1. OBJETIVO

Esta política tem como propósito, definir as regras básicas para o desenvolvimento seguro de software e sistemas.

2. AMPLITUDE

Este documento é aplicado a todo o desenvolvimento e manutenção de todos os serviços, arquitetura, software e sistemas que fazem parte do Sistema de Gestão de Segurança da Informação (SGSI).

Os usuários deste documento são os bolsistas que atuam no desenvolvimento dos produtos e soluções tecnológicas produzidos pelo NEES.

Todos os bolsistas do NEES são responsáveis pela gestão de incidentes de segurança da informação.

3. DOCUMENTOS DE REFERÊNCIA

- Norma ABNT NBR ISO 27002 - Segurança da informação, segurança cibernética e proteção à privacidade — Controles de segurança da informação.
- POP.SGSI_016 - Gestão de Risco Corporativo
- PSI_010 - Política de gestão de mudanças
- Plano de treinamento e conscientização

4. DEFINIÇÃO

Desenvolvimento Seguro: o desenvolvimento seguro é um requisito para construir serviço, arquitetura, *software* e sistema seguros.

5. DIRETRIZES

5.1 Avaliação de risco para o processo de desenvolvimento

Além da avaliação de risco executada de acordo com a Metodologia de avaliação de riscos e tratamento de riscos, o responsável pelo desenvolvimento precisa executar a avaliação dos seguintes ítens:

- Os riscos relativos ao acesso não autorizado ao ambiente de desenvolvimento.
- Os riscos relativos a mudanças não autorizadas ao ambiente de desenvolvimento.
- Os riscos que a nova tecnologia pode trazer a empresa.
- Os riscos que uma nova tecnologia pode trazer se for usada na empresa.

5.2 Tornar seguro o ambiente de desenvolvimento

É necessário que exista ao menos três ambientes distintos onde venha ser executado código de atividade em desenvolvimento, código que esteja em homologação e código em ambiente produtivo. O ambiente de desenvolvimento deve ser utilizado para que o time de desenvolvimento tenha possibilidade de validar o código gerado em ambiente próximo do ambiente **produtivo**. O acesso será possível apenas após fornecimento de credenciais com duplo fator de autenticação - 2FA.

O ambiente de homologação será o mais próximo possível do ambiente produtivo, onde será homologado pelo Product Owner. O acesso será possível apenas após fornecimento de credenciais com duplo fator de autenticação - 2FA, ou credenciais gov.br quando se tratar de um software de acesso público.

O ambiente de produção, deve executar o código que será disponibilizado aos usuários finais. O acesso a infraestrutura que executa o código de produção deve ser acessível somente ao time responsável, que terá apenas os acessos estritamente necessários.

Quanto a rotina de backup, deve seguir ao documento PSI_014 – Política de Backup das Informações.

5.3 Requisitos de segurança

- Não se deve armazenar senhas em texto plano sem utilizar um algoritmo de hash seguro e salt.
- Deve-se utilizar controle de usuário e senha nominais para determinar a identidade do usuário.
- Deve-se utilizar autenticação via AD / Workspace sempre que possível para autenticar usuários internos.
- Deve-se dar ciência ao usuário das permissões e níveis de acesso que possui.
- Deve-se utilizar grupos ou OU de Active Directory (AD) / Workspace para determinar as políticas de acesso e roles de usuário.

5.4 Verificar e testar a implementação dos requisitos de segurança

- Deve-se realizar testes manuais de segurança antes de cada versão do software que modifique sua estrutura (telas de login, serviços não autenticados, novos formulários com interação com o usuário, etc.).
- Deve-se garantir, através de testes automatizados, que os serviços e dados sigilosos estejam protegidos e disponíveis apenas para os usuários detentores das informações.
- Deve-se elaborar uma política de testes, automatizados ou não, visando a garantia de não vulnerabilidade aos principais ataques conhecidos em sistemas.
- Aos lideres técnicos dos projetos compete acompanhar os relatórios de vulnerabilidades disponibilizados pelas ferramentas **Gitlab** e **Wazuh** ambas utilizadas nos ambientes do NEES.
- Deve-se definir cenários de testes voltados à garantia dos requisitos não funcionais dosoftware, preferencialmente realizado por uma equipe de testes diferente da equipe de desenvolvimento do software, com intuito de se evitar vícios.
- 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.).

5.5 Repositório

Deve-se utilizar o sistema de controle de versão https://gitlab.ufal.br com controle de acesso e recuperação em caso de falhas:

5.5.1 Controle de versão

O controle de versão minimamente proteger as branchs de homologação e de, bem como mensagens de commits devem seguir o padrão definido em https://www.conventionalcommits.org/en/v1.0.0/, contudo as mensagens devem ser em Português.

5.6 Controle de mudança

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:

5.6.1 Projeto

- Deve-se empregar modelo de projeto de software que contemple:
  - Etapa de modelagem de ameaças;
  - Definição clara dos riscos de segurança;
  - Nível de severidade que o comprometimento de dados sensíveis traria ao sistema e à instituição.
- Não se deve omitir, durante o projeto de desenvolvimento de sistema e sua execução, a definição de responsabilidades pela segurança de dados do sistema e como essa responsabilidade será verificada.
- Deve-se utilizar cronograma de projeto que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

5.6.2 Codificação

- 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.

5.6.3 Manutenção

- 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.

5.7 Armazenamento de dados

Esta seção apresenta definições e diretrizes que tratam do armazenamento de informações sigilosas ou não e de sua disponibilização. Descreve procedimentos para o armazenamento seguro das informações em bancos de dados. Detalha o gerenciamento de permissões de acesso e distribuição de senhas a serem adotadas para operacionalização dessas estruturas.

5.7.1 Procedimentos e Meios para Armazenamento de Dados

- Não se deve utilizar meio de armazenamento que não possua acesso para leitura e escrita restrito por senha, salvo quando o acesso for controlado por ACL com restrição do endereço de origem.

- Deve-se preferencialmente armazenar dados criptografados. Inclusive os dados em repouso.

5.7.2 Permissões para Acesso a Informações em Banco de Dados

- Não se deve disponibilizar às aplicações acesso à algum banco de dados utilizando login de usuário com permissões de root.
- Não se deve disponibilizar às aplicações acesso à algum banco de dados utilizando login de usuário com permissões para execução de comandos em Data Definition Language (DDL).
- Não se deve disponibilizar às aplicações acesso à algum banco de dados utilizando login de usuário com permissões além das estritamente necessárias ao seu funcionamento.

5.7.3 Gerenciamento e Distribuição de Senhas para Acesso a Dados

- Não se deve permitir a elaboração de senhas que não sigam os padrões estabelecidos pelo NEES.
- Não se deve utilizar o armazenamento de senhas em código-fonte. As mesmas devem ser armazenadas sob a forma de **secrets** ou preferencialmente em https://vault.nees.ufal.br.
- Deve-se armazenar de forma segura os dados de usuários e os sistemas que utilizam cada senha fornecida.
- Não se devem utilizar as mesmas senhas para ambientes de desenvolvimento, teste, homologação e produção.

5.8 Proteção dos dados

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

5.8.1 Criptografia e Hash

- Deve-se utilizar um método criptográfico que siga o princípio de Kerckhoffs. 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 (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.
- Não se deve utilizar algoritmos considerados obsoletos para criptografia e hash criptográfico. Exemplos: MD5, SHA1, DES/3DES, RC2, RC4, MD4.
- Não se deve distribuir chaves criptográficas sem a utilização de uma infraestrutura de chave pública e, portanto, sem a utilização de um cifrador assimétrico.
- Não se deve utilizar um tamanho da chave menor que 256 bits (cifrador simétrico) ou 4096 bits (cifrador assimétrico).

5.8.2 Senhas

- Tamanho da senha: Não se deve utilizar senhas com menos de 8 caracteres.
- Variação de símbolos: Deve-se utilizar pelo menos letras maiúsculas e minúsculas, junto a ao menos um tipo de caractere especial e um dígito.
- Aleatoriedade: Não se deve elaborar senhas sem auxílio de software gerador de senhas aleatórias, configurado para atender aos parâmetros aqui estabelecidos.
- Testes: Não se deve utilizar senha que não tenha sido validada por um software testador de força de senhas, por exemplo: 1Password, NordPass, RoboForm, Kaspersky.
- Periodicidade de troca: Não se deve utilizar periodicidade de troca de senhas superior a 6 meses.
- Mudança e recuperação de senha: Não se deve permitir que se utilize o mesmo canal de validação da senha. Não se deve enviar a senha antiga para o usuário, em claro ou não, sob nenhuma hipótese.
- Armazenamento (usuário): Não se deve armazenar senha que não esteja criptografada seguindo o nível padrão de criptografia estabelecido neste documento.
- Número de tentativas: Não se deve permitir uma taxa de tentativas de validação de senha superior a 5 tentativas por minuto. A senha deve ser bloqueada em caso de no máximo 5 erros de validação consecutivos e sua reabilitação deve depender de processo específico.

5.9 Treinamento em segurança requerido

- Deve-se proporcionar treinamento e capacitação de programadores para aquisição e revisão de princípios de segurança computacional e desenvolvimento de software seguro.

6. REGISTROS

Os registros relacionados com este procedimento são:

- Sistema INTEGRA.

7. DISTRIBUIÇÃO E CONTROLE

Este documento está disponível e controlado através do sistema INTEGRA, módulo conhecimento/ ISO27001/Politicas. Deve ser atualizado anualmente.

8. HISTÓRICO DE ALTERAÇÕES

Revisão	Data	Descrição	Responsável
01	20/06/2024	Criação do documento.	Francisco Meneses
02	13/04/2025	Revisão da estrutura de todo o documento e inclusão do controle de documentos e assinaturas, via sistema INTEGRA	Shirley Vital