



Guia de Desenvolvimento Seguro

Poder Judiciário do Estado do Rio de Janeiro

Gabinete da Presidência

Departamento de Segurança da Informação

Versão 1.0

Sumário

1. Fundamentos.....	3
2. Disposições gerais.....	4
3. Ambientes.....	6
4. Autenticação	7
5. Sessões.....	8
6. Acesso.....	9
7. Registro de erros e informações.....	10
8. Transmissões de Dados	10
9. Configurações.....	10
10. Armazenamento e bases de dados	11
11. Entradas de usuário	11
12. Criptografia.....	13
13. Práticas gerais de codificação.....	13
14. Código-fonte.....	14
15. Testes de segurança	15

1. Fundamentos

- autenticação: validar a identidade do usuário;
- autorização: definição das permissões e/ou acessos do usuário autenticado;
- auditoria: capacidade de atribuir a ação executada a um usuário autenticado;
- confidencialidade: garantir que as informações confidenciais somente serão exibidas para usuários que tenham autorização para isso;
- integridade: garantir que as informações somente sejam modificadas por usuários que tenham autorização para isso;
- disponibilidade: garantir que as informações estejam disponíveis a usuários que tenham autorização para isso.

- *privacy by design* – sete princípios fundamentais

https://iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf

- proativo, não reativo; prevenir, não remediar:

Não esperar que riscos de privacidade cheguem a se materializar, e sim evitar que eles aconteçam;

- privacidade por padrão:

Não é necessário desativar a coleta de dados extras, já que ela deve vir desativada por padrão;

- privacidade embutida no design:

A privacidade faz parte integral de toda a estrutura do produto ou serviço;

- total funcionalidade:

Ao implementar privacidade em um produto ou serviço, ela deve ser feita de forma a somar funcionalidades ao projeto, e não prejudicar nenhuma funcionalidade. As funcionalidades que conflitem com princípios de privacidade devem ser modeladas de forma criativa para que possam funcionar juntos.

- segurança ponta-a-ponta:
A privacidade deve ser continuamente protegida através de todo o ciclo de vida dos dados em questão.
- visibilidade e transparência:
A coleta, processamento e armazenamento de dados devem ser documentados de forma totalmente transparente, incluindo informações sobre a responsabilidade dos dados em caso de algum vazamento ou invasão.
- respeito pela privacidade do usuário
Deve ser sempre respeitada a decisão e a proteção dos dados do usuário, já que os dados são propriedade dele.

2. Disposições gerais

- observar o disposto no Ato Normativo nº 10/2019 e suas atualizações, que trata da Gestão de Ativos de Segurança da Informação no âmbito do PJERJ;
- observar o disposto no Ato Normativo nº 27/2020 e suas atualizações, que trata da Gestão de Acessos aos Recursos Computacionais no âmbito do PJERJ;
- Observar a Rotina Administrativa (RAD) que trata da Codificação Segura publicada pelo Departamento de Segurança da Informação (DESEG);
- cumprir a Lei Geral de Proteção de Dados (Lei nº 13.709/18) e suas atualizações, além disso, deve adotar medidas de proteção de dados pessoais desde a fase de concepção até a fase de execução de processos e sistemas (Privacy by Design), incluindo a coleta de dados limitada ao que é estritamente necessário ao alcance do propósito definido (Privacy by Default);
- obedecer aos padrões de acessibilidade, como eMAG e WCAG;
- caso o serviço prestado envolva transações de cartões de pagamento, deverá estar em conformidade com o padrão PCI-DSS;
 - anualmente, deverá evidenciar a conformidade de acordo com as regras do PCI, bandeiras e adquirentes;
 - todas as informações de propriedade do PJERJ, bem como informações de seus clientes e colaboradores, devem ter sua utilização restrita à

prestação do serviço contratado e devem ser tratadas como confidenciais.

- softwares, incluindo seus agentes, que precisem de senha mestra (administrador ou "root") para serem executados, ou terem suas funcionalidades regulares operacionais, não devem ser instalados;
- softwares executados com privilégios administrativos, utilitários, interfaces de linha de comando, aplicativos de conexão remota e similares devem ser restritos, controlados e documentados, devendo-se implementar:
 - limitação do uso de tais sistemas a um número mínimo de usuários confiáveis e autorizados, inclusive por tempo determinado e ao estritamente necessário;
 - procedimentos de autenticação, autorização e identificação a tais sistemas, quando aplicáveis.
- durante o levantamento de requisitos de software, deve-se considerar, também, conjuntamente com o DESEG, os requisitos de segurança da informação e proteção de dados, observando a adequação do sistema desde sua concepção;
- devem ser observados os princípios de segurança da informação em toda a esteira do ciclo de desenvolvimento de software e operações;
- as técnicas de programação segura devem ser aplicadas em novos sistemas, melhorias ou em reuso de código;
- quando do desenvolvimento terceirizado, a parte externa deverá estar em conformidade com as práticas do desenvolvimento seguro, cumprindo, no mínimo, os seguintes requisitos:
 - fornecimento de evidências de que os princípios de segurança foram seguidos no desenvolvimento do software, de modo a estabelecer um nível de segurança aceitável e a qualidade da privacidade;
 - fornecimento de evidências de que testes suficientes foram realizados para assegurar a ausência de conteúdo malicioso, tanto intencional quanto não intencional, no momento da entrega do software;
 - fornecimento de evidências de que testes suficientes foram realizados e vulnerabilidades conhecidas foram mitigadas;
 - documentação efetiva da construção do ambiente usado para realizar as entregas de software.

3. Ambientes

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto a segurança nos ambientes de trabalho:

- segregar ambientes de desenvolvimento, testes, treinamento, homologação e produção, como ambientes distintos com senhas diferentes;
- Os ambientes de desenvolvimento (implementação), testes e homologação devem ser desativados se:
 - o sistema de produção (implantação) for desativado ou não mais tiver evolução;
 - formalmente solicitado pelo gestor do sistema.
- as informações disponíveis fora do ambiente de produção podem ser representativas dos cenários esperados ou que ocorrem em produção. Se houver necessidade de simulação, todos os dados sensíveis deverão ser pseudoanonimizados para evitar que se crie um ambiente sem controle de acesso;
- os ambientes que não sejam de produção e partes administrativas de qualquer ambiente, não devem aparecer nos mecanismos de busca;
- compiladores, editores de código-fonte e demais ferramentas de desenvolvimento não devem estar disponíveis quando não forem necessários ao funcionamento de um sistema;
- validar com nome de domínio correto, dentro do prazo de validade e instalados adequadamente os certificados emitidos, para as aplicações acessíveis pela internet;
- desativar as funcionalidades de listagem de diretórios;
- configurar os serviços para utilizar portas de rede distintas de suas portas padrões;
- desativar informações padrões de serviços, tais como “banners” ou páginas que contenham versões, nomes, releases etc.;
- atuar conjuntamente com o DETIC para integrar as aplicações às soluções de firewall de aplicações web (WAF) existentes, no que for tecnicamente aplicável,

com o intuito de mitigar ataques como “XSS”, “CSRF”, “SQL Injection”, dentre outras explorações às aplicações web.

4. Autenticação

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto a autenticação:

- se possível, utilizar as soluções disponíveis de single-sign-on;
- dar preferência à implementação de mais de um fator de autenticação sempre que possível para assegurar a autenticação do usuário legítimo e evitar ataques de força bruta (BFA);
 - quando não for possível utilizar soluções de autenticação multifator (MFA), considerar alternativas como senhas secundárias, PINs ou perguntas de segurança;
 - registrar o número de falhas de autenticação e associar estes dados à respectiva conta, ao invés do endereço IP, a fim de se evitar ataques de endereços IPs diferentes.
- exigir nova autenticação do usuário para acessar recursos confidenciais, como atualização de senha ou e-mail, e demais transações julgadas sensíveis;
- Conforme disposto no Ato Normativo nº 27/2020 e suas atualizações, que trata da Gestão de Acessos no âmbito do PJERJ, implementar parâmetros de senha fortes configuráveis como por exemplo, tamanho mínimo de senha, quantidade de tentativas antes de bloquear a conta, histórico de senhas, tempo mínimo para troca de senha, tempo máximo para troca de senha etc.;
- observar as versões atualizadas e estáveis quando utilizar protocolos de autenticação que não exigem senha (OAuth, OpenId, SAML etc.);
- não indicar qual parte da autenticação está errada (se o usuário ou a senha), nas mensagens de erro de autenticação, trazer uma mensagem de erro idêntica, independente de qual parte esteja errada;
- apresentar faixas de horário de autenticação, não permitindo que usuários autenticuem no sistema fora do horário especificado, nas aplicações em que for pertinente.

5. Sessões

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto ao tratamento de sessões:

- evitar expor informações internas da aplicação em qualquer situação através da sessão;
- utilizar, sempre, conexões HTTPS (TLS 1.2 ou superior) em toda a sessão web, e não somente para o processo de autenticação.
- para o tratamento de cookies, habilitar os atributos “Secure” para canal criptografado de troca de IDs de sessão, “HttpOnly” para impedir que scripts acessem os cookies e “SameSite = Strict” para mitigar vazamento de informações e falsificações;
- utilizar os gerenciadores de sessão padrão da linguagem de programação ou dos frameworks adotados, observando as melhores práticas e versões atualizadas;
- estabelecer um tempo de inatividade para que uma sessão não fique aberta por tempo indeterminado, considerando o prazo de 2 a 5 minutos para aplicações críticas e o prazo de 15 a 20 minutos para aplicações de baixo risco, invalidando a sessão no lado cliente e no lado servidor;
- renovar o ID da sessão após qualquer alteração no nível de privilégio do usuário associado, considerando cenários como alteração de senha, alteração de permissão, alternância entre uma função de usuário comum para uma função de administrador etc.;
- implementar controles para que o usuário não consiga enviar ID de sessão por meios diversos, tal como em parâmetros de URL;
- garantir, quando um usuário fizer login no sistema, que sejam gerados uma nova sessão e um novo ID de sessão, sendo esses dados únicos e suficientemente longos e aleatórios, para evitar reuso de sessão e descoberta através de método de tentativa e erro;
- implementar controle de sessão, de modo que um usuário não possa ter mais do que uma sessão ativa. Caso uma nova sessão seja aberta, a sessão antiga deve ser invalidada;

- revalidar a sessão de tempos em tempos, devendo este tempo ser definido internamente, nos casos em que as sessões durem um longo período;
- invalidar todas as sessões quando o usuário efetuar logoff do sistema;
- registrar todas as tentativas de autenticação, tendo resultado em sucesso ou não.

6. Acesso

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto aos acessos:

- observar o privilégio mínimo, restringindo o acesso a funcionalidades por padrão e liberação conforme a necessidade;
- estar preparado para trabalhar no modo "fail-secure", ou seja, em caso de falhas, o acesso seja bloqueado, quando ocorrer alguma falha no controle de acesso;
- restringir o acesso às URLs, funções, referências, serviços e dados somente a usuários autorizados;
- quando houver necessidade de que o sistema forneça informações restritas, deve-se disponibilizar tais conteúdos de forma temporária (links temporários ou URLs pré-assinadas, por exemplo), com prazo predefinido a ser estabelecido em procedimento próprio, excluindo a forma de acesso após o tempo de expiração;
- permitir que apenas usuários com os devidos privilégios possam alterar as configurações de segurança;
- possuir perfis de acesso que sejam customizáveis pelos administradores;
- a aplicação deve ser capaz de gerar trilha de auditoria para que haja rastreamento das ações realizadas pelos usuários;
- ter capacidade de registrar na trilha gerada tanto informações de login, gerenciamento de acessos (troca de senha, alteração de perfil), logoff, quanto informações de alterações sistêmicas, como por exemplo, o que foi alterado (campo e/ou valor), quando foi alterado (timestamp), quem fez a alteração (login), de onde foi feita a alteração (IP e/ou hostname), qual era o valor anterior

(manter todos os registros, não apenas o último antes da alteração) e o valor para o qual foi alterado.

7. Registro de erros e informações

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto ao registro de erros e informações:

- não exibir informações sensíveis onde não existe forma de controle (logs, cache no cliente, etc), caso seja necessário, a informação deve ser pseudoanonimizada;
- não expor o ID da sessão diretamente na URL, em mensagens de erro ou logs;
- ter acesso aos logs apenas por pessoas autorizadas (administradores do ambiente e desenvolvedores);
- observar a marcação de tempo no log ou trilha de auditoria para que esteja devidamente sincronizado com o tempo real.

8. Transmissões de Dados

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto a segurança na transmissão de dados:

- utilizar canais criptografados, principalmente quando houver tráfego de dados sensíveis e credenciais de acesso.

9. Configurações

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto a segurança nas configurações:

- utilizar armazenamento isolado ou criptografado para configurações;
- evitar armazenar senhas diretamente no código-fonte;
- proteger a senha mestre e as chaves privadas contra acessos não autorizados;
- não configurar senhas em texto claro nos arquivos de configuração no servidor (ex. arquivos “.conf”, “.xml”, etc.);

- não armazenar as informações de conexão na própria aplicação, devendo ser armazenadas em arquivo separado, com conteúdo criptografado e acesso somente a pessoal autorizado.

10. Armazenamento e bases de dados

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto a segurança no armazenamento e bases de dados:

- provisionar às aplicações apenas usuários comuns não administrativos, com as eventuais permissões necessárias e documentadas;
- utilizar um framework de persistência de dados (ORM) para manipular as informações na base de dados;
- não armazenar senhas em formato aberto, devendo ser armazenado apenas o hash e o “salt”, se a aplicação gerenciar um repositório de credenciais.

11. Entradas de usuário

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas gerais de controle das entradas de dados pelos usuários:

- validar as entradas de dados no ambiente da aplicação;
- não permitir entrada de código script em qualquer dado de texto;
- não exibir os caminhos completos dos arquivos durante a transmissão ou armazenamento dos arquivos;
- arquivos que necessitam serem armazenados nos bancos de dados devem passar por inspeção e testes através de *sandbox*. Somente aqueles que forem validados poderão ser armazenados;
- evitar que os formulários carreguem automaticamente informações de autenticação, como usuários e senhas, possuindo medida de segurança conhecida como autenticação por desafio e resposta - CAPTCHA (reCAPTCHA, hCaptcha, Honeypot Captcha, Friendly Captcha etc.);
- implementar mensagens de erro genéricas, não informando qual dado foi inserido incorretamente, tampouco se a conta for bloqueada ou inexistente;

- efetuar todos os controles de autenticação do lado do servidor, não do cliente;
- solicitar no mínimo as informações da senha antiga, da senha nova e a confirmação da senha nos campos de troca de senha;
- utilizar somente requisições do tipo "POST" para tratamento de credenciais de autenticação;
- estar visível e acessível em todas as páginas autenticadas, o botão de "sair" (logout);
- para o tratamento de dados preenchidos pelo usuário, no que for tecnicamente aplicável, validar campos de entrada e efetuar validação de caracteres;
- aplicar validações em entrada de dados para que apenas os dados corretos e esperados sejam processados pelas aplicações, utilizando, preferencialmente, bibliotecas ou componentes já consolidados ou funções de frameworks, impedindo ataques de injeção ("SQL Injection", "Command Injection", "LDAP Injection" etc.);
- as seguintes práticas para o tratamento de arquivos submetidos pelo usuário ("upload" de arquivos), no que for tecnicamente aplicável:
- utilizar, sempre que disponíveis, bibliotecas ou funções de validação consolidadas, testadas e atualizadas para resguardar a aplicação em cenários diversos, incluindo os dispostos abaixo;
- definir um diretório específico, preferencialmente em um "host" apartado ao da aplicação, para salvar os arquivos que serão submetidos. Em qualquer caso, não utilizar o diretório-raiz da aplicação;
- remover a permissão de execução do diretório que receberá os arquivos submetidos;
- definir previamente os tipos de arquivos válidos, fornecendo ao usuário a mensagem com o respectivo motivo quando não aceito e informar canais disponíveis para suporte;
- definir um limite de tamanho de arquivo para evitar ataques de negação de serviço;
- definir um limite de comprimento de nome de arquivo;
- efetuar validação de caracteres, eliminando dos nomes de arquivos e de suas respectivas extensões os caracteres especiais, os caracteres de controle e Unicode;

- renomear o arquivo submetido com um nome distinto do submetido, preferencialmente aleatório;
- identificar e rejeitar arquivos com extensão dupla (ex.: “.php.jpg”);
- verificar o conteúdo dos arquivos submetidos ao servidor e validar se corresponde ao formato de arquivo esperado. Não confiar no cabeçalho “Content-Type”, pois ele pode ser falsificado. Em casos indefinidos, os arquivos deverão ser descartados;
- evitar, sempre que possível, formatos de arquivos compactados. Quando for aplicável, verificar, separadamente, os respectivos arquivos internos;
- impedir o upload, e eventual sobrescrita por este meio, de arquivos de configuração, tais como “.htaccess”, “.web.config”, “.conf”, “.xml” etc.;
- aplicar controles para que arquivos submetidos não sejam acessados por usuários não autorizados;
- preferencialmente, limitar o acesso da funcionalidade de upload de arquivos somente a usuários autenticados e/ou autorizados.

12. Criptografia

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas quanto à utilização de criptografia:

- implementar funções de criptografia no lado do servidor para proteger os dados sensíveis das aplicações;
- deve-se observar: protocolo, algoritmo de troca de chave, assinatura digital, modo de criptografia e hash que são implementados pela solução para evitar componentes depreciados;
- utilizar técnica de “salt” nos hashes sempre que possível.

13. Práticas gerais de codificação

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas gerais de codificação:

- preferencialmente, utilizar funções, bibliotecas e códigos já disponíveis e testados;
- deve-se documentar todos os frameworks e bibliotecas adotados, e demais componentes de software de terceiros, incluindo suas versões;
- contar com mecanismos nas aplicações, que previnam condições de concorrência, evitando requisições simultâneas e situações semelhantes;
- realizar sempre o tratamento dos dados adequadamente, antes de transferir as entradas dos usuários para qualquer função;
- limitar a geração e a alteração de código aos usuários privilegiados devendo também se limitar o mínimo possível dentro da própria aplicação;
- deve-se evitar ao máximo utilizar o uso de funções ou recursos obsoletos (“deprecated”), justificando e documentando seu uso quando necessários. Devem ser estudadas novas soluções para substituir, o mais rapidamente possível, tais recursos obsoletos;
- considerar as boas práticas para ecossistema de microsserviços, especialmente quanto ao gerenciamento e controle centralizado de comunicações entre aplicativos (“gateway” de APIs) e soluções de mensageria consolidadas;
- quando do desenvolvimento de aplicações web, garantir a compatibilidade entre os navegadores (“browsers”) mais utilizados.

14. Código-fonte

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas gerais de controle do código-fonte:

- utilizar somente ferramentas de controle de versão do código homologadas pelo PJERJ;
- observar para não disponibilizar, no controle de versão, arquivos ou diretórios de configurações com dados ou informações sensíveis (Ex: “.git”, “.ssh” etc.);
- o acesso ao código-fonte das aplicações deve ser restrito somente aos desenvolvedores das aplicações e aos analistas de segurança da informação para as devidas auditorias;

- devem ser mantidos acordos de confidencialidade para desenvolvedores ou demais interessados que necessitem acessar os códigos desenvolvidos pelo PJERJ ou sob sua custódia, mesmo que de forma temporária;
- esse repositório deve concentrar todos os códigos fonte, pacotes de melhoria, scripts de bancos de dados, arquivos de novas versões, entre outros.

15. Testes de segurança

O desenvolvimento de novos sistemas de informação deve seguir as seguintes práticas gerais de mitigação de ameaças e disponibilização para testes de segurança:

- realizar testes manuais antes de liberar versões;
- deve-se definir cenários de testes, principalmente nos aspectos de segurança, para os casos de atualizações na arquitetura do sistema (servidores de aplicação, banco de dados, versões de browser, versões de sistema operacional etc.);
- Observar, não exaustivamente, as seguintes vulnerabilidades de:
 - injeção de código;
 - autenticação;
 - integração XML;
 - perfis de acesso;
 - erros de configuração de segurança;
 - cross site script - XSS;
 - falta de logs e monitoramento;
 - interface de programação de aplicativos (APIs) (Ex.: REST, GraphQL, gRPC etc.).
- após a conclusão do desenvolvimento de um novo sistema, manutenções evolutivas ou uma correção em um sistema já existente, o Departamento de Segurança da Informação (DESEG) deverá ser comunicado para que um teste de segurança possa ser realizado;
- além dos testes sugeridos no item anterior, outros poderão ser desenvolvidos pela equipe de Segurança da Informação baseado no conhecimento sobre o projeto e nos requisitos de segurança solicitados;

- excepcionalmente, os testes previstos anteriormente podem ser dispensados, desde que autorizado pelo Presidente do PJERJ, após análise da motivação e dos riscos pela dispensa que devem ficar registrados;
- todo o procedimento realizado deve ser devidamente documentado e os resultados devem ser compartilhados com as equipes da área de Segurança da Informação e de desenvolvimento, para que os itens identificados como falhas sejam corrigidas;
- após a sinalização da correção dos itens pelo Departamento de Soluções (DESOL), da SGTEC, os itens deverão ser testados novamente para garantir a efetividade da correção proposta.