Norma Interna – Desenvolvimento Seguro

JANEIRO - 2023

Sumário

Responsável:	Rafael Nantes (VP)
Aprovado por:	Suleiman Bragança (CEO)
Políticas Relacionadas:	Normas de publicação de código fonte – Sistema Cielo
Data de Aprovação:	01/2023
Data de Revisão:	04/2024
Versão atual:	2.0

1 – OBJETIVO

Este documento é o guia para desenvolvimento seguro de software da Vector. O seu objetivo é apresentar boas práticas, a serem adotadas por analistas, desenvolvedores e instaladores de software – sejam eles prestadores de serviço terceirizados, ou do quadro permanente da Vector, tornando o processo de concepção dos sistemas construídos dentro da Vector mais confiável, auditável, estável e protegido contra ameaças. O presente guia funciona como complemento da Política de Segurança da Informação da Vector, descrevendo as normas e diretrizes para um Desenvolvimento Seguro.

2 - FORMATO

O documento é estruturado em torno de “diretrizes”, recomendações de boas práticas, a serem seguidas, em cada um dos tópicos listados nesse documento.

3 - DIRETRIZES

Serão classificadas as diretrizes, em função da necessidade de proteção aos dados e das obrigações imputadas ao programador ou analista, em três níveis - cada um cumulativo com as medidas do nível anterior, salvo em sobreposição de escopo. Os níveis são:

- Mínimo: deveres a serem seguidos na construção de sistemas, para que se obtenha um nível de segurança, considerado empiricamente mínimo.

- Padrão: recomendações pertinentes, à construção de sistemas, para que se obtenha um nível de segurança, considerado empiricamente padrão à data da última revisão do documento.
- Forte: medidas adicionais pertinentes à construção de sistemas, para que se obtenha um nível de segurança empiricamente forte, à data da última revisão do documento.

Na ausência da indicação de níveis de segurança, em uma dada diretriz, seu nível é considerado mínimo.

3.1 – AMBIENTE DE DESENVOLVIMENTO

Esta seção apresenta diretrizes para a instalação, configuração e gerenciamento de ambientes de desenvolvimento de sistemas.

3.1.1 - Acesso ao Código-Fonte

Diretivas para controle de acesso dos desenvolvedores ao código-fonte das aplicações. Quanto ao sigilo do código-fonte dos sistemas desenvolvidos, devem ser, por padrão, de livre acesso aos servidores da Vector. As demais situações deverão ser analisadas, projeto a projeto, pelos gestores.

Mínimo:

- Deve-se utilizar um sistema de controle de versão, com controle de acesso e recuperação em caso de falhas.

Padrão:

- Deve-se utilizar um controle de versão distribuído, que mantém um repositório completo em cada máquina de desenvolvimento.

3.1.2 - Separação de Ambientes

Diretivas para a separação de ambientes de desenvolvimento/testes/homologação (DEV / TESTE / HOM) do ambiente de produção (PROD). As aplicações desenvolvidas devem considerar o uso de repositórios seguros de dados, especialmente na forma de banco de dados com acesso controlado ou arquivos criptografados, quando o uso de banco de dados não for possível/desejável.

***Mínimo*:**

- Deve-se utilizar bancos de dados distintos para cada ambiente;
- Deve-se utilizar servidores de aplicação/web distintos para cada ambiente;
- Deve-se prover acesso ao ambiente de desenvolvimento/ testes/ homologação apenas aos integrantes da equipe de desenvolvimento e aos interessados no produto.

***Padrão*:**

- Deve-se prover um instalador expresso para a instalação do ambiente necessário para a execução de uma dada aplicação;
- Deve-se realizar testes periódicos para assegurar a segurança do ambiente de desenvolvimento/testes/homologação.

Forte:

- Não se deve fornecer senhas de acesso ao ambiente de produção aos desenvolvedores.

3.1.3 - Procedimentos para Solicitação de Lançamento de Versões

1 - Os lançamentos em PRODUÇÃO são realizados mediante autorização do Cliente através do JIRA, que fica responsável por realizar, a formalização interna através do processo de abertura de GMUD do Cliente.

2 - As melhorias elencadas, na solicitação de lançamento, devem estar todas com desenvolvimento concluído e homologação confirmada pelo time de QA responsável, (time do Cliente).

3 - O pedido de lançamento, contendo o pacote de melhorias, deve ser avaliado e aceito pelo time de desenvolvimento e DEVSECOPS, que fica responsável por observar sobre viabilidade, complexidade, riscos, interdependências, prazo para preparação e execução, podendo este negar o pedido, ou solicitar maior prazo quando verificado ausência de informações necessárias, dependências não contempladas no pacote ou qualquer inviabilidade técnica.

4 - As janelas de lançamento em PRODUÇÃO são semanais, ocorrendo toda quinta-feira, sendo que o pedido de lançamento contendo a relação de melhorias deve ser formalizado com 1 dia útil de antecedência através das reuniões semanais de GMUD pré-agendadas para esta finalidade.

Padrão:

- Seguir as diretrizes do cliente.

3.1.4 - Procedimentos de Lançamento de Versão

1 - O pacote de lançamento deve ser preparado com antecedência pelo time de desenvolvimento e DEVSECOPS após aprovação da GMUD observando-se o checklist de dependências.

2 - Os lançamentos deverão ocorrer no dia da GMUD programada a partir das 18:00.

3 - Ao concluir o lançamento o Cliente deve ser imediatamente comunicada para que seja realizado um checklist de validação.

4 - Durante o período de conferência/validação do lançamento o time deve ficar de plantão para tratar/deliberar sobre possíveis inconsistências.

5 – O Cliente fica responsável pela comunicação aos usuários sobre o lançamento e possíveis intermitências ou indisponibilidade no período de lançamento.

Padrão:

- Seguir as diretrizes do cliente.

3.1.5 - Rollback e tratamento de inconsistências do procedimento de lançamento de versão:

1 - Após lançamento e durante a janela de validação o time de desenvolvimento deverá permanecer a disposição para eventuais ajustes até que a validação seja completa.

2 - Em casos que se determinar necessário ou que não sejam possíveis os ajustes, o procedimento de rollback pode ser solicitado pelo Cliente dentro da janela de validação do lançamento sendo necessários até 6 horas para realização do procedimento contados a partir da solicitação.

Padrão:

- Seguir as diretrizes do cliente.

3.2 – PARAMETRIZAÇÃO PARA PROTEÇÃO DE DADOS

Esta seção apresenta diretrizes para a configuração de proteção a dados sensíveis. São detalhados parâmetros para criptografia, hash e gerenciamento de senhas.

3.2.1 - Criptografia e Hash

- Diretrizes para a configuração e utilização de algoritmos de criptografia e hash visando prover confidencialidade a dados;
- Dados sigilosos e sensíveis devem ser criptografados, sempre que possível. O método de criptografia empregado deve obedecer às particularidades dos dados e de sua utilização, seguindo os parâmetros aqui listados;
- Deve-se utilizar hashes criptográficos sempre que possível, sobretudo nos seguintes casos: verificação da integridade de dados; armazenamento e verificação de senhas; provimento de identificador "único", para objetos em um sistema e geração de números pseudo-aleatórios.

Mínimo:

- Deve-se utilizar um método criptográfico que siga o princípio de Kerckhoffs10, o método de encriptação e seus parâmetros devem ser públicos e estar documentados, somente a chave criptográfica deve ser mantida em sigilo;
- Não se deve utilizar um cifrador que admita um método conhecido para quebra da chave criptográfica melhor do que a força bruta, baseada em tentativa e erro;
- Não se deve utilizar o modo de cifrador de bloco electronic codebook (ECB) ou modos menos seguros;
- Não se deve utilizar um tamanho da chave menor que 128 bits (cifrador simétrico) ou 1024 bits (cifrador assimétrico);
- Não se deve utilizar função de hash sem algum tipo de salt.

Padrão:

- Não se deve utilizar algoritmos considerados obsoletos para criptografia e hash criptográfico. Exemplos: MD5, SHA1, DES/3DES, RC2, RC4, MD4;
- Não se deve utilizar um tamanho da chave menor que 192 bits (cifrador simétrico) ou 2048 bits (cifrador assimétrico);
- Não se deve distribuir chaves criptográficas sem a utilização de uma infraestrutura de chave pública e, portanto, sem a utilização de um cifrador assimétrico.

Forte:

- Não se deve utilizar um tamanho da chave menor que 256 bits (cifrador simétrico) ou 4096 bits (cifrador assimétrico).

3.3 – CICLO DE VIDA DE SOFTWARE

Esta seção apresenta diretrizes, para reforço da segurança de software nas diferentes fases de seu ciclo de vida; projeto, codificação e manutenção. Traz, ainda, diretrizes para a aplicação com as pessoas envolvidas nestas diferentes fases.

3.3.1 - Produto

- Diretrizes para tornar seguras as práticas utilizadas, durante a etapa de elaboração de produto, inserida dentro da metodologia de desenvolvimento da Vector. As práticas aqui elencadas, estão de acordo que institui a política de segurança da informação da Vector. Deve-se empregar modelo de produto de software que contemple: a) etapa de modelagem de ameaças; b) definição clara dos riscos de segurança; e c) e o nível de severidade que o comprometimento de dados sensíveis traria ao sistema e à empresa;
- Não se deve omitir, durante o desenvolvimento do produto e sua execução, a definição de responsabilidades pela segurança de dados do sistema e como essa responsabilidade será verificada;
- Deve-se utilizar cronograma de desenvolvimento que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

3.3.2 - Codificação

Diretrizes para tornar seguras práticas utilizadas durante a etapa de codificação de sistemas:

- Deve-se documentar, inclusive no código da aplicação, as medidas protetivas aplicadas no código-fonte, de modo a indicar precisamente o procedimento utilizado e suas peculiaridades;
- Não se deve armazenar senhas em código-fonte;
- Não se deve utilizar códigos da Internet sem conhecer a fonte ou entender seu funcionamento;
- Deve-se evitar ao máximo usar funções ou recursos obsoletos (deprecated), restringindo de forma ampla o seu acesso. Novas soluções devem ser estudadas para substituir, o mais rapidamente possível, tais recursos;
- Deve-se aplicar, sempre que possível, o conceito de validação positiva, que se trata de um mecanismo que usa critérios pré-definidos, para validar o tamanho, caracteres, formato, e as regras de negócio que se aplicam sobre os dados antes de aceitar a entrada. Qualquer dado que não atenda aos critérios deve ser rejeitado;
- Todo ponto de interação de dados com o usuário do sistema (input) deve ter sua validação feita na entrada do dado e na sua apresentação (output);
- Validações de segurança devem ser realizadas no servidor (Webserver).

3.3.3 - Manutenção

- Diretrizes para tornar seguras as práticas, utilizadas durante a etapa de manutenção de sistemas:
- Não se deve habilitar as atualizações automáticas de software, ou componentes utilizados na construção de um sistema, sob pena de introdução indevida de falhas de segurança;
- Não se deve modificar software de terceiros, salvo quando estritamente necessário, controles de segurança internos podem ser invalidados. A mudança deve ser feita pelo desenvolvedor original do sistema, sempre que possível.

3.3.4 - Pessoal

- Diretrizes para a perpetuação de práticas de desenvolvimento seguro, junto às pessoas que operam as diferentes fases do ciclo de vida do software;
- Deve-se proporcionar treinamento e capacitação de programadores, para aquisição e revisão de princípios de segurança computacional e desenvolvimento de software seguro.

4 - ABRANGÊNCIA

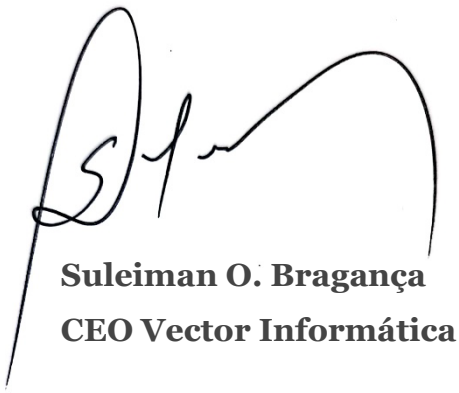
Esta Norma estabelece regras, que devem ser cumpridas por toda a equipe de Desenvolvimento.

5 - CONSIDERAÇÕES FINAIS

As dúvidas decorrentes, de fatos não descritos nesta, deverão ser encaminhadas ao responsável pelo Desenvolvimento.

Esta Norma entra em vigor a partir da data de publicação e pode ser alterada a qualquer tempo, por decisão da Direção, mediante o surgimento de fatos relevantes que apareçam ou não tenham sido contemplados neste documento.

Barueri, janeiro de 2023



Suleiman O. Bragança
CEO Vector Informática Ltda.