

Guia de Segurança na Web com Foco no CMS Joomla

Sumário

Segurança na Web com Foco no CMS Joomla	1
1) Introdução	2
2) Ambiente do Programador	3
3) Hospedagem do Site	4
4) Configurações do PHP	6
5) Configurações do Apache	9
6) Configuração do Joomla	15
7) Algumas Extensões de Terceiros que Colaboram com a Segurança.....	18
8) Ferramentas	22
9) Programação	23
10) Backup e Restore	27
11) Firewall	28
12) Senhas	30
13) E se o site foi invadido?	32
14) Checklist de Segurança para Joomla.....	34
15) Testes de Vulnerabilidade e Simulações de Ataque	37
16 – Permissões do Sistema de Arquivos	38
17) Links úteis/Referências	39

1) Introdução

Alerta Inicial - vale ressaltar que algumas características aqui relatadas podem não funcionar no seu servidor e pode até ser que todas funcionem e mais algumas. Portanto fique atento para fazer os ajustes necessários.

Alerta 2: Este manual tem alguns itens desatualizados. Você precisa conhecer do que se trata e atualizar quando for o caso. Um exemplo é a versão do PHP, que atualmente recomenda-se pelo menos a 5.6 ou 7.

Este artigo é fruto de pesquisas sobre o assunto pela Internet e de algo vindo da minha experiência com a criação de sites. Tem foco em sites com o CMS Joomla, mas muita coisa se aplica a outros CMS e outros tipos de sites em PHP.

Geralmente programadores web criam seus sites como se a Internet fosse um ambiente seguro, ou seja, sem quase nenhuma preocupação com segurança. Mas a cada dia a Internet fica ainda mais insegura. Se acompanharmos as notícias e discussões em listas e fóruns iremos perceber isso.

Nós que trabalhamos criando sites, somos também responsáveis pela segurança dos mesmos, temos obrigação de fazer tudo que estiver ao nosso alcance para tornar os sites que criamos o mais seguro possível. Estudando, testando e a cada dia aprendendo mais a zelar pelo que criamos.

Sempre que for instalar uma nova extensão o faça em ambiente de testes, onde deve efetuar diversas experiências com a mesma e somente quando se sentir seguro instale no site em produção.

2) Ambiente do Programador

Precisamos ter nosso ambiente de trabalho, nosso computador desktop onde criamos o site, isento de ameaças, para que ao enviar o conteúdo ou o nosso site para o servidor não estejamos colaborando para aumentar as ameaças do mesmo. Por isso a sugestão de usar um sistema operacional mais seguro como Linux. Mesmo usando Linux devemos usar um bom firewall para filtrar ameaças. Caso utilizemos Windows devemos nos prevenir usando um navegador menos inseguro e com um bom e atualizado anti-virus, firewall e diversos outros softwares que ajudem a manter o ambiente limpo de ameaças.

Fique atento para a atualização de todos os softwares importantes que utiliza, como antivírus, firewall, IDEs, etc. Não esquecer de atualizar o Sistema Operacional.

Use sempre senhas fortes para tudo no servidor e inclusive em seu desktop. Será perda de tempo investir em muitos cuidados com a segurança, muito tempo de trabalho, muita pesquisa e estudo, se usar senhas fracas e fáceis de serem quebradas. Senhas fortes devem ter no mínimo letras e números. Para reforçar use também símbolos. Uma recomendação importante é que nunca mantenha sua senha do servidor em arquivo texto.

Ajuda muito usar com frequência programas/sites para varredura/scan dos sites que estamos trabalhando. Veja na seção de programação algumas sugestões.

Mesmo que você esteja usando Linux, instale um antivírus como o clamav para manter sua máquina limpa de arquivos de outros sistemas operacionais frágeis e para varrer os arquivos do site quando baixar e antes de enviar. Não esquecer de varrer pendrives que vieram do Windows.

3) Hospedagem do Site

A hospedagem, o servidor que guarda os arquivos do site, é um dos itens mais importantes para a segurança do mesmo. Afinal de contas ele têm o controle sobre os servidores e assim pode implementar os mais diversos filtros de segurança que deixarão seu site em ambiente menos vulnerável ou o contrário, caso não implementem. Antes de contratar colha a maior quantidade possível de informações sobre o serviço. Contate amigos que têm sites hospedados nele, questione sobre como é o serviço que recebem, suas características, para ver se atendem ao que planeja (sempre deixe uma folga em espaço, banda e quantidade de bancos). Existem muitas características sobre uma boa hospedagem que devemos considerar: espaço em disco, banda, quantidade de bancos, de domínios, suporte, suporte a conexões via SSH, segurança (vários itens relacionados), etc. Nunca contrate somente pelo menor preço. Pesquise antes.

Uma boa hospedagem tem um suporte rápido e competente, cuja equipe conhece bem suas funções e pode ajudar a resolver problemas e a detectá-los de forma ágil.

Uma boa hospedagem adota procedimentos que tornam seguro o seu site como:

- Apache chrooted
- PHP instalado como CGI
- Apache modSecurity
- Ativa a extensão Source Guardian no PHP

Seleção da Hospedagem

Escolha criteriosamente o servidor de hospedagem e o tipo. Tenha como um dos fatores de decisão a importância que o servidor dá à segurança.

Escolher um servidor para o site que seja da sua confiança, ou pelo menos que não desconfie dele.

Se possível evite servidores compartilhados.

Caso use um VPS ou servidor dedicado fique atento à monitoração de ataques, TripWire e SAMHAIN são boas ferramentas para isso.

Cheque os logs regularmente.

Use .htaccess

Conexões ao cPanel

Use somente conexões seguras ao conectar ao servidor. Se não conectar assim sua senha poderá ser interceptada. Evite ftp, ao invés use o gerenciador de arquivos o próprio cPanel.

Versão do PHP

Configure seu site para usar pelo menos a versão 5.6 do PHP. (atualize) Para isso adicione a linha abaixo ao .htaccess:

```
AddHandler application/x-httpd-php56 .php .php3 .php4 .php5 .phtml
```

Negando Acesso

Caso queira evitar acesso web a um diretório, crie um .htaccess para o mesmo com a linha:
deny from all

Segurança equilibrada

Não descuide da segurança mas também não seja paranoico, mantenha o equilíbrio. Este é um assunto tão envolvente que pode fazer com que nos descuidemos de outras áreas para nos concentrar somente nele. A segurança inclusive depende de outras áreas para ser forte.

4) Configurações do PHP

php.ini e ini_set()

display_errors

Estes só devem estar ativos quando estamos em ambiente de testes, criando ou programando, antes de enviar para o servidor devemos desativar. Se alguma extensão ou aplicativo precisa dessas extensões ativas o prudente é não usar essas extensões. Aliás, para quando estamos instalando, testando e programando, devemos exibir ao máximo os erros. Idealmente o `php.ini` deve estar com:

```
error_reporting = E_ALL & ~E_DEPRECATED
```

Quando usar php.ini, ini_set() ou .htaccess

Isso vai depender de como o servidor está configurado.

- Algumas configurações do PHP podem ser alteradas usando a função `ini_set()`, sendo que as alterações são válidas somente enquanto o script estiver em execução.
- Outras configurações devem ser adicionadas num script `php.ini` ou no `.htaccess`.
- Outras em qualquer uma das formas acima.
- E algumas configurações não podem ser alteradas, como é o caso de `memory_limit` e `execution_time`. Caso alteremos essas podemos prejudicar a performance do site.

Quando e onde usar cada um vai depender de como o PHP está instalado no servidor, que pode ser como módulo CGI ou como um módulo do Apache.

PHP como Módulo do Apache ou como CGI

Para saber como foi instalado o PHP no seu servidor poderá consultar o help desk. Alguns servidores armazenam as respostas do suporte numa área chamada de **Knowledge Base** até com busca.

A instalação como módulo CGI é mais segura, portanto preferível. Alguns servidores oferecem as duas modalidades e geralmente para servidores compartilhados é oferecido como módulo CGI, mas para ter certeza pergunte ou faça testes para identificar. Veja detalhes abaixo.

Como CGI - mais segura, portanto mais restritiva para alterar as configurações do PHP. Não podemos criar um `php.ini` no raiz e ele valer recursivamente para todos os sub-diretórios. Temos que criar um `php.ini` para cada diretório.

Também não podemos adicionar configurações do PHP nos scripts `.htaccess`. Se o fizermos receberemos um erro e o script não funcionará. Temos que remover o que adicionamos para o site voltar. Podemos usar a função `ini_set()`.

Como módulo do Apache - Aqui podemos criar um `php.ini` no raiz do domínio e ele valerá para todos os sites, ou seja, tem efeito global. Também podemos usar as configurações do PHP em scripts `.htaccess` e podemos usar a função `ini_set()`.

Como Configurar o PHP

`ini_set()` - Esta função é muito útil para efetuar configurações em servidores onde o PHP foi instalado como CGI, especialmente para sites com Joomla. No caso entramos com as linhas da `ini_set()` no `configuration.php`, que é visto por todos os scripts.

Exemplo usando `ini_set()`

```
ini_set('extension', 'sourceguardian.so');
ini_set('session.save_path', '/home/joao/public_html/tmp');
ini_set('cgi.force_redirect', 1);
ini_set('allow_url_fopen', 0);
ini_set('display_errors', 0);
ini_set('allow_url_include', 0);
ini_set('expose_php', 0);
ini_set('magic_quotes_gpc', 0);
ini_set('post_max_size', 262144);
ini_set('upload_max_filesize', 262144);
ini_set('upload_tmp_dir', '/home/joao/public_html/tmp');

$disfunctions = 'proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec, system,
shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note, apache_setenv,
disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice,
symlink, phpinfo';

ini_set('disable_functions', $disfunctions);

// Verifique se pode usar estes abaixo em seu servidor
ini_set('zend_extension', '/usr/local/php52/lib/php/extensions/ioncube.so');
ini_set('zend_extension_manager.optimizer=', '/usr/local/Zend/lib/Optimizer-3.3.3');
ini_set('zend_extension_manager.optimizer_ts', '/usr/local/Zend/lib/Optimizer_TS-3.3.3');
ini_set('zend_optimizer.version', '3.3.3');
ini_set('zend_extension', '/usr/local/Zend/lib/ZendExtensionManager.so');
ini_set('zend_extension_ts', '/usr/local/Zend/lib/ZendExtensionManager_TS.so');
```

php.ini - Quando criamos um php.ini e adicionamos a um certo diretório, as diretivas dele sobrescreverão as existentes no script php.ini do servidor, mudando o valor das diretivas, mas perdendo alguns recursos importantes, como é o caso do ionCube. Veja exemplo abaixo para contornar isso.

Exemplo de php.ini para reforçar a segurança

```
extension=sourceguardian.so
session.save_path = "/home/ribafs03/public_html/tmp"
cgi.force_redirect = 1
allow_url_fopen= 0
display_errors = 0
expose_php = 0
magic_quotes_gpc = 0
memory_limit = 8388608
#open_basedir = 1
post_max_size = 262144
upload_max_filesize = 262144
upload_tmp_dir = "/home/ribafs03/public_html/tmp"
disable_functions = proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec,
system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note, apache_setenv,
disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice,
symlink, phpinfo

// Checar se seu servidor suporta os abaixo e as versões
zend_extension=/usr/local/php56/lib/php/extensions/ioncube.so
zend_extension_manager.optimizer=/usr/local/Zend/lib/Optimizer-3.3.3
zend_extension_manager.optimizer_ts=/usr/local/Zend/lib/Optimizer_TS-3.3.3
zend_optimizer.version=3.3.3
zend_extension=/usr/local/Zend/lib/ZendExtensionManager.so
zend_extension_ts=/usr/local/Zend/lib/ZendExtensionManager_TS.so
```

Lista de todas as diretivas só php.ini

http://www.php.net/manual/pt_BR/ini.list.php

Descrição das diretivas do principais do arquivo php.ini

http://www.php.net/manual/pt_BR/ini.core.php

5) Configurações do Apache

Sobre o .htaccess

O arquivo .htaccess, que no linux é um arquivo oculto (inicia com ponto), é um arquivo muito útil para a administração de segurança e vários outros recursos em sites que usam o Apache como servidor web. Em especial quando não temos acesso direto às configurações do Apache.

Usado para configurar o Apache e também o PHP (somente se instalado como módulo do Apache).

Veja alguns dos seus vários e úteis usos. Li certa vez: "Use e abuse do .htaccess, pois ele é seu amigo."

Listar Arquivos de Diretório

Se por exemplo você quer que o diretório onde você colocou o .htaccess liste os arquivos caso não haja um index.html da vida, você adiciona o seguinte no .htaccess:

```
Options +Indexes
```

E para tirar essa opção:

```
Options -Indexes
```

Permitir acesso somente para uma faixa de IPs:

```
<Files pagina_erro_403.php>
```

```
Order Deny,Allow
```

```
Deny from all
```

```
Allow from 192.168.
```

```
</Files>
```

Como personalizar páginas de erro:

```
ErrorDocument 403 /acesso_negado.php
```

```
ErrorDocument 404 /nao_encontrado.php
```

```
ErrorDocument 500 /erro_interno_servidor.php
```

401 - Authorization Required

400 - Bad request

403 - Forbidden

404 - Wrong page

500 - Internal Server Error

Ativar mod_rewrite

```
RewriteEngine On
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteRule ^(.*)$ index.php?pagina=$1
```

Bloqueia uma lista de IPs:

```
order allow, deny
deny from 210.140.98.160
deny from 69.197.132.70
deny from 74.14.13.236
allow from all
```

Deixa a Intranet acessar

```
Order allow,deny
allow from 192.168.0.
deny from all
```

Deixa todo mundo acessar, menos o IP 192.168.0.25

```
Order deny,allow
deny from 192.168.0.25
allow from all
```

Impedir todos de visitarem o site

```
deny from all
```

Obs.: no caso acima somente permite acesso pelo cpanel, ftp ou outro, nunca pela web.

Restringe o arquivo "secreto.html" somente para o IP 192.168.0.30

```
<Files secreto.html>
Order allow,Deny
Allow from 192.168.0.30
Deny from all
</Files>
```

Nega o acesso dos clientes ao .htaccess (bom colocar no httpd.conf)

Vem com a configuração padrão do Apache

```
<Files ~ "\.ht">
```

```
Order allow,deny
```

```
Deny from all
```

```
</Files>
```

Redirecionar todos os visitantes que chegarem na pasta /antigo para o site <http://www.novosite.com/novo>

```
Redirect /antigo http://www.site.com/novo
```

No caso, o arquivo <http://www.site.com/antigo/teste.png> será redirecionado para <http://www.site.com/novo/teste.png>

Proteger Diretório com Login e Senha

Usar o comando htpasswd para criar a senha com a seguinte sintaxe:

```
htpasswd -c arquivodasenha login
```

```
htpasswd -c senha joao
```

O arquivo gerado conterá uma linha com login:senhacriptografada

Inserir no diretório um arquivo .htaccess com o conteúdo:

```
AuthName "Acesso Restrito"
```

```
AuthType Basic
```

```
AuthUserFile /backup/www/diretorio/senha
```

```
Require valid-user
```

/backup/www/diretorio/senha - é o caminho completo para o arquivo com a senha

Exige senha para acessar o diretório administrator

```
<Directory /administrator>
```

```
AuthName "Acesso Restrito à Usuários"
```

```
AuthType Basic
```

```
AuthUserFile /etc/httpd/auth/aceso
```

```
AuthGroupFile /etc/httpd/auth/grupos
```

```
require group admin
</Directory>
```

Bloquear Perl

Muitos scripts de ataque são criados em Perl, portanto para bloquear perl e outros bots para que não acessem seu site, adicione o código abaixo em um .htaccess (no raiz do domínio):

```
SetEnvIfNoCase User-Agent libwww-perl bad_bots
order deny,allow
deny from env=bad_bots
```

bogus handler para perl

Caso não esteja usando scripts em Perl em seu site adicione um "bogus handler" para estes scripts no .htaccess:

```
##Deny access to all CGI, Perl, Python and text files
<FilesMatch "\.(cgi|pl|py|txt)">
Deny from all
</FilesMatch>
## Se não está usando um arquivo robots.txt, então comente
# as 3 linhas abaixo para evitar o acesso somente ao arquivo robots.txt
<FilesMatch robots.txt>
Allow from all
</FilesMatch>
```

Outros recursos importantes para o .htaccess:

```
#Enable mod_rewrite and insert some sample rules:
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteCond %{REQUEST_URI} ^(component/option,com) [NC,OR] ##optional - see notes##
RewriteCond %{REQUEST_URI} (/\.htm\.php\.html/[^.]*)$ [NC]
RewriteRule ^(content/|component/) index.php
```

Protegendo o acesso direto ao .htaccess e ao configuration.php:

```
<Files .htaccess>
order allow,deny
deny from all
```

```
</Files>
```

```
<FilesMatch "configuration.php">
```

```
Order allow,deny
```

```
Deny from all
```

```
</FilesMatch>
```

E outros

```
<FilesMatch "\.(htaccess|htpasswd|ini|phps|log|sh|conf)$">
```

```
Order allow,deny
```

```
Deny from all
```

```
</FilesMatch>
```

Muito Importante

Adicione ao final do .htaccess:

```
# Block out any script trying to set a mosConfig value through the URL
```

```
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|\%3D) [OR]
```

```
# Block out any script trying to base64_encode crap to send via URL
```

```
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
```

```
# Block out any script that includes a <script> tag in URL
```

```
RewriteCond %{QUERY_STRING} (<|\%3C).*script.*(\>|\%3E) [NC,OR]
```

```
# Block out any script trying to set a PHP GLOBALS variable via URL
```

```
RewriteCond %{QUERY_STRING} GLOBALS(=|\\[|\\%[0-9A-Z]{0,2}) [OR]
```

```
# Block out any script trying to modify a _REQUEST variable via URL
```

```
RewriteCond %{QUERY_STRING} _REQUEST(=|\\[|\\%[0-9A-Z]{0,2})
```

```
# Send all blocked request to homepage with 403 Forbidden error!
```

```
RewriteRule ^(.*)$ index.php [F,L]
```

```
# Dica de http://forum.codecall.net/security-tutorials/4867-joomla-hacking-script.html
```

Criando um sistema de detecção de intrusão com o .htaccess

URL encoding attacks such as SQL injection, white space, javascript, etc and redirects the URL to log.php. Log.php will then alert you via email.

Referência: <http://www.hackosis.com/simple-htaccess-intrusion-detection-system/>

```
Options +FollowSymLinks
```

```
RewriteEngine On
```

```

RewriteCond %{QUERY_STRING} (\|%22).*(>|%3E|<|%3C).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING} (<|%3C).*script.*(>|%3E) [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING} (javascript:).*(;).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING}
(\;|\'|\"\\|%22).*(union|select|insert|drop|update|md5|benchmark|or|and|if).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteRule (,;|<|>|'|) /log.php [NC]

```

Criar o arquivo log.php na raiz do site. Mudar o e-mail dmin@site.com Este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo para receber a notificação:

```

<?php
$r= $_SERVER['REQUEST_URI'];
$q= $_SERVER['QUERY_STRING'];
$i= $_SERVER['REMOTE_ADDR'];
$u= $_SERVER['HTTP_USER_AGENT'];
$mess = $r . '|' . $q . '|' . $i . '|' . $u;

mail(" admin@site.com este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo. ", "bad request", $mess, "from: bot@site.com este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo. ");

echo "Hot Damn!";

?>

```

6) Configuração do Joomla

Atualização

Mantenha o Joomla e as extensões do seu site atualizados.

Recomendação importante: sempre faça um backup full do site (todos os arquivos e o banco), antes de atualizar, pois pode ser que você tenha alterado alguma extensão ou o próprio Joomla e a atualização apague isso. Sugestões: com_simplebackup e AkeebaBackup.

Não atualize para novas versões (exemplos: da 2.5 para a 3, da 3.x para a 4) de imediato, tenha prudência

- Uma ótima extensão para colaborar com essa tarefa é o componente Admin Tools, do mesmo desenvolvedor do Akeda Backup: <http://www.akeebabackup.com/download.html>
- Sempre instale imediatamente que faça o download, mas antes de atualizar realize um backup full para se prevenir, pois algumas atualizações podem alterar o template default (que você pode estar usando) ou outra extensão que tenha personalizado, portanto backup full antes.

Aproveite para assinar também o RSS das novidades sobre segurança:

<http://feeds.joomla.org/JoomlaSecurityNews>

Extensões de Terceiros

Evite ao máximo utilizar extensões de terceiros. Sempre que o Joomla tiver uma extensão, prefira a do Joomla, como por exemplo, para URLs amigáveis prefira usar o recurso do Joomla. Somente instale extensões de terceiros se extremamente necessário. E se instalar sempre vá ao site do autor e baixe a última versão e também mantenha-a atualizada. Também tenha o cuidado de instalar antes em um computador de testes e somente após alguns testes instale no site em produção.

Muita prudência ao instalar extensões de terceiros:

- Evite usar extensões de terceiros, especialmente quando o Joomla já trazer uma nativa similar.
- Caso decida instalar visite antes a Lista de Vulnerabilidade de Extensões.
- Extensões não usadas devem ser desinstaladas.
- Instale em máquina de teste antes de colocar em produção. Instale várias vezes, teste, execute e se for um template teste em vários navegadores.
- Baixe somente do site dos criadores
- Mantenha sempre atualizada
- Caso descubra que uma extensão é insegura não somente desabilite mas desinstale e remova manualmente tudo que restar

Evite Alteração do Código do Core

Evite hackear (alterar) o código do core do Joomla tornando difícil a manutenção e atualização e

inseguro o mesmo.

Criação de Artigos

Ao criar artigos ou permitir que estranhos criem artigos é importante filtrar HTML ou configurar o usuário para não usar editor HTML, sob pena de ter o site invadido.

Download do Joomla

Faça sempre o download do site oficial.

Valores Default

Valores default são perigosos, pois são de conhecimento de todos.

Esconda o diretório administrator de curiosos usando uma extensão como oAdminExile.

Mover o configuration.php para fora da área do Apache

Mover o configuration.php para fora da área do Apache e mudar as permissões para 400. Caso queira alterar mude para 600.

Se seu site estiver em public_html/portal

Então copie o configuration.php para o diretório abaixo de public_html e o renomeie para portal.cfg, além de mudar as permissões dele para 400.

Somente mude para 600 se precisar alterar. Então edite o original deixando somente a linha abaixo:

```
require_once( dirname( __FILE__ ) . '/../portal.cfg' );
```

URLs Amigáveis

Ativar as URLs amigáveis para evitar ataques pela URL e também maior visibilidade no Google além de tornar mais amigável para os visitantes.

robots.txt

Edição de robots: libere a pasta images

Páginas de Erro

Crie páginas de erro mais bonitas e com informações úteis ao visitante, como um e-mail para feedback.

Joomla não é Perfeito

O Joomla é o software mais bem feito que já vi, mas cuidado para não chegar a pensar que o Joomla é perfeito e não precisa de ajustes na segurança.

Logs e Tmp

Diretório de Logs - Altere no Admin: Configuração Global - Sistema - Caminho para o diretório do log

Diretório Tmp - Altere também: Configuração Global - Servidor - Caminho para o diretório temporário.

Desabilitar Extensões não Usadas

Plugin XML-RPC caso não precise dele desabilite-o. Vem desabilitado por default.

7) Algumas Extensões de Terceiros que Colaboram com a Segurança

AdminTools - Componente que detecta novas atualizações do Joomla e do próprio componente, alertando quando o acessamos. Além disso ele pode instalar a nova versão com apenas 3 cliques. Além disso ele também corrige as permissões do site para 755 (diretórios) e 644 (arquivos), adiciona uma segunda camada de segurança (novo login) ao administrador, além de outros recursos.

É um verdadeiro canivete suíço para sites em Joomla.

Atualmente recebemos um e-mail tão logo sai uma nova versão do Joomla. Ao ser avisado devemos ir ao administrador e o atualizar.

Recomendação importante: sempre faça um backup full do site (todos os arquivos e o banco), antes de atualizar, pois pode ser que você tenha alterado alguma extensão ou o próprio Joomla e a atualização apague isso e eventualmente a atualização pode impedir o acesso ao site.

Plugin jHackGuard - plugin criado pelo SiteGround sites para proteger sites em Joomla de ataques de hackers. Basta instalar e ativar para ganhar proteção contra ataques tipo SQL Injections, Remote URL/File Inclusions, Remote Code Executions e XSS.

<http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233>

Veja este artigo do Júlio Coutinho sobre o JhackGuard:

JHackGuard plugin de segurança para o Joomla!

jHackGuard é um plugin desenvolvido pelo SiteGround e ajuda a proteger contra ataques de crackers. Basta adicionar ao seu Joomla e ele estará mais seguro contra sqlinjections, codeinjections e ataques baseados em xss!

Este plugin tem sido utilizado com sucesso pelos clientes SiteGround durante os últimos anos e o Siteground resolveu tornar pública a sua versão mais recente, de modo que você pode facilmente proteger o seu site Joomla. Tudo que você precisa fazer é instalar jHackGuard e habilitá-lo - nenhuma configuração adicional necessária!

O Plugin de segurança do SiteGround assegura sites Joomla, protegendo-os contra a diferentes técnicas de hacking. Ele filtra os dados de entrada dos usuários e implementa as configurações de segurança adicionais PHP.

O Plugin de segurança do SiteGround contém opções de filtragem de segurança avançadas. Ele vem com um conjunto de regras pré-definidas, que funciona na maioria dos casos. Caso deseje, você pode alterar os padrões. Felizmente, ele tem um log e você pode depurar qualquer comportamento inesperado.

O Plugin de segurança do SiteGround pode ser configurado através da área de administrador do Joomla. O plugin está desativado para os administradores autenticados para que os filtros não os impeçam de fazer tarefas administrativas.

Aconselho testar o plugin localmente e caso haja alguma instabilidade em seu website, faça o seguinte:

- 1) Acesse o banco de dados pelo phpmyadmin
- 2) Localize a tabela #__plugins
- 3) Localize o plugin JHackGuard
- 4) Clique no lápis (editar)
- 5) Altere o valor do status de 1 para 0 para desabilitar o plugin

Instalação jHackGuard

A instalação do plugin é padrão como qualquer outra extensão do Joomla.

Você baixa o plugin para sua máquina e no backend do Joomla navega pelo menu superior Extensões -> Instalar / Desinstalar.

Clique no botão Procurar e localize o pacote de extensão no seu disco rígido. Em seguida, clique no botão "Upload File & Install". O plugin será instalado e acrescentado ao Joomla.

Por último, siga por Extensões -> Gerenciamento de Plugin. Nela localize o plugin jHackGuard e clique no ícone "habilitar". Isso irá ativar o plugin.

As regras padrão de jHackGuard foram programadas pelos especialistas do Siteground, com base em sua experiência na fixação de um grande número de vulnerabilidades de diferentes sites Joomla. Recomendamos o uso das normas padrão para o melhor desempenho do plugin.

No entanto, se você quiser fazer alterações específicas para suas configurações, você pode fazer isso a partir da página Gerenciamento de Plugin em sua área administrativa do Joomla. Uma vez lá, clique no security - rótulo Plugin jHackGuard para entrar em sua página de configurações. Os parâmetros configuráveis para o plugin são separados em vários grupos:

*** Opções de Login**

* Arquivo de log - Aqui você pode digitar o nome do arquivo onde os registros sobre as atividades plugin serão mantidos. O nome do arquivo padrão é log.php-jHackGuard. Ela é armazenada sob a pasta de logs.

* Enable Login - Você pode decidir se as atividades do plugin serão registradas

*** Fluxos de Dados**

* Filtro \$_POST - Filtros variáveis provenientes do método POST HTTP.

* Filtro \$_GET - Filtros de variáveis passadas para o script através de parâmetros de URL.

* Filtro \$_COOKIE - Filtros de variáveis provenientes de HTTP Cookies.

*** Parâmetros de filtragem**

* Filtro de eval () - Filtra o resultado da avaliação de uma string como código PHP.

* Filtro base64_decode - Filtra o resultado da decodificação de dados codificados em base64.

* Filtro de comandos SQL - Filtra a execução de comandos SQL. Esta solução evita os ataques de injeção SQL.

*** Parâmetros avançados**

* Allow_url_fopen - Desativa a opção de recuperar arquivos de FTP remoto ou servidor web. Esta

solução protege seu site contra injeção de código.

* Allow_url_include - Desativa a opção de incluir URLs de pedidos PHP. Desta forma seu site estará protegido contra ataques remotos URL Inclusão.

(*) Plugin sugerido por LinkDF 2010, membro da comunidade Joomla! Brasília no Orkut.

Fonte: <http://www.siteground.com>

Artigo traduzido e adaptado por Júlio Coutinho no site:

<http://www.joomlabrasilia.org/tutoriais-de-joomla/seguranca-e-joomla/624.html>

Este plugin adiciona um link na parte inferior do template e tem um pequeno problema para quem usa o PHP mostrando Notices.

Comente a linha 80 caso queira ocultar o link.

A linha 370 deve ficar assim:

```
$chars = 'PCRE_UNICODE_PROPERTIES' ? '\pL' : 'a-zA-Z';
```

Adicionando as aspas simples na contante.

Componente com_encrypt – Adiciona criptografia para os campos dos forms, o que evita que usuários maliciosos monitorem e capturem senhas em texto claro.

Vários outros para extensões de terceiros:

<http://www.ratmil.com/downloads/category/4-encryption-configuration-plugins.html>

Plugin Marcos Interceptor SQL Injection – Um plugin para prevenir SQL injection, prevenindo ataques deste tipo, contendo:

- •.Filters requests in POST, GET, REQUEST. and blocks SQL injection / LFI attempts
- •.Notifies you by e-mail when a alert is generated
- •.Protect also from unKnown 3rd Party extensions vulnerability.
- •.White list for safe components (at your risk ;)

Enable mail report and prepare yourself to be scared!

Anyway remember that security it is a 'forma mentis', not a plugin!

<http://www.mmleoni.net/sql-iniection-lfi-protection-plugin-for-joomla>

OSOL Captcha - Use um bom plugin com Captcha para todos os forms do site.

Este adiciona captcha em qualquer form do site.

Para adicionar um captcha em form que não adicionou automaticamente, adicione no código do form:

```
<?php
```

```
global $mainframe;
```

```
//set the argument below to true if you need to show vertically( 3 cells one below the other)  
$mainframe->triggerEvent('onShowOSOLCaptcha', array(false));  
?>
```

Mude false para true se quiser os campos na vertical.

<http://www.outsource-online.net/osol-captcha-for-joomla.html>

8) Ferramentas

Para Joomla, ferramentas são softwares que não têm um instalador como as demais instalações mas que de alguma forma trabalham com o Joomla, sendo executados pela linha de comando ou mesmo pela web, como as abaixo.

Joomla Scan

Ferramenta não instalável, que precisa ser executada na linha de comando.

Yet Another Joomla Vulnerability Scanner that can detects file inclusion, sql injection, command execution vulnerabilities of a target Joomla! web site.

Download

<http://sourceforge.net/projects/joomscan/>

[http://www.owasp.org/index.php/OWASP Joomla Vulnerability Scanner Usage](http://www.owasp.org/index.php/OWASP_Joomla_Vulnerability_Scanner_Usage)

[http://www.owasp.org/index.php/Category:OWASP Joomla Vulnerability Scanner Project](http://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project)

Descompacte para o public_html

Acesse o terminal e vá para a pasta joomscal-latest

```
chmod +x joomscan.pl
```

```
./joomscan.pl -pv -u victim.com -x localhost
```

Irá mostrar vulnerabilidades encontradas no Joomla e as respectivas versões

phpSecInfo – este é um aplicativo em PHP que deve ser instalado no servidor (não usa bancos de dados) e mostra as vulnerabilidades do PHP e sugestões para corrigir.

<http://phpsec.org/projects/phpsecinfo/>

9) Programação

Quando programando para Joomla devemos utilizar seu framework, em especial suas funções de filtragem para bancos de dados e para limpeza de arquivos no sistema de arquivos, entre outras.

Não devemos esquecer as boas práticas de programação e manter o código bem organizado.

Devemos ter cuidado especial com todas as entradas de usuários: URLs, campos de formulários devem ser filtrados por caracteres especiais, especialmente campos hidden, cookies, etc para isso usando as funções do Joomla.

Devemos sempre usar criptografia em campos de senha, reforçar formulários com tokens.

Devemos usar session para bloquear o acesso direto em todos os scripts.

No caso do Joomla, devemos ativar o recurso de URLs amigáveis para maior proteção contra os ataques via URL.

Nunca usar caminhos diretos em includes e sempre preferir require a include, pois os includes não param em erros nem disparam mensagens de erro.

Preferir algo como:

```
require_once( dirname( __FILE__ ) . '/../../tiago.cfg' );
```

Que ocultam o caminho real.

Visitar frequentemente sites públicos de divulgação de vulnerabilidades como o

Bons artigos e sites sobre vulnerabilidades no código:

https://docs.joomla.org/Archived:Vulnerable_Extensions_List

http://www.owasp.org/index.php/Main_Page

<http://www.owasp.org/index.php/Category:Vulnerability>

[http://www.owasp.org/index.php/Category:OWASP Joomla Vulnerability Scanner Project](http://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project)

<http://www.invasao.com.br/2009/01/31/vulnerabilidades-em-aplicacoes-web/>

<http://www.vivaolinux.com.br/dica/Explorando-vulnerabilidades-em-websites>

<http://www.vivaolinux.com.br/artigo/Vulnerabilidade-em-formulario-PHP/>

<http://www.portugal-a-programar.org/forum/index.php?topic=43795.0>

<http://samurai.intelguardians.com/>

<http://sectools.org/web-scanners.html>

[http://wiki.locaweb.com.br/pt-](http://wiki.locaweb.com.br/pt-br/Verificando_vulnerabilidades_em_aplica%C3%A7%C3%B5es_Web)

[br/Verificando_vulnerabilidades_em_aplica%C3%A7%C3%B5es_Web](http://wiki.locaweb.com.br/pt-br/Verificando_vulnerabilidades_em_aplica%C3%A7%C3%B5es_Web)

<http://code.google.com/p/websecurify/>

[http://wiki.locaweb.com.br/pt-br/Joomla: Aprenda como manter o seu Website seguro](http://wiki.locaweb.com.br/pt-br/Joomla:_Aprenda_como_manter_o_seu_Website_seguro)

<http://www.criarweb.com/artigos/principais-vulnerabilidades-web.html>

Quem usa Debian ou derivado:

```
apt-get install w3af
```

Ferramentas para Firefox

<https://addons.mozilla.org/en-US/firefox/addon/161722/eula/88917?src=search> (Acunetix)

<https://addons.mozilla.org/en-US/firefox/addon/7597/eula/88410?src=search>

<https://addons.mozilla.org/firefox/downloads/file/77248/groundspeed-1.1-fx.xpi?src=search>

https://addons.mozilla.org/firefox/downloads/file/101322/x-forwarded-for_spoofers-10.0.2-fx.xpi?src=search

<https://addons.mozilla.org/en-US/firefox/addon/722/>

<https://addons.mozilla.org/en-US/firefox/addon/7598/eula/88414?src=search>

<https://addons.mozilla.org/en-US/firefox/addon/7595/eula/88415?src=search>

<https://addons.mozilla.org/en-US/firefox/addon/14598/eula/65681?src=search>

<https://addons.mozilla.org/en-US/firefox/addon/49858/eula/95600?src=search>

<https://addons.mozilla.org/en-US/firefox/addon/45607/eula/67793?src=search>

<https://addons.mozilla.org/en-US/firefox/addon/14600/eula/65683?src=search>

<http://github.com/codebutler/firesheep/downloads> - varre sites a procura de sites sem senha e mostra conexões wi-fi sem senha permite visualizar uma lista de contas online que estão compartilhando a rede wi-fi aberta e basta um clique para se logar como usuário da conta. Dessa forma, as contas são invadidas facilmente.

O Firesheep oferece cookies a sites como Facebook e Twitter, que os utilizam para permitir o acesso do usuário - em lugar de solicitar senhas.

O próprio desenvolvedor da extensão, Eric Butler, expôs o ponto fraco da web afirmando que qualquer pessoa que visite um site inseguro conhecido pelo Firesheep terá nome e foto exibido em uma nova janela para todos os usuários conectados à rede.

Na lista de sites que permitem acesso por cookies estão Amaxon, Flickr, Google, WordPress, Yahoo e muitos outros.

A vulnerabilidade não é específica do Firefox e mudar de browser não ajuda. A melhor opção, pelo menos por enquanto, é evitar o uso de redes Wi-Fi abertas para acessar esse tipo de conteúdo.

E muitas outras que pode encontrar no site de addons da mozilla.

Algumas Funções Úteis

```
<pre>
```

```
<?php
```

```
// Gerador de senhas aleatórias
// Autor original - Murilo Miranda
// Adaptação de Ribamar FS
```

```

function senha_aleatoria($nc){

$tamanho = $nc;

$letras = array('b','c','d','f','g','h','j','k','l','m','n','p','r','s','t','v','w','x','z');
$vogais = array('a','e','i','o','u');
$numeros = array('1','2','3','4','5','6','7','8','9');
rand(1,$cur);

    $contador = 0;
    while($contador < $tamanho){

        $controle = rand(0,2);
        if($controle == 0){
            $numeroaleatorio = rand(0,18);
            $senha .= $letras[$numeroaleatorio];
        }elseif($controle == 1){
            $numeroaleatorio = rand(0,4);
            $senha .= $vogais[$numeroaleatorio];
        }elseif($controle == 2){
            $numeroaleatorio = rand(0,8);
            $senha .= $numeros[$numeroaleatorio];
        }

        $contador++;
    }
    return $senha;
}

```

```

// Verificar se IP pertence a uma faixa de IPs
// Autor: studio24
// Adaptação de Ribamar FS

```

```

function checkIPorRange ($ip_address) {
    if (ereg("-", $ip_address)) {
        // Range
        $ar = explode("-", $ip_address);
        $your_long_ip = ip2long($_SERVER["REMOTE_ADDR"]);
        if ( ($your_long_ip >= ip2long($ar[0])) && ($your_long_ip <= ip2long($ar[1])) ) {
            return TRUE;
        }
    } else {
        // Single IP
        if ($_SERVER["REMOTE_ADDR"] == $ip_address) {
            return TRUE;
        }
    }
    return FALSE;
}

```

```
$ip= $_SERVER["REMOTE_ADDR"];

$ip_range = "198.152.0.0-198.152.4.0";
//$ip_range = "127.0.0.0-127.0.0.3";

if (checkIPorRange($ip_range)) {
    print "O IP $ip está na faixa: $ip_range";
}else{
    print "O IP $ip não está na faixa: $ip_range!";
}
//If your IP was 198.152.1.32 The above would return: IP OK

?>

</pre>
```

10) Backup e Restore

Efetuar Backups é uma das medidas mais importantes em termos de segurança, pois mesmo que o site tenha sido inteiramente destruído, se você tiver um backup confiável poderá em pouco tempo colocar seu site de volta.

Tenha um planejamento para backup e restore. Mantenha várias cópias do site (todos os arquivos e o banco) e faça testes locais de restauração.

Como você nunca sabe quando um ataque pode acontecer então faça backup e vários.

Faça backup full do site com muita frequência: simplebackup e akeba backup ajudam (mas apenas para sites abaixo de 100MB, acima disso faça um backup manual).

Guarde não apenas uma cópia do backup, mas várias, pois pode ser que a última esteja comprometida ou corrompida.

Idealmente faça teste de restauração do site em micro local para se certificar logo após o backup.

11) Firewall

Recursos importantes e que devem existir num bom servidor de hospedagem.

mod_security

É um módulo do Apache que funciona como uma aplicação de firewall incorporada. Ele oferece proteção de uma faixa de aplicativos web de ataque e permite monitoração de tráfego em tempo real e análise em tempo real e mais.

Apache Chrooted

Este é mais um firewall para servidores de hospedagem, usar o Apache chrooted.

IDS e IPS

IDS (Intrusion Detection System) e IPS (Intrusion Prevention System), assim como um kernel compilado voltado para a segurança tornam o servidor mais seguro também.

Práticas ruins

- Usar extensões desatualizadas
- Não efetuar outros cheques de segurança do PHP
- Desligar-se das permissões de arquivos e diretórios
- Não se preocupar com firewalls no servidor, como mod_rewrite e mod_security

Bugs:

- Não incluir o teste:
`defined('_JEXEC') or die('Restricted access');`
- Construções mal feitas de includes

Resetar senha de usuário da seção administrator do Joomla

Execute o phpmyadmin

Abra o banco do portal

Selecione a tabela jos_users

Edite o registro do usuário que deseja resetar a senha, geralmente o super-administrador

Apague todo o conteúdo do campo password e digite a senha desejada.

Em Funções selecione MD5 para o campo password

Clique no botão executar

Agora pode acessar o administrator

Localizando Exploits

Quem tiver acesso ao shell via SSH

Cheque os processos ativos

```
netstat -ae | grep irc
```

```
netstat -ea | grep 666
```

por portas 6666, 6667, 6668, 6669, geralmente usadas por bots IRC, que devem ter o nome irc ou http listado.

Cheque o crontab

Veja se não encontra alguma entrada estranha no crontab

Cheque por arquivos e diretórios ocultos

Outros exemplos de busca que ajudam a localizar exploits ou arquivos/pastas inesperadas:

```
find /home -type f | xargs grep -l MultiViews
```

```
find . -type f | xargs grep -l base64_encode <<< this can produce false positives, it is valid in many mail/graphics scripts
```

```
find . -type f | xargs grep -l error_reporting
```

. no Linux é o diretório atual

```
find / -name "[Bb]itch[xX]"
```

```
find / -name "psy*"
```

```
ls -lR | grep rwxrwxrwx > listing.txt
```

[Um bom sistema de IDS reforça a segurança. O PHPIDS é o mais popular destes sistemas:](http://php-ids.org/)

12) Senhas

Use senhas fortes para o cPanel, para usuários dos bancos, e-mails, para o Joomla, etc. De nada vai adiantar todo o cuidado com a segurança, se você deixa seu usuário com uma senha bem fácil ou medianamente fácil. A senha deve ser forte.

FTP

Não use FTP, pois as senhas navegam em texto claro. Em seu lugar prefira sftp (FileZilla), o gerenciador de arquivos do cPanel, etc.

Não dê oportunidade aos Crackers

Um cracker precisa ter duas coisas: oportunidade e habilidade. Vários crackers têm habilidade, não dê a eles a oportunidade.

Notícias Ruins

- Não existe segurança perfeita, portanto é melhor prevenir com backup
- Não existe a melhor forma, sempre existem várias formas de se fazer, seja criativo e esforce-se
- Nada substitui a experiência, portanto estude e pratique. Aqui o que precisamos aprender: GNU/Linux, Apache, MySQL, SQL, PHP, HTTP, CSS, XML, RSS, TCP/IP, FTP, Subversion, JavaScript e Joomla, etc.

Pequena página em JavaScript que critica a força de senhas

```
<script>
function TestaSenha(valor) {
  var d = document.getElementById('seguranca');
  // Expressões Regulares
  ERaz = /[a-z]/;
  ERAZ = /[A-Z]/;
  ER09 = /[0-9]/;
  ERxx = /[@!#$%&*+=?|-]/;
  // Teste da String
  if(valor.length == ""){
    d.innerHTML = '<b>Segurança da senha: !</b>';
  } else {
    if(valor.length < 5){
      d.innerHTML = '<b>Segurança da senha: <font color='\red\'> BAIXA</font></b>';
    } else {
      if(valor.length > 7 && valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ER09) != -1 || valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ERxx) || valor.search(ERaz) != -1 && valor.search(ERxx) != -1 &&
valor.search(ER09) || valor.search(ERxx) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ER09)){
        d.innerHTML = '<b>Segurança da senha: <font color='\green\'> ALTA</font></b>';
      } else {
```

```

    if(valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 || valor.search(ERaz) != -1 &&
valor.search(ER09) != -1 || valor.search(ERaz) != -1 && valor.search(ERxx) != -1
||valor.search(ERAZ) != -1 && valor.search(ER09) != -1 ||valor.search(ERAZ) != -1 &&
valor.search(ERxx) != -1 ||valor.search(ER09) != -1 && valor.search(ERxx) != -1){
        d.innerHTML = '<b>Segurança da senha: <font color=\'orange\'> MEDIA</font></b>';
    } else {
        d.innerHTML = '<b>Segurança da senha: <font color=\'red\'> BAIXA</font></b>';
    }
}
}
}
}
}
</script>

```

```

<body>
<h2>Teste de Segurança de Senha (JavaScript)</h2>
<form name=frm>
Login &nbsp;&nbsp;&nbsp;<input name="login"><br>
Senha <input type=password name=senha onKeyPress="TestaSenha(senha.value)"><div
id="seguranca"></div><br>
<input type=submit onClick="TestaSenha(senha.value)" value="Acessar">
</form>
<pre>

```

Teste de Segurança da senha em JavaScript

Autor: André Lourenço Pedroso

Alguns de vocês devem ter visto no Hotmail(tm), por exemplo, um recurso onde é feito um teste da senha, mostrando o seu nível de segurança.

Para aqueles que acharam esse recurso interessante, mostro nesse pequeno artigo um exemplo em JavaScript.

Os testes seguem a seguinte lógica:

- Baixa segurança - Senha que contem um tipo de caracter.
- Média segurança - Senha que tenha mais de quatro digitos e contenha no mínimo dois tipos de caracteres.
- Alta segurança - Senha que tenha mais de sete digitos e contenha no mínimo três tipos de caracteres diferentes.

Dica recebida da Dicas-L (<http://www.dicas-l.com.br>).

```

</pre>

```

```

</body>

```

13) E se o site foi invadido?

Devemos sempre prevenir. Mas e se acontecer de o site ser invadido, o que fazer?

- Mantenha o site offline para evitar outros ataques
- Baixe a última versão do Joomla
- Notifique o suporte do servidor e trabalhe com ele para fazer a limpeza do site e para ter certeza de que não ficou nenhum back door no site.
- Visite novamente o site das vulnerabilidades e veja se você ainda tem alguma extensão vulnerável
- Mude todas as senhas e se possível logins: cPanel, mysql, FTP, joomla Super Admin, etc.
- Substitua todos os templates e arquivos por cópias limpas
- Verifique atentamente os arquivos de log
- Verifique o cron se tem alguma entrada estranha
- Preferivelmente remova todo o conteúdo adicionado e todos os bancos e todos os e-mails para criar tudo novo
- Restaure com backups bem antes do ataque
- Confira as permissões de todos os arquivos. Nunca use 777, somente 644 para arquivos e 755 para diretórios
- Caso tenha acesso via SSH execute os comandos a seguir para sanear os arquivos:

```
find /home/joao03/public_html/site -type f -exec chmod 644 {} \;
```

```
find /home/joao03/public_html/site -type d -exec chmod 755 {} \;
```

Quando temos um site invadido, existe uma grande chance do invasor ter deixado backdoors para poder voltar depois. A procura pelos backdoors é algo demorado e trabalhoso, inclusive sem garantia, por isso é mais prudente remover tudo e instalar do zero, sem contar a hipótese de mudar de servidor, caso suspeite da fragilidade da sua hospedagem.

Para checar os arquivos recentemente alterados no sistema

```
find \public_html -ctime -1
```

Para proteger diretórios que precisam de permissão 777 ou por default

Crie um .htaccess no diretório images com:

```
# secure directory by disabling script execution
```

```
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi
```

```
Options -ExecCGI
```

Varrer Servidor

Faça uma varredura em todos os sites/máquinas a procura de malware, virus, trojans, spyware, etc.

* Opções disponíveis:

- ENOD32 from eSet
- Spybot Search and Destroy
- Malwarebytes
- Microsoft Malicious Software Removal Tool
- Linux AntiVirus boot cd

* Considere o Ultimate Boot CD para Windows

O que é uma extensão vulnerável?

- É uma extensão que contém ou contribui para uma vulnerabilidade de segurança.
- Quanto mais complexo for seu código maior a chance de ser vulnerável.
- As que lêem e escrevem arquivos
- Se não valida entradas de usuários
- Usam path explícito
- Permite acesso direto pela URL
- Permite inclusão de arquivos remotos
- Permite SQL injections
- Permite XSS
- Permite muito acesso a banco para usuários sem privilégios

Extensões vulneráveis são, não necessariamente extensões com código malfeito.

Projetos ativos geralmente lançam novas versões de suas extensões com as alterações.

Por estas razões é importante:

- Conhecer o número da versão de todas as extensões instaladas
- Use somente a última versão estável das extensões
- Desinstale e remova completamente todos os arquivos de extensões inseguras

Caso a última versão estável tenha sido lançada há um ano ou mais considere o projeto abandonado. Não instale extensões antigas.

[Fórum de Ajuda do Google para Webmasters](#)

- [.Aprenda com outros usuários](#)
- [.Otimização do mecanismo de pesquisa](#)
Aprimore o desempenho do seu site em pesquisas [PDF]
- [.Ferramentas de terceiros para Sitemaps](#)
Ferramentas para a criação de Sitemaps

14) Checklist de Segurança para Joomla

Listagem rápida para checar em sites antes de serem colocados em produção:

- Efetue um backup completo de todos os arquivos e do banco e restaure localmente
- Ativar URLs amigáveis e mod_rewrite
- Mover configuration.php para fora do public_html, usando:

```
require_once( dirname( __FILE__ ) . '/../../portal.cfg' );
```
- Bloquear cadastro de usuários pelo site caso não tenha necessidade: Configuração Global - Sistema - Permitir Cadastro de Usuários - Não
- Alterar metatags em Configuração Global - Configurações de Meta Dados (Trocar Joomla por outra palavra)
- Adicionar a tag <head> do template (para ocultar na origem do código HTML):

```
<?php $this->setGenerator('Ribafs - Desenvolvimento Web'); ?>
```
- Instalar algumas extensões:
 - AdminTools
 - Plugin osolcapcha
 - com_encrypt
 - jHackGuard

Usar ferramentas:

joomscan

Ativar o cache

Otimizar as tabelas do banco no phpmyadmin

Sanear permissões de arquivos:

Alterar todos os arquivos recursivamente para 644 e todas as pastas para 755 com. Veja em Permissões como fazer isso.

Depois criar algumas exceções...

configuration.php – 400

index.php do site – 400

index.php do template padrão – 400

Permissões de pastas:

includes e libraries – 500

Remover templates não usados e outras extensões também.

Adicionar ao .htaccess:

```
# Block out any script trying to set a mosConfig value through the URL
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|%3D) [OR]
# Block out any script trying to base64_encode crap to send via URL
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|[\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|[\%[0-9A-Z]{0,2})
# Send all blocked request to homepage with 403 Forbidden error!
RewriteRule ^(.*)$ index.php [F,L]
```

Adicionar ao configuration.php:

```
ini_set('extension', 'sourceguardian.so');
ini_set('session.save_path', '/home/ribafs03/public_html/tmp');
ini_set('cgi.force_redirect', 1);
ini_set('allow_url_fopen', 0);
ini_set('display_errors', 0);
ini_set('allow_url_include', 0);
ini_set('expose_php', 0);
ini_set('magic_quotes_gpc', 0);
ini_set('post_max_size', '262144');
ini_set('upload_max_filesize', '262144');
ini_set('upload_tmp_dir', '/home/joao/public_html/tmp');

$disfunctions = 'proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec, system,
shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note, apache_setenv,
disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice,
symlink, phpinfo';

ini_set('disable_functions', $disfunctions);

// Verificar se seu servidor duporta e ajustar as versões
ini_set('zend_extension', '/usr/local/php56/lib/php/extensions/ioncube.so');
```

```

ini_set('zend_extension_manager.optimizer=', '/usr/local/Zend/lib/Optimizer-3.3.3');
ini_set('zend_extension_manager.optimizer_ts', '/usr/local/Zend/lib/Optimizer_TS-3.3.3');
ini_set('zend_optimizer.version', '3.3.3');
ini_set('zend_extension', '/usr/local/Zend/lib/ZendExtensionManager.so');
ini_set('zend_extension_ts', '/usr/local/Zend/lib/ZendExtensionManager_TS.so');

```

Adicionar ao php.ini (alternativa):

Este é para o caso do servidor permitir um php.ini no raiz que será visto por todas as partas recursivamente.

```

extension=sourceguardian.so
session.save_path = "/home/ribafs03/public_html/tmp"
cgi.force_redirect = 1
allow_url_fopen= 0
display_errors = 0
expose_php = 0
magic_quotes_gpc = 0
memory_limit = 8388608
#open_basedir = 1
post_max_size = 262144
upload_max_filesize = 262144
upload_tmp_dir = "/home/ribafs03/public_html/tmp"
disable_functions = proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec,
system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note, apache_setenv,
disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice,
symlink, phpinfo
zend_extension=/usr/local/php52/lib/php/extensions/ioncube.so
zend_extension_manager.optimizer=/usr/local/Zend/lib/Optimizer-3.3.3
zend_extension_manager.optimizer_ts=/usr/local/Zend/lib/Optimizer_TS-3.3.3
zend_optimizer.version=3.3.3
zend_extension=/usr/local/Zend/lib/ZendExtensionManager.so
zend_extension_ts=/usr/local/Zend/lib/ZendExtensionManager_TS.so

```

Vários dos recursos acima você precisará confirmar com o suporte do seu servidor para ver se estão disponíveis.

15) Testes de Vulnerabilidade e Simulações de Ataque

Também é importante estar constantemente e em especial ao concluir o site, efetuando testes de vulnerabilidade para saber como está a segurança do site. Saber se ele está vulnerável ou não a certos tipos de ataques.

Existem boas ferramentas que colaboram com essa tarefa. Veja algumas extensões para Firefox e também o **w3af**.

16 – Permissões do Sistema de Arquivos

Mudando recursivamente as permissões do diretório atual:

```
find . -type f -exec chmod 644 {} \;  
find . -type d -exec chmod 755 {} \;
```

De outro diretório

```
sudo find /var/www/teste/ -type f -exec chmod 0644 {} \;  
sudo find /var/www/teste/ -type d -exec chmod 0755 {} \;
```

Isso mudará as permissões recursivamente de todos os diretórios para 755 e de todos os arquivos para 644.

Sempre antes de enviar seus arquivos para o servidor é recomendado executar estes comandos nos mesmos.

Permissões

Nunca mantenha permissões acima de 755 para diretórios e acima de 644 para arquivos. Caso sua aplicação precise disso por alguma circunstância, procure colocar estes arquivos abaixo do `public_html` ou então crie um arquivo `.htaccess`, com apenas:

```
deny from all
```

O `configuration.php` deve ficar com 400. Altere para 600 somente quando precisar alterá-lo e depois retorne para 400.

ALERTA

Nunca deixe brechas nos seus arquivos com permissões que terminem em 6 ou 7 (766 ou 777). Idealmente sempre abrigue seus sites em servidores de hospedagem que não necessitem que você altere as permissões para 777 ao instalar o Joomla ou alguma extensão. Os bons servidores permitem a instalação de forma indolor, nada de 777. Nestes as permissões ficam como 755 e podemos instalar sem problema.

Dê atenção especial aos `index.php` do site, da administração e do template, mudando suas permissões para algo como 444 ou 400, para restringir o acesso ao público. Só isso em si já evitará alguns aborrecimentos e corrigirá outros.

17) Links úteis/Referências

Lista de Vulnerabilidade em Extensões

http://docs.joomla.org/Vulnerable_Extensions_List

Exploits em Geral

<http://www.milw0rm.com/>

Diagnóstico dos arquivos do Joomla

<http://extensions.joomla.org/extensions/tools/security-tools/1146?qh=YToxOntpOjA7czoxMToiZGlhZ25vc3RpY3MiO30%3D>

RSS do Joomla

http://www.joomla.org/index.php?option=com_rss_xtd&feed=RSS2.0&type=com_frontpage&Itemid=1

RSS Segurança Joomla

<http://feeds.joomla.org/JoomlaSecurityNews>

Scanner de Vulnerabilidades Online

<https://www.joomlascan.com/> - Apenas varre e diz se tem ou não, mas não limpa nem diz qual, somente quando pagamos

<http://linkscanner.explabs.com/linkscanner/default.aspx>

Scanners offline

<http://sectools.org/web-scanners.html>

Aplicativo de Scanner

<http://www.acunetix.com/>

Checagem/Teste de DNS Online

<http://dnscheck.iis.se/>

<http://www.squish.net/dnscheck/>

Listas negras

Consulte se seu site está nas listas negras

<http://rbld.org>

mandville PhilD fw116 JeffChannell dynamicnet

http://docs.joomla.org/Security_Checklist_1_-_Getting_Started

Programando com Segurança:

<http://www.ibm.com/developerworks/opensource/tutorials/os-php-lockdown/os-php-lockdown-pdf.pdf>

<http://www.ibm.com/developerworks/br/library/os-php-secure-apps/>

<http://phpsecurity.org/contents#ch01>

Consultei vários outros sites e em especial os dois abaixo:

Documentação do Joomla

<http://docs.joomla.org/Developers#Security>

A recomendação para proteger com permissão de 444 alguns arquivos importantes recebi na lista AjudaJoomla

<http://br.groups.yahoo.com/group/ajudajoomla/> do colega DJ.

KnowledgeBase do siteground.com.

<http://kb.siteground.com>

Ribamar FS - <http://ribafs.org>