EVERESTE
INSTITUTO DE TECNOLOGIA E INOVAÇÃO

GUIA DE

DESENVOLVIMENTO DE SOFTWARE SEGURO

MANAUS
2024

SUMÁRIO

APRESENTAÇÃO E OBJETIVOS 3
FORMATO 3
DIRETRIZES 3
ESTRUTURA 4
DIRETRIZES PARA DESENVOLVIMENTO DE SOFTWARE SEGURO 4
GERENCIAMENTO DE SESSÃO 4
VALIDAÇÃO DOS DADOS DE ENTRADA 5
CODIFICAÇÃO DE SAÍDA 7
AUTENTICAÇÃO E GERENCIAMENTO DE CREDENCIAIS 8
GERENCIAMENTO DE SENHAS 11
GERENCIAMENTO DE SESSÕES 14
CONTROLE DE SESSÃO 14
MANUTENÇÃO DE SESSÃO 15
TÉRMINO DA SESSÃO 16
CONTROLE DE ACESSOS 17
PRÁTICAS DE CRIPTOGRAFIA 18
PROTEÇÃO DE DADOS 20
PROCEDIMENTOS E MEIOS PARA ARMAZENAMENTO DE DADOS ABERTOS 20
PROCEDIMENTOS E MEIOS PARA ARMAZENAMENTOS DE DADOS FECHADOS 21
PERMISSÕES PARA ACESSO A INFORMAÇÃO EM BANCOS DE DADOS 21
TRATAMENTO DE DADOS E APLICAÇÕES 22
SEGURANÇA NAS COMUNICAÇÕES 23
CONFIGURAÇÕES DE SISTEMA 25
CONFIGURAÇÕES DE PLATAFORMA E TRATAMENTO DE REQUISIÇÕES 25
ACESSO AO CÓDIGO-FONTE 26
SEPARAÇÃO DE AMBIENTES 27
SEGURANÇA EM BANCO DE DADOS 27
GERENCIAMENTOS DE ARQUIVOS 29
DIRETRIZES PARA DESENVOLVIMENTO SEGURO DE SOFTWARE 30
PREVENÇÃO, REAÇÃO E MITIGAÇÃO DE FALHAS DE SEGURANÇA 30
BACKUPS 30
TESTES 31
OCORRÊNCIAS 32
PROJETO 33
CODIFICAÇÃO 34
MANUTENÇÃO 34

PESSOAL 34
GLOSSÁRIO 35
REFERÊNCIAS E INDICAÇÕES 38

APRESENTAÇÃO E OBJETIVOS

Como Instituto de Tecnologia, é fundamental observar os princípios de segurança da informação ao desenvolver sistemas e aplicações em um ambiente digital, especialmente quando se trata de serviços disponibilizados por meio de aplicações Web. Isso evita a possibilidade de ataques cibernéticos que podem ter impactos devastadores.

Para garantir a qualidade e segurança dos sistemas desenvolvidos no âmbito de nossa instituição, criamos um guia de boas práticas para o desenvolvimento seguro de software. Destinado a analistas, técnicos, desenvolvedores e instaladores de software, o objetivo deste documento é tornar os processos de concepção de sistemas mais seguros, confiáveis, auditáveis e estáveis.

Para tornar nossos sistemas e serviços em ambiente Web ainda mais seguros, estamos sempre atualizando nossas diretrizes com as melhores práticas de segurança recomendadas pelo setor, tais como as diretivas do OWASP Secure Coding Practices Quick Reference Guide, com material adicional extraído do OWASP Application Security Verification Standard 4.02 e outras referências relevantes.

FORMATO

O documento é estruturado em torno de “diretrizes”, recomendações de boas práticas a serem seguidas. Seu formato é detalhado abaixo.

DIRETRIZES

As diretrizes são regras claras e classificadas em dois níveis: mínimo (M) e complementar (C). Cada diretriz tem um identificador único formado por uma letra indicando o nível e uma numeração. As diretrizes são apresentadas em formato tabular, com as de nível mínimo primeiro e as de nível complementar depois.CC

ESTRUTURA

Este guia é destinado aos desenvolvedores de software envolvidos na construção de sistemas do Instituto Evereste, fornecendo informações detalhadas e de fácil acesso. O guia é composto por dois tópicos principais: "Diretrizes para Desenvolvimento de Software Seguro", com orientações sobre práticas de programação, criptografia, senhas fortes, entre outros, e "Diretrizes para Desenvolvimento Seguro de Software", com boas práticas para versionamento, controle de acesso ao código-fonte e testes de software. Além disso, um glossário está disponível para detalhamento adicional.

DIRETRIZES PARA DESENVOLVIMENTO DE SOFTWARE SEGURO

Todos os itens que dizem respeito ao desenvolvimento de sistemas Web assumem que as operações são realizadas em ambientes municiados por uma infraestrutura segura a priori.

ID	DIRETRIZ
M1	Não se deve utilizar diretrizes sem o respectivo ID.
M2	Não se deve atribuir o mesmo ID a duas ou mais diretrizes distintas.
C1	Deve-se preservar o ID de diretrizes declaradas obsoletas.
C2	Não se deve aproveitar IDs de diretrizes declaradas obsoletas em novas diretrizes.

GERENCIAMENTO DE SESSÃO

Diretrizes para auxílio na construção de um projeto arquitetural de software seguro, focando-se em práticas gerais de programação e no desenho de seus componentes.

ID	DIRETRIZ
M3	Deve-se isolar no código da aplicação os trechos de código que contêm lógica privilegiada
M4	Deve-se 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. *Eg*., reconhecer representação de sinal, valores do tipo “Not-A-Number” (NaN), valores especiais, etc.
M5	Deve-se proteger as variáveis e os recursos compartilhados contra acessos concorrentes inapropriados.
M6	Deve-se 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*).
M7	Deve-se 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.

VALIDAÇÃO DOS DADOS DE ENTRADA

Diretrizes para validação de dados de entrada do usuário e do recebimento de dados de outros sistemas a ser realizada antes do processamento dos dados.

ID	DIRETRIZ

M8	Deve-se efetuar a validação de dados de entrada de fontes não-confiáveis. Eg., dados inseridos por usuários, base de dados externas, fluxos de arquivos, etc.
M9	Deve-se especificar o conjunto de caracteres apropriado — *eg*., UTF-8 —, para todas as fontes de entrada de dados.
M10	Deve-se codificar os dados de entrada para um conjunto de caracteres comuns antes de sua validação (*canonicalize*).
M11	Deve-se rejeitar dados de entrada quando há falha no sistema de validação.
M12	Deve-se validar todos os dados provenientes de clientes antes do processamento, incluindo todos os parâmetros, campos de formulário, mecanismos de *postback* automáticos em código embutidos — *eg*., Javascript —, conteúdos das URLs e cabeçalhos HTTP — *eg*., os nomes e os valores dos *Cookies*.
M13	Deve-se validar os dados advindos de redirecionamentos.
M14	Deve-se validar o tipo dos dados de entrada recebidos contra os esperados.
M15	Deve-se validar o intervalo dos dados de entrada recebidos contra os esperados.
M16	Deve-se validar o tamanho dos dados de entrada recebidos contra os esperados.
M17	Deve-se validar o formato dos dados de entrada, limitando quais caracteres são permitidos e garantindo que os dados seguem um padrão esperado. *Eg*., CPF, endereço de *e-mail*, número de telefone, CEP.
M18	Deve-se validar que URLs de redirecionamento dinâmico — *eg*., URLs recebidas por parâmetros — estejam em uma lista de URLs permitidas pelo sistema antes de realizar o redirecionamento ou, alternativamente, deve-

	se mostrar um aviso de redirecionamento para conteúdo potencialmente não confiável.
M19	Deve-se codificar as rotinas de validação de dados de entrada de maneira centralizada na aplicação.
M20	Deve-se efetuar a descodificação UTF-8 caso o sistema suporte o conjunto de caracteres extendidos de UTF-8.
M21	Deve-se rejeitar requisições e respostas cujos valores de cabeçalho não contenham apenas caracteres ASCII.
M22	Deve-se implementar controles adicionais de segurança caso caracteres potencialmente perigosos — precisem ser permitidos na entrada de dados da aplicação.
M23	Deve-se aplicar verificações padrão para os dados de entrada, checando: a existência de *bytes* nulos como %00; a existência de caracteres de nova linha como %0d, %0a, \r, \n; a existência de caracteres "ponto-ponto barra" como "../" ou "..\" e a existência de alteradores de caminhos. No caso de um conjunto de caracteres em UTF-8, o sistema deve utilizar representações alternativas como %c0%ae%c0%ae/
C3	Deve-se efetuar a validação de dados de entrada de qualquer fonte.
C4	Não se deve aceitar dados de entrada cujos caracteres estejam fora de uma lista de caracteres ou expressões regulares permitidas.

CODIFICAÇÃO DE SAÍDA

Diretrizes sobre a adequação de formato dos dados de saída e o preparo desses dados para interações com outros sistemas.

ID	DIRETRIZ
M24	Deve-se codificar todos os caracteres, a menos que sejam conhecidos por serem seguros para o interpretador de destino.
M25	Deve-se escapar todos os dados provenientes de fontes não confiáveis, considerando o contexto em que serão usados. *Eg.*, construção de consultas SQL, XML, LDAP e telas em HTML.
M26	Deve-se preservar o ID de diretrizes declaradas obsoletas.

AUTENTICAÇÃO E GERENCIAMENTO DE CREDENCIAIS

Diretrizes para a verificação da identidade de usuários ao realizarem operações nos sistemas e para determinação de identidade do usuário e seu correspondente nível de acesso às informações.

Observação. Autenticação AD versus OAuth2.

No OAuth2, ao contrário do AD, a autenticação é feita diretamente em uma página externa, via HTTPS, sendo que o sistema web não tem acesso às credenciais pelo usuário. Depois do login, a página externa retorna um token para a aplicação. Esse token garante que o usuário foi autenticado corretamente

No OAuth2 é exigida uma conexão ativa com a internet. Em situações de contingência onde o usuário não tenha acesso à página externa, não será possível autenticar - se no sistema.

ID	DIRETRIZ
M27	Não se deve armazenar senhas em texto plano sem utilizar um algoritmo de *hash* seguro com *salt*.
M28	Deve-se utilizar controle de usuário e senha nominais para determinar a identidade unívoca do usuário, vedando-se o uso de credenciais por múltiplos usuários.
M29	Deve-se utilizar autenticação via AD e/ou o *framework* OAuth2 sempre que possível para autenticar usuários internos.
M30	Deve-se utilizar grupos de *Active Directory* (AD) para determinar as políticas de acesso e roles de usuário.
M30	Deve-se utilizar grupos de *Active Directory* (AD) para determinar as políticas de acesso e roles de usuário.
M31	Deve-se dar ciência ao usuário das permissões e níveis de acesso que possui.
M32	Deve-se utilizar HTTPS em todas as telas do sistema.
M33	Deve-se requerer autenticação para todas as páginas e recursos, exceto para aqueles que são intencionalmente públicos.
M34	Deve-se estabelecer e utilizar serviços de autenticação padronizados e testados.
M35	Deve-se utilizar uma implementação centralizada para realizar os procedimentos de autenticação, disponibilizando bibliotecas que invoquem os serviços externos de autenticação.
M36	Deve-se separar a lógica de autenticação do recurso que está sendo requisitado e usar redirecionadores nos controladores de autenticação centralizados.

M37	Não se deve indicar qual parte dos dados de autenticação está incorreta nas mensagens de falha na autenticação. *Eg*., em vez de exibir mensagens como “Nome de usuário incorreto” ou “Senha incorreta”, utilizar apenas mensagens como “Usuário e/ou senha inválidos” para ambos os casos de erro.
M38	Deve-se utilizar autenticação para conexão a sistemas externos que envolvam tráfego de informação sensível ou acesso a funções.
M39	Deve-se cifrar e armazenar em um local protegido de um sistema confiável as credenciais de autenticação para acessar serviços externos à aplicação.
M40	Não se deve armazenar as credenciais de autenticação no código-fonte da aplicação.
M41	Não se deve armazenar as credenciais de autenticação ou qualquer outro dado sensível em imagem de container.
M42	Deve-se notificar o usuário quando a sua senha for alterada.
M43	Deve-se comunicar a data/hora da última utilização — bem ou mal sucedida — de uma conta de usuário no próximo acesso ao sistema.
M44	Deve-se modificar todas as senhas e os identificadores de usuários (IDs) que, por padrão, são definidas pelos fornecedores.
M45	Deve-se exigir nova autenticação dos usuários antes da realização de operações críticas.
M46	Deve-se garantir que código de terceiros utilizado para o processo de autenticação não contém código malicioso.
M47	Não se deve validar os dados de autenticação antes do final de todas as entradas de dados, especialmente nas implementações de autenticação sequencial.

M48	Deve-se utilizar apenas requisições POST para transmitir credenciais de autenticação.
M49	Deve-se exigir a mudança de senhas temporárias na próxima vez que o usuário realizar a autenticação no sistema.
M50	Deve-se desativar a funcionalidade de lembrar a senha nos campos de senha do navegador.
M51	Deve-se armazenar de forma segura os dados de usuários e de sistemas que utilizam cada senha fornecida.
M52	Não se deve utilizar as mesmas senhas para ambientes de desenvolvimento, homologação ou produção.
C5	Deve-se utilizar certificado digital para determinar a identidade do usuário.
C6	Deve-se realizar monitoramento para identificar ataques contra várias contas de usuários que utilizem a mesma senha.
C7	Deve-se utilizar autenticação de múltiplos fatores (utilizando simultaneamente *token*, senha, biometria etc.).
C8	Deve-se garantir que, uma vez autenticado, o usuário não possa acessar o sistema de outro endereço IP, a menos que se autentique novamente.

GERENCIAMENTO DE SENHAS

Diretrizes para geração, distribuição e uso de senhas em sistemas computacionais. Os fatores examinados para o uso de senhas em software desenvolvido de maneira segura são:

Geração e parametrização de senhas. Explana critérios para a escolha de senhas cuja finalidade é dificultar sua quebra por ataques de força-bruta ou adivinhação. O

principal parâmetro considerado é o comprimento (tamanho) da senha. Inclui procedimentos para testes de força de senhas.

Armazenamento e distribuição de senhas. Explana métodos para armazenamento seguro de senhas geradas tanto no lado validado (usuário, programa cliente, etc.) quanto no lado validador (software, sistema autenticador, etc.). Inclui métodos para a transmissão segura de senhas via rede e parâmetros para os procedimentos de mudança de senhas.

Interface. Explana medidas necessárias à interface para as tentativas de validação de senhas. Inclui parametrização para determinar a frequência de tentativas permitidas para validação de uma senha e a apresentação da senha parcialmente submetida para o usuário.

ID	DIRETRIZ
M53	Não se deve utilizar senhas com menos de 6 caracteres.
M54	Deve-se utilizar pelo menos letras maiúsculas e minúsculas, junto a ao menos um tipo de caractere (dígito, símbolo)
M55	Não se deve usar palavras comumente utilizadas para senhas (ou variantes destas). Eg., nome do animal de estimação, membro da família ou pessoa significativa; datas de aniversário; nome do feriado favorito; algo relacionado ao time esportivo favorito e as palavras “senha” e “password”.
M56	Não se deve armazenar senhas em claro.
M57	Deve-se armazenar ao menos o hash criptográfico com salt.
M58	Não se deve usar um canal em claro para a transmissão da senha ou elemento correspondente.

M59	Não se deve utilizar método de conferência menos seguro que desafios baseados em hash ou o uso de hashes armazenados.
M60	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.
M61	Não se deve elaborar senhas sem auxílio de software gerador de senhas aleatórias, configurado para atender aos parâmetros aqui estabelecidos.
M62	Não se deve utilizar senha que não tenha sido validada por um software testador de força de senhas.
M63	Não se deve enviar a senha antiga para o usuário, em claro ou não.
M64	Não se deve armazenar senha que não esteja criptografada seguindo o nível mínimo de criptografia estabelecido neste documento.
M65	Não se deve permitir uma taxa de tentativas de validação de senha superior a 5 tentativas por minuto.
M66	Deve-se bloquear a conta de usuário em caso de 5 erros de autenticação consecutivos e sua reabilitação deve depender de processo específico.
C9	Não se deve utilizar senhas com menos de 20 caracteres.
C10	Não se deve utilizar senha que não tenha sido validada por um software testador de força de senhas diferente do software gerador de senhas.
C11	Não se deve armazenar senha que não esteja criptografada seguindo o nível forte de criptografia estabelecido neste documento.
C12	Deve-se utilizar um método de prova com conhecimento zero3 de senha,
C13	Deve-se exigir prova de origem da requisição. Eg., captchas para demonstrar que o usuário é humano; assinatura digital para provar que requisição veio do sistema permitido.

3 Blum, Manuel; Feldman, Paul; Micali, Silvio. Non-Interactive Zero-Knowledge and Its Applications.

Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988): 103–112.

GERENCIAMENTO DE SESSÕES

Diretrizes 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.

Nesse escopo específico, as diretrizes estão subdivididas de acordo com o momento da sessão (início, manutenção, término), além de tópico específico sobre controle da sessão.

CONTROLE DE SESSÃO

ID	DIRETRIZ
M67	Deve-se utilizar controles de gerenciamento de sessão baseados no servidor ou em frameworks.
M68	Não se deve definir o domínio e o caminho para os cookies que contenham identificadores de sessão autenticados para um endereço externo ao site.
M69	Deve-se configurar o atributo "secure" para cookies transmitidos através de uma conexão TLS.
M70	Deve-se 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.
C14	Deve-se utilizar mecanismos complementares ao mecanismo padrão de gerenciamento de sessões para operações sensíveis do lado servidor. Eg.,

ID	DIRETRIZ
M71	Não se deve permitir o estabelecimento de sessão caso a aplicação não consiga ter acesso às informações contidas na configuração de segurança.
M72	Não se deve reconhecer identificadores gerados por controles fora do servidor ou do framework de controle como válidos.
M73	Não se deve permitir logins persistentes (sem prazo de expiração).

MANUTENÇÃO DE SESSÃO

ID	DIRETRIZ
M74	Não se deve reaproveitar uma sessão estabelecida antes do login em caso de nova autenticação.
M75	Não se deve reaproveitar um identificador de sessão quando houver uma nova autenticação.
M76	Não se deve expor os identificadores de sessão em URLs, mensagens de erro ou logs.
M77	Deve-se proteger os dados de sessão do lado servidor contra acessos não autorizados por outros usuários do mesmo servidor, inclusive durante a sessão vigente.

M78	Deve-se notificar o usuário clara e constantemente a respeito do tempo de encerramento de sessão.
C15	Não se deve permitir conexões simultâneas com o mesmo identificador de sessão.

TÉRMINO DA SESSÃO

ID	DIRETRIZ
M79	Deve-se encerrar completamente a sessão ou conexão associada no logout.
M80	Deve-se disponibilizar a funcionalidade de logout em todas as páginas que requerem autenticação.
M81	Deve-se estabelecer um tempo de expiração da sessão que seja o mais curto possível, baseado no balanceamento dos riscos e requisitos funcionais do negócio.
C16	Deve-se realizar o encerramento da sessão periodicamente, mesmo quando ela estiver ativa.

CONTROLE DE ACESSOS

Diretrizes e definições para a realização do controle de acessos a recursos do sistema, artefatos de desenvolvimento e partes sensíveis da aplicação.

ID	DIRETRIZ

M82	Deve-se restringir o acesso às URLs protegidas somente aos usuários autorizados.
M83	Deve-se restringir o acesso às funções protegidas, às referências diretas aos objetos, aos serviços e aos dados da aplicação somente aos usuários autorizados.
M84	Deve-se 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.
M85	Deve-se restringir o acesso a arquivos somente aos usuários autorizados.
M86	Não se deve utilizar o campo “referer” do cabeçalho como forma de verificação principal.
M87	Deve-se fazer a revalidação periódica da autorização do usuário para garantir que os privilégios não foram modificados.
M88	Não se deve atribuir privilégios além do mínimo às contas de serviço ou contas de suporte a conexões provenientes ou destinadas a serviços externos.
M89	Deve-se 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.
M90	Deve-se exigir nova autenticação caso os privilégios do usuário tenham sido modificados durante uma sessão.
M91	Deve-se prover suporte à desativação de contas e ao encerramento das sessões quando terminar a autorização do usuário.

C17	Não se deve aplicar regras de controle de acesso representadas pela camada de apresentação divergentes das regras presentes no lado servidor.
C18	Deve-se implementar a auditoria das contas de usuário e assegurar a desativação de contas não utilizadas.
C19	Deve-se utilizar mecanismos de criptografia e verificação de integridade no lado servidor para detectar possíveis adulterações em dados armazenados no lado do cliente.
C20	Deve-se limitar o número de transações que um único usuário ou dispositivo pode executar em determinado período de tempo.

PRÁTICAS DE CRIPTOGRAFIA

Diretrizes para a configuração e utilização de algoritmos de criptografia e *hash* visando prover confidencialidade a dados.

ID	DIRETRIZ
M94	Deve-se criptografar dados sigilosos e sensíveis.
M95	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 e somente a chave criptográfica deve ser mantida em sigilo.
M96	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.

M97	Não se deve utilizar o modo de cifrador de bloco Electronic Codebook (ECB) ou modos menos seguros.
M98	Não se deve utilizar um tamanho da chave menor que 128 bits (cifrador simétrico) ou 1024 bits (cifrador assimétrico).
M99	Não se deve utilizar função de hash sem algum tipo de salt.
M100	Não se deve utilizar módulos de criptografia incompatíveis com a FIPS 140-2 ou com um padrão equivalente.
M101	Não se deve utilizar algoritmos considerados obsoletos para criptografia e hash criptográfico. Eg., MD5, SHA1, DES/3DES, RC2, RC4, MD4.
M102	Não se deve distribuir chaves criptográficas sem a utilização de uma infraestrutura de chave pública e, portanto, sem a utilização de um cifrador assimétrico.
C21	Não se deve utilizar um tamanho da chave menor que 256 bits (cifrador simétrico) ou 4096 bits (cifrador assimétrico).
C22	Deve-se utilizar módulos criptográficos com geradores de números pseudo-aleatórios de alta aleatoriedade para a geração de todos os números, nomes de arquivos, GUIDs e strings aleatórias.
C23	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.

PROTEÇÃO DE DADOS

Diretrizes que tratam do armazenamento de informações com grau de sigilo e de sua disponibilização. A seção define taxonomia para classificação de dados e

descreve procedimentos para o armazenamento seguro dessa informação em *bancos de dados*. Também é detalhado o gerenciamento de permissões de acesso e distribuição de senhas a serem adotadas para operacionalização dessas estruturas.

No escopo deste documento os dados serão classificados, quanto ao seu *sigilo*, como:

Abertos. Dados públicos

Fechados. Dados cujo acesso é restrito a um grupo específico de pessoas

PROCEDIMENTOS E MEIOS PARA ARMAZENAMENTO DE DADOS ABERTOS

ID	DIRETRIZ
M103	Deve-se utilizar meio de armazenamento que possua acesso para escrita restrito por senha.

PROCEDIMENTOS E MEIOS PARA ARMAZENAMENTOS DE DADOS FECHADOS

ID	DIRETRIZ
M104	Deve-se utilizar meio de armazenamento que possua acesso para escrita restrito por senha.

PERMISSÕES PARA ACESSO A INFORMAÇÃO EM BANCOS DE DADOS

ID	DIRETRIZ
M105	Não se deve disponibilizar às aplicações acesso a algum banco de dados utilizando login de usuário com permissões de usuário *root* ou equivalente.
M106	Não se deve disponibilizar às aplicações acesso a algum banco de dados utilizando login de usuário com permissões para execução de comandos em *Data*
M107	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.
C25	Deve-se estabelecer correspondência um-para-um entre cada usuário de uma dada aplicação e do banco de dados

TRATAMENTO DE DADOS E APLICAÇÕES

ID	DIRETRIZ
M108	eve-se prover à aplicação a faculdade de remover dados sensíveis quando estes não forem mais necessários

M109	Deve-se desativar a cache realizada no lado cliente das páginas que contenham informações sensíveis.
M110	Não se deve incluir informações sensíveis nos parâmetros de requisição HTTP GET.
M111	Não se deve publicar documentação do sistema que possa revelar informações importantes para potenciais atacantes.
M112	Deve-se implementar uma política de privilégio mínimo, restringindo os usuários apenas às funcionalidades, dados e informações do sistema que são necessárias para executarem suas tarefas.
M113	Deve-se proteger contra acesso não autorizado todas as cópias temporárias ou registradas em cache que contenham dados sensíveis e estejam armazenadas no servidor
M114	Deve-se criptografar informações altamente sensíveis quando armazenadas.
M115	Deve-se proteger o código-fonte presente no servidor para que não seja acessado por algum usuário sem permissão
M116	Não se deve armazenar senhas, strings de conexão ou outras informações confidenciais em texto claro/legível ou em qualquer forma criptograficamente insegura no lado cliente.
M117	Deve-se remover comentários do código de produção que podem ser acessados pelos usuários.
M118	Deve-se excluir todas as cópias temporárias ou registradas em cache que contenham dados sensíveis e estejam armazenadas no servidor logo que não forem mais necessários.
M119	Deve-se desativar a funcionalidade de auto-completar nos formulários que

	contenham informações sensíveis, inclusive no formulário de autenticação.

SEGURANÇA NAS COMUNICAÇÕES

Diretrizes que tratam da transmissão segura de dados sensíveis entre sistemas, de modo a salvaguardar a integridade, autenticidade e demais atributos pertinentes ao uso dos dados comunicados.

ID	DIRETRIZ
M120	Deve-se utilizar criptografia na transmissão de todas as informações sensíveis.
M121	Deve-se empregar canal de comunicação dos dados transmitidos. *Eg*, HTTPS.
M122	Deve-se empregar canal de comunicação dados transmitidos. *Eg*, HTTPS, VPNs.
M123	Não se deve incluir informações sensíveis nos parâmetros de requisição HTTP GET.
M124	Não se deve publicar documentação do sistema que possa revelar informações importantes para potenciais atacantes.
M125	Deve-se utilizar TLS para conexões com sistemas externos que envolvam funções ou informações sensíveis

M126	Deve-se empregar um canal de comunicação com controle de autenticação. *Eg*, HTTPS, certificados digitais gerados por autoridades confiáveis, VPNs.
M127	Deve-se armazenar de maneira segura os dados a serem transmitidos em ambas as extremidades da comunicação.
M128	Deve-se especificar a codificação dos caracteres para todas as conexões.
M129	Deve-se filtrar os parâmetros que contenham informações sensíveis, provenientes do “HTTP referer”, nos links para sites externos.
C26	Deve-se empregar canal de comunicação que provenha garantia de não-repúdio dos dados transmitidos. *Eg*, certificados digitais emitidos por entidades confiáveis.
C27	Deve-se utilizar logs confiáveis das informações transmitidas, com confirmação de entrega e recepção das mensagens. *Eg*., *WS-ReliableMessaging* para SOAP WS.
C28	Deve-se utilizar um padrão único de implementação TLS, configurado de modo apropriado.

CONFIGURAÇÕES DE SISTEMA

Diretrizes para a instalação, configuração e gerenciamento de ambientes de desenvolvimento de sistemas.

CONFIGURAÇÕES DE PLATAFORMA E TRATAMENTO DE REQUISIÇÕES

ID	DIRETRIZ
M130	Deve-se restringir para o mínimo possível os privilégios do servidor *Web*, dos processos e das contas de serviços
M131	Deve-se remover código de teste ou qualquer funcionalidade desnecessária para o ambiente de produção antes da instalação do sistema no servidor de produção.
M132	eve-se definir quais métodos de requisição (*eg*., HTTP, GET ou POST) a aplicação irá suportar e se serão tratados de modo diferenciado nas diversas páginas da aplicação.
M133	o se deve 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.
M134	Deve-se isolar o ambiente de desenvolvimento da rede de produção e conceder acesso somente para grupos de desenvolvimento e testes.
M135	Deve-se garantir que os servidores, *frameworks* e componentes do sistema estão executando a última versão aprovada, com as atualizações de segurança mais recentes e que sejam compatíveis com as necessidades do sistema.
M136	Deve-se desativar a listagem de diretórios do servidor *Web*

M137	Deve-se 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.
M138	Deve-se desativar as extensões HTTP desnecessárias. *Eg*.,WebDAV.
C29	Deve-se remover todas as funcionalidades e arquivos desnecessários.
C30	ve-se certificar de que, no caso do servidor processar tanto requisições HTTP 1.0 como HTTP 1.1, ambas as versões estarão configuradas de modo semelhante.
C31	Deve-se implementar um sistema de gestão de ativos para manter o registro dos componentes e programas.

ACESSO AO CÓDIGO-FONTE

Quanto ao sigilo do código-fonte dos sistemas desenvolvidos, devem ser, por padrão, de livre acesso aos desenvolvedores alocados no projeto. As demais situações deverão ser analisadas, projeto a projeto, pela chefia

ID	DIRETRIZ
M139	Deve-se utilizar um sistema controle de versões para provimento de rastreabilidade de modificações e controle de acesso ao código-fonte.

SEPARAÇÃO DE AMBIENTES

ID	DIRETRIZ
M140	Deve-se utilizar um sistema controle de versões para provimento de rastreabilidade de modificações e controle de acesso ao código-fonte.
M141	Deve-se utilizar servidores de aplicação/*Web* distintos para cada ambiente.
M142	Deve-se prover acesso ao ambiente de desenvolvimento /testes /homologação apenas aos integrantes da equipe de desenvolvimento e aos interessados no projeto (*stakeholders*).
M143	Deve-se prover um manual para a instalação do ambiente necessário para a execução de uma dada aplicação.
C32	Não se deve fornecer as senhas de acesso ao ambiente de produção aos desenvolvedores.
C33	Deve-se realizar testes periódicos para assegurar a segurança do ambiente de desenvolvimento/testes/homologação.

SEGURANÇA EM BANCO DE DADOS

Diretrizes para reforço de práticas seguras na interação entre aplicações e bancos de dados.

ID	DIRETRIZ
M144	Deve-se utilizar um sistema controle de versões para provimento de rastreabilidade de modificações e controle de acesso ao código-fonte.

M145	Deve-se usar consultas parametrizadas fortemente tipadas.
M146	Deve-se utilizar validação de entrada e codificação de saída; se houver falha, o comando não deverá ser executado no banco de dados.
M147	Deve-se realizar a codificação (*escaping*) de meta caracteres em instruções SQL.
M148	Não se deve incluir *strings* de conexão no código da aplicação.
M149	Deve-se eliminar o conteúdo desnecessário incluído por padrão pelo fornecedor. *Eg*., esquemas e bancos de dados de exemplo.
M150	Deve-se desativar todas as contas criadas por padrão e que não sejam necessárias para suportar os requisitos de negócio.
M151	Deve-se atribuir à aplicação o menor nível possível de privilégios ao acessar o banco de dados.
M152	Deve-se, sempre que possível, usar procedimentos armazenados (*stored procedures*) para abstrair o acesso aos dados e permitir a remoção de permissões das tabelas no banco de dados.
M153	Não se deve criar SQLs concatenando parâmetros textuais de origem não-segura, como parâmetros preenchidos pelo usuário ou mesmo armazenados no banco de dados.
M154	Deve-se utilizar tratamento especial para consultas que não podem se parametrizadas, como escapes em hexadecimal.
C34	Deve-se encerrar a conexão com o banco de dados assim que possível.

GERENCIAMENTOS DE ARQUIVOS

Diretrizes para edição, distribuição, armazenamento e concessão de permissões de arquivos.

ID	DIRETRIZ
M155	Deve-se utilizar um sistema controle de versões para provimento de rastreabilidade de modificações e controle de acesso ao código-fonte.
M156	Não se deve salvar arquivos no mesmo diretório de contexto da aplicação *Web*.
M157	Deve-se prevenir ou restringir o carregamento de qualquer arquivo que possa ser interpretado e/ou executado pelo servidor *Web*.
M158	Deve-se desativar privilégios de execução nos diretórios de armazenamento de arquivos.
M159	Não se deve passar caminhos de diretórios ou de arquivos em requisições.
M160	Não se deve enviar o caminho absoluto do arquivo para o lado cliente de uma aplicação ou para o usuário.
M161	Deve-se certificar de que os arquivos da aplicação e os recursos estão definidos somente com o atributo de leitura.
M162	Deve-se requerer autenticação antes de se permitir que seja feito o carregamento de arquivos.
M163	Deve-se validar se os arquivos enviados são do tipo esperado através da validação dos cabeçalhos.
M164	Deve-se usar uma lista branca (*whitelist*) de nomes e de tipos de arquivos permitidos ao referenciar arquivos

C35	Deve-se limitar os tipos de arquivos que podem ser enviados para aceitar somente os necessários ao propósito do negócio.
C36	Deve-se implantar o carregamento seguro de arquivos nos ambientes UNIX por meio da montagem do diretório de destino como uma unidade lógica.
C37	Deve-se verificar os arquivos que os usuários submeterem através do mecanismo de carregamento em busca de vírus e *malwares*.

DIRETRIZES PARA DESENVOLVIMENTO SEGURO DE SOFTWARE

PREVENÇÃO, REAÇÃO E MITIGAÇÃO DE FALHAS DE SEGURANÇA

Diretrizes para a realização de procedimentos que garantam uma reação adequada à ocorrência de falhas de segurança. Detalha-se o emprego de *backups*, testes e tratamento de ocorrências.

Diretrizes para a realização de procedimentos que garantam uma reação adequada à ocorrência de falhas de segurança. Detalha-se o emprego de *backups*, testes e tratamento de ocorrências.

BACKUPS

A adequação às diretrizes de *backup* depende, muitas vezes, de políticas e atuação da área de infraestrutura, mas são importantes aspectos a serem considerados e monitorados no desenvolvimento de aplicações seguras.

permissões de arquivos.

ID	DIRETRIZ
M165	Deve-se incluir no plano de projeto a especificação da necessidade e a atribuição da responsabilidade de realização de backups do banco

	de dados e dos códigos-fonte do sistema, bem como as políticas de acesso a este *backup*.
M166	Deve-se definir um procedimento estruturado para a restauração de *backups*.
M167	Deve-se definir e capacitar responsáveis pela recuperação dos *backups*.
C38	Deve-se criar *baselines* das versões do sistema, facilitando a recuperação ágil para uma versão anterior.
C39	Deve-se realizar simulações de restauração de dados continuamente.

TESTES

ID	DIRETRIZ
M168	Deve-se realizar testes manuais de segurança antes de cada versão do *software* em que sua estrutura tenha sido modificada. *Eg*., telas de *login*, serviços não autenticados, novos formulários com interação com o usuário, *etc*.
M169	eve-se garantir, através de testes automatizados, que os serviços e dados sigilosos estão protegidos e disponíveis apenas para os usuários detentores das informações.
M170	Deve-se elaborar uma política de testes, automatizados ou não, visando a garantia de não vulnerabilidade aos principais ataques conhecidos em sistemas.

M171	Deve-se 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 evitar vícios.
M172	Deve-se definir cenários de testes, principalmente nos aspectos de segurança, para os casos de atualizações na arquitetura do sistema. *Eg*., servidores de aplicação, banco de dados, versões de navegador *Web*, versões de sistema operacional, etc.
C40	Deve-se propor constantes desafios entre as equipes para testar a segurança dos sistemas em formato de competição
C41	Deve-se submeter os sistemas a ferramentas de testes de invasão.
C42	Deve-se submeter imagens de container, de sistemas que utilizam essa tecnologia, à análise de vulnerabilidades.
C43	Deve-se submeter o código do sistema a ferramentas de análise estática de segurança (SAST).

OCORRÊNCIAS

ID	DIRETRIZ
M173	eve-se manter procedimento planejado para imediata indisponibilização do sistema e realização de manutenção corretiva.
M174	Deve-se definir uma política de acompanhamento pós-correção de ocorrências de falha de segurança.

C44	Deve-se utilizar lições aprendidas nas ocorrências passadas para revisar a política de testes e incrementar a segurança dos sistemas.

PROJETO

ID	DIRETRIZ
M175	Deve-se empregar modelo de projeto de *software* que contemple etapa de modelagem de ameaças; definição clara dos riscos de segurança e nível de severidade que o comprometimento de dados sensíveis traria ao sistema e à instituição.
M176	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.
M177	Deve-se utilizar cronograma de projeto que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

CODIFICAÇÃO

ID	DIRETRIZ
M178	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.

M177	Deve-se utilizar cronograma de projeto que contemple pontos de verificação de segurança do sistema desenvolvido ao longo de sua construção.

MANUTENÇÃO

ID	DIRETRIZ
M180	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.
M181	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.

PESSOAL

ID	DIRETRIZ
M182	Deve-se proporcionar treinamento e capacitação de programadores para aquisição e revisão de princípios de segurança computacional e desenvolvimento de *software* seguro.

GLOSSÁRIO

AD. Microsoft *Active Directory* - base de dados de identidades, autenticação e autorização utilizada internamente no INSTITUTO EVERESTE.

Análise de vulnerabilidade. Atividade que tem por objetivo buscar por fragilidades existentes na aplicação e nas bibliotecas, componentes e infraestrutura por ela utilizados.

Autenticação. Ato de comprovação da identificação por meio de um ou mais fatores (senha, biometria, certificado digital, token, *One-Time Password*, etc).

Autorização. Uma vez autenticado com sucesso, o usuário passa a ter acesso a recursos e informações previamente concedidos para ele e/ou seu perfil.

Ataque de força-bruta. Tipo de ataque que busca alcançar seu objetivo (quebra de senha, tentativa de acesso indevido, etc) por meio de um número expressivo de tentativas e combinações possíveis.

Cifrador. É um par de algoritmos que realizam a encriptação e a decriptação.

Cifrador Assimétrico. É um cifrador que usa chaves diferentes, uma *pública*, uma *privada*, para encriptação e decriptação. Mais lentos, em geral, mas com usos para assinatura e verificação de autenticidade. Exemplos: Rivest-Shamir-Adleman (RSA)[1] e *Elliptic Curve Cryptography* (ECC)[2].

Cifrador de bloco. Cifrador que opera sobre blocos de bits de tamanho fixo com uma transformação invariável que é especificada por uma chave simétrica.

1

2

Cifrador Simétrico. É um cifrador que usa a mesma chave para encriptação e decriptação. Mais rápidos, em geral. Exemplos: *Advanced Encryptation Standard* (AES)[3] e *Data*
Encryptation Standard (DES)[4].

Chave. É uma sequência de bits utilizada como parâmetro secreto no cifrador, necessária para realizar encriptação e/ou decriptação. A única maneira de descobrir uma chave deve ser por *força-bruta*; tentar todas as alternativas no espaço de chaves possíveis. O algoritmo utilizado pelo cifrador deve garantir que chaves longas implicam em um tempo impraticável para descobrir a chave por tentativa-e-erro.

Decriptação. É o processo de converter texto cifrado em seu texto em claro original.

Encriptação. É o processo de converter texto em claro em texto cifrado.

Identificação. Ato de informar uma credencial de um usuário.

***Hash* Criptográfico.** É uma função matemática que mapeia uma entrada de tamanho arbitrário, em *bits*, para um saída de tamanho fixo e que é utilizada para criptografia. A função também é de "mão única", no sentido de que é impossível invertê-la. A função deve ser determinística, de rápida computação e de alta entropia.

Não-repúdio. Diz respeito à impossibilidade de negar a autoria de determinada ação.

3

4

***OAuth2*.** É um protocolo de autorização que possibilita que aplicativos/aplicações obtenham acesso limitado a contas de usuários em um serviço HTTP sem a necessidade de enviar seu usuário e senha.

***Open Relay*.** Os servidores de correio eletrônico são classificados como Open Relay quando ele processa um e-mail onde o remetente e o destinatário não são usuários do servidor em questão.

Princípio de Kerckhoffs. Princípio que estabelece que a segurança deve ser estabelecida pela força da chave e não pelo segredo do método de criptografia.

REST. Protocolo para comunicação entre sistemas utilizando os métodos do protocolo HTTP.

***Salted Hash*.** Fragmento adicionado ao conteúdo original do hash para que a saída mude mesmo que o conteúdo original seja o mesmo.

SOAP. Protocolo para troca de mensagens em formato XML.

***SQL Injection*.** É uma forma de ataque em sistemas, realizado via interface, no qual o usuário informa trechos de SQL em campos de texto (ou até mesmo em telas de login ou de pesquisa), alterando a consulta prevista pelo desenvolvedor, sendo que o atacante poderá receber privilégios especiais ou poderá manipular indevidamente o banco de dados[5].

Static Application Security Testing (SAST) - Um conjunto de tecnologias desenvolvidas para analisar o código fonte, *byte code* e binários de aplicações buscando por indicativos de vulnerabilidades de segurança. Soluções SAST analisam a aplicação em um estado de não execução.

[5] Maisinformaçõessobre *SQL Injection* podem ser encontrados em [https://www.owasp.org/index.php/ SQL_Injection]

Teste de invasão. Atividade que tem por objetivo explorar falhas e vulnerabilidades existentes na aplicação e nas bibliotecas, componentes e infraestrutura por ela utilizados, com vistas a obter acesso indevido.

WS-ReliableMessaging. Protocolo para entrega segura de mensagens SOAP.

REFERÊNCIAS E INDICAÇÕES

Algumas referências listadas abaixo não foram explicitamente usadas no texto deste documento, porém contribuíram de alguma forma para a produção deste material e ficarão registradas para consulta em futuras revisões.

Blum, Manuel; Feldman, Paul; Micali, Silvio. **Non-Interactive Zero-Knowledge and Its Applications**. Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988): 103–112. Disponível em https://dl.acm.org/doi/10.1145/62212.62222. Acesso em: 13/01/2023.

GovBr**. Guia de Segurança em Aplicações Web**. Disponível em: https://www.gov.br/go vernodigital/pt-br/seguranca-e-protecao-de-dados/guias/guia_seguranca_aplicacoesweb.pdf . Acesso em: 10/01/2023.

Institute for Security and Open Methodologies. **The Open Source Security Testing Methodology Manual**. Disponível em: https://www.isecom.org/OSSTMM.3.pdf. Acesso em:10/01/2023.

Joan Daemen, Steve Borg e Vincent Rijmen, **The Design of Rijndael: AES - The Advanced Encryption Standard.** Springer-Verlag, 2002.

Hibernate Community Documentation. **Hibernate Envers Reference Documentation**.
Disponível em: https://docs.jboss.org/envers/docs/. Acesso em: 12/01/2023.

KOBLITZ, Neal. Elliptic curve cryptosystems. Mathematics Of Computation, [S.L.], v. 48, n. 177, p. 203-209, 1987. American Mathematical Society (AMS). http://dx.doi.org/10.1090/ s0025-5718-1987-0866109-5.

National Bureau of Standards, **Data Encryption Standard**, FIPS-Pub.46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.

National Cyber Security Centre. **Secure development and deployment guidance**. Disponível em: https://www.ncsc.gov.uk/collection/developers-collection. Acesso em: 10/01/2023.

NIST800-18. **Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities**. Disponível em: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf. Acesso em: 10/01/2023.

Oracle. **The Essentials of Filters**. Disponível em: https://www.oracle.com/java/technologies /filters.html. Acesso em 12/01/2023.

OWASP ASVS. **Application Security Verification Standard v4.0.3**. Disponível em: https://github.com/OWASP/ASVS/raw/v4.0.3/4.0/OWASP%20Application%20Security%20Ve rification%20Standard%204.0.3-en.pdf. Acesso em: 10/01/2023.

OWASP Cheat Sheet Series. **Sql Injection Prevention Cheat Sheet**. Disponível em: https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet. Acesso: 10/01/2023.

OWASP. **SQL Injection**. Disponível em: https://www.owasp.org/index.php/SQL_Injection. Acesso: 12/01/2023.

OWASP Proactive Controls. **OWASP Top 10 Proactive Controls 2018**. Disponível em: https://owasp-top-10-proactive-controls-2018.readthedocs.io/en/latest/index.html. Acesso em: 10/01/2023.

OWASP SCP. Melhores Práticas de Codificação Segura OWASP: Guia de Referência

Rápida. Disponível em: https://github.com/OWASP/secure-coding-practices-quick-reference -guide/releases/download/v2.0.1/OWASP_SCP_Quick_Reference_Guide.pt-BR.pdf. Acesso em: 10/01/2023.

OWASP WTSG. OWASP Web Security Testing Guide v4.2. Disponível em: https://owasp.org/www-project-web-security-testing-guide/v42/. Acesso em: 13/01/2023.

PostgreSQL Wiki. **Audit trigger 91plus**. Disponível em: https://wiki.postgresql.org/wiki/Audit_trigger_91plus. Acesso em: 12/01/2023.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L.. A method for obtaining digital signatures and public-key cryptosystems. **Communications Of The Acm**, [S.L.], v. 21, n. 2, p. 120-126, fev.
1978. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/359340.359342.

EVERESTE

INSTITUTO DE TECNOLOGIA E INOVAÇÃO

WWW.EVERESTE.ORG.BR