Web Invasion

AULA 05

Por Luiz Alvarenga

Sumário

Invasões Web

Introdução

Mais pessoas têm acesso à Internet do que nunca. Isso levou muitas organizações a desenvolver aplicativos baseados na Web que os usuários podem usar online para interagir com a organização.

Um aplicativo da web (também conhecido como site, portal, extranet) é um aplicativo baseado no modelo cliente-servidor.

O servidor fornece o acesso ao banco de dados e a lógica de negócios. Está hospedado em um servidor web. O aplicativo cliente é executado no navegador da web do cliente. Os aplicativos da Web geralmente são escritos em linguagens como Java, C # e VB.Net, PHP, ColdFusion Markup Language etc. Os mecanismos de banco de dados usados nos aplicativos da Web incluem MySQL, MS SQL Server, PostgreSQL, SQLite, etc.

Código mal escrito para aplicativos da Web pode ser explorado para obter acesso não autorizado a dados confidenciais e servidores da Web.

A maioria dos aplicativos da web é hospedada em servidores públicos acessíveis via Internet. Isso os torna vulneráveis a ataques devido à fácil acessibilidade. A seguir, são apresentadas ameaças comuns a aplicativos da web.

Tipos de Invasões WEB

- **Injeção de SQL** - o objetivo dessa ameaça pode ser ignorar os algoritmos de login, sabotar os dados etc.
- **Ataques de negação de serviço** - o objetivo dessa ameaça pode ser negar o acesso legítimo de usuários ao recurso
- **XSS de script entre sites** - o objetivo dessa ameaça pode ser injetar código que pode ser executado no navegador do cliente.
- **Envenenamento por cookie / sessão** - o objetivo desta ameaça é modificar os cookies / dados da sessão por um invasor para obter acesso não autorizado.
- **Violação de formulário** - o objetivo dessa ameaça é modificar dados de formulário, como preços em aplicativos de comércio eletrônico, para que o invasor possa obter itens a preços reduzidos.
- **Injeção de código** - o objetivo desta ameaça é injetar código como PHP, Python etc. que pode ser executado no servidor. O código pode instalar backdoors, revelar informações confidenciais etc.
- **Desfiguração** - o objetivo desta ameaça é modificar a página exibida em um site e redirecionar todas as solicitações de página para uma única página que contém a mensagem do invasor.

SQL Injection

Oque é?

É uma vulnerabilidade existente nos dias de hoje, que si usa de uma manipulação em códigos sql. Esta vulnerabilidade permite ao atacante executar consultas ao banco de dados inserindo querys (comandos Sql) na url do site ou ate mesmo em campos de text. obtendo, assim, informações confidenciais como logins e senhas, dentre outros.
Hoje em dia são usadas muitas técnicas para explorar um banco de dados de um site servidor… Citarei algumas das técnicas.
1 – Sql Injection

O que é uma SQL Injection?

SQL Injection é um ataque que envenena instruções SQL dinâmicas para comentar certas partes da instrução ou anexar uma condição que sempre será verdadeira. Ele tira proveito das falhas de design em aplicativos Web mal projetados para explorar instruções SQL e executar códigos SQL maliciosos.

Neste tutorial, você aprenderá técnicas de injeção SQL e como proteger aplicativos da Web contra esses ataques.

Como funciona a SQL Injection

Os tipos de ataques que podem ser executados usando injeção SQL variam de acordo com o tipo de mecanismo de banco de dados. **O ataque funciona em instruções SQL dinâmicas** . Uma declaração dinâmica é uma declaração gerada no tempo de execução usando a senha de parâmetros de um formulário da Web ou sequência de caracteres de consulta URI.

Vamos considerar um aplicativo da web simples com um formulário de login. O código para o formulário HTML é mostrado abaixo.

```
<formulário action = 'index.php' method = "post">

<input type = "email" name = "email" obrigatório = "obrigatório" />

<input type = "password" name = "password" />

<input type = "checkbox" name = "remember_me" value = "Lembrar de mim" />

<input type = "submit" value = "Submit" />

</form>
```

O formulário acima aceita o endereço de email e a senha os envia para um arquivo PHP chamado index.php.

Tem uma opção de armazenar a sessão de login em um cookie. Deduzimos isso da caixa de seleção remember_me. Ele usa o método de postagem para enviar dados. Isso significa que os valores não são exibidos no URL.

Vamos supor que a instrução no back-end para verificação do ID do usuário seja a seguinte

> ***SELECT * FROM users WHERE email = $ _POST ['email'] AND password = md5 ($ _ POST ['password']);***

A instrução acima usa os valores da matriz $ _POST [] diretamente sem limpá-los.

A senha é criptografada usando o algoritmo MD5.

Ilustraremos o ataque de SQL Injection usando o sqlfiddle. Abra o URL http://sqlfiddle.com/ no seu navegador da web. Você receberá a seguinte janela.

Nota: você terá que escrever as instruções SQL

Etapa 1) Digite este código no painel esquerdo

CREATE TABLE `usuários` (

`id` INT NÃO NULL AUTO_INCREMENT,

`email` VARCHAR (45) NULL,

`password` VARCHAR (45) NULL,

CHAVE PRIMÁRIA (`id `));

insira nos valores dos usuários (email, senha) (' m@m.com

', md5 ('abc'));

Etapa 2) Clique em Criar esquema

Etapa 3) Digite este código no painel direito

selecione * dos usuários;

Etapa 4) Clique em Executar SQL. Você verá o seguinte resultado

ID	EMAIL	PASSWORD
1	m@m.com	900150983cd24fb0d6963f7

Suponha que o usuário forneça **admin@admin.sys** e **1234** como a senha. A instrução a ser executada no banco de dados seria

SELECT * FROM users WHERE email = ' admin@admin.sys ' AND password = md5 ('1234');

O código acima pode ser explorado comentando a parte da senha e anexando uma condição que sempre será verdadeira. Suponhamos que um invasor forneça a seguinte entrada no campo de endereço de email.

xxx@xxx.xxx 'OR 1 = 1 LIMITE 1 -']

xxx para a senha.

A instrução dinâmica gerada será a seguinte.

SELECT * FROM users WHERE email = ' xxx@xxx.xxx ' OU 1 = 1 LIMITE 1 - '] AND senha = md5 (' 1234 ');

xxx@xxx.xxx termina com uma única citação que completa a citação da string

OR 1 = 1 LIMIT 1 é uma condição que sempre será verdadeira e limita os resultados retornados a apenas um registro.

- 'AND ... é um comentário SQL que elimina a parte da senha.

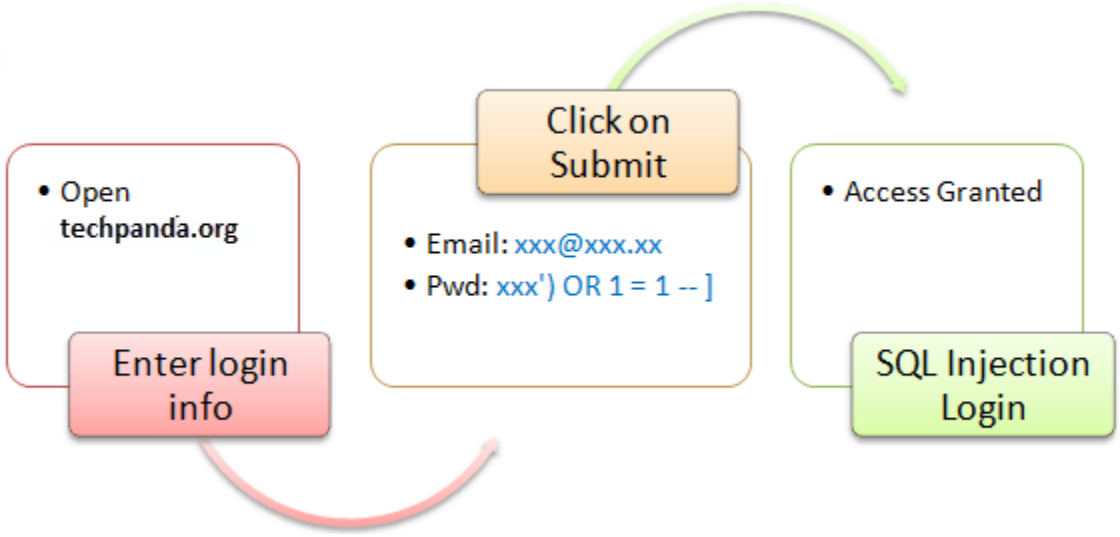
Copie a instrução SQL acima e cole-a na caixa SQL FiddleRun SQL Text, como mostrado abaixo

Atividade de hackers: SQL Injection em Sites e Aplicativo da Web

Temos um aplicativo da web simples em http://www.techpanda.org/ **que é vulnerável a ataques de SQL Injection apenas para fins de demonstração.**

O código do formulário HTML acima é retirado da página de login. O aplicativo fornece segurança básica, como higienizar o campo de email. Isso significa que o código acima não pode ser usado para ignorar o login.

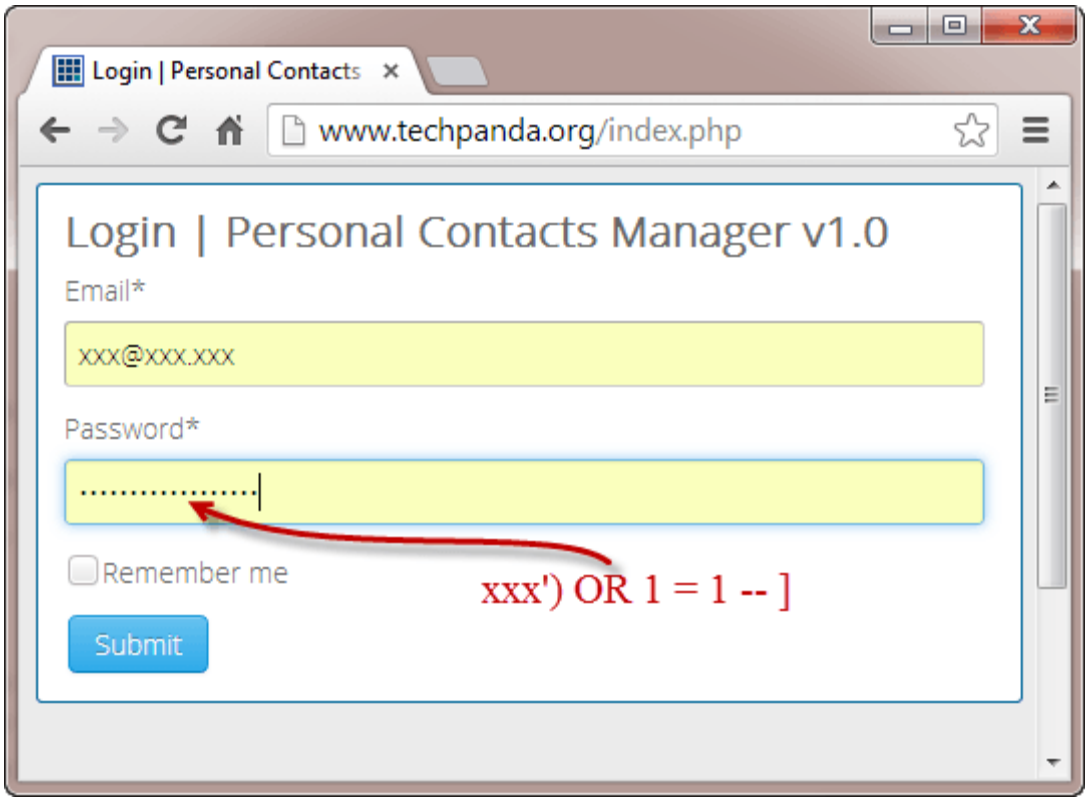
Para contornar isso, podemos explorar o campo de senha. O diagrama abaixo mostra as etapas que você deve seguir



Vamos supor que um invasor forneça a seguinte entrada

Etapa 1: digite xxx@xxx.xxx como o endereço de email

Etapa 2: insira xxx ') OR 1 = 1 -]



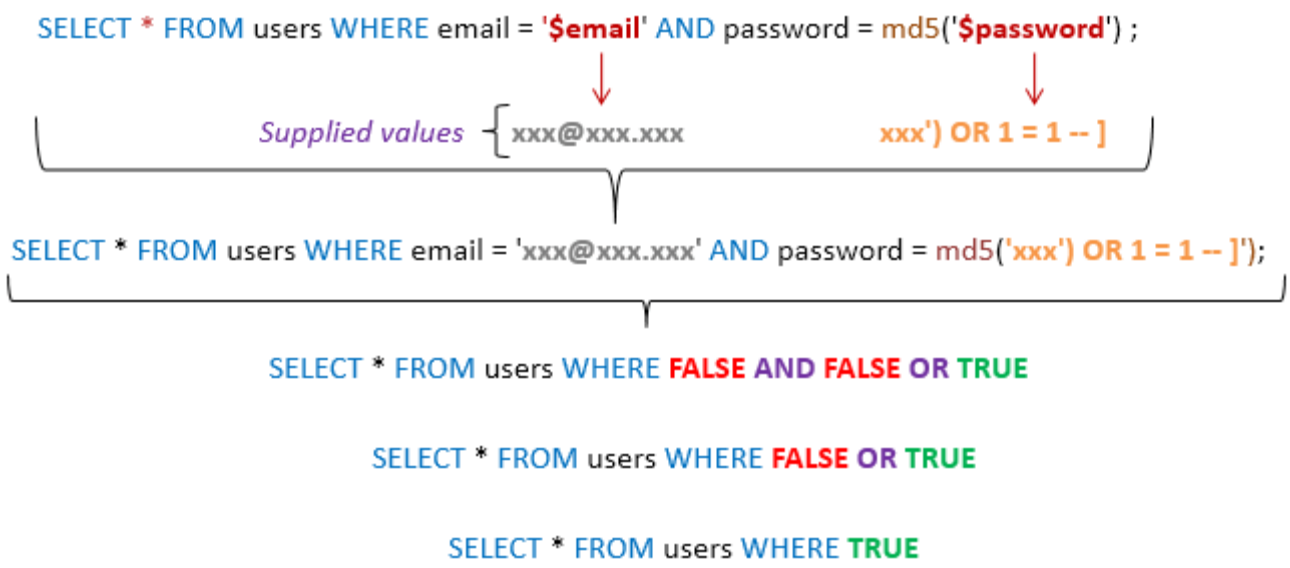
Clique no botão Enviar

Você será direcionado para o painel

A instrução SQL gerada será a seguinte

SELECT * FROM users WHERE email = ' xxx@xxx.xxx ' AND password = md5 ('xxx') OR 1 = 1 -] ');

O diagrama abaixo ilustra a declaração que foi gerada.



A instrução assume de forma inteligente que a criptografia md5 é usada

Conclui a aspas simples e o colchete de fechamento

Anexa uma condição à declaração que sempre será verdadeira

Em geral, um ataque bem-sucedido de SQL Injection tenta várias técnicas diferentes, como as demonstradas acima, para executar um ataque bem-sucedido.

Outros tipos de ataque de injeção SQL

As injeções de SQL podem causar mais danos do que apenas passar os algoritmos de login. Alguns dos ataques incluem

Excluindo dados

Atualizando dados

Inserindo dados

Executando comandos no servidor que podem baixar e instalar programas maliciosos, como cavalos de Tróia

Exportar dados valiosos, como detalhes do cartão de crédito, email e senhas, para o servidor remoto do invasor

Obtendo detalhes de login do usuário etc.

A lista acima não é exaustiva; apenas dá uma idéia do que SQL Injection

Ferramentas de automação para SQL Injection

No exemplo acima, usamos técnicas de ataque manual com base em nosso vasto conhecimento de SQL. Existem ferramentas automatizadas que podem ajudá-lo a realizar os ataques com mais eficiência e no menor tempo possível. Essas ferramentas incluem

SQLSmack - http://www.securiteam.com/tools/5GP081P75C.html

SQLPing 2 - http://www.sqlsecurity.com/downloads/sqlping2.zip?attredirects=0&d=1

SQLMap - http://sqlmap.org/

Como evitar ataques de injeção SQL

Uma organização pode adotar a seguinte política para se proteger contra ataques de SQL Injection.

A entrada do usuário nunca deve ser confiável - sempre deve ser higienizada antes de ser usada em instruções SQL dinâmicas.

Procedimentos armazenados - eles podem encapsular as instruções SQL e tratar todas as entradas como parâmetros.

Instruções preparadas **- instruções** preparadas para funcionar, criando a instrução SQL primeiro e depois tratando todos os dados do usuário enviados como parâmetros. Isso não afeta a sintaxe da instrução SQL.

Expressões regulares - elas podem ser usadas para detectar possíveis códigos nocivos e removê-los antes de executar as instruções SQL.

Direitos de acesso do usuário à conexão com o banco de dados **-** somente os direitos de acesso necessários devem ser concedidos às contas usadas para conectar-se ao banco de dados. Isso pode ajudar a reduzir o que as instruções SQL podem executar no servidor.

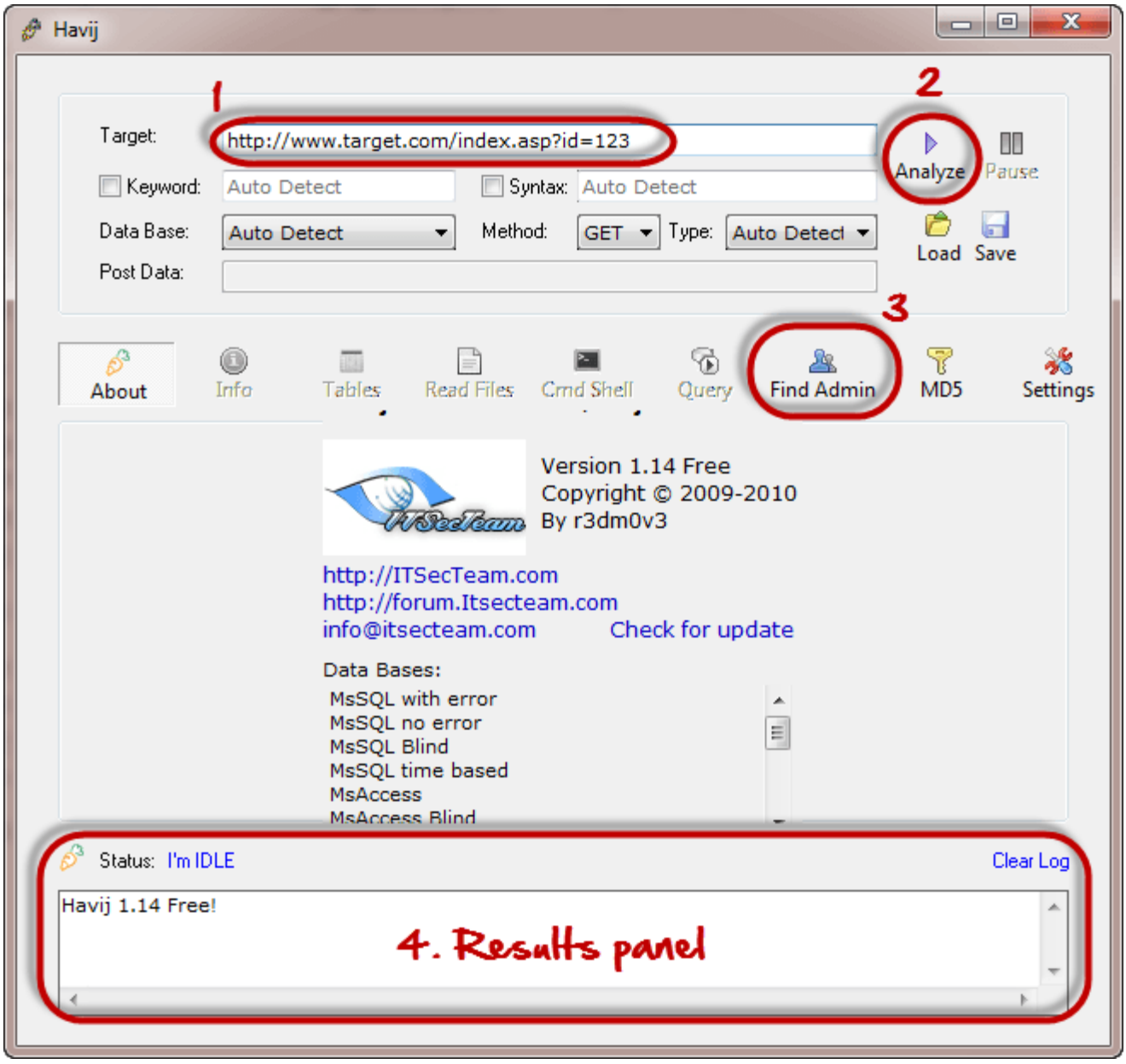
Mensagens de erro - elas não devem revelar informações confidenciais e onde exatamente ocorreu um erro. Mensagens de erro personalizadas simples, como "Desculpe, estamos com erros técnicos. A equipe técnica foi contatada. Tente novamente mais tarde ”pode ser usado em vez de exibir as instruções SQL que causaram o erro.

Atividade de hackers: Use Havij para SQL Injection

Nesse cenário prático, usaremos o programa Havij Advanced SQL Injection para verificar se há vulnerabilidades em um site.

Nota: seu programa antivírus pode sinalizá-lo devido à sua natureza. Você deve adicioná-lo à lista de exclusões ou pausar o software antivírus.

A imagem abaixo mostra a janela principal do Havij



A ferramenta acima pode ser usada para avaliar a vulnerabilidade de um site / aplicativo.

Como proteger seu site contra hackers?

Uma organização pode adotar a seguinte política para se proteger contra ataques a servidores da web.

- **Injeção de SQL** - limpar e validar parâmetros do usuário antes de enviá-los ao banco de dados para processamento pode ajudar a reduzir as chances de serem atacados via Injeção de SQL. Mecanismos de banco de dados, como MS SQL Server, MySQL, etc. suportam parâmetros e instruções preparadas. Eles são muito mais seguros que as instruções SQL tradicionais
- **Ataques de negação de serviço - os** firewalls podem ser usados para eliminar o tráfego de endereços IP suspeitos, se o ataque for um DoS simples. A configuração adequada das redes e o Sistema de detecção de intrusões também podem ajudar a reduzir as chances de um ataque DoS ter êxito.
- **Script entre sites -** validação e limpeza de cabeçalhos, parâmetros transmitidos pela URL, parâmetros de formulário e valores ocultos podem ajudar a reduzir ataques XSS.
- **Envenenamento por cookie / sessão** - isso pode ser evitado criptografando o conteúdo dos cookies, expirando o tempo limite depois de algum tempo, associando os cookies ao endereço IP do cliente que foi usado para criá-los.
- **Tempera do formulário -** isso pode ser evitado validando e verificando a entrada do usuário antes de processá-la.
- **Injeção de código -** isso pode ser evitado tratando todos os parâmetros como dados e não como código executável. Sanitização e validação podem ser usadas para implementar isso.
- **Desfiguração -** uma boa política de segurança de desenvolvimento de aplicativos da web deve garantir que sela as vulnerabilidades mais usadas para acessar o servidor da web. Essa pode ser uma configuração adequada do sistema operacional, do software do servidor da web e das melhores práticas de segurança ao desenvolver aplicativos da web.