Práticas de Programação Segura

Aula 3

Tiago Lage Payne de Pádua

❑ Controle de Acessos

- Somente usuários autorizados devem ter acesso a URLs protegidas, serviços, dados de aplicativos, dados de usuários, atributos etc;
- A auditoria de contas deve ser implementada e as contas não utilizadas devem ser excluídas;
- As contas associadas a diferentes serviços para sistemas externos ou internos que são usados principalmente para tarefas não críticas devem ter privilégios mínimos;
- O número de transações deve ser limitado para um único usuário ou dispositivo durante um determinado período de tempo;

❑ Práticas de Criptografia

- Um sistema confiável deve ser usado para implementar funções criptográficas para manter a confidencialidade de dados confidenciais no aplicativo;
- A geração de números aleatórios, nomes de arquivos, GUIDs e strings deve usar um gerador de números aleatórios aprovado;
- O gerenciamento de chaves criptográficas deve ser empregado desenvolvendo e usando processos e políticas;
- As chaves mestras devem ser protegidas contra acesso não autorizado;

❑ Tratamento de Erros e Log

- Manipuladores de erro que não descartam informações de depuração em caso de entrada não solicitada devem ser usados;
- Quando ocorrem condições de erro, a memória deve ser liberada adequadamente;
- Os logs não devem armazenar informações confidenciais relacionadas a sistemas, sessões etc;
- Os logs relacionados a falhas de validação de entrada, tentativas de autenticação, controle de acesso, exceções do sistema, alterações inesperadas nos dados e alterações feitas nas configurações de segurança devem ser mantidos e verificados minuciosamente;

❑ Proteção de Dados

- Siga o princípio do privilégio mínimo, limitando os direitos e privilégios do usuário aos sistemas, informações e usabilidade necessários para atingir as metas e objetivos exigidos;

- Exclua dados confidenciais de requisições GET em HTTP;

- Proteja o código do lado do servidor e evite que ele seja acessível pelo usuário comum. O controle de acesso adequado deve ser implementado para dados críticos e confidenciais;

- O recurso de preenchimento automático deve ser excluído ao inserir dados em formulários em sites ou aplicativos;

❑ Segurança nas Comunicações

- No caso do TLS, as conexões com falha não devem fazer downgrade para protocolos não seguros;
- Para proteção de informações confidenciais sobre fontes externas, use TLS;
- Todas as conexões devem ser especificadas com codificação de caracteres;

❑ Configuração do Sistema

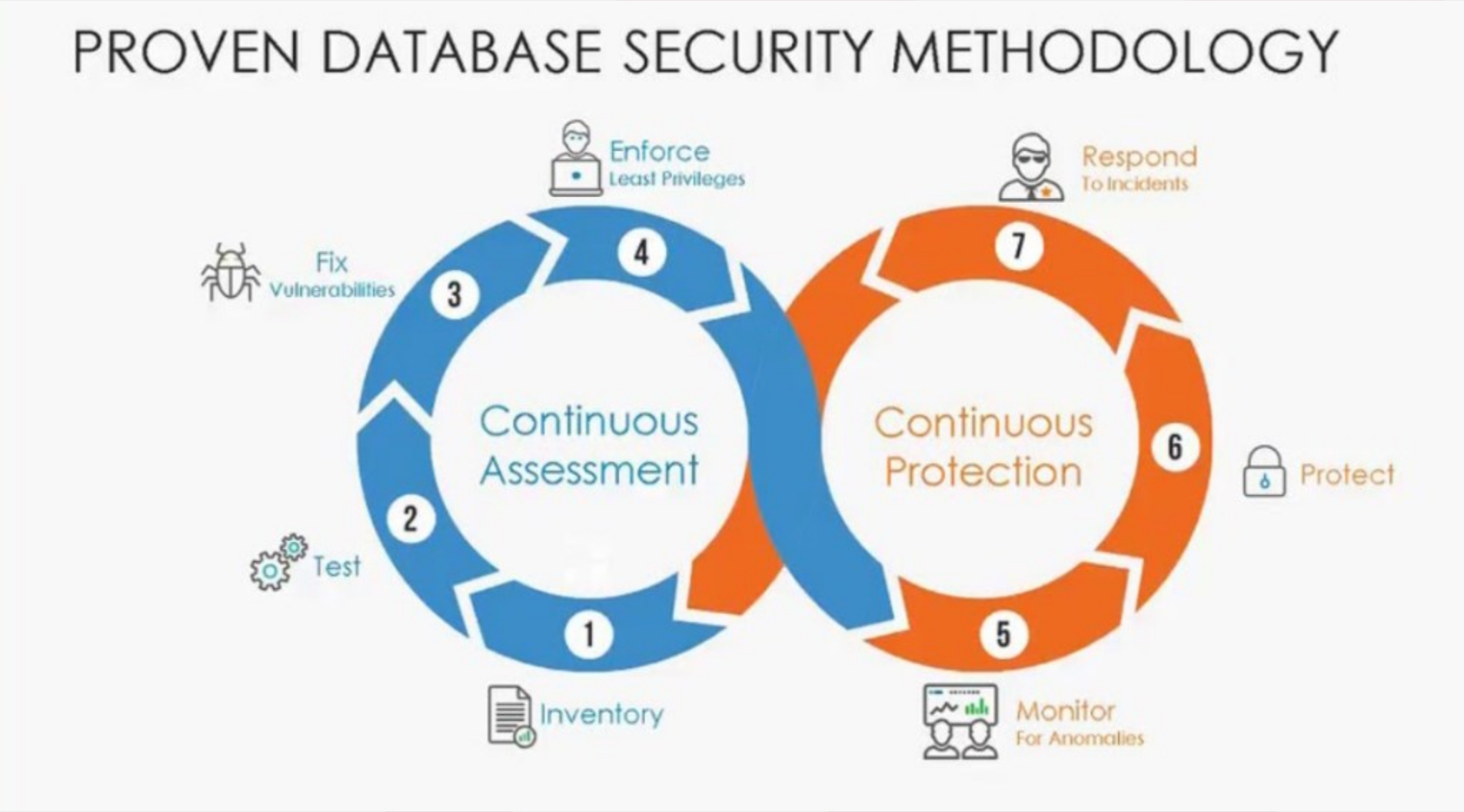
- Certifique-se de que os sistemas, estruturas e componentes do sistema estejam executando as versões e patches mais recentes;

- A aplicação de privilégio mínimo deve ser em contas de serviços, servidores web e processos;

- Os cabeçalhos de resposta HTTP devem incluir apenas informações relevantes. Informações sobre SO, versão do servidor web e estruturas de software não devem ser incluídas;

- Os ambientes de teste e desenvolvimento devem ser isolados do ambiente de produção;

❑ Segurança em Banco de Dados



❑ Segurança em Banco de Dados

- Ao acessar o banco de dados, o aplicativo deve usar o nível de privilégio mais baixo possível;
- As senhas padrão devem ser alteradas imediatamente. A higiene de senha deve ser mantida em mente ao atribuir senhas a contas;
- Habilite a autenticação multifator quando aplicável. Desative as contas padrão que não são necessárias para os requisitos de negócios;

❑ Gerenciamento de Arquivos

- Garanta a autenticação ao carregar um arquivo no servidor;
- Os arquivos carregados no servidor devem ser validados verificando os cabeçalhos dos arquivos;
- Os privilégios de execução devem ser desativados nos diretórios onde os arquivos são carregados;
- O caminho absoluto do arquivo nunca deve ser enviado ao cliente;

❑ Gerenciamento de Memória

- Funções vulneráveis como print, strcat, strcpy etc. devem ser evitadas;
- O tamanho do buffer deve ser verificado quanto a estouros;
- As strings de entrada devem ser truncadas corretamente antes que funções como cópia e concatenação sejam usadas;

❑ Práticas Gerais de Codificação

- Para tarefas comuns, utilizar sempre código testado, gerenciado e aprovado ao invés de criar código novo e não gerenciado;

- Utilizar APIs que executem tarefas específicas para realizar operações do sistema operacional. Não permitir que a aplicação execute comandos diretamente no sistema operacional, especialmente através da utilização de “shells” de comando iniciadas pela aplicação;

- Utilizar mecanismos de verificação de integridade por “checksum” ou “hash” para verificar a integridade do código interpretado, bibliotecas, arquivos executáveis e arquivos de configuração;

- Utilizar mecanismos de bloqueio para evitar requisições simultâneas para a aplicação ou utilizar um mecanismo de sincronização para evitar condições de concorrência (race conditions);

- Proteger as variáveis compartilhadas e os recursos contra acessos concorrentes inapropriados;

❑ Práticas Gerais de Codificação

- Instanciar explicitamente todas as variáveis e dados persistentes durante a declaração, ou antes da primeira utilização;

- Quando a aplicação tiver que ser executada com privilégios elevados, aumentar os privilégios o mais tarde possível e revogá-los logo que seja possível;

- 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. Prestar bastante atenção nas discrepâncias de tamanho de byte, precisão, distinções de sinal (signed/unsigned), truncamento, conversão e “casting” entre os tipos, cálculos que devolvam erros do tipo “not-a-number” e, também, como a linguagem de programação trata a representação interna de números muito grandes ou muito pequenos;

- Não transferir, diretamente, dados fornecidos pelo usuário para qualquer função de execução dinâmica sem realizar o tratamento dos dados de modo adequado;

❑ Práticas Gerais de Codificação

- Restringir a geração e a alteração de código por parte dos usuários;

- Revisar todas as aplicações secundárias, códigos e bibliotecas de terceiros para determinar a necessidade do negócio e validar as funcionalidades de segurança, uma vez que estas podem introduzir novas vulnerabilidades;

- Implementar atualizações de modo seguro. Se a aplicação precisar realizar atualizações automáticas, utilizar mecanismos de assinatura digital para garantir a integridade do código e garantir que os clientes façam a verificação da assinatura após descarregarem as atualizações. Usar canais criptografados para transferir o código a partir do host do servidor;

Ano: 2019 Banca: VUNESP Órgão: Prefeitura de Valinhos - SP Cargo: Analista de Tecnologia da Informação – SAI

Ao se deparar com uma tela de login em um sistema web, o usuário notou que o acesso não ocorria por um canal seguro. Pensando em garantir a segurança da aplicação, nessa situação, o usuário deve

a) acessar o site normalmente, pois não há risco de ter as credenciais de acesso violadas.

b) desistir do acesso, pois suas credenciais podem ser interceptadas por terceiros.

c) trocar a senha imediatamente, pois suas credenciais foram comprometidas.

d) utilizar recursos do sistema operacional, como copiar e colar, para não ter que digitar a senha.

e) utilizar um usuário com permissões de somente-leitura ao sistema.

Ano: 2019 Banca: VUNESP Órgão: Prefeitura de Valinhos - SP Cargo: Analista de Tecnologia da Informação – SAI

Ao se deparar com uma tela de login em um sistema web, o usuário notou que o acesso não ocorria por um canal seguro. Pensando em garantir a segurança da aplicação, nessa situação, o usuário deve

a) acessar o site normalmente, pois não há risco de ter as credenciais de acesso violadas.

b) desistir do acesso, pois suas credenciais podem ser interceptadas por terceiros.

c) trocar a senha imediatamente, pois suas credenciais foram comprometidas.

d) utilizar recursos do sistema operacional, como copiar e colar, para não ter que digitar a senha.

e) utilizar um usuário com permissões de somente-leitura ao sistema.

Ano: 2021 Banca: CESPE / CEBRASPE Órgão: BANESE Cargo: Técnico Bancário III - Área de Informática - Desenvolvimento

No que se refere a qualidade de software e segurança no desenvolvimento, julgue o item que se segue.

Um dos mecanismos de proteção para aplicações e servidores web é a autenticação, que permite o acesso apenas a usuários com as devidas credenciais.

( ) Certo

( ) Errado

Ano: 2021 Banca: CESPE / CEBRASPE Órgão: BANESE Cargo: Técnico Bancário III - Área de Informática - Desenvolvimento

No que se refere a qualidade de software e segurança no desenvolvimento, julgue o item que se segue.

Um dos mecanismos de proteção para aplicações e servidores web é a autenticação, que permite o acesso apenas a usuários com as devidas credenciais.

( ) Certo

(X) Errado