Sistemas Satélites

Manual de Segurança

29/09/2021 – Versão 4.0

Sicredi

Sumário

1. Introdução

Este documento deve servir como um manual para implementação segura de sistemas satélites. O documento possui os seguintes objetivos:

- Auxiliar as Cooperativas a implementarem de maneira segura os seus sistemas satélites, visando a homologação dos mesmos;
- Auxiliar as empresas parceiras do Sicredi no processo de homologação dos sistemas satélites, servindo como base de testes a serem realizados nos sistemas e servidores.

2. Estrutura do documento

Esta seção tem como objetivo servir como um guia de navegação no documento e para facilitar o entendimento de como ele está estruturado quanto aos seus controles técnicos. Muitas das dúvidas que podem surgir ao ler o documento podem ser sanadas através da leitura desta seção.

Este documento está estruturado em três principais seções (Seções 3, 4 e 5), as quais descrevem os controles que devem ser implementados com o objetivo de homologar os sistemas satélites. Cada uma destas seções está organizada em subseções, nas quais os controles estão agrupados conforme aplicabilidade ou categoria.

As subseções estão organizadas de forma a apresentar cada controle técnico, uma referência para um material de auxílio que pode ser utilizado para consulta de maiores informações e a aplicabilidade daquele controle em questão. Abaixo pode ser visto um exemplo de estrutura de um controle do manual:

1) Nesta linha é descrito o controle técnico que deve ser implementado, seja ele no ambiente, na infraestrutura ou no próprio sistema.
    a. No primeiro sub-item é feita a referência para um documento de apoio e, possivelmente, o número do controle no documento, podendo este estar anexo ou ser um documento referência de mercado (maiores informações ver Seção 6).
    b. O segundo sub-item apresenta a aplicabilidade do controle. A aplicabilidade pode variar conforme local em que o sistema está implementado e o tipo de sistema.

O documento possui, também, uma seção de anexos e referências (Seção 6). Nessa seção são apresentados documentos utilizados como apoio na elaboração do presente manual e que podem contribuir para uma melhor compreensão do processo de homologação de sistemas satélites e dos controles técnicos. Todas as normas mencionadas neste documento também estão anexas.

Por tratar-se de um documento com viés técnico, na parte final foram adicionadas duas seções (Seções 7 e 8), as quais contêm os principais acrônimos e explicação dos termos técnicos utilizados. Estas seções devem ser utilizadas como apoio na leitura do documento.

Por fim, há uma seção de dúvidas (Seção 9). Essa seção visa direcionar o colaborador que estiver lendo o documento para o canal adequado em caso de dúvidas.

3. Controles de ambiente e infraestrutura da Cooperativa

Seguindo as boas práticas de segurança como NIST SP 800-115 e ISO/IEC 27002, os controles descritos neste capítulo devem ser implementados na infraestrutura de sustentação do sistema, no ambiente da Cooperativa, e quando aplicável, na infraestrutura de nuvem/SaaS. Os controles podem ser consultados, em forma de planilha, no anexo *Controles de Ambiente e de Infraestrutura* ao final deste documento.

3.1. Infraestrutura do sistema

Os controles descritos nesta seção são aplicáveis à infraestrutura do sistema.

1) Realizar backup: devem ser adotadas medidas para proteção das informações contra perda e indisponibilidade. Cópias de segurança das informações devem ser efetuadas e testadas.
   a. Controle: item 3.3 da Norma – Segurança Cibernética
   b. **Aplicabilidade:** infraestrutura do Sistema.

2) Implementar funcionalidades mínimas de geração de registros (*logs*) de eventos da aplicação e servidores que a sustentam.
   a. Controle: Item 3.2 da Norma – Segurança Cibernética.
   b. **Aplicabilidade:** infraestrutura do Sistema.

3) Aplicar atualizações de segurança periódicas contemplando os serviços e servidores que sustentam a aplicação.
   a. **Controle:** Item 3.4 da Norma – Segurança Cibernética.
   b. **Aplicabilidade:** infraestrutura do Sistema.

4) Desativar compartilhamentos de arquivos com acesso anônimo, irrestrito e/ou público em estações de trabalho e servidores: não pode haver compartilhamento de arquivos em pastas com acesso anônimo, irrestrito e/ou público em estações de trabalho e servidores. O compartilhamento indevido de pastas e arquivos pode ser utilizado para obter informações confidenciais e consequentemente ocorrer vazamento de informações corporativas.
   a. Controle: item 3.20 da Norma – Segurança Cibernética.
   b. **Aplicabilidade:** infraestrutura do sistema.

5) Utilizar somente usuários nominais para acessos administrativos nos servidores.
   a. Controle: item 3.5 da Norma – Segurança Cibernética.
   b. Aplicabilidade: infraestrutura do sistema.

6) Utilizar antivírus corporativo em servidores Windows que sustentam a aplicação.
   a. Controle: Item 3.5 da Norma – Segurança Cibernética.
   b. **Aplicabilidade:** infraestrutura do sistema.

7) Integrar o processo de solicitação, autenticação e autorização de acesso da aplicação com os serviços de diretórios corporativos.
   a. Controle: itens 2.1.2 da Norma – Gestão de Identidades e Acessos
   b. **Aplicabilidade:** infraestrutura do sistema.

8) Adequar credenciais de usuários locais à política de senha corporativa.
   c. Controle: itens 2.2.1.1 da Norma – Gestão de Identidades e Acessos
   d. **Aplicabilidade:** infraestrutura do sistema.

3.2. Infraestrutura SaaS

Os controles descritos nesta seção são aplicáveis somente às aplicações hospedadas em nuvem, seguindo o modelo SaaS.

1) Solução Anti-DDoS: o ambiente deve estar protegido contra ataques de negação de serviço distribuídas visando manter o desempenho e disponibilidade do serviço. Essa proteção pode ser adquirida junta ao provedor de serviços em nuvem.
   a. **Controle**: NIST-800-53 – SC5 - *Denial of Service Protection*.
   b. **Aplicabilidade:** infraestrutura SaaS.

2) Certificação Tier 2 ou superior: o datacenter do provedor de serviços deve possuir, no mínimo, certificação Tier II considerando seu desempenho e tempo de atividade (*uptime*).
   a. **Controle**: ANSI/TIA-942-A *Telecommunications Infrastructure Standard for Data Centers*.
   b. **Aplicabilidade:** infraestrutura SaaS.

3) LDAP/federação: as aplicações e sistemas do Sicredi devem ser autenticados de forma integrada em serviços de diretórios corporativos.
   a. Controle: itens 3.20 da Norma – Segurança Cibernética.
   b. **Aplicabilidade:** infraestrutura SaaS.

4) Solução WAF: a aplicação deve estar protegida por alguma tecnologia de WAF (*Web Application Firewall*). Essa proteção pode ser adquirida junto ao provedor de serviços em nuvem.
   a. **Controle**: NIST-800-95 – *Guide to Secure Web Services*.
   b. Aplicabilidade: infraestrutura SaaS.

5) *Firewall* IDS/IPS: o ambiente que sustenta a aplicação deve estar protegido por alguma tecnologia de *IDS* (Sistema de Detecção de Intrusão) ou *IPS* (Sistema de Prevenção de Intrusão). Essa proteção pode ser adquirida junto ao provedor de serviços em nuvem.

a. **Controle**: NIST-800-94 – *Guide to Intrusion Detection and Prevention Systems*.
b. **Aplicabilidade:** infraestrutura SaaS.

6) Isolamento de ambiente/dados Sicredi: o ambiente que sustenta a aplicação deve ser dedicado, isto é, não deve ser compartilhado com outros clientes do provedor de serviços. Isso também se aplica para o armazenamento de informações. Isso pode ser alinhado junto ao provedor de serviços em nuvem.
   a. **Controle**: NIST-800-53 – SC39 *Process Isolation* e SC32 *Information System Partitioning*.
   c. **Aplicabilidade:** infraestrutura SaaS.

7) Utilizar antivírus em servidores que sustentam a aplicação.
   a. **Controle**: Item 3.5 da Norma – Segurança Cibernética
   b. **Aplicabilidade:** infraestrutura SaaS.

8) Multifator para acesso de administradores: O acesso de usuários para administração do ambiente que hospeda o sistema deve ser realizado através mais de um fator de autenticação.
   a. **Controle**: Item 3.20 da Norma – Segurança Cibernética
   b. **Aplicabilidade:** infraestrutura SaaS.

9) O armazenamento e processamento de dados somente deve ser realizado através de países/regiores autorizadas pelo Banco Central, conforme comunicado: https://www.bcb.gov.br/estabilidadefinanceira/exibenormativo?tipo=Comunicado&numero=31999
   a. **Controle**: Normativo regulatório – Banco Central
   b. **Aplicabilidade:** infraestrutura SaaS.

4. Controles para aplicações web

Essa seção do documento apresenta os controles que devem ser implementados contemplando todas as camadas e aspectos relacionados ao desenvolvimento de aplicações web, descritos no projeto *Application Security Verification Standard* (ASVS) da OWASP: Arquitetura, Autenticação, Gerenciamento de Sessão, Controle de Acessos, Validação, Criptografia, Tratamento de Erros, Proteção de Dados, Comunicação, Entrada de Dados, API, Lógica de Negócio, Recursos, Arquivos e Configuração.

Os controles estão organizados conforme as categorias listadas acima, com uma breve descrição, aplicabilidade e vínculo com o correspondente controle no projeto ASVS, o qual pode ser encontrado no anexo *OWASP Application Security Verification Standard 4.0 (ASVS)* ao final deste documento.

4.1. Arquitetura

Esta subseção cobre os aspectos primários de uma arquitetura de segurança: disponibilidade, confidencialidade, integridade, não-repúdio e privacidade. Todos esses princípios de segurança devem ser construídos e estar presentes em todas as aplicações.

1) Verificar que a aplicação não usa tecnologias *client-side* sem suporte, inseguras ou depreciadas.
   a. **Controle ASVS:** 1.14.6.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

2) Verificar que são usadas contas de sistema operacional únicas ou com privilégios mínimos para todos os componentes, serviços e servidores de aplicação.
   a. **Controle ASVS:** 1.2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar a existência de mecanismos que reforcem os controles de acesso, tais como *gateways* de controle acesso, servidores e funções *serverless*. Não aplicar controles de acessos somente em *client-side*.
   a. **Controle ASVS:** 1.4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que a aplicação usa um mecanismo de controle de acessos único e robusto para restringir o acesso a dados e recursos protegidos.
   a. **Controle ASVS:** 1.4.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar que são considerados atributos e características na autorização do acesso a funcionalidades e dados, não somente a função (*role*).
   a. **Controle ASVS:** 1.4.5.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

6) Verificar que não é utilizada serialização na comunicação com destinos não confiáveis. Se não for possível, garantir que são aplicados controles de integridade e criptografia para transferência de dados sensíveis.
   a. **Controle ASVS:** 1.5.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que a validação de entrada de dados é realizada adequadamente em uma camada confiável de serviços.
   a. **Controle ASVS:** 1.5.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

8) Verificar que existe uma abordagem e formato padronizado para registro de logs da aplicação.
   a. **Controle ASVS:** 1.7.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.2. Autenticação

Esta subseção trata de autenticação, a forma de estabelecer ou confirmar a autenticidade de algo ou alguém. Quando possível, o sistema deve possuir autenticação integrada com os serviços de diretório corporativos.

1) Verificar que controles anti automação são efetivos na mitigação de ataques do tipo *credential stuffing*, força-bruta e bloqueios de contas.
   a. **Controle ASVS:** 2.2.1.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

2) Verificar que contas compartilhadas ou padrão não são utilizadas, tais como "*root*", "*admin*" ou "*sa*".
   a. **Controle ASVS:** 2.5.4.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

4.3. Gerenciamento de Sessão

Esta subseção trata de um dos principais componentes de qualquer aplicação *web* ou API *stateful*. O gerenciamento de sessão é o mecanismo através do qual se controla e mantém o estado de um usuário ou dispositivo que esteja interagindo com a aplicação.

1) Verificar que a aplicação nunca revela os *tokens* de sessão nos parâmetros da URL ou mensagens de erro.
   a. **Controle ASVS:** 3.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que a aplicação gera um novo *token* de sessão assim que o usuário se autentica.
   a. **Controle ASVS:** 3.2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que a aplicação somente armazena o *token* de sessão no navegador usando métodos seguros, por exemplo *cookies* seguros ou armazenamento de sessão do HTML 5.
   a. **Controle ASVS:** 3.2.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que os *tokens* de sessão são gerados usando algoritmos criptográficos seguros.
   a. **Controle ASVS:** 3.2.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar que o processo de *logout* e expiração de sessão invalidam o *token* de sessão de tal forma que usar o botão "voltar" ou as aplicações conectadas não consigam retomar uma sessão autenticada, incluindo aplicações confiáveis.
   a. **Controle ASVS:** 3.3.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que *tokens* de sessão baseados em *cookies* possuem o atributo "*Secure"* configurado.
   a. **Controle ASVS:** 3.4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que *tokens* de sessão baseados em *cookies* possuem o atributo "*HttpOnly"* configurado.
   a. **Controle ASVS:** 3.4.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

8) Verificar que *tokens* de sessão baseados em *cookies* utilizam o atributo "*SameSite"* para limitar a exposição a ataques CSRF.
   a. **Controle ASVS:** 3.4.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

9) Verificar que *tokens* de sessão baseados em *cookies* utilizam o prefixo *"__Host-"* para garantir confidencialidade ao *cookie* de sessão.
   a. **Controle ASVS:** 3.4.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

10) Verificar que tokens de sessão *stateless* usem assinatura digital, criptografia e outras medidas para proteger contra ataques de adulteração, *enveloping*, repetição, cifras nulas e substituição de chaves.
   a. **Controle ASVS:** 3.5.3.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

4.4. Controle de Acessos

Esta subseção trata do controle de acessos, ou autorização, cujo conceito é o de conceder acessos a recursos somente para aqueles que possuem permissão.

1) Verificar que as regras de controle de acessos da aplicação são aplicadas em uma camada confiável de serviços.
   a. Controle ASVS: 4.1.1.
   b. Aplicabilidade: todos os sistemas satélites.

2) Verificar se todos os atributos de dados, usuários e políticas usadas pelos controles de acesso não podem ser manipulados pelos usuários finais a menos que especificamente autorizado.
   a. Controle ASVS: 4.1.2.
   b. Aplicabilidade: todos os sistemas satélites.

3) Verificar que é aplicado o princípio do menor privilégio – usuário deve somente ter acesso a funções, arquivos, URLs, controladores, serviços e outros recursos que ele possua autorização específica.
   a. Controle ASVS: 4.1.3.
   b. Aplicabilidade: todos os sistemas satélites.

4) Verificar que é aplicado o princípio de negar por padrão através do qual novos usuários/funções iniciam com a mínima ou nenhuma permissão e que usuários/funções não recebam acesso a novas funcionalidades a menos que ele seja explicitamente concedido.
   a. **Controle ASVS:** 4.1.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar se os controles de acesso falham de forma segura (*fail safe*) incluindo quando ocorre uma exceção.
   a. **Controle ASVS:** 4.1.5.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que dados sensíveis e APIs são protegidos contra ataques direcionados de criação de objetos, leitura, atualização e deleção de registros.
   a. **Controle ASVS:** 4.2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que a aplicação ou *framework* possui um mecanismo anti CSRF para proteger funcionalidades autenticadas, e um mecanismo anti automação ou anti CSRF para proteger funcionalidades não autenticadas.
   a. **Controle ASVS:** 4.2.2.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

8) Verificar se a navegação pelos diretórios está desabilitada a menos que seja explicitamente desejada. Adicionalmente, as aplicações não devem permitir a descoberta e divulgação de metadados de arquivos ou diretórios, tais como *Thumbs.db, .DS_Store, .git* ou *.svn*.
   a. **Controle ASVS:** 4.3.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.5. Validação

Esta subseção trata de validação. O problema de segurança mais presente em aplicações *web* é a falta de validação adequada dos dados enviados pelo cliente ou ambiente antes de usá-los. Essa fragilidade leva a quase todas as vulnerabilidades relevantes em aplicações *web*, tais como *Cross-Site Scripting* (XSS), *SQL Injection* e *buffer overflows*.

1) Verificar que a aplicação possui proteções contra ataques de manipulação de parâmetros HTTP, principalmente se o *framework* da aplicação não faz distinção sobre a origem dos parâmetros da requisição (GET, POST, *cookies*, cabeçalho ou variáveis de ambiente).
   a. **Controle ASVS:** 5.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que todas as entradas (campos de formulários HTML, requisições REST, parâmetros de URL, cabeçalhos HTTP, *cookies*, arquivos *batch*, *feeds RSS*, etc) são validados através de validação positiva (*whitelist*).

   a. **Controle ASVS:** 5.1.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que dados estruturados são fortemente *tipados* e validados através de um esquema bem definido de caracteres, tamanho e padrão (por exemplo, números de cartão de crédito e telefone).
   a. **Controle ASVS:** 5.1.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que dados não estruturados são sanitizados de maneira segura, permitindo somente alguns caracteres e tamanhos.
   a. **Controle ASVS:** 5.2.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar que a aplicação sanitiza as entradas de usuário antes de repassar para sistemas de e-mail, protegendo contra ataques de injeção SMTP ou IMAP.
   a. **Controle ASVS:** 5.2.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que a aplicação não usa a função *eval()* ou outras funcionalidades de execução dinâmica de código. Quando não houver alternativa, toda entrada de usuário incluída precisa ser *sanitizada* ou executada em uma *sandbox* antes de ser, de fato, executada.
   a. **Controle ASVS:** 5.2.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que a aplicação protege contra ataques de injeção de *template* garantindo que toda entrada de usuário é *sanitizada* ou executada em uma *sandbox*.
   a. **Controle ASVS:** 5.2.5.
   b. **Aplicabilidade:** todos os sistemas satélites.

8) Verificar que a aplicação protege contra ataques SSRF, fazendo validação ou *sanitizando* dados não confiáveis ou metadados de arquivos HTTP, tais como nomes de arquivos e campos de entrada em URLs, usando *whitelist* de protocolos, domínios, diretórios e portas.
   a. **Controle ASVS:** 5.2.6.
   b. **Aplicabilidade:** todos os sistemas satélites.

9) Verificar que a aplicação sanitiza, desabilita ou executa em *sandbox* conteúdos *scriptáveis* ou expressões em linguagem de *template* fornecidas pelo usuário, tais como *Markdown, CSS, XSL, BBCode* ou similares.
   a. **Controle ASVS:** 5.2.8.
   b. **Aplicabilidade:** todos os sistemas satélites.

10) Verificar que há validação de contexto, automatizado ou manual, que faça o *escape* apropriado protegendo contra XSS refletido, armazenado ou *DOM-based*.

   a. **Controle ASVS:** 5.3.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

11) Verificar que a seleção de dados ou consultas em bancos de dados (*SQL, HQL, ORM, NoSQL*) são feitas com consultas parametrizadas, *ORMs*, *frameworks* de entidade, ou são protegidas de alguma outra forma contra ataques de injeção em bancos de dados.
   a. **Controle ASVS:** 5.3.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

12) Verificar que quando parametrização ou mecanismos mais seguros não são usados, codificação de saídas específicas para o contexto são usadas para proteger contra ataques de injeção, tais como o *escape* de SQL para proteger contra injeção de SQL.
   a. **Controle ASVS:** 5.3.5.
   b. **Aplicabilidade:** todos os sistemas satélites.

13) Verificar que a aplicação está protegida contra ataques de injeção *JavaScript* ou *JSON*.
   a. **Controle ASVS:** 5.3.6.
   b. **Aplicabilidade:** todos os sistemas satélites.

14) Verificar que a aplicação está protegida contra ataques de injeção LDAP, ou que controles específicos de segurança para prevenir injeção LDAP foram implementados.
   a. **Controle ASVS:** 5.3.7.
   b. **Aplicabilidade:** todos os sistemas satélites.

15) Verificar que a aplicação está protegida contra injeção de comandos em SOs e que chamadas de sistemas operacionais são parametrizadas.
   a. **Controle ASVS:** 5.3.8.
   b. **Aplicabilidade:** todos os sistemas satélites.

16) Verificar que a aplicação está protegida contra ataques de Inclusão Local de Arquivos (*LFI*) ou Inclusão Remota de Arquivos (*RFI*).
   a. **Controle ASVS:** 5.3.9.
   b. **Aplicabilidade:** todos os sistemas satélites.

17) Verificar que a aplicação está protegida contra ataques de injeção *XPath* e *XML*.
   a. **Controle ASVS:** 5.3.10.
   b. **Aplicabilidade:** todos os sistemas satélites.

18) Verificar que a aplicação usa *strings memory-safe*, cópias de memória seguras e ponteiros aritméticos para detectar ou prevenir *stack*, *buffer* ou *heap overflow*.
   a. **Controle ASVS:** 5.4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

19) Verificar que sinais, variação e técnicas de validação de entradas são usadas para prevenir *integer overflows*.
   a. **Controle ASVS:** 5.4.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

20) Verificar que é feita verificação de integridade de objetos serializados ou que estes são criptografados para prevenir a criação de objetos maliciosos ou adulteração de dados.
   a. **Controle ASVS:** 5.5.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

21) Verificar que a aplicação restringe corretamente os *parsers* de XML para usarem as configurações mais restritivas possíveis e para garantir que funcionalidades inseguras são desabilitadas, tal como resolver entidades externas, prevenindo ataques XXE.
   a. **Controle ASVS:** 5.5.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

22) Verificar que se utiliza *JSON.parse* quando é feito *parser* de *JSON* em navegadores ou em *backends* baseados em *JavaScript*. Não deve ser usado *eval()* para analisar JSON.
   a. **Controle ASVS:** 5.5.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.6. Criptografia

Esta subseção trata de requisitos de criptografia. O objetivo é que sejam usados para validar a encriptação da aplicação, o gerenciamento de chaves, números aleatórios e operações de *hash*.

1) Verificar que são utilizados algoritmos criptográficos, métodos e bibliotecas validados ou aprovados por órgãos e instituições, em vez de codificados internamente.
   a. **Controle ASVS:** 6.2.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que módulos de blocos conhecidamente inseguros (ex. *ECB*), modos de preenchimento (ex. *PKCS#1 v1.5*), cifras com blocos pequenos (ex. *Triple-DES, Blowfish*), e algoritmos de *hash* fracos (ex. *MD5, SHA1*) não são usados a menos que necessários para garantir compatibilidade com versões anteriores.
   a. **Controle ASVS:** 6.2.5.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

3) Verificar que todos os números aleatórios, nomes aleatórios de arquivos, *GUIDs* aleatórios e *strings* aleatórias são geradas usando o gerador de números aleatórios criptograficamente seguro.
   a. **Controle ASVS:** 6.3.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.7. Tratamento de Erros e Registros de Logs

Esta subseção trata do tratamento de erros e registro de informações em *logs*, cujo objetivo principal é fornecer informações úteis e importantes aos usuários, administradores e equipes de resposta a incidentes. O objetivo não é criar quantidades massivas de *logs*, mas sim que eles tenham alta qualidade e possam ser bem utilizados.

1) Verificar que a aplicação não registra *logs* de credenciais ou detalhes de pagamento. *Tokens* de sessão só devem ser armazenados em formato de *hash*.
   a. **Controle ASVS:** 7.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que a aplicação não registra *logs* de dados sensíveis conforme leis de privacidade locais ou políticas de segurança.
   a. **Controle ASVS:** 7.1.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que os *logs* de segurança estão protegidos contra acesso e modificação não autorizadas.
   a. **Controle ASVS:** 7.3.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que quando um erro inesperado ou um erro de segurança ocorre é exibida uma mensagem genérica e com um identificador único que possa auxiliar o suporte na investigação.
   a. **Controle ASVS:** 7.4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.8. Proteção de Dados

Esta subseção trata da proteção de dados, que se resume em basicamente três elementos chave: confidencialidade, integridade e disponibilidade. A aplicação é responsável por garantir que os dados armazenados nos dispositivos estão criptografados e não podem ser facilmente obtidos, alterados ou divulgados ilicitamente.

1) Verificar que a aplicação define cabeçalhos *anti-cache* para que dados sensíveis não sejam armazenados em *cache* dos navegadores mais modernos.
   a. **Controle ASVS:** 8.2.1.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

2) Verificar se todos os dados sensíveis são enviados para o servidor no corpo da mensagem ou cabeçalho HTTP e que parâmetros URL nunca são usados para enviar dados sensíveis.
   a. **Controle ASVS:** 8.3.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que informações sensíveis e privadas são criptografadas usando algoritmos validados ou aprovados por órgãos e instituições que fornecem confidencialidade e integridade.
   a. **Controle ASVS:** 8.3.7.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.9. Segurança na Comunicação

Esta subseção trata da segurança na comunicação. As boas práticas recomendam alterações frequentes nas configurações de TLS, seguidamente por razões de quebras nos algoritmos e cifras criptográficas. Dessa forma, as configurações devem ser revisadas periodicamente para garantir que configurações seguras estão sendo aplicadas na comunicação.

1) Verificar que versões seguras do TLS são usadas em todas conexões com clientes, e que falhas de conexão não recaem em conexões inseguras.
   a. **Controle ASVS:** 9.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que versões antigas dos protocolos SSL e TLS, algoritmos, cifras e configurações estão desabilitadas, tais como SSLv2, SSLv3, TLS 1.0 e TLS 1.1.
   a. **Controle ASVS:** 9.1.3.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

3) Verificar que todas as conexões do servidor usam certificados TLS confiáveis. Nos casos de certificados internos ou auto assinados, o servidor deve ser configurado para confiar somente em CAs e certificados auto assinados internos específicos.
   a. **Controle ASVS:** 9.2.1.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

4) Verificar se todas as conexões criptografadas com sistemas internos que envolvem informações ou funções sensíveis são autenticadas.
   a. **Controle ASVS:** 9.2.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.10. Controle de Códigos Maliciosos

Esta subseção trata de requisitos de verificação contra códigos maliciosos. A identificação de todos os códigos maliciosos é algo praticamente impossível, então os esforços devem ser direcionados para garantir que os códigos utilizados e executados não sejam conhecidamente maliciosos ou possuam funcionalidades indesejadas.

1) Verificar que a aplicação não solicita permissões desnecessárias ou excessivas a dispositivos ou sensores relacionados a privacidade, tais como contatos, câmeras, microfones ou localização.
   a. **Controle ASVS:** 10.2.2.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

4.11. Lógica de Negócios

Esta subseção trata de controles para lógica de negócios. Essa categoria é particular para cada aplicação, então há poucos controles que se apliquem a todas as aplicações. A segurança para lógica de negócios deve ser observada para cada aplicação para proteger contra ameaças externas, pois não é algo que um *web application firewall* (WAF) ou canal criptografado de comunicação possa proteger.

1) Verificar que a aplicação somente irá processar os fluxos de negócio para os usuários em ordem sequencial e sem pular etapas.
   a. **Controle ASVS:** 11.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.12. Arquivos e Recursos

Esta subseção trata da proteção de arquivos e recursos. O objetivo é garantir que arquivos não confiáveis sejam tratados adequadamente e de forma segura, e que arquivos obtidos de fontes não confiáveis sejam armazenados fora do diretório raiz do servidor e com permissões limitadas.

1) Verificar que a aplicação não aceita arquivos muito grandes que podem encher o armazenamento ou causar um ataque de negação de serviço.
   a. **Controle ASVS:** 12.1.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que arquivos obtidos de fontes não confiáveis são validados quanto ao seu tipo baseado em seu conteúdo.
   a. **Controle ASVS:** 12.2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que o nome do arquivo submetido pelo usuário não é usado diretamente no sistema ou *framework* de arquivos para proteger contra *path traversal*.
   a. **Controle ASVS:** 12.3.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que o nome do arquivo submetido pelo usuário é validado ou ignorado para prevenir o vazamento, criação, atualização ou remoção de arquivos locais.
   a. **Controle ASVS:** 12.3.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar que metadados de arquivos não confiáveis não são usados diretamente com as APIs do sistema ou bibliotecas para proteger contra injeção de comandos em SO (*OS command injection*).
   a. **Controle ASVS:** 12.3.5.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que arquivos obtidos de fontes não confiáveis são armazenados fora do diretório *web root*, com permissões limitadas, preferencialmente com fortes validações.
   a. **Controle ASVS:** 12.4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que arquivos obtidos de fontes não confiáveis são escaneados por antivírus para prevenir o *upload* de arquivos maliciosos.
   a. **Controle ASVS:** 12.4.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

8) Verificar que a camada da *web* está configurada para exibir somente arquivos com extensões específicas, de modo a prevenir vazamento não intencional de informações e código-fonte. Por exemplo, arquivos de backup (ex.: *.bak*), arquivos temporários (ex.: *.swp*), arquivos comprimidos (ex.: *.zip, .tar.gz*), e outras extensões comuns usadas por editores devem ser bloqueadas.
   a. **Controle ASVS:** 12.5.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

9) Verificar que requisições feitas diretamente a arquivos carregados não são executados como conteúdo HTML/JavaScript.
   a. **Controle ASVS:** 12.5.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

10) Verificar que o servidor *web* ou de aplicação está configurado com uma *whitelist* de recursos ou sistemas com os quais o servidor pode enviar requisições ou trocar dados e arquivos.
   a. **Controle ASVS:** 12.6.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

4.13. API

Esta subseção trata de controles para APIs. Os principais objetivos são garantir que: todos os *web services* possuam autenticação, gerenciamento de sessão e autorização adequados; todos os parâmetros que transitam de uma camada insegura para uma camada segura são validados; e que existem controles de segurança para todos os tipos de API.

1) Verificar que todos os componentes da aplicação usam a mesma codificação (*encoding*) e analisadores (*parsers*) para evitar ataques que explorem diferentes *URIs* ou comportamentos de *parse* de arquivos que podem ser usados em ataques SSRF e RFI.
   a. **Controle ASVS:** 13.1.1.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

2) Verificar que o acesso a funções de administração e gerenciamento estão limitadas a administradores autorizados.
   a. **Controle ASVS:** 13.1.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que as URLs das APIs não expõem informações sensíveis, tais como chaves de API, *tokens* de sessão, etc.
   a. **Controle ASVS:** 13.1.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que requisições cujo tipo de conteúdo esteja faltando ou seja inesperado são rejeitadas com os cabeçalhos apropriados (*HTTP response status 406 Unacceptable* ou *415 Unsupported Media Type*).
   a. **Controle ASVS:** 13.1.5.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

5) Verificar que os métodos *HTTP RESTful* são possibilidades válidas para o usuário ou ação, para prevenir que usuários comuns usem os métodos DELETE ou PUT em APIs ou recursos protegidos.
   a. **Controle ASVS:** 13.2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que há validação da estrutura do *JSON* antes de aceitar qualquer entrada.
   a. **Controle ASVS:** 13.2.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que os *web services RESTful* que utilizem *cookies* estão protegidos contra *CSRF* através do uso de pelo menos um dos seguintes controles: padrão duplo ou triplo de envio de *cookies*, *CSRF nonces* ou verificações dos cabeçalhos de origem.
   a. **Controle ASVS:** 13.2.3.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

8) Verificar que os serviços *REST* explicitamente verificam se o cabeçalho *Content-Type* é o esperado, tal como *application/xml* ou *application/JSON*.
   a. **Controle ASVS:** 13.2.5.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

9) Verificar que existe validação da estrutura *XSD* para garantir a formatação do documento XML, seguido pela validação de cada campo de entrada antes de processar qualquer dado.
   a. **Controle ASVS:** 13.3.1.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

4.14. Configuração

Esta subseção trata da configuração das aplicações. O objetivo é garantir que: o ambiente de construção da aplicação seja seguro e automatizável; componentes inseguros e desatualizados não sejam utilizados pela aplicação; e que a configuração seja segura por padrão e as liberações sejam feitas conforme necessidade.

1) Verificar que todas as funcionalidades desnecessárias, documentação, exemplos e configurações foram removidas, tais como aplicações, documentação de plataforma e usuários padrão ou de exemplos.
    a. **Controle ASVS:** 14.2.2.
    b. **Aplicabilidade:** todos os sistemas satélites.

2) Verificar que o modo de *debug* está desabilitado no servidor *web* ou de aplicação e no *framework* em ambiente produtivo, para eliminar a possibilidade de *debug* de funcionalidades, consoles de desenvolvedores, e divulgação de informações de segurança não intencionais.
    a. **Controle ASVS:** 14.3.2.
    b. **Aplicabilidade:** todos os sistemas satélites.

3) Verificar que foi definido um *content security policy* (CSPv2) para ajudar a mitigar o impacto de ataques do tipo XSS, tais como HTML, DOM, JSON e *JavaScript*.
    a. **Controle ASVS:** 14.4.3.
    b. **Aplicabilidade:** todos os sistemas satélites.

4) Verificar que todas as respostas contêm o cabeçalho *X-Content-Type-Options: nosniff*.
    a. **Controle ASVS:** 14.4.4.
    b. **Aplicabilidade:** todos os sistemas satélites.

5) Verificar que o cabeçalho *HSTS* está incluído em todas as respostas e para todos os subdomínios, tal como *Strict-Transport-Security: max-age=15724800 includeSubdomains*.
    a. **Controle ASVS:** 14.4.5.
    b. **Aplicabilidade:** todos os sistemas satélites.

6) Verificar que o servidor de aplicação somente aceita os métodos HTTP utilizados pela aplicação ou API.
    a. **Controle ASVS:** 14.5.1.
    b. **Aplicabilidade:** todos os sistemas satélites.

7) Verificar que o cabeçalho *Origin* não é usado para autenticação ou decisões de controle de acesso, considerando que esse cabeçalho pode facilmente ser manipulado por um atacante.
    a. **Controle ASVS:** 14.5.2.
    b. **Aplicabilidade:** todos os sistemas satélites.

8) Verificar que o cabeçalho de compartilhamento de recursos entre domínios (*CORS*) *Access-Control-Allow-Origin* usa uma *whitelist* restrita de domínios confiáveis, e não suporta origem nula (*null*).
    a. **Controle ASVS:** 14.5.3.
    b. **Aplicabilidade:** todos os sistemas satélites.

9) Verificar que cabeçalhos HTTP adicionados por um *proxy* confiável ou dispositivos de SSO, tais como *token bearer*, são autenticados pela aplicação.
   a. **Controle ASVS:** 14.5.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

5. Controles para aplicativos móveis

Esta seção do documento apresenta os controles que devem ser implementados contemplando todas as camadas e aspectos relacionados ao desenvolvimento de aplicativos móveis, descritos no projeto *Mobile Application Security Verification Standard* (MASVS) da OWASP: Arquitetura, Design, Modelagem de Ameaças, Armazenamento de Dados e Privacidade, Criptografia, Autenticação e Gerenciamento de Sessão, Comunicação, Interação com as Plataformas, Qualidade de Código, Configurações de *Build* e Engenharia Reversa.

Os controles estão organizados conforme categorias listadas acima, com uma breve descrição, aplicabilidade e vínculo com o correspondente controle no projeto MASVS, o qual pode ser encontrado no anexo *OWASP Mobile Application Security Verification Standard 1.1 (MASVS)* ao final deste documento.

5.1. Arquitetura, Design e Modelagem de Ameaças

Esta subseção lista requisitos que pertencem à arquitetura e design do aplicativo. Além dos controles técnicos, devem haver processos estabelecidos para garantir que segurança foi considerada ao planejar a arquitetura dos aplicativos móveis, e que o papel funcional e de segurança de cada componente do sistema é conhecido.

1) Os controles de segurança nunca devem ser executados somente em *client-side*, mas sim nos respectivos *endpoints* remotos.
   a. **Controle MASVS:** 1.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Existe um mecanismo para forçar atualizações do aplicativo móvel.
   a. **Controle MASVS:** 1.9.
   b. **Aplicabilidade:** sistemas expostos para a Internet e sistemas em nuvem.

5.2. Armazenamento e Privacidade de Dados

Esta subseção trata da proteção de dados sensíveis, tais como credenciais de acesso e dados pessoais. Em primeiro lugar, dados sensíveis podem ser expostos a outros aplicativos sendo executados no mesmo dispositivo sem intenção, caso o sistema operacional use mecanismos como *Interprocess Communication* (IPC) inadequadamente. Além disso, dispositivos móveis podem ser perdidos ou roubados mais facilmente

comparado a outros dispositivos, então ter acesso físico aos dispositivos é um cenário mais provável.

Dessa forma, proteções adicionais devem ser implementadas para fazer com que a recuperação de dados sensíveis seja um processo mais difícil.

1) As funcionalidades de armazenamento do sistema são usadas adequadamente para armazenar dados sensíveis, tais como informações de identificação pessoal, credenciais de usuário ou chaves criptográficas.
   a. **Controle MASVS:** 2.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Nenhum dado sensível é armazenado nos *logs* da aplicação.
   a. **Controle MASVS:** 2.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Nenhum dado sensível é compartilhado com terceiros.
   a. **Controle MASVS:** 2.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) *Cache* de teclado é desabilitada em entradas de texto que processem dados sensíveis.
   a. **Controle MASVS:** 2.5.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Nenhum dado sensível, como senhas ou *pins*, é exposto através da interface de usuário.
   a. **Controle MASVS:** 2.7.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Nenhum dado sensível está incluído em *backups* gerados pelo sistema operacional do dispositivo móvel.
   a. **Controle MASVS:** 2.8.
   b. **Aplicabilidade:** todos os sistemas satélites.

5.3. Criptografia

Esta subseção trata da criptografia, um recurso essencial quando se trata de proteger dados armazenados em dispositivos móveis. O objetivo dos controles abaixo é garantir que a aplicação utilize protocolos, algoritmos e bibliotecas aderentes às melhores práticas de mercado.

1) O aplicativo não depende de criptografia simétrica com chaves *hardcoded* como único método de criptografia.
   a. **Controle MASVS:** 3.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Todos os números aleatórios são gerados usando um gerador de números aleatórios seguro.

a. **Controle MASVS:** 3.6.
b. **Aplicabilidade:** todos os sistemas satélites.

5.4. Autenticação e Gerenciamento de Sessão

Esta subseção trata da autenticação e do gerenciamento de sessão. A autenticação de usuários está presente na arquitetura da grande maioria dos aplicativos móveis, então deve haver controles específicos para tratar sobre como ela é feita e como a sessão dos usuários é tratada em cada acesso.

1) Se o aplicativo fornece aos usuários acesso a serviços remotos, alguma forma de autenticação, tal como usuário e senha, é realizada no *endpoint* remoto.
   a. **Controle MASVS:** 4.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Caso o gerenciamento de sessão seja *stateful*, o *endpoint* remoto utiliza identificadores de sessão gerados aleatoriamente para autenticar as requisições dos clientes sem enviar suas credenciais.
   a. **Controle MASVS:** 4.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Caso a autenticação seja baseada em *tokens stateless*, o servidor fornece um *token* que seja assinado usando um algoritmo seguro.
   a. **Controle MASVS:** 4.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

5.5. Comunicação de Rede

Esta subseção trata da comunicação de rede. O objetivo dos controles é garantir a confidencialidade e integridade das informações trocadas entre o aplicativo móvel e os serviços remotos do servidor. Minimamente, o aplicativo móvel deve estabelecer um canal seguro e criptografado para a comunicação utilizando protocolo TLS com configurações adequadas.

1) Os dados são criptografados na rede usando TLS 1.2 ou superior. O canal seguro é usado consistentemente em todo o aplicativo.
   a. **Controle MASVS:** 5.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) O aplicativo verifica o certificado X.509 no *endpoint* remoto quando um canal seguro é estabelecido. Somente certificados assinados por CAs confiáveis são aceitos.
   a. **Controle MASVS:** 5.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) O aplicativo usa sua própria loja de certificados ou faz a pinagem do certificado ou chave pública do servidor, e posteriormente não estabelece conexões com

dispositivos que apresentem certificados ou chaves diferentes, mesmo que assinados por CAs confiáveis.

a. **Controle MASVS:** 5.4.
b. **Aplicabilidade:** todos os sistemas satélites.

5.6. Interação com as Plataformas

Esta subseção trata da interação do aplicativo com as plataformas. Os controles têm como objetivo garantir que o aplicativo usa as APIs e os componentes padrões das plataformas de maneira segura. Tratam, também, de controles sobre a comunicação entre os aplicativos (IPC).

1) O aplicativo solicita somente as permissões necessárias.
   a. **Controle MASVS:** 6.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) Todas as entradas de fontes externas e do usuário são validadas e, se necessário, sanitizadas. Isso inclui dados recebidos pela interface de usuário, mecanismos de IPC e fontes de rede.
   a. **Controle MASVS:** 6.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) O aplicativo não expõe funções sensíveis através de URLs customizadas.
   a. **Controle MASVS:** 6.3.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) *WebView* está configurado para permitir somente o menor conjunto de manipuladores de protocolos necessários (idealmente, somente *https* é suportado). Manipuladores potencialmente perigosos, como *file*, *tel* e *app-id* estão desabilitados.
   a. **Controle MASVS:** 6.6.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) Se métodos nativos do aplicativo estão expostos a algum *WebView*, verificar que o *WebView* somente renderiza *JavaScript* contido no pacote do aplicativo.
   a. **Controle MASVS:** 6.7.
   b. **Aplicabilidade:** todos os sistemas satélites.

6) Se houver desserialização de objetos, deve ser implementado usando APIs seguras de serialização.
   a. **Controle MASVS:** 6.8.
   b. **Aplicabilidade:** todos os sistemas satélites.

5.7. Qualidade de Código e Configurações de *Build*

Esta subseção trata da qualidade de código e das configurações feitas em tempo de compilação. O objetivo é garantir que boas práticas de desenvolvimento seguro são

seguidas ao desenvolver o aplicativo e que as funcionalidades de segurança oferecidas pelos compiladores estão ativadas.

1) O aplicativo é assinado e provisionado com um certificado válido, cuja chave privada é protegida adequadamente.
   a. **Controle MASVS:** 7.1.
   b. **Aplicabilidade:** todos os sistemas satélites.

2) O aplicativo foi construído em modo *release* com configurações apropriadas para publicação (ex.: *debug* desativado).
   a. **Controle MASVS:** 7.2.
   b. **Aplicabilidade:** todos os sistemas satélites.

3) Os códigos para *debug* foram removidos, e o aplicativo não registra em *logs* mensagens de *debug* e informações detalhadas de erros.
   a. **Controle MASVS:** 7.4.
   b. **Aplicabilidade:** todos os sistemas satélites.

4) O aplicativo captura e trata as possíveis exceções.
   a. **Controle MASVS:** 7.6.
   b. **Aplicabilidade:** todos os sistemas satélites.

5) A lógica de tratamento de erros em controles de segurança nega os acessos por padrão.
   a. **Controle MASVS:** 7.7.
   b. **Aplicabilidade:** todos os sistemas satélites.

5.8. Engenharia Reversa

Esta subseção trata de engenharia reversa do aplicativo. O objetivo é impedir que um atacante consiga explorar o aplicativo de forma a obter acesso ao código-fonte original, podendo alterar o código e analisar o comportamento do aplicativo.

1) É aplicada ofuscação para proteger o código-fonte com o objetivo de impedir a visualização do código através de análise dinâmica.
   a. **Controle MASVS:** 8.9.
   b. **Aplicabilidade:** todos os sistemas satélites.

6. Anexos e Referências

6.1. OWASP Application Security Verification Standard 4.0.2 (ASVS)

https://owasp.org/www-project-application-security-verification-standard/

6.2. OWASP Mobile Application Security Verification Standard 1.2 (MASVS)

https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide

6.3. Referências do NIST

NIST 800-44

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-44ver2.pdf

NIST 800-52

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf

NIST 800-53

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

NIST 800-57

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

NIST 800-94

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf

NIST 800-95

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf

NIST 800-115

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

NIST 800-171

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171.pdf

7. Acrônimos

API: *Application Programming Interface*.

ASVS: *Application Security Verification Standard*.

CA: *Certificate Authority*.

CDN: *Content Delivery Network*.

CORS: *Cross-Origin Resource Sharing*.

CSP: *Content Security Policy*.

CSRF: *Cross-Site Request Forgery*.

DNS: *Domain Name System*.

HQL: *Hibernate Query Language*.

HSTS: HTTP *Strict Transport Security*.

IMAP: *Internet Message Access Protocol*.

IPC: *Interprocess Communication*.

MASVS: *Mobile Application Security Verification Standard*.

NIST: *National Institute of Standards and Technology*.

NoSQL: *Not Only SQL*.

ORM: *Object-Relational Mapping*.

OTP: *One-Time Password*.

OWASP: *Open Web Application Security Project*.

PSTN: *Public Switched Telephone Network*.

SaaS: *Software as a Service*.

SMS: *Short Message Service*.

SMTP: *Simple Mail Transfer Protocol*.

SO/OS: Sistema Operacional/*Operating System*.

SQL: *Structured Query Language*.

SSL: *Secure Sockets Layer*.

SSO: *Single Sign-On*.

SSRF: *Server-Side Request Forgery*.

SVG: *Scalable Vector Graphics*.

TLS: *Transport Layer Security*.

XSD: XML *Schema Definition*.

XSS: *Cross-Site Scripting*.

8. Glossário

Build: processo de compilação e construção do software.

Cache: área de memória de acesso rápido.

Checklist: lista de controle.

Client-side: executado no lado cliente, ou seja, no dispositivo do usuário.

Cookies: pacote de dados enviado através de um site de Internet.

Debug: depuração – encontrar e reduzir defeitos em software ou hardware.

Encoding: padrão de codificação.

Endpoint: ponto de extremidade, podendo ser um servidor ou dispositivo de usuário.

Enveloping: inserção de código malicioso de forma mascarada.

Escape: tratamento dado a conjuntos de caracteres.

Gateway: porta de entrada centralizada.

Hardcoded: fixo e inalterável, geralmente no código-fonte.

Hash: função que mapeia dados de comprimento variável para dados de comprimento fixo através uma série de cálculos.

Stack, buffer e *heap overflow*: diferentes formas de estouro de representação, isto é, são necessários mais bits para armazenamento ou processamento do que os que foram disponibilizados.

Mainframe: computador de grande porte, dedicado a processamentos de grandes volumes de informações.

Man-in-the-middle: ataque em que o fraudador intercepta e manipula a comunicação entre dispositivos.

Memory-safe: protegidos contra erros de acesso de memória, por exemplo, *overflows*.

Parse: quebra ou divisão de uma entrada em porções menores para serem armazenadas ou manipuladas.

Path traversal: uma das vulnerabilidades mais comuns, presente no CWE/SANS Top 25, utilizado para obter acesso não autorizado a arquivos e diretórios.

Plugins: programa usado para adicionar funções a outros programas maiores.

Proxy: pode ser definido como um intermediário que atua entre o usuário e o servidor.

RESTful: estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados.

Salt: dado aleatório usado como entrada adicional para uma função unidirecional.

Sandbox: utilizado para isolar a execução de programas.

Sanitizado: eliminar trechos de texto que tenham características de metadados e possam causar algum problema de segurança.

Scriptáveis: que pode se tornar ou ser usado como *script* (código executável).

Serverless: arquitetura de computação orientada a eventos.

Server-side: termo usado para designar operações que, em arquiteturas cliente-servidor, são feitas no servidor.

Stateful: utilizado quando se deve manter o controle do estado das interações com o sistema.

Stateless: utilizado quando não se deseja armazenar registros das interações anteriores com o sistema.

String: sequência de caracteres, geralmente usado para representar palavras.

Takeover: assumir o controle.

Token: sequência de caracteres hexadecimais utilizado para autenticação.

Token Bearer: esquema de autenticação HTTP que envolve tokens de segurança chamados tokens de portador.

Tipado: variáveis com tipos de dados específicos.

Truncagem: remoção de partes do texto.

Uptime: quantidade de tempo que um sistema de computador está ligado e funcionando.

Web service: solução utilizada na integração de sistemas e na comunicação entre diferentes aplicações.

Whitelist: lista com dispositivos, componentes ou recursos já previamente aprovados.

9. Dúvidas

Em caso de dúvidas em relação ao processo de homologação de sistemas satélites, favor consultar o documento anexo *Processo de Homologação de Sistemas Satélites*.

Caso haja dúvidas não sanadas no documento atual nem no documento mencionado acima, favor consultar o documento anexo *FAQ – Perguntas Frequentes*, que contempla as dúvidas mais frequentes envolvendo o processo.

Para todas as demais dúvidas, favor enviar e-mail para sistemas_satelites@sicredi.com.br.