSIMM:</b> SR3.3, CR1.4, CR3.5</p> <p><b>EO14028:</b> 4e(iv), 4e(ix)</p> <p><b>IDASOAR:</b> 2</p> <p><b>IEC62443:</b> SI-1, SI-2</p> <p><b>ISO27034:</b> 7.3.5</p> <p><b>MSSDL:</b> 9</p> <p><b>OWASPASVS:</b> 1.1.7, 1.5, 1.7, 5, 7</p> <p><b>OWASPMASVS:</b> 7.6</p> <p><b>SCFPSSD:</b> Estabelecer requisitos de registro e práticas de auditoria, usar ferramentas de análise de código para encontrar problemas de segurança antecipadamente, tratar os dados com segurança, tratar os erros, usar somente funções seguras</p> <p><b>SP800181:</b> SP-DEV-001; T0013, T0077, T0176; K0009, K0016, K0039, K0070, K0140, K0624; S0019, S0060, S0149, S0172, S0266; A0036, A0047</p>
	<p><b>PW.5.2:</b> <i>Movido para o PW.5.1 como exemplo</i></p>		

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
<p><b>Configurar os Processos de Compilação, Interpretação e Construção para Melhorar a Segurança do Executável (PW.6):</b> Diminuir o número de vulnerabilidades de segurança no software e reduzir os custos, eliminando as vulnerabilidades antes da realização dos testes.</p>	<p><b>PW.6.1:</b> Usar ferramentas de compilação, interpretação e construção que ofereçam recursos para melhorar a segurança do executável.</p>	<p><b>Exemplo 1:</b> Usar versões atualizadas das ferramentas de compilação, interpretação e construção.</p> <p><b>Exemplo 2:</b> Seguir os processos de gerenciamento de alterações ao implantar ou atualizar ferramentas de compilação, interpretação e construção e auditar todas as alterações inesperadas nas ferramentas.</p> <p><b>Exemplo 3:</b> Validar regularmente a autenticidade e a integridade das ferramentas de compilação, interpretação e construção. Veja <a href="#">PO.3</a>.</p>	<p><b>BSAFSS:</b> DE.2-1</p> <p><b>BSIMM:</b> SE2.4</p> <p><b>CNCFSSCP:</b> Proteção de Pipelines de Compilação—Verificação, Automação</p> <p><b>EO14028:</b> 4e(iv), 4e(ix)</p> <p><b>IEC62443:</b> SI-2</p> <p><b>MSSDL:</b> 8</p> <p><b>SCAGILE:</b> Tarefa de Segurança Operacional 3</p> <p><b>SCFPSSD:</b> Usar as versões atuais do compilador e da cadeia de ferramentas e opções seguras do compilador</p> <p><b>SCSIC:</b> Controles de Integridade do Desenvolvimento de Software do Fornecedor</p> <p><b>SP80053:</b> SA-15</p> <p><b>SP800161:</b> SA-15</p>
	<p><b>PW.6.2:</b> Determinar quais recursos das ferramentas de compilação, interpretação e construção devem ser usados e como cada um deve ser configurado e, em seguida, implementar e usar as configurações aprovadas.</p>	<p><b>Exemplo 1:</b> Ativar os recursos do compilador que geram avisos sobre códigos mal protegidos durante o processo de compilação.</p> <p><b>Exemplo 2:</b> Implementar o conceito de "compilação limpa", em que todos os avisos do compilador são tratados como erros e eliminados, exceto aqueles determinados como falsos positivos ou irrelevantes.</p> <p><b>Exemplo 3:</b> Realizar todas as compilações em um ambiente de compilação dedicado e altamente controlado.</p> <p><b>Exemplo 4:</b> Habilitar recursos do compilador que randomizem ou ofusquem características de execução, como o uso de locais de memória, que, de outra forma, seriam previsíveis e, portanto, potencialmente exploráveis.</p> <p><b>Exemplo 5:</b> Testar para garantir que os recursos estejam funcionando conforme o esperado e não estejam causando inadvertidamente problemas operacionais ou outros problemas.</p> <p><b>Exemplo 6:</b> Verificar continuamente se as configurações aprovadas estão sendo usadas.</p> <p><b>Exemplo 7:</b> Disponibilizar as configurações de ferramentas aprovadas como configuração como código para que os desenvolvedores possam usá-las prontamente.</p>	<p><b>BSAFSS:</b> DE.2-3, DE.2-4, DE.2-5</p> <p><b>BSIMM:</b> SE2.4, SE3.2</p> <p><b>CNCFSSCP:</b> Proteção de Pipelines de Compilação—Verificação, Automação</p> <p><b>EO14028:</b> 4e(iv), 4e(ix)</p> <p><b>IEC62443:</b> SI-2</p> <p><b>IR8397:</b> 2.5</p> <p><b>MSSDL:</b> 8</p> <p><b>OWASPASVS:</b> 14.1, 14.2.1</p> <p><b>OWASPMASVS:</b> 7.2</p> <p><b>PCISSLC:</b> 3.2</p> <p><b>SCAGILE:</b> Tarefa de Segurança Operacional 8</p> <p><b>SCFPSSD:</b> Usar as versões atuais do compilador e da cadeia de ferramentas e opções seguras do compilador</p> <p><b>SCSIC:</b> Controles de Integridade do Desenvolvimento de Software do Fornecedor</p> <p><b>SP80053:</b> SA-15, SR-9</p> <p><b>SP800161:</b> SA-15, SR-9</p> <p><b>SP800181:</b> K0039, K0070</p>
<p><b>Revisão e/ou Análise de Código Legível por Humanos para Identificar Vulnerabilidades e Verificar a Conformidade com os Requisitos de Segurança (PW.7):</b> Ajudar a identificar vulnerabilidades para que possam ser corrigidas antes do lançamento do software para evitar a exploração. Usar métodos automatizados reduz o esforço e os recursos necessários para detectar vulnerabilidades. O código legível por humanos inclui código-fonte, scripts e qualquer outra forma de código que uma organização considere legível por humanos.</p>	<p><b>PW.7.1:</b> Determinar se a <i>revisão de código</i> (uma pessoa olha diretamente para o código para encontrar problemas) e/ou a <i>análise de código</i> (ferramentas são usadas para encontrar problemas no código, seja de forma totalmente automatizada ou em conjunto com uma pessoa) deve ser usada, conforme definido pela organização.</p>	<p><b>Exemplo 1:</b> Seguir as políticas ou diretrizes da organização para saber quando a revisão de código deve ser realizada e como ela deve ser conduzida. Isso pode incluir código de terceiros e módulos de código reutilizáveis escritos internamente.</p> <p><b>Exemplo 2:</b> Seguir as políticas ou diretrizes da organização para saber quando a análise de código deve ser realizada e como ela deve ser conduzida.</p> <p><b>Exemplo 3:</b> Escolher métodos de revisão e/ou analisar código com base no estágio do software.</p>	<p><b>BSIMM:</b> CR1.5</p> <p><b>EO14028:</b> 4e(iv), 4e(ix)</p> <p><b>IEC62443:</b> SM-5, SI-1, SVV-1</p> <p><b>NISTLABEL:</b> 2.2.2.2</p> <p><b>SCSIC:</b> Revisões Por Pares e Testes de Segurança</p> <p><b>SP80053:</b> SA-11</p> <p><b>SP800161:</b> SA-11</p> <p><b>SP800181:</b> SP-DEV-002; K0013, K0039, K0070, K0153, K0165; S0174</p>
	<p><b>PW.7.2:</b> Realizar a revisão e/ou a análise do código com base nos padrões de codificação segura da organização e registrar e fazer a triagem de todos os problemas descobertos e das correções recomendadas no fluxo de trabalho da equipe de desenvolvimento ou no sistema de rastreamento de problemas.</p>	<p><b>Exemplo 1:</b> Realizar revisão por pares do código e revisar qualquer revisão de código existente, análise ou resultados de testes como parte da revisão por pares.</p> <p><b>Exemplo 2:</b> Usar revisores especializados para verificar se há backdoors e outros conteúdos maliciosos no código.</p> <p><b>Exemplo 3:</b> Usar ferramentas de revisão por pares que facilitem o processo de revisão por pares e documentar todas as discussões e outros comentários.</p> <p><b>Exemplo 4:</b> Usar uma ferramenta de análise estática para verificar automaticamente se há vulnerabilidades no código e se ele está em conformidade com os padrões de codificação segura da organização, com uma pessoa revisando os problemas relatados pela ferramenta e corrigindo-os conforme necessário.</p> <p><b>Exemplo 5:</b> Usar listas de verificação de revisão para verificar se o código está</p>	<p><b>BSAFSS:</b> TV.2, PD.1-4</p> <p><b>BSIMM:</b> CR1.2, CR1.4, CR1.6, CR2.6, CR2.7, CR3.4, CR3.5</p> <p><b>EO14028:</b> 4e(iv), 4e(v), 4e(ix)</p> <p><b>IDASOAR:</b> 3, 4, 5, 14, 15, 48</p> <p><b>IEC62443:</b> SI-1, SVV-1, SVV-2</p> <p><b>IR8397:</b> 2.3, 2.4</p> <p><b>ISO27034:</b> 7.3.6</p> <p><b>MSSDL:</b> 9, 10</p> <p><b>NISTLABEL:</b> 2.2.2.2</p> <p><b>OWASPASVS:</b> 1.1.7, 10</p> <p><b>OWASPMASVS:</b> 7.5</p> <p><b>OWASPSAMM:</b> IR1-B, IR2-A, IR2-B, IR3-A</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
		<p>em conformidade com os requisitos.</p> <p><b>Exemplo 6:</b> Usar ferramentas automatizadas para identificar e corrigir práticas de software inseguras documentadas e verificadas de forma contínua, à medida que o código legível por humanos é verificado no repositório de códigos.</p> <p><b>Exemplo 7:</b> Identificar e documentar as causas básicas dos problemas descobertos.</p> <p><b>Exemplo 8:</b> Documentar as lições aprendidas com a revisão e análise do código em um wiki que os desenvolvedores possam acessar e pesquisar.</p>	<p><b>PCISSLC:</b> 3.2, 4.1</p> <p><b>SCAGILE:</b> Tarefas de Segurança Operacional 4, 7; Tarefas que Exigem a Ajuda de Especialistas em Segurança 10</p> <p><b>SCFPSSD:</b> Usar ferramentas de análise de código para encontrar problemas de segurança antecipadamente, usar ferramentas de teste de segurança de análise estática, realizar verificação manual de recursos/mitigações de segurança</p> <p><b>SCSIC:</b> Revisões Por Pares e Testes de Segurança</p> <p><b>SP80053:</b> SA-11, SA-11(1), SA-11(4), SA-15(7)</p> <p><b>SP800161:</b> SA-11, SA-11(1), SA-11(4), SA-15(7)</p> <p><b>SP800181:</b> SP-DEV-001, SP-DEV-002; T0013, T0111, T0176, T0267, T0516; K0009, K0039, K0070, K0140, K0624; S0019, S0060, S0078, S0137, S0149, S0167, S0174, S0242, S0266; A0007, A0015, A0036, A0044, A0047</p>
<p><b>Testar o Código Executável para Identificar Vulnerabilidades e Verificar a Conformidade com os Requisitos de Segurança (PW.8):</b> Ajudar a identificar vulnerabilidades para que possam ser corrigidas antes do lançamento do software, a fim de evitar a exploração. O uso de métodos automatizados reduz o esforço e os recursos necessários para detectar vulnerabilidades e melhora a rastreabilidade e a repetibilidade. O código executável inclui binários, bytecode e código-fonte executados diretamente e qualquer outra forma de código que uma organização considere executável.</p>	<p><b>PW.8.1:</b> Determinar se o teste de código executável deve ser realizado para encontrar vulnerabilidades não identificadas por revisões, análises ou testes anteriores e, em caso afirmativo, quais tipos de teste devem ser usados.</p>	<p><b>Exemplo 1:</b> Seguir as políticas ou diretrizes da organização para saber quando o teste de código deve ser realizado e como ele deve ser conduzido (por exemplo, em um ambiente de sandbox). Isso pode incluir código executável de terceiros e módulos de código executável reutilizáveis escritos internamente.</p> <p><b>Exemplo 2:</b> Escolher métodos de teste com base no estágio do software.</p>	<p><b>BSAFSS:</b> TV.3</p> <p><b>BSIMM:</b> PT2.3</p> <p><b>EO14028:</b> 4e(ix)</p> <p><b>IEC62443:</b> SVV-1, SVV-2, SVV-3, SVV-4, SVV-5</p> <p><b>NISTLABEL:</b> 2.2.2.2</p> <p><b>SCSIC:</b> Revisões Por Pares e Testes de Segurança</p> <p><b>SP80053:</b> SA-11</p> <p><b>SP800161:</b> SA-11</p> <p><b>SP800181:</b> SP-DEV-001, SP-DEV-002; T0456; K0013, K0039, K0070, K0153, K0165, K0342, K0367, K0536, K0624; S0001, S0015, S0026, S0061, S0083, S0112, S0135</p>
	<p><b>PW.8.2:</b> Escopo dos testes, projeto dos testes, execução dos testes e documentação dos resultados, incluindo o registro e a triagem de todos os problemas descobertos e correções recomendadas no fluxo de trabalho da equipe de desenvolvimento ou no sistema de rastreamento de problemas.</p>	<p><b>Exemplo 1:</b> Realizar testes funcionais robustos dos recursos de segurança.</p> <p><b>Exemplo 2:</b> Integrar o teste de vulnerabilidade dinâmica ao conjunto de testes automatizados do projeto.</p> <p><b>Exemplo 3:</b> Incorporar testes para vulnerabilidades relatadas anteriormente no conjunto de testes do projeto para garantir que os erros não sejam reintroduzidos.</p> <p><b>Exemplo 4:</b> Levantar em consideração as infraestruturas e as stacks de tecnologia com as quais o software será usado na produção ao desenvolver planos de teste.</p> <p><b>Exemplo 5:</b> Usar ferramentas de teste de fuzz para encontrar problemas com o tratamento de entrada.</p> <p><b>Exemplo 6:</b> Se houver recursos disponíveis, usar testes de penetração para simular como um invasor pode tentar comprometer o software em cenários de alto risco.</p> <p><b>Exemplo 7:</b> Identificar e registrar as causas básicas dos problemas descobertos.</p> <p><b>Exemplo 8:</b> Documentar as lições aprendidas com os testes de código em um wiki que os desenvolvedores possam acessar e pesquisar.</p> <p><b>Exemplo 9:</b> Usar código-fonte, registros de projeto e outros recursos ao desenvolver planos de teste.</p>	<p><b>BSAFSS:</b> TV.3, TV.5, PD.1-4</p> <p><b>BSIMM:</b> ST1.1, ST1.3, ST1.4, ST2.4, ST2.5, ST2.6, ST3.3, ST3.4, ST3.5, ST3.6, PT1.1, PT1.2, PT1.3, PT3.1</p> <p><b>EO14028:</b> 4e(iv), 4e(v), 4e(ix)</p> <p><b>IDASOAR:</b> 7, 8, 10, 11, 38, 39, 43, 44, 48, 55, 56, 57</p> <p><b>IEC62443:</b> SM-5, SM-13, SI-1, SVV-1, SVV-2, SVV-3, SVV-4, SVV-5</p> <p><b>IR8397:</b> 2.6, 2.7, 2.8, 2.9, 2.10, 2.11</p> <p><b>ISO27034:</b> 7.3.6</p> <p><b>MSSDL:</b> 10, 11</p> <p><b>NISTLABEL:</b> 2.2.2.2</p> <p><b>OWASPMASVS:</b> 7.5</p> <p><b>OWASPSAMM:</b> ST1-A, ST1-B, ST2-A, ST2-B, ST3-A</p> <p><b>PCISSLC:</b> 4.1</p> <p><b>SCAGILE:</b> Tarefas de Segurança Operacional 10, 11; Tarefas que Exigem a Ajuda de Especialistas em Segurança 4, 5, 6, 7</p> <p><b>SCFPSSD:</b> Realizar testes de segurança de análise dinâmica, analisadores de fuzz, varredura de vulnerabilidade de rede, realizar testes funcionais automatizados de recursos/mitigações de segurança, realizar testes de penetração</p> <p><b>SCSIC:</b> Revisões Por Pares e Testes de Segurança</p> <p><b>SP80053:</b> SA-11, SA-11(5), SA-11(8), SA-15(7)</p> <p><b>SP800161:</b> SA-11, SA-11(5), SA-11(8), SA-15(7)</p> <p><b>SP800181:</b> SP-DEV-001, SP-DEV-002; T0013, T0028, T0169, T0176, T0253, T0266, T0456, T0516; K0009, K0039, K0070, K0272, K0339, K0342, K0362, K0536, K0624; S0001, S0015, S0046, S0051, S0078, S0081, S0083, S0135, S0137, S0167, S0242; A0015</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
<p><b>Configurar o Software para Ter Configurações Seguras Por Padrão (PW.9):</b> Ajudar a melhorar a segurança do software no momento da instalação para reduzir a probabilidade de o software ser implantado com configurações de segurança fracas, colocando-o em maior risco de comprometimento.</p>	<p><b>PW.9.1:</b> Definir uma linha de base segura determinando como configurar cada definição que tenha efeito sobre a segurança ou uma definição relacionada à segurança, de modo que as configurações padrão sejam seguras e não enfraqueçam as funções de segurança fornecidas pela plataforma, pela infraestrutura de rede ou pelos serviços.</p>	<p><b>Exemplo 1:</b> Realizar testes para garantir que as configurações, incluindo as configurações padrão, estejam funcionando conforme o esperado e não estejam causando inadvertidamente nenhuma falha de segurança, problemas operacionais ou outros problemas.</p>	<p><b>BSAFSS:</b> CF.1  <b>BSIMM:</b> SE2.2  <b>EO14028:</b> 4e(iv), 4e(ix)  <b>IDASOAR:</b> 23  <b>IEC62443:</b> SD-4, SVV-1, SG-1  <b>ISO27034:</b> 7.3.5  <b>SCAGILE:</b> Tarefas que Exigem a Ajuda de Especialistas em Segurança 12  <b>SCSIC:</b> Controles de Integridade de Entrega de Software do Fornecedor, Controles de Integridade de Desenvolvimento de Software do Fornecedor  <b>SP800181:</b> SP-DEV-002; K0009, K0039, K0073, K0153, K0165, K0275, K0531; S0167</p>
	<p><b>PW.9.2:</b> Implementar as configurações padrão (ou grupos de configurações padrão, se aplicável) e documentar cada configuração para os administradores de software.</p>	<p><b>Exemplo 1:</b> Verificar se a configuração aprovada está em vigor para o software.  <b>Exemplo 2:</b> Documentar a finalidade de cada configuração, as opções, o valor padrão, a relevância da segurança, o possível impacto operacional e o relacionamento com outras configurações.  <b>Exemplo 3:</b> Usar mecanismos técnicos programáticos autorizados para registrar como cada configuração pode ser implementada e avaliada pelos administradores de software.  <b>Exemplo 4:</b> Armazenar a configuração padrão em um formato utilizável e seguir as práticas de controle de alterações para modificá-la (por exemplo, configuração como código).</p>	<p><b>BSAFSS:</b> CF.1  <b>BSIMM:</b> SE2.2  <b>EO14028:</b> 4e(iv), 4e(ix)  <b>IDASOAR:</b> 23  <b>IEC62443:</b> SG-3  <b>OWASPSAMM:</b> OE1-A  <b>PCISL:</b> 8.1, 8.2  <b>SCAGILE:</b> Tarefas que Exigem a Ajuda de Especialistas em Segurança 12  <b>SCFPSSD:</b> Verificar as Configurações Seguras e o Uso da Mitigação da Plataforma  <b>SCSIC:</b> Controles de Integridade de Entrega de Software do Fornecedor, Controles de Integridade de Desenvolvimento de Software do Fornecedor  <b>SP80053:</b> SA-5, SA-8(23)  <b>SP800161:</b> SA-5, SA-8(23)  <b>SP800181:</b> SP-DEV-001; K0009, K0039, K0073, K0153, K0165, K0275, K0531</p>
<b>Responder a vulnerabilidades (RV)</b>			
<p><b>Identificar e confirmar vulnerabilidades em uma base contínua (RV.1):</b> Ajudar a garantir que as vulnerabilidades sejam identificadas mais rapidamente para que possam ser corrigidas mais rapidamente de acordo com o risco, reduzindo a janela de oportunidade para os invasores.</p>	<p><b>RV.1.1:</b> Reunir informações de adquirentes de software, usuários e fontes públicas sobre possíveis vulnerabilidades no software e em componentes de terceiros que o software usa e investigar todos os relatórios confiáveis.</p>	<p><b>Exemplo 1:</b> Monitorar os bancos de dados de vulnerabilidade<sup>9</sup>, as listas de discussão de segurança e outras fontes de relatórios de vulnerabilidade por meios manuais ou automatizados.  <b>Exemplo 2:</b> Usar fontes de inteligência contra ameaças para entender melhor como as vulnerabilidades em geral estão sendo exploradas.  <b>Exemplo 3:</b> Analisar automaticamente os dados de proveniência e composição de software de todos os componentes de software para identificar quaisquer novas vulnerabilidades que eles tenham.</p>	<p><b>BSAFSS:</b> VM.1-3, VM.3  <b>BSIMM:</b> AM1.5, CMVM1.2, CMVM2.1, CMVM3.4, CMVM3.7  <b>CNCFSSCP:</b> Proteção de Materiais—Verificação  <b>EO14028:</b> 4e(iv), 4e(vi), 4e(viii), 4e(ix)  <b>IEC62443:</b> DM-1, DM-2, DM-3  <a href="#">ISO29147:</a> 6.2.1, 6.2.2, 6.2.4, 6.3, 6.5  <a href="#">ISO30111:</a> 7.1.3  <b>OWASPSAMM:</b> IM1-A, IM2-B, EH1-B  <b>OWASPSCVS:</b> 4  <b>PCISL:</b> 3.4, 4.1, 9.1  <b>SCAGILE:</b> Tarefa de Segurança Operacional 5  <b>SCFPSSD:</b> Resposta e divulgação de vulnerabilidades  <b>SCTPC:</b> MONITOR1  <b>SP80053:</b> SA-10, SR-3, SR-4  <b>SP800161:</b> SA-10, SR-3, SR-4  <b>SP800181:</b> K0009, K0038, K0040, K0070, K0161, K0362; S0078</p>
	<p><b>RV.1.2:</b> Revisar, analisar e/ou testar o código do software para identificar ou confirmar a presença de vulnerabilidades não detectadas anteriormente.</p>	<p><b>Exemplo 1:</b> Configurar o conjunto de ferramentas para realizar análises e testes automatizados de código de forma regular ou contínua para todas as versões compatíveis.  <b>Exemplo 2:</b> Consultar <a href="#">PW.7</a> e <a href="#">PW.8</a>.</p>	<p><b>BSAFSS:</b> VM.1-2, VM.2-1  <b>BSIMM:</b> CMVM3.1  <b>EO14028:</b> 4e(iv), 4e(vi), 4e(viii), 4e(ix)  <b>IEC62443:</b> SI-1, SVV-2, SVV-3, SVV-4, DM-1, DM-2  <b>ISO27034:</b> 7.3.6</p>

<sup>9</sup> Um exemplo é o National Vulnerability Database (NVD) (<https://nvd.nist.gov/>).

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
	<p><b>RV.1.3:</b> Ter uma política que trate da divulgação e correção de vulnerabilidades e implementar as funções, responsabilidades e processos necessários para apoiar essa política.</p>	<p><b>Exemplo 1:</b> Estabelecer um programa de divulgação de vulnerabilidades e facilitar para os pesquisadores de segurança conhecerem seu programa e relatarem possíveis vulnerabilidades.</p> <p><b>Exemplo 2:</b> Ter uma Equipe de Resposta a Incidentes de Segurança de Produtos (PSIRT) e processos para lidar com as respostas a relatórios e incidentes de vulnerabilidade, incluindo planos de comunicação para todas as partes interessadas.</p> <p><b>Exemplo 3:</b> Ter um manual de resposta de segurança para lidar com uma vulnerabilidade genérica relatada, um relatório de dias zero, uma vulnerabilidade que está sendo explorada na natureza e um grande incidente em andamento envolvendo várias partes e componentes de software de código aberto.</p> <p><b>Exemplo 4:</b> Realizar periodicamente exercícios dos processos de resposta a incidentes de segurança do produto.</p>	<p><b>ISO29147:</b> 6.4  <b>ISO30111:</b> 7.1.4  <b>PCISSLC:</b> 3.4, 4.1  <b>SCAGILE:</b> Tarefas de Segurança Operacional 10, 11  <b>SP80053:</b> SA-11  <b>SP800161:</b> SA-11  <b>SP800181:</b> SP-DEV-002; K0009, K0039, K0153</p> <p><b>BSAFSS:</b> VM.1-1, VM.2  <b>BSIMM:</b> CMVM1.1, CMVM2.1, CMVM3.3, CMVM3.7  <b>EO14028:</b> 4e(viii), 4e(ix)  <b>IEC62443:</b> DM-1, DM-2, DM-3, DM-4, DM-5  <b>ISO29147:</b> Todos  <b>ISO30111:</b> Todos  <b>MSSDL:</b> 12  <b>NISTLABEL:</b> 2.2.2.3  <b>OWASPMASVS:</b> 1.11  <b>OWASPSAMM:</b> IM1-A, IM1-B, IM2-A, IM2-B  <b>PCISSLC:</b> 9.2, 9.3  <b>SCFPSSD:</b> Resposta e Divulgação de Vulnerabilidades  <b>SP80053:</b> SA-15(10)  <b>SP800160:</b> 3.3.8  <b>SP800161:</b> SA-15(10)  <b>SP800181:</b> K0041, K0042, K0151, K0292, K0317; S0054; A0025  <b>SP800216:</b> Tudo</p>
<p><b>Avaliar, Priorizar e Corrigir as Vulnerabilidades (RV.2):</b> Ajudar a garantir que as vulnerabilidades sejam corrigidas de acordo com o risco para reduzir a janela de oportunidade para os invasores.</p>	<p><b>RV.2.1:</b> Analisar cada vulnerabilidade para reunir informações suficientes sobre o risco e planejar sua correção ou outra resposta ao risco.</p>	<p><b>Exemplo 1:</b> Usar o software de rastreamento de problemas existente para registrar cada vulnerabilidade.</p> <p><b>Exemplo 2:</b> Realizar cálculos de risco para cada vulnerabilidade com base em estimativas de sua capacidade de exploração, o impacto potencial se for explorado e quaisquer outras características relevantes.</p>	<p><b>BSAFSS:</b> VM.2  <b>BSIMM:</b> CMVM1.2, CMVM2.2  <b>EO14028:</b> 4e(iv), 4e(viii), 4e(ix)  <b>IEC62443:</b> DM-2, DM-3  <b>ISO30111:</b> 7.1.4  <b>NISTLABEL:</b> 2.2.2.2  <b>PCISSLC:</b> 3.4, 4.2  <b>SCAGILE:</b> Tarefa de Segurança Operacional 1, Tarefas que Exigem a Ajuda de Especialistas em Segurança 10  <b>SP80053:</b> SA-10, SA-15(7)  <b>SP800160:</b> 3.3.8  <b>SP800161:</b> SA-15(7)  <b>SP800181:</b> K0009, K0039, K0070, K0161, K0165; S0078</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
	<p><b>RV.2.2:</b> Planejar e implementar respostas de risco para vulnerabilidades.</p>	<p><b>Exemplo 1:</b> Tomar uma decisão baseada em riscos sobre se cada vulnerabilidade será corrigida ou se o risco será tratado por outros meios (por exemplo, aceitação de riscos, transferência de riscos) e priorizar as ações a serem tomadas.</p> <p><b>Exemplo 2:</b> Se uma atenuação permanente para uma vulnerabilidade ainda não estiver disponível, determinar como a vulnerabilidade pode ser atenuada temporariamente até que a solução permanente esteja disponível e adicionar essa correção temporária ao plano.</p> <p><b>Exemplo 3:</b> Desenvolver e liberar avisos de segurança que forneçam as informações necessárias aos adquirentes de software, incluindo descrições de quais são as vulnerabilidades, como encontrar instâncias do software vulnerável e como resolvê-las (por exemplo, onde obter patches e o que os patches alteram no software; quais definições de configuração podem precisar ser alteradas; como soluções temporárias podem ser implementadas).</p> <p><b>Exemplo 4:</b> Entregar correções aos adquirentes por meio de um mecanismo de entrega automatizado e confiável. Uma única correção pode abordar várias vulnerabilidades.</p> <p><b>Exemplo 5:</b> Atualizar os registros de decisões de projeto, respostas a riscos e exceções aprovadas, conforme necessário. Veja <a href="#">PW.1.2</a>.</p>	<p><b>BSAFSS:</b> VM.1-1, VM-2  <b>BSIMM:</b> CMVM2.1  <b>EO14028:</b> 4e(iv), 4e(vi), 4e(viii), 4e(ix)  <b>IEC62443:</b> DM-4  <b>ISO30111:</b> 7.1.4, 7.1.5  <b>NISTLABEL:</b> 2.2.2.2  <b>PCISSLC:</b> 4.1, 4.2, 10.1  <b>SCAGILE:</b> Tarefa de Segurança Operacional 2  <b>SCFPSSD:</b> Corrigir a Vulnerabilidade, Identificar Fatores Mitigantes ou Soluções Alternativas  <b>SCTPC:</b> MITIGAR  <b>SP80053:</b> SA-5, SA-10, SA-11, SA-15(7)  <b>SP800160:</b> 3.3.8  <b>SP800161:</b> SA-5, SA-8, SA-10, SA-11, SA-15(7)  <b>SP800181:</b> T0163, T0229, T0264; K0009, K0070</p>
<p><b>Analisar as Vulnerabilidades para Identificar Suas Causas Principais (RV.3):</b> Ajudar a reduzir a frequência das vulnerabilidades no futuro.</p>	<p><b>RV.3.1:</b> Analisar as vulnerabilidades identificadas para determinar suas causas principais.</p>	<p><b>Exemplo 1:</b> Registrar a causa raiz dos problemas descobertos.</p> <p><b>Exemplo 2:</b> Registrar as lições aprendidas por meio da análise de causa raiz em um wiki que os desenvolvedores possam acessar e pesquisar.</p>	<p><b>BSAFSS:</b> VM.2-1  <b>BSIMM:</b> CMVM3.1, CMVM3.2  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> DM-3  <b>ISO30111:</b> 7.1.4  <b>OWASPSAMM:</b> IM3-A  <b>PCISSLC:</b> 4.2  <b>SCFPSSD:</b> Feedback do ciclo de vida do desenvolvimento seguro  <b>SP800181:</b> T0047, K0009, K0039, K0070, K0343</p>
	<p><b>RV.3.2:</b> Analisar as causas-raiz ao longo do tempo para identificar padrões, como uma determinada prática de codificação segura que não está sendo seguida de forma consistente.</p>	<p><b>Exemplo 1:</b> Registrar as lições aprendidas por meio da análise de causa raiz em um wiki que os desenvolvedores possam acessar e pesquisar.</p> <p><b>Exemplo 2:</b> Adicionar mecanismos à cadeia de ferramentas para detectar automaticamente instâncias futuras da causa raiz.</p> <p><b>Exemplo 3:</b> Atualizar os processos manuais para detectar instâncias futuras da causa raiz.</p>	<p><b>BSAFSS:</b> VM.2-1, PD.1-3  <b>BSIMM:</b> CP3.3, CMVM3.2  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> DM-4  <b>ISO30111:</b> 7.1.7  <b>OWASPSAMM:</b> IM3-B  <b>PCISSLC:</b> 2.6, 4.2  <b>SCFPSSD:</b> Feedback do ciclo de vida do desenvolvimento seguro  <b>SP800160:</b> 3.3.8  <b>SP800181:</b> T0111, K0009, K0039, K0070, K0343</p>
	<p><b>RV.3.3:</b> Analisar o software em busca de vulnerabilidades semelhantes para erradicar uma classe de vulnerabilidades e corrigi-las proativamente em vez de esperar por relatórios externos.</p>	<p><b>Exemplo 1:</b> Consulte <a href="#">PW.7</a> e <a href="#">PW.8</a>.</p>	<p><b>BSAFSS:</b> VM.2  <b>BSIMM:</b> CR3.3, CMVM3.1  <b>EO14028:</b> 4e(iv), 4e(viii), 4e(ix)  <b>IEC62443:</b> SI-1, DM-3, DM-4  <b>ISO30111:</b> 7.1.4  <b>PCISSLC:</b> 4.2  <b>SP80053:</b> SA-11  <b>SP800161:</b> SA-11  <b>SP800181:</b> SP-DEV-001, SP-DEV-002; K0009, K0039, K0070</p>
	<p><b>RV.3.4:</b> Revisar o processo de SDLC e atualizá-lo, se for o caso, para evitar (ou reduzir a probabilidade de) que a causa raiz se repita nas atualizações do software ou no novo software que for criado.</p>	<p><b>Exemplo 1:</b> Registrar as lições aprendidas por meio da análise de causa raiz em um wiki que os desenvolvedores possam acessar e pesquisar.</p> <p><b>Exemplo 2:</b> Planejar e implementar mudanças nas práticas apropriadas do SDLC.</p>	<p><b>BSAFSS:</b> PD.1-3  <b>BSIMM:</b> CP3.3, CMVM3.2  <b>EO14028:</b> 4e(ix)  <b>IEC62443:</b> DM-6</p>

Práticas	Tarefas	Exemplos de implementação ilustrativos	Referências
			<p><b>ISO30111:</b> 7.1.7  <b>MSSDL:</b> 2  <b>PCISL:</b> 2.,6, 4.2  <b>SCFPSSD:</b> Feedback do Ciclo de Vida do Desenvolvimento Seguro  <b>SP80053:</b> SA-15  <b>SP800161:</b> SA-15  <b>SP800181:</b> K0009, K0039, K0070</p>

**Referências**

- [BSAFSS] BSA (2020) *A estrutura da BSA para software seguro: A New Approach to Securing the Software Lifecycle (Uma nova abordagem para proteger o ciclo de vida do software)*, versão 1.1. Disponível em [https://www.bsa.org/files/reports/bsa\\_framework\\_secure\\_software\\_update\\_2020.pdf](https://www.bsa.org/files/reports/bsa_framework_secure_software_update_2020.pdf)
- [BSIMM] Miguez S, Erlikhman E, Ewers J, Nassery K (2021) *BSIMM12 2021 Foundations Report*. Disponível em <https://www.bsimm.com/content/dam/bsimm/reports/bsimm12-foundations.pdf>
- [CNCFSSCP] Cloud Native Computing Foundation (2021) *Software Supply Chain Best Practices (Práticas recomendadas da cadeia de suprimentos de software)*. Disponível em <https://github.com/cncf/tag-security/tree/main/supply-chain-security/supply-chain-security-paper>
- [EO14028] Ordem Executiva 14028 (2021) Melhorando a segurança cibernética do país. (Casa Branca, Washington, DC), DCPD-202100401, 12 de maio de 2021. <https://www.govinfo.gov/app/details/DCPD-202100401>
- [IDASOAR] Hong Fong EK, Wheeler D, Henninger A (2016) State-of-the-Art Resources (SOAR) for Software Vulnerability Detection, Test, and Evaluation 2016. (Institute for Defense Analyses [IDA], Alexandria, VA), IDA Paper P-8005. Disponível em <https://www.ida.org/research-and-publications/publications/all/s/st/stateoftheart-resources-soar-for-software-vulnerability-detection-test-and-evaluation-2016>
- [IEC62443] International Electrotechnical Commission (IEC), Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements (Requisitos do ciclo de vida de desenvolvimento de produtos seguros), IEC 62443-4-1, 2018. Disponível em <https://webstore.iec.ch/publication/33615>
- [IR7692] Waltermire DA, Scarfone KA, Casipe M (2011) Specification for the Open Checklist Interactive Language (Especificação para a Linguagem Interativa de Lista de Verificação Aberta), (OCIL) Versão 2.0. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), NIST Interagency ou Internal Report (IR) 7692. <https://doi.org/10.6028/NIST.IR.7692>
- [IR7864] LeMay E, Scarfone KA, Mell PM (2012) The Common Misuse Scoring System (O Sistema de Pontuação de Uso Indevido Comum), (CMSS): Métricas para vulnerabilidades de uso indevido de recursos de software. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), NIST Interagency ou Internal Report (IR) 7864. <https://doi.org/10.6028/NIST.IR.7864>

- [IR8397] Black P, Guttman B, Okun V (2021) Guidelines on Minimum Standards for Developer Verification of Software. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), NIST Interagency ou Internal Report (IR) 8397. <https://doi.org/10.6028/NIST.IR.8397>
- [ISO27034] International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), Information technology – Security techniques - Application security – Part 1: Overview and concepts, ISO/IEC 27034-1:2011, 2011. Disponível em <https://www.iso.org/standard/44378.html>
- [ISO29147] Organização Internacional de Normalização (ISO)/Comissão Eletrotécnica Internacional (IEC), Tecnologia da informação – Técnicas de segurança – Divulgação de vulnerabilidades, ISO/IEC 29147:2018, 2018. Disponível em <https://www.iso.org/standard/72311.html>
- [ISO30111] Organização Internacional de Normalização (ISO)/Comissão Eletrotécnica Internacional (IEC), Tecnologia da informação – Técnicas de segurança – Processos de tratamento de vulnerabilidades, ISO/IEC 30111:2019, 2019. Disponível em <https://www.iso.org/standard/69725.html>
- [MSSDL] Microsoft (2021) *Security Development Lifecycle (Ciclo de vida de desenvolvimento de segurança)*. Disponível em <https://www.microsoft.com/en-us/securityengineering/sdl/>
- [NISTCSF] National Institute of Standards and Technology (2018) Framework for Improving Critical Infrastructure Cybersecurity, Versão 1.1. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD). <https://doi.org/10.6028/NIST.CSWP.04162018>
- [NISTLABEL] Ogata M, Haney J, Merkel W, Phelps A (2022) Recommended Criteria for Cybersecurity Labeling of Consumer Software. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD). Disponível em <https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity>
- [NTIASBOM] Administração Nacional de Telecomunicações e Informações (NTIA) (2021) *The Minimum Elements For a Software Bill of Materials (SBOM)*. Disponível em <https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom>
- [OWASPASVS] Open Web Application Security Project (2021) *OWASP Application Security Verification Standard 4.0.3. (Padrão de verificação de segurança de aplicativos OWASP 4.0.3.)* Disponível em <https://github.com/OWASP/ASVS>
- [OWASPMASVS] Open Web Application Security Project (2021) *OWASP Mobile Application Security Verification Standard, Version 1.4.2. (Padrão de verificação de segurança de aplicativos móveis OWASP, versão 1.4.2.)* Disponível em <https://github.com/OWASP/owasp-masvs/releases>

- [OWASPSAMM] Open Web Application Security Project (2017) *Software Assurance Maturity Model Version 1.5. (Modelo de maturidade de garantia de software versão 1.5.)* Disponível em [https://www.owasp.org/index.php/OWASP\\_SAMM\\_Project](https://www.owasp.org/index.php/OWASP_SAMM_Project)
- [OWASPSCVS] Open Web Application Security Project (2020) *OWASP Software Component Verification Standard, Version 1.0. (Padrão de verificação de componentes de software OWASP, versão 1.0.)* Disponível em <https://github.com/OWASP/Software-Component-Verification-Standard>
- [PCISSLC] Payment Card Industry (PCI) Security Standards Council (2021) *Secure Software Lifecycle (Secure SLC) Requirements and Assessment Procedures Version 1.1.* Disponível em [https://www.pcisecuritystandards.org/document\\_library?category=sware\\_sec#results](https://www.pcisecuritystandards.org/document_library?category=sware_sec#results)
- [SCAGILE] Software Assurance Forum for Excellence in Code (2012) *Practical Security Stories and Security Tasks for Agile Development Environments.* Disponível em [http://www.safecode.org/publication/SAFECode\\_Agile\\_Dev\\_Security0712.pdf](http://www.safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf)
- [SCFPSSD] Software Assurance Forum for Excellence in Code (2018) *Fundamental Practices for Secure Software Development: Essential Elements of a Secure Development Lifecycle Program (Elementos essenciais de um programa de ciclo de vida de desenvolvimento seguro), terceira edição.* Disponível em [https://safecode.org/wp-content/uploads/2018/03/SAFECode\\_Fundamental\\_Practices\\_for\\_Secure\\_Software\\_Development\\_March\\_2018.pdf](https://safecode.org/wp-content/uploads/2018/03/SAFECode_Fundamental_Practices_for_Secure_Software_Development_March_2018.pdf)
- [SCSIC] Software Assurance Forum for Excellence in Code (2010) *Software Integrity Controls: An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain (Uma abordagem baseada em garantia para minimizar riscos na cadeia de suprimentos de software).* Disponível em [http://www.safecode.org/publication/SAFECode\\_Software\\_Integrity\\_Controls0610.pdf](http://www.safecode.org/publication/SAFECode_Software_Integrity_Controls0610.pdf)
- [SCTPC] Software Assurance Forum for Excellence in Code (2017) *Managing Security Risks Inherent in the Use of Third-Party Components (Gerenciando riscos de segurança inerentes ao uso de componentes de terceiros).* Disponível em [https://www.safecode.org/wp-content/uploads/2017/05/SAFECode\\_TPC\\_Whitepaper.pdf](https://www.safecode.org/wp-content/uploads/2017/05/SAFECode_TPC_Whitepaper.pdf)
- [SCTTM] Software Assurance Forum for Excellence in Code (2017) *Tactical Threat Modeling.* Disponível em [https://www.safecode.org/wp-content/uploads/2017/05/SAFECode\\_TM\\_Whitepaper.pdf](https://www.safecode.org/wp-content/uploads/2017/05/SAFECode_TM_Whitepaper.pdf)

- [SP80053] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations (Controles de segurança e privacidade para sistemas de informação e organizações). (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), NIST Special Publication (SP) 800-53, Rev. 5. Inclui atualizações a partir de 10 de dezembro de 2020. <https://doi.org/10.6028/NIST.SP.800-53r5>
- [SP800160] Ross R, McEvelley M, Oren J (2016) Engenharia de segurança de sistemas: Considerações sobre uma abordagem multidisciplinar na engenharia de sistemas seguros confiáveis. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), Publicação Especial do NIST (SP) 800-160, Volume 1. Inclui atualizações a partir de 21 de março de 2018. <https://doi.org/10.6028/NIST.SP.800-160v1>
- [SP800161] Boyens J, Smith A, Bartol N, Winkler K, Holbrook A, Fallon M (2021) Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), Second Draft NIST Special Publication (SP) 800-161, Rev. 1. <https://doi.org/10.6028/NIST.SP.800-161r1-draft2>
- [SP800181] Newhouse W, Keith S, Scribner B, Witte G (2017) National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework. (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), Publicação Especial do NIST (SP) 800-181. <https://doi.org/10.6028/NIST.SP.800-181>
- [SP800216] Schaffer K, Mell P, Trinh H (2021) Recommendations for Federal Vulnerability Disclosure Guidelines (Recomendações para diretrizes federais de divulgação de vulnerabilidades). (Instituto Nacional de Padrões e Tecnologia, Gaithersburg, MD), Draft NIST Special Publication (SP) 800-216. <https://doi.org/10.6028/NIST.SP.800-216-draft>

## Apêndice A—A SSDF e a Ordem Executiva 14028

A Ordem Executiva do Presidente (EO) sobre "Melhoria da Segurança Cibernética da Nação (14028)", emitida em 12 de maio de 2021 [\[EO14028\]](#), encarregou vários órgãos - incluindo o NIST - de melhorar a segurança cibernética por meio de várias iniciativas relacionadas à segurança e à integridade da cadeia de suprimentos de software.

A seção 4 do EO instruiu o NIST a solicitar contribuições do setor privado, do meio acadêmico, de órgãos governamentais e outros, e a identificar padrões, ferramentas, práticas recomendadas e outras diretrizes existentes ou desenvolver novos padrões para aprimorar a segurança da cadeia de suprimentos de software. A Tabela 2 mapeia as subseções da Seção 4e do EO para as práticas e tarefas da SSDF que podem ajudar a abordar cada subseção como parte de uma abordagem baseada em riscos.

**Tabela 2: Práticas de SSDF Correspondentes às Subseções da EO 14028**

Subseção EO 14028	Práticas e tarefas da SSDF
<a href="#">4e(i)(A)</a>	PO.5.1
<a href="#">4e(i)(B)</a>	PO.5.1
<a href="#">4e(i)(C)</a>	PO.5.1, PO.5.2
<a href="#">4e(i)(D)</a>	PO.5.1
<a href="#">4e(i)(E)</a>	PO.5.2
<a href="#">4e(i)(F)</a>	PO.3.2, PO.3.3, PO.5.1, PO.5.2
<a href="#">4e(ii)</a>	PO.3.2, PO.3.3, PO.5.1, PO.5.2
<a href="#">4e(iii)</a>	PO.3.1, PO.3.2, PO.5.1, PO.5.2, PS.1.1, PS.2.1, PS.3.1, PW.4.1, PW.4.4
<a href="#">4e(iv)</a>	PO.4.1, PO.4.2, PS.1.1, PW.2.1, PW.4.4, PW.5.1, PW.6.1, PW.6.2, PW.7.1, PW.7.2, PW.8.2, PW.9.1, PW.9.2, RV.1.1, RV.1.2, RV.2.1, RV.2.2, RV.3.3
<a href="#">4e(v)</a>	PO.3.2, PO.3.3, PO.4.1, PO.4.2, PO.5.1, PO.5.2, PW.1.2, PW.2.1, PW.7.2, PW.8.2, RV.2.2
<a href="#">4e(vi)</a>	PO.1.3, PO.3.2, PO.5.1, PO.5.2, PS.3.1, PS.3.2, PW.4.1, PW.4.4, RV.1.1, RV.1.2
<a href="#">4e(vii)</a>	PS.3.2
<a href="#">4e(viii)</a>	RV.1.1, RV.1.2, RV.1.3, RV.2.1, RV.2.2, RV.3.3
<a href="#">4e(ix)</a>	Todas as práticas e tarefas consistentes com uma abordagem baseada em riscos
<a href="#">4e(x)</a>	PS.2.1, PS.3.1, PS.3.2, PW.4.1, PW.4.4

Para coincidir com o lançamento deste documento, o NIST também [publicou orientações](#) sobre como os produtores e adquirentes de software podem se comunicar entre si em relação ao atestado de conformidade com a Seção 4e da EO 14028.

## Apêndice B—Acrônimos

Os acrônimos e abreviações selecionados usados neste documento estão definidos abaixo.

BSIMM	Criação de Segurança no Modelo de Maturidade
CISQ	Consórcio para a Qualidade da Informação e do Software
CNCF	Fundação Para Computação Nativa em Nuvem
COTS	Pronto Para Uso Comercial
CPS	Sistema Ciberfísico
DevOps	Desenvolvimento e Operações
EO	Ordem Executiva
GOTS	Pronto Para Uso Governamental
GSA	Administração de Serviços Gerais
ICS	Sistema de Controle Industrial
IDA	Instituto de Análises de Defesa
IEC	Comissão Eletrotécnica Internacional
IoT	Internet das Coisas
IR	Relatório Interno ou Entre Agências
ISO	Organização Internacional de Padronização
ISPAB	Conselho Consultivo de Segurança da Informação e Privacidade
TI	Tecnologia da Informação
ITL	Laboratório de Tecnologia da Informação
KPI	Indicador-chave de desempenho
KRI	Indicador-Chave de Risco
MITA	Aliança de Tecnologia e Imagens Médicas
NAVSEA	Comando de Sistemas Marítimos Navais
NICE	Iniciativa Nacional para Educação em Segurança Cibernética
NIST	Instituto Nacional de Padrões e Tecnologia
NTIA	Administração Nacional de Telecomunicações e Informações
OLIR	Programa Nacional de Referências Informativas Online
OWASP	Projeto aberto de segurança de aplicativos da Web
PCI	Setor de Cartões de Pagamento
PSIRT	Equipe de resposta a incidentes de segurança de produtos
Código SAFEC	Fórum de Garantia de Software para Excelência em Código
SAMM	Modelo de maturidade de garantia de software
SBOM	Lista de Materiais do Software
SDL	[Microsoft] Ciclo de Vida de Desenvolvimento de Segurança
SDLC	Ciclo de Vida de Desenvolvimento de Software
SEI	Instituto de Engenharia de Software
SLC	Ciclo de Vida do Software

SOAR  
SSDF

Recursos de Última Geração  
Estrutura de Desenvolvimento de Software Seguro

## Apêndice C—Registro de Alterações

Este apêndice resume as alterações mais notáveis feitas no SSDF desde a [SSDF original](#) publicada em abril de 2020.

### Esta versão (publicada em fevereiro de 2022)

- Tarefas
  - Eliminado o PW.4.5 (incorporado ao PW.4.4)
- Exemplos de Implementação Ilustrativos
  - Acrescentados vários exemplos sugeridos por meio de comentários do público
  - Adicionado "Exemplo X" ao início de cada exemplo informativo ilustrativo
- Referências
  - Adicionado EO14028, NISTLABEL, SP800161, SP800216
  - BSIMM, OWASPASVS, OWASPMASVS atualizados
  - Atualização do NISTDVS para IR8397
- Editorial
  - Foram feitas pequenas alterações de redação em todo o documento
  - Adicionadas definições de "proveniência", "artefato" e "evidência"

### Versão preliminar publicada em setembro de 2021

- Práticas
  - Adicionado PO.5
  - Eliminado o PW.3 (incorporado ao PW.4)
- Tarefas
  - Adicionados PO.1.2, PO.5.1, PO.5.2, PS.3.2, PW.1.2
  - PW.3.1 movido para PO.1.3; PW.3.2 movido para PW.4.5; PW.4.3 movido para PW.1.3
  - PW.5.2 rebaixado para um exemplo de PW.5.1
- Referências
  - Adicionados CNCFSSCP, IEC62443, ISO29147, ISO30111, NISTDVS, OWASPMASVS, OWASPSCVS
  - Atualizados BSAFSS, BSIMM, OWASPASVS, PCISSLC
  - Eliminado OWASPTTEST
- Convenções da tabela SSDF
  - Identificadores retirados para práticas e tarefas excluídas/movidas (PW.3, PW.3.1, PW.3.2, PW.4.3 e PW.5.2)
  - Adicionadas bordas coloridas e linhas sombreadas para cada grupo de práticas; indicadas práticas e tarefas retiradas por falta de sombreado

- Convertido o conteúdo de um white paper em um documento da série Publicação Especial 800
- Apêndice A Adicionado