Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para APIs

PROGRAMA DE PRIVACIDADE E SEGURANÇA DA INFORMAÇÃO (PPSI)

Versão 2.0

Brasília, abril de 2023

GUIA DE REQUISITOS MÍNIMOS DE PRIVACIDADE E SEGURANÇA DA INFORMAÇÃO PARA APIs

MINISTÉRIO DA GESTÃO E DA INOVAÇÃO EM SERVIÇOS PÚBLICOS

Esther Dweck

Ministra

SECRETARIA DE GOVERNO DIGITAL

Rogério Souza Mascarenhas

Secretário de Governo Digital

DIRETORIA DE PRIVACIDADE E SEGURANÇA DA INFORMAÇÃO

Leonardo Rodrigo Ferreira

Diretor de Privacidade e Segurança da Informação

COORDENAÇÃO-GERAL DE PROTEÇÃO DE DADOS

Loriza Andrade Vaz de Melo

Coordenadora-Geral de Proteção de Dados

Equipe Técnica de Elaboração

Álvaro Sergio de Souza Junior

Amaury C. da Silveira Junior

Bruno Pierre Rodrigues de Sousa

Eder Ferreira de Andrade

Francisco Magno Felix Nobre

Heráclito Ricardo Ferreira Gomes

Ivaldo Jeferson de Santana Castro

Raphael César Estevão

Yuri Arcanjo de Carvalho

Equipe Revisora

Marcelo de Lima

Marcus Paulo Barbosa Vasconcelos

Equipe Técnica de Revisão - Versão 2.0

Adriano de Andrade Moura

Rafael Da Silva Ribeiro

Raphael César Estevão

Rogério Vinícius Matos Rocha

Histórico de Versões

Data	Versão	Descrição	Autor
04/10/2021	1.0	Primeira versão do Guia de Requisitos Mínimos de Segurança e Privacidade para APIs.	Equipe Técnica de Elaboração
15/04/2023	2.0	Atualização para alinhamento com o Guia do Framework de Privacidade e Segurança da Informação, conforme destacado no Anexo II.	Equipe Técnica de Revisão

SUMÁRIO

AVISO PRELIMINAR E AGRADECIMENTOS

O presente Guia, especialmente recomendado e dirigido aos órgãos e às entidades da Administração Pública Federal - APF, visa a auxiliar na especificação de Requisitos Mínimos de Privacidade e Segurança da Informação para APIs (*Application Programming Interface*), em atendimento ao previsto no Capítulo VII - DA SEGURANÇA E DAS BOAS PRÁTICAS Lei nº 13.709, de 14 de agosto de 2018 - Lei Geral de Proteção de Dados Pessoais (LGPD), que determina que a Administração Pública, ao prestar diversos serviços que tratam dados pessoais à sociedade, deve adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito dos dados que estão sob sua custódia, bem como assegurar o mínimo de segurança nas interfaces de programação de aplicação. Adicionalmente, a especificação de Requisitos Mínimos de Privacidade e Segurança da Informação para APIs visa a atender, além da LGPD, a outros normativos vigentes sobre o tema de privacidade e segurança da informação.

Este documento é de autoria exclusiva da Secretaria de Governo Digital (SGD) do Ministério da Gestão e da Inovação em Serviços Públicos e tem como referência fundamental o Guia do Framework de Privacidade e Segurança da Informação baseado em diversas publicações e documentos técnicos já existentes que são utilizados amplamente por profissionais da área de privacidade e segurança da informação. Destacam-se as publicações do *Center for Internet Security* (CIS), da *International Organization for Standardization* (ISO), do *National Institute of Standards and Technology* (NIST). Em complemento ao Guia do Framework de Privacidade e Segurança da Informação, este **Guia** foi inspirado em publicações da *Open Web Application Security Project* (OWASP) e *World Wibe Web Consortium* (W3C). Com o objetivo de facilitar a difusão de conhecimentos sobre privacidade e segurança da informação, tais referências, quando escritas em línguas estrangeiras, foram traduzidas para o português pela equipe técnica da Diretoria de Privacidade e Segurança da Informação da Secretaria de Governo Digital.

Nesse cenário, a Secretaria de Governo Digital enfatiza que:

a) não representa, tampouco se manifesta em nome do CIS, da ISO, do NIST, do OWASP, da W3C e vice-versa;
b) não se manifesta em nome de autoridades de privacidade e segurança da informação;
c) não é coautora das publicações internacionais abordadas;
d) não assume nenhuma responsabilidade administrativa, técnica ou jurídica por usos ou interpretações inadequadas, fragmentados ou parciais do presente guia; e
e) caso o leitor deseje se certificar de que atende integralmente os requisitos das publicações das instituições mencionadas, deverá consultar diretamente as fontes oficiais de informação ofertadas por elas, que foram listadas na seção "Referências Bibliográficas" deste documento.

Finalmente, um agradecimento especial deve ser registrado ao CIS, à ISO, ao NIST, à OWASP, à W3C e aos profissionais de privacidade e segurança da informação consultados, por suas valiosas contribuições para a comunidade e para elaboração **deste documento**.

Este Guia será atualizado frequentemente, de acordo com as novas diretrizes determinadas pelas autoridades em privacidade e segurança da informação ou segundo eventuais alterações que ocorram nos normativos vigentes relacionados a privacidade e segurança da informação e outras referências utilizadas **neste documento**

INTRODUÇÃO

Este Guia tem por finalidade apresentar orientações com o intuito de auxiliar os órgãos e entidades da Administração Pública Federal, direta, autárquica e fundacional a especificar os Requisitos Mínimos de Privacidade e Segurança da Informação para APIs no âmbito institucional.

Os Controles 16 (p. 55) e 22 (p. 63) do Guia do Framework de Privacidade e Segurança da Informação, estabelecem que:

Controle 16: Segurança de Aplicações *- Gerenciar o ciclo de vida de segurança de todos os softwares desenvolvidos e adquiridos internamente, a fim de prevenir, detectar e corrigir falhas de segurança.*

Controle 22: Políticas, Processos e Procedimentos *- Definir, desenvolver, divulgar, implementar e atualizar políticas, processos e procedimentos operacionais, internos e externos que regem as ações relativas à proteção de dados pessoais e privacidade, e controles para programas, sistemas de informação ou tecnologias que envolvam o tratamento de dados pessoais.*

O presente Guia serve como um modelo prático a ser utilizado para auxiliar na adoção de medidas dos Controles 16 e 22 do Guia do Framework de Privacidade e Segurança da Informação[1] v1 e respectivas evoluções desta versão (1.1, 1.2 etc.) elaborado e publicado pela SGD. As medidas dos Controles 16 e 22 que estão contempladas por este Guia são: 16.1, 16.2, 16.3, 16.4, 16.5, 16.6, 16.7, 16.10 e 22.7.

As APIs (*Application Programming Interface*, em inglês, ou Interface de Programação de Aplicação, em português) são um conjunto de rotinas e padrões estabelecidos por um software ou sistema para utilizar as suas funcionalidades por aplicativos que não pretendem se envolver com os detalhes da implementação do software fonte, mas apenas utilizar os seus serviços.

É notável a importância do papel das APIs na materialização das estratégias de negócio e na transformação digital. Atualmente, muitas aplicações recorrem às APIs para garantir a integração com múltiplos sistemas, o que melhora a governança de suas

[1] < https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/ppsi/guia_framework_psi.pdf >. Acesso: 10 abr. 2023

atividades de interoperabilidade, estabelece relações interorganizacionais, agiliza processos de suporte a serviços digitais de ponta a ponta, e fornece os meios para garantir visibilidade, que é um princípio subjacente da transparência, mediante a obtenção de interfaces externas para os serviços públicos.

Ainda sobre a importância do uso de APIs, pode ser citada a Política de Dados Abertos (Decreto nº 8.777, de 11 de maio de 2016), que se assenta em dois pilares fundamentais do mercado interno: transparência e livre concorrência.

Tal quadro enfatiza a importância da reutilização de dados no setor público, no que concerne à iniciativa "*once only*". O princípio "uma única vez" que permite que cidadãos e empresas sejam capazes de fornecer informações apenas uma vez e tenham esses dados compartilhados e reutilizados por outros órgãos e entidades públicas.

Saber como estruturar uma boa política de gestão de APIs reutilizáveis e específicas é essencial para evitar complexidade excessiva e para permitir a ligação mais fácil e rápida entre dados, aplicações e sistemas. Quando bem desenhadas e implementadas, as APIs são um excelente mecanismo para aumentar a produtividade das equipes e possibilitar a fruição de ferramentas mais avançadas. Contudo, é preciso observar alguns pontos relacionados à privacidade e segurança da informação de APIs antes de incorporar o uso de tais serviços.

A proteção e a confiabilidade de informações sensíveis que trafegam por meio das APIs devem ser aspectos prioritários.

1 Requisitos gerais

Em meio ao processo de transformação digital, a profusão da mobilidade e da Internet das Coisas (IoT), as APIs ganham um papel estratégico para o desenvolvimento de aplicativos web e mobile em ambientes compartilhados. No entanto, quando se mencionam novas formas de interação, é inevitável abordar os riscos inerentes nesse processo. Diante de tal contexto, os conceitos de *Privacy by Design* e *Security by Design* vêm ganhando destaque, induzindo que a privacidade e a segurança da informação sejam embutidas na construção de uma nova aplicação já em estágio inicial.

Assim, o presente capítulo trata de requisitos gerais que compreendem a adoção de práticas como *Privacy by Design* e *Security by Design*.

A *General Data Protection Regulation* (GDPR)[2] foi uma das primeiras normas jurídicas a apresentar dois princípios que passaram a ser relevantes na produção de aplicativos e outros serviços digitais, dentro do contexto moderno de proteção de dados. Trata-se do *Privacy by Design* e do *Privacy by Default*, que permitem a adequada governança de dados. Esses princípios encontram-se hoje contemplados na Lei Geral de Proteção de Dados (LGPD) brasileira, valendo-se de conceitos similares.

A LGPD estabelece que instituições públicas e privadas devem utilizar medidas técnicas aptas a proteger os dados contra acessos não autorizados, no que concerne a situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão. A legislação estabelece igualmente que as organizações devem contar com ações para prevenir a ocorrência de danos aos dados tratados, comprovando que atendem esses requisitos.

Pode-se dizer que o *Privacy by Default* é uma decorrência do *Privacy by Design,* que seria garantir medidas de proteção aos dados pessoais como padrão no desenvolvimento de produtos ou serviços, contemplando a proteção mínima necessária, como também a observância aos princípios da LGPD. Isso significa que, uma vez que um produto ou serviço

[2] O Regulamento Geral sobre a Proteção de Dados (RGPD) (UE) 2016/679 é um regulamento do direito europeu sobre privacidade e proteção de dados pessoais, aplicável a todos os indivíduos na União Europeia e Espaço Económico Europeu que foi criado em 2018. https://gdpr-info.eu/ Acesso: 04 abr. 2023

foi lançado ao público, as configurações de privacidade mais restritas já estarão aplicadas por padrão, sem qualquer necessidade de ação do usuário.

O conceito de *Security by Design*, quando em referência à segurança de sistemas, descreve as melhores práticas e padrões de segurança a serem aplicados pela equipe. Por isso, é considerado uma das principais abordagens para garantir a privacidade e segurança da informação dos sistemas. Nesse contexto, a segurança é construída no sistema desde o seu início e começa com um design de arquitetura adequado, levando em consideração os pilares da Segurança da Informação como requisitos[3].

1.1 Privacy by Design

A ideia de *Privacy by Design* é incorporar medidas protetivas de privacidade e dados pessoais, em todas as fases dos projetos em desenvolvimento. A princípio, não seria permitido desenvolver nenhum projeto, produto ou serviço sem que os princípios de proteção da privacidade estejam no centro desse desenvolvimento.

Em outras palavras, o produto ou serviço deve ser lançado e recebido pelo usuário com todas as salvaguardas que foram concebidas durante o seu desenvolvimento. Logo, todas as medidas para proteger a privacidade, idealizadas desde o início do desenvolvimento do projeto, atendem esse princípio.

O *Privacy by Design* pode ser mais bem compreendido por intermédio da análise dos princípios basilares de privacidade, segundo a Nota do Grupo de Trabalho W3C[4]. Tais princípios, que são relevantes para APIs, efetuam uma correlação com os princípios da LGPD:

[3] OWASP – Security by Design Principles (https://www.owasp.org/index.php/Secure_Coding_Principles). Acesso: 04 abr. 2023

[4] https://www.w3.org/TR/dap-privacy-reqs/#privacy-requirements Acesso: 05 abr. 2023

Figura 1: Correlação de princípios

A tabela a seguir resume a divisão de como cada elemento é coberto na definição de requisitos para APIs e os relacionados às expectativas do usuário quanto ao uso de dados:

Elemento de Privacidade	Requisitos para definições de API	Requisitos relacionados às expectativas do usuário quanto ao uso de dados
Aviso	X	
Consentimento	X	
Minimização	X	
Controle	X	
Acesso	X	
Retenção		X
Uso Secundário		X
Compartilhamento		X

Tabela 1: Divisão dos elementos de privacidade

1.2 Security by Design

O Glossário disponibilizado pelo Gabinete de Segurança Institucional (GSI)[5] de termos-chave de segurança de informação define "segurança de informação" como " ações que objetivam viabilizar e assegurar a disponibilidade, a integridade, a confidencialidade e a autenticidade das informações". O OWASP recomenda que todos os controles de segurança sejam projetados com os pilares fundamentais da segurança da informação a fim de fornecer:

1) **integridade:** proteger contra modificação indevida de informações ou destruição, incluindo a garantia do não-repúdio às informações e a autenticidade;

2) **confidencialidade**: preservar as restrições autorizadas sobre acesso e divulgação, incluindo meios para proteger a privacidade pessoal e as informações proprietárias; e

3) **disponibilidade**: garantir acesso oportuno e confiável e uso de informação.

Os princípios a seguir estão todos relacionados a esses três pilares. De fato, deve-se considerar cada pilar ao se considerar como construir um controle. Isso certamente ajudará a produzir um controle de segurança robusto.

É importante salientar, ainda, que a listagem abaixo não representa um conjunto de requisitos exaustivos:

1.2.1 Reduzir a superfície de ataque

Cada vez que um programador adiciona um recurso ao seu aplicativo, ele aumenta o risco de uma vulnerabilidade de segurança. O princípio de reduzir a superfície de ataque é usado para restringir as funções que os usuários têm permissão para acessar, contribuindo com a redução de vulnerabilidades. Com a integração de ferramentas de proteção já existentes, é possível desenvolver um ecossistema de monitoramento e correções em tempo real.

[5] https://www.gov.br/gsi/pt-br/composicao/SSIC/dsic/glossario-de-seguranca-da-informacao-1 Acesso: 05 de br. 2023

1.2.2 Estabelecimento de padrões

Este princípio afirma que o aplicativo deve ser seguro por padrão. Os requisitos de segurança devem estar presentes nas práticas de desenvolvimento e implantação de código. Deve haver regras de segurança fortes sobre como os registros de usuários são tratados, com que frequência as senhas devem ser atualizadas, quão complexas as senhas devem ser, e assim por diante.

1.2.3 Princípio do menor privilégio

O princípio do menor privilégio recomenda fornecer as permissões necessárias e suficientes para que um usuário realize suas tarefas, com tempo determinado e os direitos mínimos estabelecidos. A atribuição de permissões a um usuário pode impedir que ele execute tarefas para as quais não está autorizado, como acessar, obter ou modificar informações.

1.2.4 Princípio da defesa em profundidade

A defesa em profundidade é um conjunto de práticas que se concentram na proteção, detecção e reação de invasões. Para isso, são usados softwares de segurança e ferramentas para a construção de uma estratégia de contra-ataques.

Esse princípio afirma que vários controles de segurança que abordem os riscos de maneiras diferentes são a melhor opção para proteger um aplicativo. Portanto, em vez de existir um controle de segurança para acesso do usuário, haveria várias camadas de validação, ferramentas adicionais de auditoria de segurança, e ferramentas de registro. Ferramentas de segurança como firewalls, antivírus, filtragem de conteúdo, criptografia e controle de acesso colaboram para a prevenção contra-ataques.

1.2.5 Falhar com segurança

A manipulação segura de erros é um aspecto importante para um software seguro e para o *Security by Design*. Falhas não devem dar ao usuário privilégios adicionais. É importante que tais exceções não permitam comportamentos que o sistema normalmente

não permitiria. Um software desenvolvido com segurança deve considerar a existência de três resultados possíveis de um mecanismo de segurança: proibir ou permitir a operação, ou lançar uma exceção.

1.2.6 Não confie nos serviços

Muitos aplicativos web utilizam serviços de terceiros para acessar funcionalidades ou obter dados adicionais. Conhecido também como *Zero Trust*, este princípio afirma que nunca deve haver confiança nesses serviços do ponto de vista da segurança. Isso significa que o aplicativo deve sempre verificar a validade dos dados que os serviços de terceiros enviam e não dar a esses serviços permissões de alto nível dentro do aplicativo.

1.2.7 Segregação de funções

A segregação de funções e responsabilidades é o controle de acesso baseado no papel, na atividade ou na função de um usuário dentro de um sistema. Certas funções têm níveis de confiança diferentes dos usuários normais. Em particular, os administradores são diferentes dos usuários normais; em geral, os administradores não devem ser usuários do aplicativo. O perfil por função consegue agrupar os acessos, possibilitando uma visão geral dos privilégios e controlando os acessos de uma forma segura para o *Security by Design*.

1.2.8 Evitar a segurança por obscuridade

A segurança através da obscuridade é um controle de segurança fraco e quase sempre falha quando é o único controle. Isso significa que a segurança dos sistemas principais não deve depender da ocultação de detalhes. Por exemplo, a segurança de um aplicativo independe do conhecimento do código-fonte mantido em segredo. A segurança deve se basear em muitos outros fatores, incluindo políticas de senha razoáveis, defesa em profundidade, limites de transações de negócios, arquitetura de rede sólida e controles de auditoria e fraude.

1.2.9 Mantenha a segurança simples

No início de uma implementação de *Security by Design*, é comum fazer uso de ferramentas, processos e controles em favor da segurança de sistemas, mas é necessário refletir sobre a relevância de todos esses controles. Eles acrescentam mais segurança ou burocracia aos sistemas? A existência de muitas ferramentas pode aumentar as brechas de segurança em vez de extingui-las, da mesma forma que procedimentos pouco documentados ou falta de automações que podem deixar usuários excessivamente em espera por um acesso.

1.2.10 Segurança no processo de manutenção do software

Assim que um problema de segurança for identificado, é importante desenvolver um teste para entender a causa-raiz do problema. As vulnerabilidades em sistemas precisam ser estudadas pelo time de desenvolvimento para uma correção eficiente. É preciso entender o comportamento da vulnerabilidade de forma estrutural no sistema e verificar se existem outros componentes que podem ser afetados pela mesma vulnerabilidade.

Fique atento

A falta de um processo ou controle para realizar as correções de problemas pode causar o surgimento de novos problemas e brechas de segurança nos sistemas. Um processo contínuo de gestão de vulnerabilidades é um aliado para as equipes de desenvolvimento, atuando na identificação, análise, classificação e tratamento de tais pontos sensíveis. Esse processo busca medir o progresso e avaliar os riscos aos quais os sistemas estão submetidos, colaborando com uma estratégia de *Security by Design*.

2 Requisitos específicos

2.1 Privacidade

A metodologia *Privacy By Design* é uma grande aliada no auxílio à adequação à LGPD, em especial ao artigo 46, que prevê:

> *Art. 46. Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito.*
>
> *(...)*
>
> *§ 2º As medidas de que trata o caput deste artigo deverão ser observadas desde a fase de concepção do produto ou do serviço até a sua execução.*

Este capítulo descreve boas práticas de privacidade para APIs, conforme recomendações da Nota do Grupo de Trabalho W3C. Ele não repete os princípios e os requisitos de privacidade documentados na Nota de Requisitos de Privacidade da API do Dispositivo (Requisito geral 1: *Privacy by Design*), que também deve ser consultado. Ademais, é importante salientar que a listagem abaixo não representa um conjunto exaustivo de requisitos.

2.1.1 Requisitos para definições de API

Muitos dos requisitos aqui listados são recomendações, não requisitos absolutos. Isso ocorre porque, em muitos casos, tornar um requisito absoluto é apropriado apenas para um subconjunto de APIs, mas não para todas. Conforme apropriado, algumas APIs podem impor requisitos normativos mais específicos que os requisitos neste documento.

2.1.1.1 Aviso

Certificar-se de que os usuários entendam as implicações de usar um aplicativo que depende de uma API, pois é fundamental para garantir a proteção de seus dados. Os seguintes requisitos podem ajudar a garantir que os usuários sejam devidamente notificados:

- As APIs devem permitir que os agentes de software notifiquem os usuários de que seus dados estão sendo coletados por meio da API. Essa notificação deve identificar o aplicativo (por exemplo, exibindo sua origem do documento) e os dados precisos que estão sendo coletados.
- Recomenda-se que as APIs forneçam suporte para indicadores visuais, com o intuito de demonstrar que os dados estão sendo coletados por meio das APIs.

2.1.1.2 Consentimento

A semântica de algumas APIs é definida de forma que a interação do usuário seja essencial para fazer uso da API. Um exemplo é uma API de câmera que permite ao usuário tirar uma fotografia, mas é definida para exigir que o usuário pressione uma tecla do obturador para tirar a fotografia. Outro exemplo é uma API que produz um modelo de mensagem, mas exige que o usuário pressione "enviar" para realmente enviar a mensagem.

Essas ações do usuário constituem **consentimento implícito** para a coleta de dados por meio da API, uma vez que o usuário tem a opção de realizar essas ações e isso implica no consentimento para que o aplicativo acesse os recursos do(s) dispositivo(s) associado(s). Em tais situações, em que é óbvio que a execução da ação envolve o compartilhamento de dados com o aplicativo e o uso pretendido dos dados pelo aplicativo, diálogos adicionais que solicitam o consentimento dos usuários podem não ser necessários.

As APIs também podem ser definidas de forma que o **consentimento seja explícito, não implícito.** Os exemplos são uma API de câmera que tira uma fotografia sem o envolvimento do usuário ou uma API de mensagens que envia uma mensagem sem que o usuário pressione "enviar". Nesses casos, é necessário o consentimento explícito do usuário.

Para garantir que os dados não sejam coletados sem que os usuários saibam ou percebam, as APIs devem ser projetadas com a presunção de que o modelo de

consentimento explícito será usado e devem explicar as circunstâncias específicas sob as quais o consentimento implícito pode ser aceitável. Isso dá origem aos seguintes requisitos:

- As APIs devem permitir que os agentes de software obtenham o consentimento do usuário antes de compartilhar quaisquer dados por meio das APIs.
- As APIs devem ser definidas de forma que o consentimento explícito seja assumido e recomenda-se articular as circunstâncias em que o consentimento implícito é aceitável.

Uma ressalva importante ao modelo de consentimento, suportado pelas APIs do dispositivo, está relacionada aos dados sobre outras pessoas que o usuário pode ter em seu dispositivo e pode compartilhar por meio das APIs (contato de outras pessoas ou informações de calendário, por exemplo). Um usuário não deve ser capaz de consentir, por meio das APIs do dispositivo, o uso de informações de outras pessoas além da interação original com a API. Assim, por exemplo, um usuário deve poder consentir que um aplicativo entre em contato com outra pessoa mencionada em uma entrada de calendário (talvez para dizer “estou atrasado”), mas o usuário não deve também consentir que o aplicativo faça uso posterior das informações de contato da pessoa (talvez para enviar mensagens de marketing para essa pessoa). Essa ressalva deve ser transmitida quando apropriado nas APIs, práticas recomendadas e outros documentos de APIs de dispositivos.

2.1.1.3 Minimização

Para reduzir os riscos de superexposição dos dados dos usuários, é importante projetar APIs para que os desenvolvedores da Web possam solicitar o mínimo de informações necessárias para atingir seus objetivos.

Um caso de uso de exemplo é um caso de rede social em que a API de contatos é usada para contatos que também são membros de uma rede social. Endereços de e-mail servem como identificadores de redes sociais. Neste caso, limitar os resultados aos endereços e não a outras informações pessoais é um exemplo de minimização.

Isso dá origem aos seguintes requisitos:

- As APIs devem facilitar a solicitação do mínimo de informações necessárias para o uso pretendido.

Por exemplo, uma chamada de API deve exigir que parâmetros específicos sejam definidos para obter mais informações e deve ter como padrão pouca ou nenhuma informação.

- Recomenda-se que as APIs permitam que os agentes de software transmitam apenas as informações que o solicitante está solicitando.

Por exemplo, se um desenvolvedor só precisa acessar um campo específico de um catálogo de endereços do usuário, deve ser possível marcar explicitamente esse campo na chamada da API para que o agente de software possa informar ao usuário que esse único campo de dados será compartilhado.

- Recomenda-se que as APIs possibilitem que os agentes de software deixem o usuário selecionar, filtrar e transformar as informações antes de serem compartilhadas com o solicitante.

O agente do usuário pode então atuar como um intermediário para dados confiáveis e só transmitirá ao solicitante os dados explicitamente permitidos pelo usuário.

2.1.1.4 Controle

Dada a confidencialidade dos dados disponibilizados pelas APIs do dispositivo, é importante que os usuários possam controlar quais aplicativos têm acesso a esses dados. Os requisitos a seguir garantem que os usuários tenham controle sobre seus dados mesmo depois de compartilhá-los com um aplicativo e os usuários tenham controles robustos sobre quais aplicativos podem obter seus dados:

- As APIs devem permitir que os agentes de software suportem a revogação do consentimento do usuário para o compartilhamento de dados por meio das APIs.
- Recomenda-se que as APIs ofereçam suporte à capacidade dos agentes de software de permitir que os usuários coloquem aplicativos confiáveis na lista branca e aplicativos não confiáveis na lista negra.

2.1.1.5 Acesso

O aviso e o controle não podem ser totalmente implementados, a menos que os usuários possam revisar como foi realizado o compartilhamento de dados no passado. Os requisitos a seguir sugerem como as APIs podem oferecer suporte ao acesso dos usuários a essas informações:

- Recomenda-se que as APIs permitam que os agentes de software visualizem os aplicativos com os quais compartilharam dados.
- Recomenda-se que as APIs permitam que os agentes de software visualizem, editem e excluam o histórico de compartilhamento de dados do usuário com cada um.

2.1.2 Requisitos relacionados às Expectativas do Usuário no Uso de Seus Dados

Os usuários podem ter expectativas sobre o uso de seus dados, em particular quanto a retenção dos dados, uso para outros fins e compartilhamento. Como os aplicativos planejam atender a essas expectativas (política de aplicativo), como os usuários expressam seus desejos (política de usuário) e as restrições no uso de dados podem estar relacionados ao gerenciamento dessas expectativas.

Como a retenção, o uso secundário e o compartilhamento estão amplamente fora do controle das APIs, não é possível afirmar que deva existir quaisquer requisitos de API sobre esses aspectos. Por outro lado, pode-se imaginar o seguinte requisito:

- As APIs devem oferecer suporte para que os usuários transmitam suas preferências sobre retenção, uso secundário e compartilhamento para aplicativos no contexto de uma interação envolvendo API.

Da mesma forma, caso exista a necessidade de transmitir suas políticas sobre esses aspectos, então surge mais um requisito:

- As APIs devem oferecer suporte a um mecanismo para que os aplicativos transmitam suas políticas sobre retenção, uso secundário e compartilhamento aos usuários antes ou durante as interações da API.

2.1.2.1 Retenção

A retenção está relacionada às expectativas do usuário sobre quanto tempo os dados fornecidos serão retidos e se os aplicativos devem descartar os dados coletados após cumprir a finalidade para a qual foram coletados.

2.1.2.2 Uso Secundário

O Uso Secundário relaciona-se a atender às expectativas do usuário sobre se os aplicativos podem usar dados para fins diferentes daqueles para os quais eles foram coletados.

2.1.2.3 Compartilhamento

O compartilhamento é sobre as expectativas do usuário de como os dados serão compartilhados. Assim que os dados forem disponibilizados a um solicitante, o solicitante poderá armazenar e redistribuir esses dados, com ou sem o consentimento do usuário, conforme legislação. A granularidade dos dados compartilhados é um aspecto importante do compartilhamento.

2.2 Segurança

É consenso entre profissionais de segurança que as APIs representam um importante vetor de vulnerabilidade, em observância aos diversos incidentes que passam pelas APIs.

Parte desses problemas está vinculada à inobservância de sua segurança, por se supor que estariam mais protegidas – seja por permanecerem mais embutidas no contexto das aplicações, seja por alguns cenários em que não estejam de fato expostas de forma geral na Internet. Por natureza, as APIs expõem a lógica dos aplicativos e dados, inclusive de dados pessoais sensíveis. Por essa razão, as APIs têm se tornado um alvo crescente de ataques.

Diante dessa realidade e com enfoque na mudança na classificação do risco das APIs em comparação a outras aplicações, o OWASP, por meio da atividade da comunidade de

segurança, elaborou o projeto API Security Top 10[6] com ênfase em estratégias e soluções para a compreensão e mitigação de vulnerabilidades únicas associadas às APIs. O objetivo é orientar os profissionais que estejam envolvidos no desenvolvimento e na manutenção de APIs.

Este capítulo expõe os 10 maiores riscos extraídos do guia de "OWASP API Security Top 10". Os riscos serão apresentados e medidos pelas categorias abaixo:

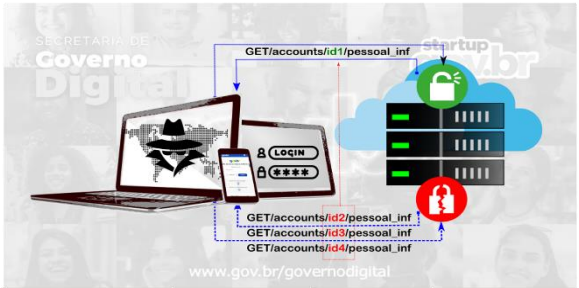
Explorabilidade	**Prevalência da Fraqueza**	**Detecção da Fraqueza**	**Impacto Técnico**
Fácil: 3	Difundida: 3	Fácil: 3	Severo: 3
Médio: 2	Comum: 2	Média: 2	Moderado: 2
Difícil: 1	Difícil: 1	Difícil: 1	Menor: 1

Tais riscos devem ser analisados ao longo do ciclo de vida das APIs.

Considerando o objetivo deste capítulo, para fins de facilitar sua utilização, os 10 riscos serão apresentados abaixo em forma de cartões (*cards*) que representam um recorte do documento original, composto pelas seguintes partes: risco, nível do risco, conceito, motivo e prevenção.

[6] OWASP API Security - Top a | OWASP - https://github.com/OWASP/API-Security/tree/master/2023/en/src Acesso: 06 abr. 2023

API1. Quebra de Autorização a Nível de Objeto



Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Difundida: 3	Média: 2	Severo: 3

Conceito

Os invasores podem explorar pontos de extremidade de API que são vulneráveis à autorização de nível de objeto quebrada manipulando a ID de um objeto que é enviado dentro da solicitação. Isso pode levar ao acesso não autorizado a dados confidenciais.

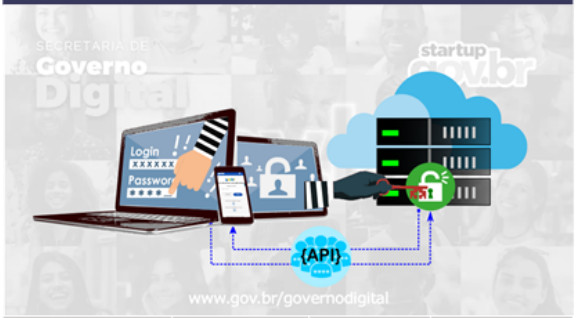
Como acontece?

- ✓ Descobrindo API, ID de recurso ou objeto e utilizando-o para fins maliciosos.
- ✓ Os dados não têm validação do usuário e são acessíveis a qualquer pessoa.
- ✓ Mensagens de erros retornam muitas informações, apresentando para os atacantes como abusar da API.
- ✓ Atacantes utilizam técnicas de enumeração para coletar dados.

Como prevenir?

- ✓ Implementar verificações de autorização com políticas e hierarquia do usuário.
- ✓ O desenvolvimento de aplicativos deve colaborar com a segurança para implementar uma autorização forte para garantir que a API valide os privilégios do usuário para todas as funções.
- ✓ A API deve ser projetada para usar IDs aleatórios.
- ✓ Verificar a autorização cada vez que houver uma requisição do cliente para acessar o banco de dados.
- ✓ Construir casos de teste específicos de segurança a nível de função para recursos relacionados à autorização.

API2. Quebra de Autenticação



Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Comum: 2	Média: 2	Severo: 3

Conceito

Políticas de autenticação de usuário quando mal implementadas, permitem que os atacantes se passem por usuários legítimos, podendo tirar vantagens das falhas de implementação nos mecanismos de autenticação.

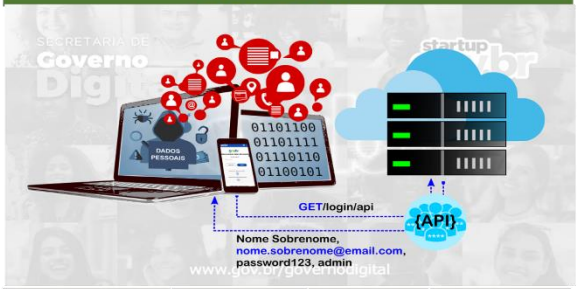
Como acontece?

- ✓ Descobrindo APIs de autenticação desprotegidas (login, cadastro e reset de senha) e com falhas e utilizando-os para propósitos maliciosos. Tais falhas de API que expõem essa ameaça incluem APIs que permitem senhas fracas, que utilizam mensagens de erros que retornam muitas informações e que não possuem validação de token e utilizam criptografia fraca ou sem criptografia.

Como prevenir?

- ✓ Grupos de desenvolvimento, segurança e negócios de aplicativos devem compreender, concordar e documentar o fluxo de trabalho de autenticação e os requisitos associados.
- ✓ Use autenticação baseada em padrões (forte), incluindo multifator quando possível.
- ✓ Verifique todas as formas possíveis de autenticação em todas as APIs.
- ✓ Use autenticação padrão, geração de token, armazenamento de senha e multifator de autenticação.
- ✓ APIs de redefinição de senha e links únicos também permitem que os usuários estejam autenticados e também estejam protegidos com a mesma seriedade.

API3. Autorização de Nível de Propriedade de Objeto Quebrado



Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Comum: 2	Média: 2	Moderado: 2

Conceito

Uma API publicada pode expor mais dados do que o necessário, contando com o aplicativo cliente para realizar a filtragem necessária. Se um atacante consultar diretamente a API subjacente, o mesmo poderá acessar dados confidenciais.

Como acontece?

- ✓ O atacante descobre uma API, rouba os dados ou utiliza as informações para um ataque maior.

Como prevenir?

- ✓ Nunca confie no cliente para filtrar dados.
- ✓ Projete a API para sempre retornar dados mínimos; nunca confie na filtragem do lado do cliente - inclua requisitos no esquema de API.
- ✓ Revise todas as respostas e adapte as respostas aos consumidores de API que realmente precisam.
- ✓ Evite utilizar métodos genéricos, como "to_json()" ou "to_string()".
- ✓ Defina esquemas de todas as respostas da API.
- ✓ Não se esqueça das respostas de erro.
- ✓ Identifique todas as informações sensíveis ou PII e justifique seu uso.
- ✓ Aplicar verificações de resposta para prevenir vazamentos de dados acidentais e de exceção.

API4. Consumo irrestrito de recursos

Explorabilidade	Prevalência	Detecção	Impacto
Médio: 2	Difundida: 3	Fácil: 3	Moderado: 2

Conceito

Muitas vezes, as APIs não impõem quaisquer restrições sobre o tamanho ou número de recursos que podem ser solicitados pelo cliente/usuário. Isso não só pode impactar o desempenho do servidor da API, levando à negação de serviço (DoS), mas também deixar a porta aberta para falhas de autenticação, tal como força bruta.

Como acontece?

- ✓ Inadequado ou nenhuma limitação de taxa (limite de resposta, memória, tamanho da payload, número de processos, gravações e requisições), permite que os atacantes submetam muitas requisições de API, podendo causar indisponibilidade no serviço (DoS ou DDoS).

Como prevenir?

- ✓ Definir e aplicar limites de consumo de chamada de API.
- ✓ Limitar tamanho do payload.
- ✓ Definir e aplicar o tamanho máximo dos dados em todas as entradas de parâmetros e payloads (tamanho para strings, números de elementos do array entre outros).
- ✓ Limites de taxa específicos para métodos de API, clientes e endereços.
- ✓ Verificar as taxas de compressão.
- ✓ Limites nos recursos de contêiner.

API5. Autorização de nível de função quebrada

Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Comum: 2	Difícil: 1	Médio: 2

Conceito

Políticas de controle de acesso complexas com diferentes hierarquias, grupos e funções e uma separação pouco clara entre funções administrativas e regulares, tendem a levar a falhas de autorização. Ao explorar esses problemas, os invasores obtêm acesso aos recursos de outros usuários e/ou funções administrativas.

Como acontece?

- ✓ Privilégios de usuário que não estão segregados ou aplicados adequadamente (admin, superuser, helpdesk etc), permite que um atacante obtenha acesso a dados, comandos privilegiados ou funções confidenciais (por exemplo, PUT, DELETE, OPTIONS etc) permitindo que o roubo de dados ocorra.

Como prevenir?

- ✓ Não confie no aplicativo para impor acesso de administrador.
- ✓ Defina e implemente um mecanismo forte e consistente de controle de acesso/autorização.
- ✓ Negar, por padrão, todos os acessos.
- ✓ Modelo positivo de segurança por padrão – Negar tudo, exceto o que precisa ser permitido.
- ✓ Garantir acesso com base em funções específicas.
- ✓ Projetar e testar adequadamente a autorização.

API6. Falsificação de solicitação do lado do servidor

Explorabilidade	Prevalência	Detecção	Impacto
Médio: 2	Comum: 2	Difícil: 1	Médio: 2

Conceito

Os dados não filtrados fornecidos por meio de APIs para aplicativos clientes, permitem que os atacantes adivinhem as propriedades do objeto por meio de requisições.

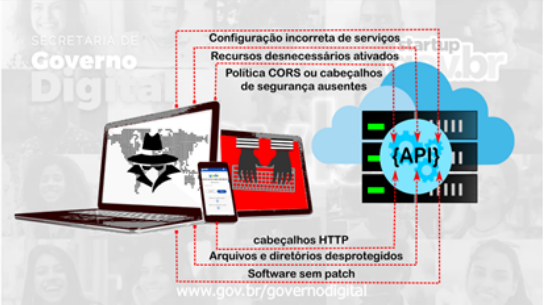
Como acontece?

- ✓ Variáveis do lado do servidor que deveriam ser restritas (mas não são) podem ser inicializadas ou substituídas pelo atacante.
- ✓ O atacante descobre parâmetros modificáveis e os explora criando novos usuários com privilégios de administrador ou com outros privilégios elevados.

Como prevenir?

- ✓ Não vincule dados de entrada e objetos internos automaticamente.
- ✓ Evite usar funções que vinculam entradas a objetos ou variáveis de código.
- ✓ Defina explicitamente todos os parâmetros e payloads que esteja esperando.
- ✓ Para esquemas de objeto, use readOnly definido como true para todas as propriedades que podem ser recuperadas por meio de APIs, mas nunca devem ser modificadas.
- ✓ Definir com precisão no momento do design os schemas, types, patterns que será aceito em requisições e aplique-os em tempo de execução.

API7. Configuração incorreta de segurança



Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Difundida: 3	Fácil: 3	Moderado: 2

Conceito

A configuração incorreta da segurança é geralmente resultado de configurações padrão inseguras, configurações incompletas ou ad-hoc, armazenamento em nuvem aberta, cabeçalhos HTTP configurados incorretamente, métodos HTTP desnecessários, compartilhamento permissivo de recursos de origem cruzada (CORS) e mensagens de erro detalhadas contendo informações confidenciais.

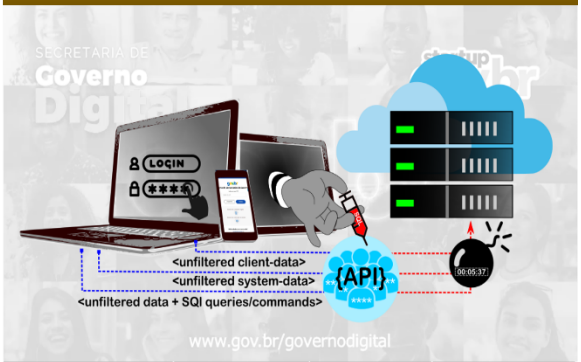
Como acontece?

- ✓ A aplicação está sem patch, falta de hardening ou tem serviços configurados incorretamente que expõem gaps de segurança.
- ✓ Mensagem em modo verbose, sem criptrografia, funcionalidades desnecessarias ou metodos HTTP verbose permitido/expostos.
- ✓ Por falta da política de Cross-Origin Resource Sharing (CORS).

Como prevenir?

- ✓ Realizar frequentes processos de patching e hardening.
- ✓ Avalie e reforce continuamente a conformidade do esquema.
- ✓ Defina um processo repetível para hardening, patch e teste funcional de segurança.
- ✓ Processo automatizado para localizar falhas de configuração.
- ✓ Desativar recursos desnecessários.
- ✓ Restringir o acesso administrativo.
- ✓ Definir e aplicar todas as saídas incluindo erros.

API8. Falta de proteção contra ameaças automatizadas



Explorabilidade	Prevalência	Detecção	Impacto
Fácil: 3	Difundida: 3	Difícil: 1	Severo: 3

Conceito

Envolve a compreensão do modelo de negócios da API, a localização de fluxos de negócios confidenciais e a automação de acesso a esses fluxos, causando danos aos negócios.

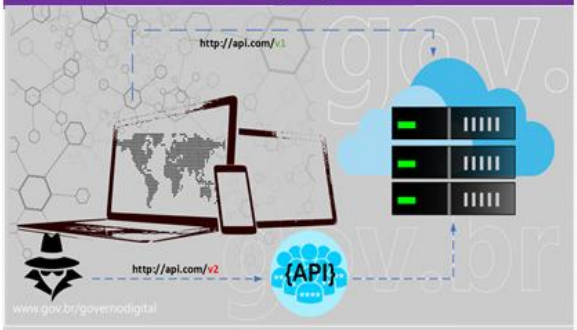
Como acontece?

- ✓ As APIs vulneráveis simplesmente expõe um fluxo de negócios mesmo sem bugs de implementação. O excesso de requisições de forma automatizada pode prejudicar o negócio.

Como prevenir?

- ✓ Identificar os fluxos de negócios que podem prejudicar o negócio se utilizados excessivamente.
- ✓ Escolher mecanismos de proteção corretos para mitigar o negócio risco.
- ✓ Defina, limite e aplique as saídas da API para evitar vazamentos de dados.
- ✓ Proteja e limite o acesso a APIs que são consumidas diretamente por máquinas (como a do desenvolvedor de aplicações).

API9. Gestão imprópria de ativos



Conceito

O gerenciamento insuficiente e segregação do ambiente permite que o atacante acesse os endpoints da API desprotegida.

Como acontece?

- ✓ O processo de publicação de API não documentada/inexistente permite que diferentes grupos publiquem APIs sem supervisão.
- ✓ APIs de shadow, obsoletas e end-of-life implantadas fora da visão de segurança introduzem novos vetores de ataque.
- ✓ Os atacantes descobrem APIs ocultas, obsoletas e de fim de vida que foram implantadas fora da visão de segurança.

Como prevenir?

- ✓ Faça o inventário de todos os hosts de API
- ✓ Limite o acesso a tudo que não deve ser público.
- ✓ Trate todas as APIs como se fossem internas para maximizar as proteções de segurança.
- ✓ O desenvolvimento de aplicativos, segurança e grupos de negócios devem concordar, documentar e seguir um processo de publicação de API.
- ✓ Limite o acesso aos dados de produção. Segregar o acesso a dados de produção e não produção.
- ✓ Implementar controles externos adicionais, tal como firewalls de API.
- ✓ Retire adequadamente versões antigas ou correções de segurança backport.
- ✓ Implementar autenticação estrita, redirecionamentos, CORS, etc.

API10. Consumo inseguro de APIs

Explorabilidade	Prevalência	Detecção	Impacto
Médio: 2	Comum: 2	Difícil: 1	Severo: 3

Conceito

Os desenvolvedores tendem a confiar, mas não verificar, em seus endpoints, que interagem com APIs externas ou de terceiros. A exploração bem-sucedida de falhas de segurança nessas APIs pode afetar aqueles que dependem delas.

Como acontece?

- ✓ Os desenvolvedores tendem adotar padrões de seguranças mais fracos, no que diz respeito à validação de entrada, deixando a API vulnerável para ataques.

Como prevenir?

- ✓ Avaliar a postura de segurança da API junto aos provedores de serviços.
- ✓ Certifique-se de que todas as interações da API aconteçam em um canal de comunicação seguro (TLS);
- ✓ Sempre valide os dados recebidos de APIs integradas antes de usá-los.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 27001:2013**: Tecnologia da informação — Técnicas de segurança — Sistemas de gestão da segurança da informação - Requisitos. Rio de Janeiro, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 27002:2013**: Tecnologia da informação — Técnicas de segurança — Código de prática para controles de segurança da informação. Rio de Janeiro, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 27005:2019**: Tecnologia da informação — Técnicas de segurança — Gestão de riscos de segurança da informação. Rio de Janeiro, 2019.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 27701:2019**: Técnicas de segurança — Extensão da ABNT NBR ISO/IEC 27001 e ABNT NBR ISO/IEC 27002 para gestão da privacidade da informação — Requisitos e diretrizes. Rio de Janeiro, 2019.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 31000:2018**: Gestão de Riscos — Diretrizes. Rio de Janeiro, 2018.

BRASIL. Presidência da República. Casa Civil. Subchefia para Assuntos Jurídicos. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais.** Disponível em: < http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2018/Lei/L13709.htm >. Acesso em: 03 abr. 2023.

API Security - About API.gov.au – Governo da Austrália. Disponível em < https://api.gov.au/sections/api-security.html >. Acesso: 03 abr. 2023.

API Standards - Tech at GSA. Disponível em < https://tech.gsa.gov/guides/api_standards/ >. Acesso: 10 abr. 2023. Application programming interfaces (APIs) - Service Manual - GOV.UK (www.gov.uk). Disponível em < https://www.gov.uk/service-manual/technology/application-programming-interfaces-apis >. Acesso: 10 abr. 2023.

BRASIL. Presidência da República. Gabinete de Segurança Institucional. Portaria nº 93, de 26 de setembro de 2019. **Glossário de Segurança da Informação**. Disponível em: <

https://www.in.gov.br/en/web/dou/-/portaria-n-93-de-26-de-setembro-de-2019-219115663 >. Acesso em: 10 abr. 2023.

BRASIL. Presidência da República. Gabinete de Segurança Institucional. Departamento de Segurança da Informação e Comunicações. **Instrução Normativa nº 01**, de 27 de maio de 2020. Brasília, DF, GSI/PR, 2020. Disponível em: < https://www.gov.br/gsi/pt-br/composicao/SSIC/dsic/legislacao/copy_of_IN01_consolidada.pdf>. Acesso em: 10 abr. 2023.

BRASIL. Ministério da Economia. **Estratégia de Governo Digital 2020-2022.** Brasília, DF, GSI/PR, 2020. Disponível em: < https://www.gov.br/governodigital/pt-br/EGD2020 >. Acesso em: 10 abr. 2023.

CENTER INTERNET SECURITY. **CIS Controls**, versão 8, janeiro de 2022. Disponível em: < https://learn.cisecurity.org/cis-controls-download >. Acesso em: 10 abr. 2023.

COMITÊ CENTRAL DE GOVERNANÇA DE DADOS - CCGD. **Guia de Boas Práticas LGPD.** Agosto de 2020. Disponível em: < https://www.gov.br/governodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_lgpd.pdf >. Acesso em: 10 abr. 2023.

CYBER SECURITY AGENCY OF SINGAPORE (CSA). **Importance of Securing Your Application Programming Interface (API)**. Singapura, 2022. Disponível em: < https://www.csa.gov.sg/alerts-advisories/Advisories/2022/ad-2022-011 >. Acesso em: 10 abr. 2023.

INFORMATION COMMISSIONER'S OFFICE - ICO. Disponível em < https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-by-design-and-default/ > Acesso: 10 abr. 2023.

INTERNATIONAL STANDARD. **ISO/IEC 29100:2011**: Information technology — Security techniques — Privacy framework. Genebra, 2011.

INTERNATIONAL STANDARD. **ISO/IEC 29134:2017**: Information technology – Security techniques – Guidelines for privacy impact assessment. Genebra, 2017.

INTERNATIONAL STANDARD. **ISO/IEC 29151:2017**: Information technology — Security techniques — Code of practice for personally identifiable information protection. Genebra, 2017.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **NIST Special Publication 800-95**: Guide to Secure Web Services, 2007.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **NIST Special Publication 800-204**: Security Strategies for Microservices-based Application Systems, 2019.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **NIST Special Publication 800-53 revisão 5**: Security and Privacy Controls for Information Systems and Organizations. Gaithersburg, 2020.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **NIST Framework for Improving**

Critical Infrastructure Cybersecurity: Version 1.1, disponível em < https://www.nist.gov/cyberframework > Acesso em: 10 de abr. 2023.

THE OPEN WEB APPLICATION SECURITY PROJECT (OWASP). **OWASP API Security Top 10 2023**: The Ten Most Critical API Security Risks, 2023. Disponível em: < https://github.com/OWASP/API-Security/tree/master/2023/en/src >. Acesso em: 10 de abr. 2023.

THE OPEN WEB APPLICATION SECURITY PROJECT (OWASP). **Security by Design Principles**. Disponível em: < https://wiki.owasp.org/index.php/Security_by_Design_Principles>. Acesso em: 10 de mai. 2023.

WORLD WIDE WEB CONSORTIUM (W3C). **Device API Privacy Requirements**. Disponível em: < https://www.w3.org/TR/dap-privacy-reqs/ >. Acesso em: 06 de abr. 2023.

https://www.w3.org/TR/2012/NOTE-app-privacy-bp-20120703/>.

Anexo I

Mudanças da Versão 2.0

Este anexo tem a finalidade de fornecer os destaques das mudanças inseridas nesta versão do Guia de Requisitos Mínimos de Privacidade e Segurança para APIs.

Primeiramente, ressalta-se que as mudanças inseridas nesta versão em comparação com a anterior visam a adequação com o Guia do Framework de Privacidade e Segurança da Informação v1 elaborado e publicado pela SGD em novembro de 2022.

Foram realizadas modificações nas seguintes seções: seção sobre aviso preliminar e agradecimentos; introdução e referência de que controle e medidas do Framework de Privacidade e Segurança da Informação são atendidos pelo Guia de Requisitos Mínimos de Privacidade e Segurança para APIs.

Dentre os ajustes pontuais cumpre destacar:

- Remoção do tópico 1" Diretrizes Gerais" para melhor adequação com o Guia do Framework de Privacidade e Segurança da Informação;
- Inserção de tabela com os Elementos de Privacidade na seção "Requisito geral 1.1: Priv*acy by Design*" afim de esclarecer os Requisitos para definições de API e os Requisitos relacionados às expectativas do usuário quanto ao uso de dados;
- Na seção "Requisito 2.1: Privacidade", o texto foi alterado com a finalidade de trazer maior clareza e estar mais adequado ao tema específico de Requisitos Mínimos para privacidade e Segurança para APIs;
- Na seção "Requisito 2.2: Segurança", foram realizadas atualizações nos cards referentes ao top 10, conforme novo documento publicado em 2023;
- Removido a Anexo I anterior, sem prejuízo para o conteúdo e qualidade do documento.