OWASP Proactive Controls

Pedro Gonçalves, A82313
Roberto Cachada, A81012

Introdução

- Quem é a OWASP?
  - Organização sem fins lucrativos com o objetivo de ajudar organizações a criar/manter software seguro.
- O que são os OWASP Proactive Controls?
  - Lista de categorias com técnicas de segurança que devem ser tidas em conta durante a construção de software seguro.
- Para que servem?
  - Ajudar desenvolvedores a construir software seguro.

C1. Definir Requisitos de Segurança

- O que são Requisitos de Segurança?
  - Funcionalidades que estando no software asseguram certas propriedades de segurança.
- Qual a importância de definir tais requisitos?
  - Cria uma base de funcionalidades seguras antes da escrita de código.
- Implementação de Requisitos:
  - Descoberta e Seleção.
  - Investigação e Documentação.
  - Implementação e Confirmação.

C2. Utilização de Frameworks e Bibliotecas de Segurança

- Vantagens da utilização destas frameworks/bibliotecas:
  - Poupa recursos às equipas de desenvolvimento.
  - Ferramentas mantêm-se atualizadas com o passar do tempo.
  - Manutenção simples.
- Quais as melhores práticas de implementação?
  - Utilizar bibliotecas de fontes confiáveis, que sejam mantidas ativamente.
  - Manter o código atualizado conforme as atualizações na biblioteca.
  - Encapsular as bibliotecas para reduzir os vetores de ataque.
  - Utilizar apenas o necessário das bibliotecas, para garantir a segurança do software.

C3. Acesso Seguro a Bases de Dados

- Queries:
  - Parametrização Queries.
    - Bobby-Tables.
    - OWASP Cheat Sheet on Query Parameterization.
- Comunicação:
  - A execução de queries, e os seus resultados devem ser sempre encriptados.
- Autenticação segura:
  - Todos os acessos à base de dados devem ser sempre autenticados.
- Configuração segura dos SGBD.

C4. Codificação E Formatação Dados

- Serve para proteger o software de ataques de injeção.
- Codificação:
  - Tradução de caracteres especiais em formas equivalentes.
    - Por exemplo, < para &lt, em HTML.
- Escaping :
  - Utilização de caracteres especiais antes de certos elementos, para permitir que sejam interpretados da forma correta.
    - Por exemplo, utilizar \ antes de “, para ser interpretadas como parte do texto ao invés de uma string.
- Estas técnicas devem ser utilizadas apenas antes da transmissão ao interpretados.

C5. Validação de todos os Inputs

- Validade Sintática:
  - Verifica se os inputs se encontram na forma esperada, através de duas abordagens:
    - Whitelisting:
      - Verifica se o input cumpre com um conjunto de regras “boas”.
    - Blacklisting:
      - Verifica a existência de conteúdo malicioso.
- Validade Semântica.
  - Verifica se os inputs fazem sentido no contexto da aplicação.
- Em aplicações servidor-cliente, as validações devem ser feitas do lado do servidor.

C5. Validação de todos os Inputs

- Expressões Regulares:
  - Úteis para validar input.
  - Devem ser utilizadas com caução, pois podem originar DoS caso não sejam implementadas corretamente.
  - Podem ser de difícil compreensão, o que pode dificultar o seu uso/manutenção.
- Dados Serializados.
  - Evitar o uso deste tipo de dados, devido a ser complicado validar os mesmos.

C6. Implementação de Identidade Digital

- Authentication Assurance Levels (ALLs)
  - Nível 1: Autenticação Single-Factor
    - Ex: Password
  - Nível 2: Autenticação Multi-Factor (MFA)
    - Algo que a pessoa saiba - password ou PIN
    - Algo que a pessoa tenha - token ou telemóvel
    - Algo que faça parte da pessoa - dados biométricos (ex.: impressão digital)
  - Nível 3: Criptographic Based Authentication

C6. Implementação de Identidade Digital

- Session Management
  - Session IDs devem ser longos, únicos e aleatórios
  - Gerada nova sessão ou trocar ID na autenticação e re-autenticação
  - A sessão deve expirar após algum tempo de inatividade
- Browser Cookies
  - Devem expirar quando a sessão expira (ou logo de seguida)
  - Flags “secure” e “HttpOnly”
  - Atributo “samesite”
- Tokens

C7. Implementação de Controlos de Acesso

- Design de Controlos de Acesso
  - Controlos de Acesso devem ser planeados antes do desenvolvimento
  - Todos os pedidos são sujeitos a Controlo de Acesso
  - Deny by Default
  - Dar o menor acesso possível
  - Don’t Hardcode Rules
  - Todos os eventos de Controlo de Acesso devem ser registados em logs

C8. Proteção de Todos os Dados

- Classificação de Dados
- Encriptação de Dados em Trânsito
- Encriptação de Dados Armazenados

C9. Implementação de Logging de Segurança e Monitorização

- Design Seguro de Logging
  - Validação e escaping antes do logging
  - Não guardar informação sensível nos logs
  - Ter um serviço de logging central (em sistemas distribuídos)

C10. Tratamento de Erros e Exceções

- Possíveis consequências
  - Fuga de informação
  - DoS
- Recomendações
  - Tratar exceções de forma central
  - Tratar todo o comportamento inesperado
  - Fazer logging com informação suficiente
  - Testar código

Conclusão

OWASP Proactive Controls

Pedro Gonçalves, A82313
Roberto Cachada, A81012