# Vulnerabilidades

# Ementa

- Identificação de Vulnerabilidades
  - Métodos de Identificação
- Tipos
  - Exposição de Informação
    - Comentários HTML e versionamento
    - Base de dados locais
    - Mensagens de Erro e de Exceção
    - Descoberta de Path
    - Indexação de Diretório e Leitura de Arquivos Locais (Local File Read)
    - Inclusão de Arquivos Remotos (Remote File Inclusion)
    - Path Traversal e Null Bytes
    - Injeções
- Configurações e Manutenção
  - Arquivos de configuração
  - Arquivos de Backup ou de versões antigas
  - Acesso Externo à Interface de Administração
  - Quebra de Autenticação / Roubo de Sessão
  - HTTP Response Splitting
- Validação de Dados
  - Cross-Site Scripting (XSS)
  - Comment Spam
  - Redirecionamento sem Validação
  - Cross-Site Request Forgery (CSRF)
  - Referência insegura à objetos

# Vulnerabilidade

- *Vulnerabilidade é uma fraqueza que possa ser explorada de modo a comprometer a segurança de um ativo.*

# Identificação de Vulnerabilidades

- A identificação de vulnerabilidades é a última etapa da fase de coleta de informações, e antecede a fase de ataques ao sistema/rede sob avaliação.
- Um ataque nada mais é do que a execução de uma seqüência de rotinas que visam explorar vulnerabilidades previamente identificadas.

# Métodos de Identificação de Vulnerabilidades

- Tentativa-erro
- Automatização (várias tentativas-erro por segundo)
- Análise de código-fonte

# Vamos as vulnerabilidades?

# 1 Exposição de Informação

# Comentários HTML e versionamento

- ...
- Impacto
  - As informações contidas em comentários HTML podem levar a descoberta de itens como:
  - 1. o fluxo de funcionamento da aplicação
  - 2. rotinas de verificação de conteúdo
  - 3. os métodos de controle de sessão
  - 4. observações sobre a codificação
  - 5. versões de frameworks de apoio utilizados

# Comentários HTML e versionamento

- Medidas
  - Inspecione o código da aplicação e dos templates buscando por comentários que possam ser impressos no código HTML. É interessante buscar também por possíveis comentários em funções de debug ou que venham a descrever erros.

# Base de dados locais

- ...
- Impacto
  - 1. Roubo de informações pessoais, que podem ser usadas para compras, por exemplo
  - 2. Acesso a informações de configuração da aplicação
  - 3. Acesso a estrutura interna da aplicação, entre outros acessos.

# Bases de dados locais

- Medidas
  - Realizar a proteção dos arquivos de base de dados através da configuração do servidor web é uma das possíveis medidas. O uso de bases locais é desencorajado, devido, além dos problemas de segurança, a problemas de performance.

# Mensagens de Erro e de Exceção

- Impacto (informações que podem ser obtidas)
  - erros da aplicação
  - erros em requisições da aplicação
  - erros de componentes da aplicação
  - informações sobre a arquitetura da aplicação

# Descoberta de Path

- permite a um indivíduo mal-intencionado saber em qual diretório se encontram os arquivos da aplicação Web

# Indexação de Diretório e Leitura de Arquivos Locais

- permite que diretórios e arquivos internos da aplicação fiquem acessíveis externamente.
- Impacto
  - arquivos internos da aplicação podem revelar informações internas do serviço ou do ambiente que podem ser utilizadas para a elaboração de ataques mais sofisticados. Além disso, informações internas sobre o funcionamento da aplicação web podem ser descobertas ou divulgadas fora do previsto.

# Indexação de Diretório e Leitura de Arquivos Locais

- Medidas
  - O acesso a recursos desnecessários do ambiente deve ser bloqueado. A listagem de diretórios deve ser desativada no servidor web e seções com informações internas da aplicação devem estar restritas.

# Inclusão de Arquivos Remotos

- é uma das vulnerabilidades mais procuradas, pois permite:
  - 1. Realizar execução de código no servidor web
  - 2. Executar código client-side como um JavaScript, que pode levar a outros ataques, como XSS
  - 3. Causar Negação de Serviço (DoS)
  - 4. Pode realizar manipulações nas informações fornecidas

# Inclusão de Arquivos Remotos

- Impacto
  - Além de expor completamente informações e conteúdo da aplicação, uma vulnerabilidade de Inclusão de Arquivos Remota pode expor também todos os serviços executados no servidor por permitir a inclusão de um arquivo que pode ser um shell via web.

# Inclusão de Arquivos Remotos

- Exemplo
  - ```
    <?php
        $file =$_GET['page']; //The page we wish to display
        include($file);
    ?>
    ```
    ```

# Inclusão de Arquivos Remotos

- Medidas
  - 1. Nunca use includes baseados em informações fornecidas pelo usuário. Quando for o caso, não use if/elseif/else use switch/case
  - 2. Independente dos dados, faça sempre trimming das strings de include, procurando por http, /, //
  - 3. Desabilite as opções registrer_global, allow_url_fopen e allow_url_include no arquivo php.ini (no caso do PHP)
  - 4. Valide fortemente os dados fornecidos pelos usuários

# Path Traversal e Null Bytes

- permite ao atacante acessar arquivos e diretórios que são armazenados fora da pasta raiz da web.
- Impacto
  - elevação de privilégio
  - podem ser obtidos
    - 1. código fonte do aplicativo
    - 2. configuração
    - 3. arquivos do sistema

# Path Traversal e Null Bytes

```
http://some_site.com.br/get-files.jsp?file=report.pdf
http://some_site.com.br/get-page.php?home=aaa.html
  http://some_site.com.br/../../../../etc/shadow
  http://some_site.com.br/get-files?file=/etc/passwd
```

# Path Traversal e Null Bytes

- Medidas
  - 1. Independente dos dados, faça sempre trimming das strings de include, procurando por http, /, //
  - 2. Desabilite as opções registrer_global, allow_url_fopen e allow_url_include no arquivo php.ini
  - 3. Valide fortemente os dados fornecidos pelos usuários
  - 4. Veja se existe a possibilidade de executar o servidor web em chroot

# Injeções

- Falhas de injeção permitem que atacantes insiram código malicioso através de uma aplicação web para outro sistema.
- Exemplos de sistemas: SQL (mais famoso), LDAP, XML, SMTP...

# Injeções

- Impacto
  - comprometimento do sistema ou sua destruição

2

# Configurações e Manutenção

# Arquivos de Configuração

- ...
- Informações que podem ser encontradas em arquivos de configuração, podem dizer respeito a:
  - 1. informações para acesso a serviços externos (banco de dados, serviços de log, etc)
  - 2. comentários sobre configurações
  - 3. versões de software, como última atualização ou referências externas

# Arquivos de Configuração

- Medidas
  - Verificar se os arquivos de configuração são acessíveis externamente
  - Se for possível, colocar os arquivos de configuração em um diretório protegido

# Arquivos de Backup ou de versões antigas

- arquivos sem referência e / ou esquecidos
- versões antigas renomeadas de arquivos modificados
- Impacto
  - Todos esses arquivos podem conceder o acesso a um indivíduo mal-intencionado as funcionalidades internas da aplicação, backdoors, interfaces administrativas, ou até mesmo as credenciais para se conectar à interface administrativa ou o servidor de banco de dados.

# Arquivos de Backup ou de versões antigas

- Medidas
  - evitar edição local de arquivos
  - verificar que tipo de atividades são feitas no sistema de arquivos onde esta a aplicação web
  - Arquivos de dados, arquivos de log, arquivos de configuração, etc, devem ser armazenadas em diretórios não acessíveis pelo servidor web
  - Instantâneos do sistema de arquivos não devem ser acessíveis via web

# Acesso Externo à Interface de Administração

- funções administrativas devem ser separadas das funcionalidades normais, pois é um controle chave de fraudes
- Impacto
  - ataques de força bruta
  - tentativas de exploração
  - bypass dos mecanismos de autenticação

# Acesso Externo à Interface de Administração

- Medidas
  - Recomenda-se que a interface de administração da aplicação web seja segregada e com acesso restrito somente aos administradores da aplicação.
  - Importante: Usuários da aplicação não são administradores logo o acesso deles a interface de administração não é necessário.

# Quebra de Autenticação / Roubo de Sessão

- As aplicações web também devem estabelecer sessões para acompanhar o fluxo de pedidos de cada usuário.
- Como o HTTP não fornece esse recurso, a aplicação deve se encarregar dele, e também deve fornecer meios de controlar o fluxo de requisições de seus usuários.

# Quebra de Autenticação / Roubo de Sessão

- Impacto
  - Um usuário (ou não) pode assumir a identidade de outro
    - Isso vai contra a lógica de negócio da organização

# Quebra de Autenticação / Roubo de Sessão

- Medidas
  - 1. o uso de senhas fortes pelos usuários
  - 2. a definição de uma política de senha, que possua a capacidade de bloquear o login do usuário depois de certo número de conexões falhas
  - 3. mecanismos de controle de mudanças, onde as funções relacionadas com o gerenciamento de contas sempre exijam a re-autenticação do usuário
  - 4. armazenamento seguro das senhas, onde as senhas serão armazenadas com o uso de um algoritmo de hash criptográfico forte
  - 5. proteção das credenciais em transito, onde as credenciais utilizadas são enviadas para o usuário de forma segura
  - 6. proteção do session ID ou do Identificador de Sessão de modo que, se alguma outra pessoa não autorizado tiver posse dele, essa pessoa não consiga asssumir a identidade de outra pessoa
  - 7. não permita a enumeração das contas disponíveis
  - 8. remova todos os códigos de demonstração disponíveis
  - 9. elabore trilhas de auditoria e logs completos do uso administrativo da aplicação

# HTTP Response Splitting

- O ataque consiste em fazer o servidor imprimir um carriage return (CR, ASCII 0x0D) e um avanço de linha (LF, ASCII 0x0A) seguido do conteúdo fornecido pelo atacante na seção de cabeçalho da sua resposta, normalmente, incluindo-os em campos de entrada enviados para a aplicação.
- Os cabeçalhos são separados por um CRLF e os cabeçalhos da resposta são separados a partir do seu corpo por dois. Portanto, falhando ao remover CRs e LFs, a aplicação permite que o atacante defina cabeçalhos arbitrários, assuma o controle do corpo, ou quebre a resposta em duas ou mais.

# HTTP Response Splitting

- Impacto
  - Um atacante em posse de uma vulnerabilidade de HTTP Response Splitting obtem praticamente o controle total do que será exibido para o cliente.

- Os ataques que podem ser efetuados são:
  - 1. Cross-site scripting (XSS)
  - 2. Defacement
  - 3. Envenenamento de cache
  - 4. Hijacking, entre outros semelhantes

# HTTP Response Splitting

- Medidas
  - codificar strings na URL antes da inclusão no cabeçalhos HTTP, como nos campos Location ou Set-Cookie
  - conversão para inteiros ou substituição usando expressões regulares

# 3 Validação de Dados

# Cross-Site Scripting (XSS)

- ocorre quando scripts maliciosos são injetados dentro do contexto de seções aparentemente confiáveis de um site
  - XSS Reflected ocorre quando o código injetado é passado e refletido pelo Servidor Web
  - Stored XSS ocorre quando o código injetado é armazenado persistentemente no lado do servidor, dentro de um banco, fórum de mensagens, campo de comentários, e etc
  - XSS baseado em DOM

# Cross-Site Scripting (XSS)

- Impacto
  - Um atacante pode usar um XSS e enviar uma URL maliciosa para um usuário final
  - O navegador recebe os parâmetros e executa automaticamente os scripts, que podem acessar cookies, sequestrar seções e reter informações sensíveis.
  - Os scripts podem, inclusive, manipular e modificar o conteúdo da página HTML, exibindo um conteúdo arbitrário. É possível criar frames maliciosos dentro do contexto original da página, manipular notícias, controlar o navegador do usuário, entre outros.

# Cross-Site Scripting (XSS)

- Medidas
  - 1. Todo parâmetro recebido pela aplicação deve ser validado quanto ao tipo e tamanho para verificar se a entrada é compatível com o dado esperado. ("Whitelist" ao invés de "Blacklist")
  - 2. Todo dado entrado deve sofrer codificação "HTML Encode" antes de ser copiado para respostas da aplicação.
  - 3. É importante que todo o código da aplicação seja revisado para proteger a aplicação contra falhas de XSS em outras URLs e páginas. A expressão regular de validação de e-mail deve ser revalidada em todos os campos em que é usada. Um controle de tamanho das entradas e formulários também é útil para mitigar e dificultar a elaboração de ataques.

# Comment Spam

- ...
- Impacto
  - 1.diminuir a qualidade do site
  - 2.reduzir a qualidade da experiência do usuário
  - 3.afetar a disponibilidade da aplicação web, pois em alguns casos podem ser executados ataques automatizados com a finalidade de deixar a aplicação mais lenta, ou de comprometer o mecanismo de armazenamento
  - 4.afetar o posicionamento do website em mecanismos de busca

# Comment Spam

- Medidas
  - Uso de CAPTCHA's
  - Existem módulos que verificam o conteúdo da mensagem de comentário
  - Assegurar que o site não está comprometido. Quando o site está vulnerável, scripts automáticos podem explorar falhas e acessar seções administrativas, inserindo mensagens e auto-aprovando os comentários, por exemplo.
  - Proibir a inserção de URL`s nos campos de comentários (incluindo nas informações de usuário).

# Redirecionamento sem Validação

- ocorre quando a aplicação web recebe um parâmetro e redireciona o usuário sem qualquer tipo de validação
- Impacto
  - um usuário pode abusar da aplicação para criar links maliciosos para campanhas de phishing ou páginas falsas

# Redirecionamento sem Validação

- Medidas
  - Não usar dados provenientes do cliente
  - Usar uma codificação interna da aplicação para criptografar a URL e evitar que o usuário consiga manipular os parâmetros.
  - Verificar se o usuário pode acessar o recurso que está tentando.

# Cross-Site Request Forgery (CSRF)

- O Cross-Site Request Forgery (CSRF) é uma classe de ataques que explora a relação de confiança entre uma aplicação Web e seu usuário legítimo.
- Etapas do CSRF
  - 1. O usuário se autentica ou está autenticado na aplicação alvo do ataque;
  - 2. O usuário recebe um link ou utiliza o mesmo navegador para acessar um aplicativo malicioso;
  - 3. O link ou aplicativo malicioso navegado inclui uma requisição à aplicação alvo, carregando todos os parâmetros necessários para a execução da transação;
  - 4. Pela SameOriginPolicy do navegador, como existe uma sessão autenticada válida para o usuário no aplicativo alvo, a aplicação recebe a requisição e executa a operação conforme a solicitação enviada.

# Cross-Site Request Forgery (CSRF)

- Impacto
  - Com uma aplicação web vulnerável a CSRF é possível através de outra aplicação web realizar requisições devido a existência de uma sessão sem controle na aplicações vulnerável a CSRF.

# Cross-Site Request Forgery (CSRF)

- Medidas
  - Synchronizer Token Pattern - token de "desafio" aleatório associado com a sessão atual do usuário.

# Dúvidas?