PORTARIA PRE Nº 224, DE 16 DE SETEMBRO DE 2024

Institui Norma de Segurança Cibernética — NSC11 — Desenvolvimento de Software Seguro do Tribunal Regional Eleitoral de Minas Gerais.

O PRESIDENTE DO TRIBUNAL REGIONAL ELEITORAL DE MINAS GERAIS, no uso de suas atribuições conferidas pelo inciso XV do art. 22 da Resolução TRE-MG nº 1.277, de 29 de maio de 2024, o Regimento Interno, considerando o disposto no art. 4° da Resolução TRE-MG nº 1.240, de 6 de fevereiro de 2023, que determina que a "revisão e a atualização das normas complementares de Segurança da Informação ocorrerão sempre que necessário, por meio de portaria da Presidência do Tribunal.",

RESOLVE:

CAPÍTULO I

DAS DISPOSIÇÕES PRELIMINARES

Art. 1° Fica instituída a Norma de Segurança Cibernética — NSC11 — Desenvolvimento de *Software* Seguro do Tribunal Regional Eleitoral de Minas Gerais.

Parágrafo único. A norma de segurança de que trata o *caput* tem por objetivos:

I — apresentar as regras a serem seguidas por desenvolvedores de sistemas, profissionais responsáveis pela infraestrutura de TIC, administradores de bancos de dados e responsáveis por procedimentos relacionados à continuidade de serviços de TIC (*backup*, *restore*), para a garantia de disponibilização de serviços seguros de Tecnologia da Informação;

II — prover diretrizes para especificação de artefatos e processos da metodologia de desenvolvimento de sistemas.

Art. 2° Esta portaria integra a Política de Segurança da Informação do Tribunal Regional Eleitoral de Minas Gerais, regulamentada pela Resolução TRE-MG nº 1.240 de 6 de fevereiro de 2023.

Art. 3° Para os efeitos desta portaria, aplicam-se os termos e definições da Norma de Segurança Cibernética — NSC1 — Termos e Siglas de Segurança da Informação.

Art. 4° Esta portaria aplica-se aos magistrados, servidores efetivos e requisitados, ocupantes de cargo em comissão sem vínculo efetivo, estagiários, prestadores de serviço, colaboradores e usuários externos que utilizam os ativos de informação e de

processamento na Justiça Eleitoral de Minas Gerais.

CAPÍTULO II

DA ARQUITETURA DE *SOFTWARE*

Art. 5° Serão observadas as seguintes diretrizes para a construção de um projeto arquitetural de *software* seguro, focando-se em práticas gerais de programação e no desenho de seus componentes:

I — evitar erros de cálculo decorrentes da falta de entendimento da representação interna da linguagem de programação usada e de como é realizada a interação com os aspectos de cálculo numérico como, por exemplo, reconhecer representação de sinal, valores do tipo "*Not-A-Number*" (NaN), valores especiais, etc.;

II — proteger as variáveis e os recursos compartilhados contra acessos concorrentes inapropriados;

III — utilizar mecanismos de bloqueio que evitem a ocorrência de requisições simultâneas feitas à aplicação ou utilizar um mecanismo de sincronização para evitar condições de concorrência (*race conditions*);

IV — aumentar os privilégios da aplicação para um patamar mais elevado o mais tardiamente possível em relação ao fluxo de execução que necessita dos privilégios adicionais e revogar esses privilégios adicionais assim que não forem mais necessários.

CAPÍTULO III

DA VALIDAÇÃO DOS DADOS

Art.6° As orientações a seguir referem-se à validação de dados de entrada do usuário e do recebimento de dados de outros sistemas a ser realizada antes do processamento dos dados:

I — efetuar a validação de dados de entrada de fontes não-confiáveis;

II — especificar o conjunto de caracteres apropriado para todas as fontes de entrada de dados;

III — codificar os dados de entrada para um conjunto de caracteres comuns antes de sua validação (*canonicalize*);

IV — rejeitar dados de entrada quando há falha no sistema de validação;

V — validar todos os dados provenientes de clientes antes do processamento, incluindo todos os parâmetros, campos de formulário, mecanismos de *postback* automáticos em código embutidos;

VI — validar os dados advindos de redirecionamentos;

VII — validar o tipo dos dados de entrada recebidos contra os esperados;

VIII — validar o intervalo dos dados de entrada recebidos contra os esperados;

IX — validar o tamanho dos dados de entrada recebidos contra os esperados;

X — validar o formato dos dados de entrada, ao limitar quais caracteres são permitidos, garantindo que os dados seguirão um padrão esperado;

XI — validar as *Uniform Resource Locator* — URLs — de redirecionamento dinâmico;

XII — codificar as rotinas de validação de dados de entrada de maneira centralizada na aplicação;

XIII — rejeitar requisições e respostas cujos valores de cabeçalho não

contenham apenas caracteres *American Standard Code for Information Interchange* — ASCII;

XIV — implementar controles adicionais de segurança no caso de caracteres potencialmente perigosos que precisem ser permitidos na entrada de dados da aplicação;

XV — aplicar verificações padrão para os dados de entrada, checando:

a) a existência de bytes nulos como %00;

b) a existência de caracteres de nova linha como %0d, %0a, \r, \n;

c) a existência de caracteres "ponto-ponto barra" como "../" ou "..\" ;

d) a existência de alteradores de caminhos;

XVI—utilizar representações alternativas como %c0%ae%c0%ae/ no caso de um conjunto de caracteres em UTF-8;

XVII — efetuar a validação de dados de entrada de qualquer fonte;

XVIII — rejeitar dados de entrada cujos caracteres estejam fora de uma lista de caracteres ou expressões regulares permitidas.

CAPÍTULO IV

DA CODIFICAÇÃO DE SAÍDA

Art. 7º As regras a seguir referem-se à adequação de formato dos dados de saída e o preparo desses dados para interações com outros sistemas:

I — escapar todos os dados provenientes de fontes não confiáveis, considerando o contexto em que serão usados como, por exemplo, construção de consultas SQL, XML, LDAP e telas em HTML;

II — tratar dados provenientes de fontes não confiáveis e que gerem comandos para o sistema operacional.

CAPÍTULO V

DA AUTENTICAÇÃO E DO GERENCIAMENTO DE CREDENCIAIS

Art. 8º Quanto à autenticação e ao gerenciamento de credenciais, será observado o seguinte:

I — não se devem armazenar senhas em texto plano sem utilizar um algoritmo de *hash* seguro com *Salt*;

II — utilizar controle de usuário e senha nominais para determinar a identidade unívoca do usuário, vedando-se o uso de credenciais por múltiplos usuários, exceto para usuários utilizados por aplicações e para acesso a APIs;

III — utilizar autenticação via ferramenta de gerenciamento de credenciais e de permissões de acesso que implemente *framework* e protocolos aprovados pelo mercado e por órgãos governantes superiores para autenticação de usuários internos;

IV — dar ciência ao usuário das permissões e níveis de acesso que possui;

V — utilizar *Hypertext Transfer Protocol Secure* — HTTPS — em todas as páginas do sistema;

VI — requerer autenticação para todas as páginas e recursos, exceto para aqueles que são intencionalmente públicos;

VII — estabelecer e utilizar serviços de autenticação padronizados e testados;

VIII — utilizar uma implementação centralizada para realizar os procedimentos de autenticação, disponibilizando bibliotecas que invoquem os serviços externos de

autenticação;

IX — separar a lógica de autenticação do recurso que está sendo requisitado e usar redirecionadores nos controladores de autenticação centralizados;

X — evitar indicar qual parte dos dados de autenticação está incorreta nas mensagens de falha na autenticação, evitando, por exemplo, exibir mensagens como "Nome de usuário incorreto" ou "Senha incorreta", utilizando apenas mensagens como "Usuário e/ou senha inválidos" para ambos os casos de erro e evitar também a função e método em que o erro ocorreu;

XI — utilizar autenticação para conexão a sistemas externos que envolvam tráfego de informação sensível ou acesso a funções;

XII — cifrar e armazenar em local protegido de um sistema confiável as credenciais de autenticação para acessar serviços externos à aplicação;

XIII — evitar armazenar as credenciais de autenticação no código-fonte da aplicação e em arquivos de configuração;

XIV — evitar armazenar as credenciais de autenticação ou qualquer outro dado sensível em imagem de *container*;

XV — notificar o usuário quando a sua senha for alterada;

XVI — comunicar a data/hora da última utilização — bem ou malsucedida — de uma conta de usuário no próximo acesso ao sistema;

XVII — modificar todas as senhas e os identificadores de usuários — Ids — que, por padrão, são definidas pelos fornecedores;

XVIII — exigir nova autenticação dos usuários antes da realização de operações críticas;

XIX — garantir que o código utilizado para o processo de autenticação não contenha código malicioso;

XX — evitar validar os dados de autenticação antes do final de todas as entradas de dados, especialmente nas implementações de autenticação sequencial;

XXI — utilizar apenas requisições POST para transmitir credenciais de autenticação;

XXII — exigir a mudança de senhas temporárias na próxima vez que o usuário realizar a autenticação no sistema;

XXIII — desativar a funcionalidade de lembrar a senha nos campos de senha do navegador;

XXIV — armazenar de forma segura os dados de usuários e de sistemas que utilizam cada senha fornecida;

XXV — evitar utilizar as mesmas senhas para ambientes de desenvolvimento, homologação ou produção;

XXVI — utilizar certificado digital para determinar a identidade do usuário;

XXVII — realizar monitoramento para identificar ataques contra várias contas de usuários que utilizem a mesma senha;

XXVIII — utilizar autenticação de múltiplos fatores;

XXIX — garantir que, uma vez autenticado, o usuário esteja impedido de acessar o sistema de outro endereço *Internet Protocol* — IP —, a menos que se autentique novamente.

CAPÍTULO VI

DO GERENCIAMENTO DE SENHAS

Art. 9º Quanto ao gerenciamento de senhas será observado o seguinte:

I — não se devem utilizar senhas com menos de 12 (doze) caracteres;

II — utilizar pelo menos letras maiúsculas e minúsculas, junto a ao menos um tipo de caractere (dígito, símbolo);

III — não se devem usar palavras comumente utilizadas para senhas ou variantes destas;

IV — não se devem armazenar senhas não cifradas;

V — armazenar ao menos o *hash* criptográfico com *Salt*;

VI — não se deve usar um canal em claro para a transmissão da senha ou elemento correspondente;

VII — não se deve utilizar método de conferência menos seguro que desafios baseados em *hash* ou o uso de *hashes* armazenados;

VIII — não se deve mostrar diretamente a senha quando esta necessita ser digitada pelo usuário — deve haver opção de habilitar e desabilitar a visualização da senha digitada até então;

IX — não se deve utilizar periodicidade de troca de senhas superior a 6 (seis) meses;

X — não se deve enviar a senha antiga para o usuário, em claro ou não;

XI — não se deve armazenar senha criptografada abaixo do nível mínimo de criptografia estabelecido neste documento;

XII — não se deve permitir uma taxa de tentativas de validação de senha superior a 3 (três) tentativas por minuto;

XIII — bloquear a conta de usuário em caso de 5 (cinco) erros de autenticação consecutivos;

XIV — elaborar senhas com auxílio de *software* gerador de senhas aleatórias, configurado para atender aos parâmetros aqui estabelecidos;

XV — utilizar senha que tenha sido validada por um *software* testador de força de senhas.

Art. 10. Quanto ao gerenciamento de senhas é recomendável:

I — utilizar senhas com mais de 20 (vinte) caracteres;

II — utilizar senha que tenha sido validada por um *software* testador de força de senhas diferente do *software* gerador de senhas;

III — armazenar senha que esteja criptografada seguindo o nível forte de criptografia estabelecido neste documento;

IV — utilizar um método de prova com conhecimento zero de senha;

V — exigir prova de origem da requisição.

CAPÍTULO VII

DO GERENCIAMENTO DE SESSÕES

Art. 11. Os dispositivos a seguir são orientados para o gerenciamento de sessões de usuário, visando garantir a autenticidade e o correto exercício de permissões do usuário enquanto durar sua sessão no sistema.

Art. 12. Em relação ao controle de sessões, será observado o seguinte:

I — utilizar controles de gerenciamento de sessão baseados no servidor ou em

frameworks;

II — não se deve definir o domínio e o caminho para os *cookies* que contenham identificadores de sessão autenticados para um endereço externo ao *site*;

III — configurar o atributo "*secure*" para *cookies* transmitidos através de uma conexão *Transport Layer Security* — TLS;

IV — configurar os *cookies* com o atributo "*HttpOnly*", a menos que seja explicitamente necessário ler ou definir os valores dos mesmos através de *scripts* do lado cliente da aplicação;

V — convém utilizar mecanismos complementares ao mecanismo padrão de gerenciamento de sessões para operações sensíveis do lado servidor como, por exemplo, no caso de operações de gerenciamento de contas através da utilização de *tokens* aleatórios ou parâmetros associados à sessão.

Art. 13. Em relação à criação de sessões, será observado o seguinte:

I — não se deve permitir o estabelecimento de sessão caso a aplicação não consiga ter acesso às informações contidas na configuração de segurança;

II — não se devem reconhecer identificadores gerados por controles fora do servidor ou do *framework* de controle como válidos;

III — não se devem permitir *logins* persistentes (sem prazo de expiração).

Art. 14. Em relação à manutenção de sessões, será observado o seguinte:

I — não se deve reaproveitar uma sessão estabelecida antes do *login* em caso de nova autenticação;

II — não se deve reaproveitar um identificador de sessão quando houver uma nova autenticação;

III — não se devem expor os identificadores de sessão em URLs, mensagens de erro ou *logs*;

IV — proteger os dados de sessão do lado servidor contra acessos não autorizados por outros usuários do mesmo servidor, inclusive durante a sessão vigente;

V — notificar o usuário clara e constantemente a respeito do tempo de encerramento de sessão;

VI — não convém permitir conexões simultâneas com o mesmo identificador de sessão.

Art. 15. Em relação ao término das sessões, será observado o seguinte:

I — encerrar completamente a sessão ou conexão associada no *logout*;

II — disponibilizar a funcionalidade de *logout* em todas as páginas que requerem autenticação;

III — estabelecer um tempo de expiração da sessão que seja o mais curto possível, baseado no balanceamento dos riscos e requisitos funcionais do negócio;

IV — convém realizar o encerramento da sessão periodicamente, mesmo quando estiver ativa.

CAPÍTULO VIII

DO CONTROLE DE ACESSOS

Art. 16. Em relação à implementação de controle de acesso, será observado o

seguinte:

I — restringir o acesso às URLs protegidas somente aos usuários autorizados;

II — restringir o acesso às funções protegidas, às referências diretas aos objetos, aos serviços e aos dados da aplicação somente aos usuários autorizados;

III — restringir o acesso aos atributos e dados do usuário, às informações de políticas dos mecanismos de controle de acesso e às configurações de segurança relevantes somente aos usuários autorizados;

IV — restringir o acesso a arquivos somente aos usuários autorizados;

V — não se deve utilizar o campo *"referer"* do cabeçalho como forma de verificação principal;

VI — implementar uma política de privilégio mínimo para as contas de acesso a sistemas, contas de serviço e contas de suporte a conexões provenientes ou destinadas a serviços externos;

VII — utilizar um único componente em toda a aplicação *Web* para realizar o processo de verificação de autorização de acesso — isto inclui bibliotecas que invocam os serviços externos de autorização;

VIII — exigir nova autenticação caso os privilégios do usuário tenham sido modificados durante uma sessão;

IX — prover suporte à desativação de contas e ao encerramento das sessões quando terminar a autorização do usuário;

X — garantir o controle de autorização em todas as requisições, inclusive em *scripts* do lado servidor, *includes* e requisições provenientes de tecnologias do lado cliente, como *Asynchronous Javascript and XML* — AJAX;

XI — convém fazer a revalidação periódica da autorização do usuário para garantir que os privilégios não foram modificados;

XII — não convém aplicar regras de controle de acesso representadas pela camada de apresentação divergentes das regras presentes no lado servidor;

XIII — convém implementar a auditoria das contas de usuário e assegurar a desativação de contas não utilizadas;

XIV — convém utilizar mecanismos de criptografia e verificação de integridade no lado servidor para detectar possíveis adulterações em dados armazenados no lado do cliente;

XV — convém limitar o número de transações que um único usuário ou dispositivo pode executar em determinado período de tempo.

Art. 17. As práticas de criptografia seguirão as normas elencadas na NSC12 – Uso de Recursos Criptográficos.

CAPÍTULO IX

DO TRATAMENTO E REGISTRO DE ERROS

Art. 18. No que se refere à manutenção de *logs* para posterior auditoria, rastreamento e consulta de incidentes ligados à segurança dos sistemas, será observado o seguinte:

I — definir no documento de especificação de requisitos do sistema quais informações deverão ser registradas, o local e o tempo mínimo de armazenamento dos dados da auditoria;

II — não se devem expor informações sensíveis nas respostas aos erros,

inclusive detalhes de sistema, identificadores de sessão ou informação da conta do usuário;

III — não se devem armazenar informações sensíveis nos registros de *logs*, como detalhes desnecessários do sistema, identificadores de sessão e senhas;

IV — restringir o acesso aos *logs* apenas para pessoal autorizado;

V — utilizar mensagens de erro genéricas, sem especificações que permitam inferir em que parte do sistema o erro ocorreu, sua causa, informações de depuração (*debug*) ou informações da pilha de exceção;

VI — convém registrar em *log* todas as falhas de acesso;

VII — convém utilizar um serviço centralizado para realizar todas as operações de *log*;

VIII — convém utilizar páginas de erro personalizadas, ou seja, substituir as páginas de erro https do navegador por páginas próprias;

IX — convém registrar em *log* todas as falhas de conexão TLS com o *back-end*;

X — convém registrar em *log* todas as falhas que ocorreram nos módulos de criptografia;

XI — convém registrar em *log* todas as exceções lançadas pelo sistema;

XII — convém registrar em *log* todo o uso de funções administrativas, inclusive as mudanças realizadas nas configurações de segurança;

XIII — convém utilizar uma função de *hash* criptográfica para validar a integridade dos registros de *log*;

XIV — convém registrar em *log* todas as falhas de validação de entrada de dados;

XV — convém definir no documento de especificação de requisitos do sistema quais são as políticas de retenção;

XVI — convém registrar em *log* as tentativas de conexão com *tokens* de sessão inválidos ou expirados;

XVII — convém registrar em *log* todos os eventos suspeitos de adulteração, inclusive alterações inesperadas no estado dos dados.

CAPÍTULO X

DA PROTEÇÃO DE DADOS

Art. 19. Os dispositivos a seguir tratam do armazenamento de informações com grau de sigilo e de sua disponibilização.

Art. 20. Quanto ao sigilo, os dados serão classificados da forma apresentada a seguir, conforme a Resolução TRE-MG n° 1.172, de 12 de maio de 2021, que regulamenta o acesso e a classificação da informação quanto à confidencialidade no Tribunal, e a Lei n° 13.709, de 14 de agosto de 2018, a Lei Geral de Proteção de Dados Pessoais:

I— como "Abertos": os dados considerados como informação pública, assim considerados por força de lei ou cuja divulgação não cause qualquer dano, podendo seu acesso ser franqueado a qualquer pessoa;

II— como "Fechados": os dados relativos a informação sigilosa, classificada como secreta, ultrassecreta ou reservada, nos termos da Lei nº 12.527, de 18 de novembro de 2011, e os dados pessoais considerados sensíveis ou não públicos, tratados conforme especifica a Lei Geral de Proteção de Dados Pessoais.

Art. 21. A correta classificação dos domínios de dados existentes nas bases de

dados do Tribunal requer que tais domínios estejam registrados, classificados e disponíveis em um dicionário de dados passível de fácil alteração e administração.

Art. 22. Em relação aos procedimentos e meios para armazenamento de dados abertos, será utilizado meio de armazenamento que possua acesso para escrita restrito por senha.

Art. 23. Quanto aos procedimentos e meios para armazenamento de dados fechados, será observado o seguinte:

I — utilizar meio de armazenamento que possua acesso para leitura e escrita restrito por senha;

II — armazenar dados criptografados, sempre que possível.

Art. 24. Quanto à concessão para acesso a informações em bancos de dados, será o observado o seguinte:

I — não se deve disponibilizar às aplicações acesso a algum banco de dados utilizando *login* de usuário com permissões de usuário *root* ou equivalente;

II — não se deve disponibilizar às aplicações acesso a algum banco de dados utilizando *login* de usuário com permissões para execução de comandos em *Data Definition Language* — DDL;

III — não se deve disponibilizar às aplicações acesso a algum banco de dados utilizando *login* de usuário com permissões além das estritamente necessárias ao seu funcionamento;

IV — estabelecer correspondência "um-para-um" entre cada usuário de uma dada aplicação e do banco de dados, sempre que possível.

Art. 25. Quanto ao tratamento de dados em aplicações, será observado o seguinte:

I — não se devem incluir informações sensíveis nos parâmetros de requisição HTTP GET;

II — não se deve publicar documentação do sistema que possa revelar informações importantes para potenciais atacantes;

III — criptografar informações altamente sensíveis quando armazenadas;

IV — proteger o código-fonte presente no servidor para que não seja acessado por algum usuário sem permissão;

V — não se devem armazenar senhas, *strings* de conexão ou outras informações confidenciais em texto claro/legível ou em qualquer forma criptograficamente insegura no lado cliente;

VI — remover comentários do código de produção que podem ser acessados pelos usuários;

VII — convém desativar a *cache* realizada no lado cliente das páginas que contenham informações sensíveis;

VIII — convém proteger contra acesso não autorizado todas as cópias temporárias ou registradas em *cache* que contenham dados sensíveis e estejam armazenadas no servidor;

IX — convém prover à aplicação a faculdade de remover dados sensíveis, quando desnecessários.

CAPÍTULO XI

DA SEGURANÇA NAS COMUNIDADES

Art. 26. No que se refere à transmissão segura de dados sensíveis entre sistemas, de modo a salvaguardar a integridade, autenticidade e demais atributos pertinentes ao uso dos dados comunicados, será observado o seguinte:

I — utilizar criptografia na transmissão de todas as informações sensíveis;

II — empregar canais de comunicação com controle de duplicação e perda de informações/mensagens;

III — empregar canal de comunicação que forneça controle de integridade dos dados transmitidos;

IV — empregar canal de comunicação que forneça confidencialidade dos dados transmitidos;

V — não se devem utilizar certificados TLS inválidos, com nome de domínio incorreto ou expirados;

VI — não se deve fornecer uma conexão insegura quando ocorrer alguma falha nas conexões TLS;

VII — utilizar TLS para conexões com sistemas externos que envolvam funções ou informações sensíveis;

VIII — empregar um canal de comunicação com controle de autenticação;

IX — armazenar de maneira segura os dados a serem transmitidos em ambas as extremidades da comunicação;

X — especificar a codificação dos caracteres para todas as conexões;

XI — filtrar os parâmetros que contenham informações sensíveis, provenientes do "HTTP referer", nos *links* para *sites* externos;

XII — convém empregar canal de comunicação que garanta "não-repúdio" dos dados transmitidos;

XIII — convém utilizar *logs* confiáveis das informações transmitidas, com confirmação de entrega e recepção das mensagens;

XIV — convém utilizar um padrão único de implementação TLS, configurado de modo apropriado.

CAPÍTULO XII

DAS CONFIGURAÇÕES DE SISTEMA

Art. 27. Para configurações de plataforma e tratamento de requisições, será observado o seguinte:

I — restringir para o mínimo possível os privilégios do servidor *Web*, dos processos e das contas de serviços;

II — remover código de teste ou qualquer funcionalidade desnecessária para o ambiente de produção antes da instalação do sistema no servidor de produção;

III — definir quais métodos de requisição a aplicação irá suportar e se serão tratados de modo diferenciado nas diversas páginas da aplicação;

IV — não se devem manter informações desnecessárias presentes nos cabeçalhos de resposta HTTP e que podem estar relacionadas com o sistema operacional, versão do servidor *Web* e *frameworks* de aplicação;

V — isolar o ambiente de desenvolvimento da rede de produção e conceder

acesso somente para grupos de desenvolvimento e testes;

VI — garantir que os servidores, *frameworks* e componentes do sistema estejam executando a última versão aprovada, com as atualizações de segurança mais recentes e que sejam compatíveis com as necessidades do sistema;

VII — desativar a listagem de diretórios do servidor *Web*;

VIII — configurar o arquivo *robots.txt* adequadamente de forma a prevenir a divulgação da estrutura de diretórios e impedir que robôs de busca façam indexação de arquivos que não devem ser indexados;

IX — desativar as extensões HTTP desnecessárias.

Art. 28. Quanto às configurações de plataforma e tratamento de requisições é recomendável:

I — remover todas as funcionalidades e arquivos desnecessários;

II — certificar de que, no caso de o servidor processar tanto requisições HTTP 1.0 como HTTP 1.1, ambas as versões estejam configuradas de modo semelhante;

III — implementar um sistema de gestão de ativos para manter o registro dos componentes e programas.

Art. 29. Quanto à utilização de cabeçalhos de segurança, será observado o seguinte:

I — configurar o cabeçalho "*Content-Type*" em todas as respostas do servidor, sendo que o conteúdo da resposta deve corresponder ao "*Content-Type*" configurado;

II — configurar o cabeçalho "*Content-Security-Policy*" — CSP — nas respostas HTTP para ajudar a reduzir o impacto de ataques XSS que usam injeção de HTML, DOM, JSON ou *JavaScript*;

III — configurar o cabeçalho "*X-Content-Type-Options: nosniff*" em todas as respostas do servidor;

IV — configurar nas respostas do servidor o cabeçalho "*X-Frame-Options*" e/ou o cabeçalho "*Content-Security-Policy: frame-ancestor*" para prevenir a possibilidade da aplicação ser embutida em *iframes* de *sites* de terceiros não autorizados.

Art. 30. Quanto à utilização de cabeçalhos de segurança, será observado o seguinte:

I — configurar o cabeçalho "*Strict-Transport-Security*" em todas as respostas e para todos os subdomínios, tal como "*Strict-Transport-Security: max-age*=15724800; includeSubdomains";

II — configurar o cabeçalho "*Referrer-Policy*" para evitar o vazamento de informações presentes na URL para *websites* não confiáveis por meio do cabeçalho "*Referer*".

Art. 31. Quanto ao acesso ao código-fonte, será observado o seguinte:

I — utilizar um sistema controle de versões para provimento de rastreabilidade de modificações e controle de acesso ao código-fonte compatível com *git*;

II — franquear o livre acesso aos códigos-fontes pelos desenvolvedores, sendo que outras situações devem ser analisadas projeto a projeto.

Art. 32. Quanto à separação de ambientes, será observado o seguinte:

I — utilizar bancos de dados distintos para cada ambiente;

II — utilizar servidores de aplicação/*Web* distintos para cada ambiente;

III — prover acesso ao ambiente de desenvolvimento/testes/homologação apenas aos integrantes da equipe de desenvolvimento e aos interessados no projeto (*stakeholders*);

IV — prover um instalador expresso para a instalação do ambiente necessário para a execução de uma dada aplicação;

V — não se devem fornecer as senhas de acesso ao ambiente de produção aos desenvolvedores, sempre que possível;

VI — realizar testes periódicos para assegurar a segurança do ambiente de desenvolvimento/testes/homologação, sempre que possível.

CAPÍTULO XIII

DA SEGURANÇA EM BANCOS DE DADOS

Art. 33. Quanto à segurança em bancos de dados, será observado:

I — usar consultas parametrizadas fortemente "tipadas";

II — certificar de que as variáveis são fortemente "tipadas";

III — utilizar validação de entrada e codificação de saída e, se houver falha, o comando não deverá ser executado no banco de dados;

IV — realizar a codificação (*escaping*) de meta caracteres em instruções SQL;

V — não se devem incluir *strings* de conexão no código da aplicação;

VI — eliminar o conteúdo desnecessário incluído por padrão pelo fornecedor, como em esquemas e bancos de dados de exemplo;

VII — desativar todas as contas criadas por padrão e que não sejam necessárias para suportar os requisitos de negócio;

VIII — atribuir à aplicação o menor nível possível de privilégios ao acessar o banco de dados;

IX — não se devem criar SQLs concatenando parâmetros textuais de origem não segura, como parâmetros preenchidos pelo usuário ou mesmo armazenados no banco de dados;

X — utilizar tratamento especial para consultas que não podem ser parametrizadas, como *escapes* ou codificação em hexadecimal;

XI — convém usar, sempre que possível, procedimentos armazenados (*stored procedures*) para abstrair o acesso aos dados e permitir a remoção de permissões das tabelas no banco de dados;

XII — convém encerrar a conexão com o banco de dados assim que possível.

CAPÍTULO XIV

DO GERENCIAMENTO DE ARQUIVOS

Art. 34. Quanto à edição, distribuição, armazenamento e concessão de permissões de arquivos utilizados e gerados pelas aplicações, será observado o seguinte:

I — prevenir ou restringir o carregamento de qualquer arquivo que possa ser interpretado e/ou executado pelo servidor *Web*;

II — desativar privilégios de execução nos diretórios de armazenamento de arquivos;

III — não se devem passar caminhos de diretórios ou de arquivos em requisições;

IV — não se deve enviar o caminho absoluto do arquivo para o lado cliente de uma aplicação ou para o usuário;

V — certificar de que os arquivos da aplicação e os recursos estão definidos com atributo de "somente leitura";

VI — requerer autenticação antes de se permitir que seja feito o carregamento de arquivos;

VII — validar se os arquivos enviados são do tipo esperado através da validação dos cabeçalhos;

VIII — convém usar uma lista branca (*whitelist*) de nomes e de tipos de arquivos permitidos ao referenciar arquivos;

IX — convém limitar, se possível, os tipos de arquivos que podem ser enviados para aceitar somente os necessários ao propósito do negócio;

X — convém implantar, se possível, o carregamento seguro de arquivos nos ambientes UNIX por meio da montagem do diretório de destino como uma unidade *lógica*;

XI — convém verificar, se possível, os arquivos que os usuários submeterem através do mecanismo de carregamento em busca de vírus e *malwares*;

XII — não convém repassar dados fornecidos pelos usuários diretamente a uma função de inclusão dinâmica;

XIII — não convém salvar arquivos no mesmo diretório de contexto da aplicação *Web.*

CAPÍTULO XV

DO GERENCIAMENTO DE MEMÓRIA

Art. 35. Quanto ao gerenciamento de memória, será observado o seguinte:

I — verificar se o *buffer* é, de fato, tão grande quanto o especificado;

II — verificar os limites do *buffer* caso as chamadas à função sejam realizadas em ciclos;

III — verificar se há algum risco de ocorrer gravação de dados além do espaço reservado em *buffer*;

IV — liberar a memória alocada de modo apropriado após concluir a sub-rotina (função/método) e em todos os pontos de saída;

V — truncar todas as *strings* de entrada para um tamanho razoável antes de passar para as funções de cópia e concatenação;

VI — utilizar, se possível, pilhas não-executáveis, quando disponíveis.

CAPÍTULO XVI

DA PREVENÇÃO, REAÇÃO E MITIGAÇÃO DE FALHAS DE SEGURANÇA

Art. 36. Serão observados os procedimentos previstos na Norma de Geração e Restauração de Cópias de Segurança — NSC6 — para preservação e recuperação dos conteúdos dos repositórios de projetos de desenvolvimento de sistemas.

Parágrafo único. É recomendável criar *baselines* das versões do sistema, para facilitar a recuperação ágil para uma versão anterior.

Art. 37. Quanto aos testes, será observado o seguinte:

I — realizar testes manuais de segurança antes de cada versão do *software* em que sua estrutura tenha sido modificada;

II — garantir, através de testes automatizados, que os serviços e dados sigilosos estão protegidos e disponíveis apenas para os usuários detentores das informações;

III — elaborar uma política de testes, automatizados ou não, visando à garantia de não vulnerabilidade aos principais ataques conhecidos em sistemas;

IV — definir cenários de testes voltados à garantia dos requisitos não funcionais do *software*, preferencialmente realizado por uma equipe de testes diferente da equipe de desenvolvimento do *software*, com intuito de se evitarem vícios;

V — definir cenários de testes, principalmente nos aspectos de segurança, para os casos de atualizações na arquitetura do sistema;

VI — convém propor constantes desafios entre as equipes para testar a segurança dos sistemas em formato de competição;

VII — convém submeter os sistemas a ferramentas de testes de invasão;

VIII — convém submeter imagens de container, de sistemas que utilizam essa tecnologia, à análise de vulnerabilidades;

IX — convém submeter o código do sistema a ferramentas de análise estática de segurança (*SAST*).

Art. 38. Quanto a ocorrências, será mantido procedimento planejado para imediata indisponibilização do sistema e realização de manutenção corretiva.

CAPÍTULO XVII

DO CICLO DE VIDA DO *SOFTWARE*

Art. 39. A lista de riscos do projeto deverá ser mantida atualizada durante a fase de projeto com as ameaças, a definição clara dos riscos de segurança e o nível de severidade que um eventual comprometimento de dados sensíveis traria ao sistema e à instituição.

Art. 40. Na fase de codificação, serão documentadas, inclusive no código da aplicação, as medidas protetivas aplicadas no código-fonte, de modo a indicar precisamente o procedimento utilizado e suas peculiaridades.

Parágrafo único. É recomendável na fase de codificação utilizar mecanismos de verificação de integridade por *checksum* ou *hash* para verificar a integridade do código interpretado, bibliotecas, arquivos executáveis e arquivos de configuração.

Art. 41. Quanto à fase de manutenção, será observado o seguinte:

I — desabilitar as atualizações automáticas de *software* ou componentes utilizados na construção de um sistema, sob pena de introdução indevida de falhas de segurança e de incidentes;

II — restringir a modificação dos *softwares* de terceiros, de modo a evitar o risco de invalidação de controles de segurança internos, salvo quando estritamente necessário, reforçando-se que toda mudança, sempre que possível, deve ser feita pelo desenvolvedor original do *software*.

CAPÍTULO XVIII

DAS DISPOSIÇÕES FINAIS

Art. 42.O descumprimento desta portaria será imediatamente registrado como incidente de segurança, e caberá ao Gestor de Segurança da Informação, caso julgue necessário e a depender da gravidade do incidente, a comunicação à Comissão de Segurança da Informação, para apuração e consequente adoção das providências cabíveis.

Art. 43. Os casos omissos serão resolvidos pela Comissão de Segurança da Informação.

Art. 44. Esta portaria entra em vigor na data de sua publicação.

Belo Horizonte, 16 de setembro de 2024.

Desembargador Ramom Tácio de Oliveira
Presidente

Documento assinado eletronicamente por **RAMOM TÁCIO DE OLIVEIRA**, **Presidente**, em 16/09/2024, às 16:51, conforme art. 1º, III, "b", da Lei 11.419/2006.

A autenticidade deste documento pode ser conferida no site https://sei.tre-mg.jus.br/controlador_externo.php?acao=documento_conferir&acao_origem=documento_conferir&lang=pt_BR&id_orgao_acesso_externo=0, informando o código verificador **5680314** e o código CRC **E83DB851**.