Security Essentials

Choosing the right hosting company is essential to ensuring the smooth operation of your Joomla site.

We're naturally based and would recommend you **host it with us** . We currently host over 1500 Joomla websites for all of our clients without any server issues whatsoever. Our servers are setup to facilitate both **auto-installations** and **manual installations** of Joomla. If you host your site with us, you can basically ignore the information below.

Please take the time to read this entire document. Some of it may not make sense to you now, but it will prove important as you progress with **building websites using Joomla**.

This information is from the Joomla Documentation Wiki. Visit the following URL to learn more about securing your Joomla site: ***http://docs.joomla.org/Security_Checklist_1_-_Getting_Started***

Security matters

Internet security is a fast moving challenge and ever present threat. There is no one right way to secure a website, and all security methods are subject to instant obsolescence, incremental improvement, constant revision. All public facing website are open to constant attack. Are you willing and able to invest the time it takes to administer a dynamic, 24x7, world-accessible, database-driven, interactive, user-authenticated website? Do you have the time and resources to respond to the constant flow of new Internet security issues?

The **Top 10 Stupidest Administrator Tricks** is a comic/tragic look at what can go wrong. Don't learn these tricks the hard way! Depending on your own experience, reading the *Stupidest Tricks* will either make you laugh or cry. Luckily, there are some well-established principles upon which to base your defensive plans. The following checklists point you toward current best practices for Joomla security.

The most important guidelines

These checklists are long and growing because the full plot is thick, complex, and expanding, but don't despair! Here are a few essential guidelines for securing any website. Following them will protect you from most catastrophes.

1. **Backup early and often:**
2. **Update early and often:**
3. **Use a secure host**
4. **Use the community**

The bad news

1. **There is no perfect security on the Web!**
2. **There's no one right way!**
3. **There's no substitute for experience!**

Choose a Qualified Hosting Provider

The most important decision

Probably no decision is more critical to site security than the choice of hosts and servers. However, due to the wide variety of hosting options and configurations, it's not possible to

provide a complete list for all situations. Check this unbiased **list of recommended hosts** who fully meet the security requirements of a typical Joomla site. (
FAQ
)

Shared server risks

If you are on a tight budget and your site does not process highly confidential data, you can probably get by with a shared server, but you must understand the unavoidable risks. Most of the tips listed below are appropriate for securing sites on shared server environments.

Avoid sloppy server configurations

For a real eye-opener, **read this report** on thousands of sites that allowed Google to index the results of phpinfo(). Don't make this mistake on your site! The report includes alarming statistics on the percentage of sites that use depreciated settings such as register_globals ON or that don't have open_basedir set at all: By the way, if *phpini* and
register_globals
are unfamiliar terms you are probably not ready to securely manage your own site.

Configuring Apache

Use Apache .htaccess

See also ***.htaccess examples***

Block typical exploit attempts with local Apache *.htaccess* files. This option is not enabled on all servers. Check with your host if you run into problems. Using
.htaccess
, you can password protect sensitive directories, such as administrator, restrict access to sensitive directories by IP Address, and depending on your server's configuration, you may be able to increase security by switching from PHP4 to PHP5.

Joomla ships with a **preconfigured .htaccess** file, but *you* need to choose to use it. The file is called htaccess.txt; to use it rename it to .htaccess and place it in the root of your webpage.

Consider following the "Least Privilege" principle for running PHP using tools such as PHPsuExec, php_suexec or suPHP. (Note: These are advanced methods that require agreement and coordination with your hosting provider. Such options are enabled or disabled on a server-wide basis and are not individually adjustable on shared servers.)

Use Apache mod_security

Configure Apache mod_security and mod_rewrite filters to block PHP attacks. See **Google search for mod_security**
and
Google search for mod_rewrite
. (Note: These are advanced methods that usually require agreement and coordination with your hosting provider. Such options are enabled or disabled on a server-wide basis and are not individually adjustable on shared servers.)

Configuring MySQL

Secure the database

Be sure MySQL accounts are set with limited access. The initial install of MySQL is insecure and careful configuration is required. (See the **MySQL Manuals**) Note: This item applies only to those administering their own servers, such as dedicated servers. Users of shared servers are dependent on their hosting provider to set proper database security.)

Configuring PHP

Understand how PHP works

Understand how to work with the php.ini file, and how PHP configurations are controlled. Study the **Official List of php.ini Directives** at **http://www.php.net** , and the well-documented default php.ini file included with every PHP install. Here is the **latest default php.ini file** on the official PHP site.

Use PHP5

Currently, both PHP4 and PHP5 are maintained, and both are often available on servers. Before PHP4 becomes obsolete, upgrade your custom scripts to PHP5. Don't worry about core Joomla code; all current versions are PHP5 compatible. (See **PHP News**)

Use local php.ini files

On shared servers you can't edit the main php.ini file, but you may be able to add custom, local

php.ini files. If so, you'll need to copy the php.ini files to every sub-directory that requires custom settings. Luckily a **set of scripts at B & T Scripts and Tips** can do the hard work for you.

There are a few important things to keep in mind.

1. Local *php.ini* files ***only*** have an effect if your server is configured to use them. This includes a *php.ini* file in your *http_root* directory. You can test whether or not these file affect your site by setting an obvious directive in the local *php.ini* file to see if it affects your site.

2. Local *php.ini* files only effect *.php* files that are located within the same directory (or included() or required() from those files). This means that there are normally only two Joomla! directories in which you would want to place a *php.ini* file. They are your *http_root* (your actual directory name may vary), which is where Joomla's Front-end *index.php* file is located, and the Joomla! *administrator* directory, which is where the Back-end administrator *index.php* file is located. Other directories that don't have files called via the Web do not need local *php.ini* files.

3. If you have a *php.ini* file in every directory, some script probably did this for you. If you didn't intend it to happen, you probably should root them out, but given #2 above, you probably only have to panic about the *php.ini* files in *http_root* and the *administrator* directories.

Use PHP disable_functions

Use *disable_functions* to disable dangerous PHP functions that are not needed by your site. Here is a typical setup for a Joomla! site:

disable_functions = show_source, system, shell_exec, passthru, exec, phpinfo, popen, proc_open

Use PHP open_basedir

open_basedir should be enabled and correctly configured. This directive limits the files that can be opened by PHP to the specified directory-tree. This directive is NOT affected by whether Safe Mode is ON or OFF.

The restriction specified with open_basedir is a prefix, not a directory name. This means that *open_basedir = /dir/incl*
allows access to
/dir/include
and
/dir/incls
if they exist. To restrict access to only the specified directory, end with a slash. For more information, see
PHP Security and Safe Mode Configuration Directives
.

open_basedir = /home/users/you/public_html

In some system configurations, at least with PHP 4.4.8, the use of the trailing slash to restrict the access to only the specified directory may cause Joomla to warn*JFolder::create: Infinite loop detected* when saving the Back-End Global Configuration. This warning is triggered because PHP
file_exists()
function fails, for example, when asked if
/home/user/public_html/joomla_demo
exists and
open_basedir
is set to
/home/user/public_html/joomla_demo/
(see the trailing slash).

Additionally, if *open_basedir* is set it may be necessary to set PHP *upload_tmp_dir* configuration directive to a path that falls within the scope of
open_basedir
or, alternatively, add the
upload_tmp_dir
path to
open_basedir
using the appropriate path separator for the host system.

```
open_basedir = /home/users/you/public_html:/tmp
```

PHP will use the system's temporary directory when *upload_tmp_dir* is not set or when it is set but the directory does not exist, therefore it may be necessary to add it to
open_basedir
as above to avoid uploading errors within Joomla.

Adjust magic_quotes_gpc

Adjust the *magic_quotes_gpc* directive as needed for your site. The recommended setting for Joomla! 1.0.x is ON to protect against poorly-written third-party extensions. The safest method is to turn *magic_quotes_gpc* off and avoid all poorly-written extensions, period.

Joomla! 1.5 ignores this setting and works fine either way.

For more information, see either **Magic quotes and security** or **PHP Manual, Chapter 31. Magic Quotes** .

```
magic_quotes_gpc = 1
```

Don't use PHP safe_mode

Avoid the use of PHP safe_mode. This is a valid but incomplete solution to a deeper problem and provides a false sense of security. See the official PHP site for an explanation of this issue.

safe_mode = 0

Don't use PHP register_globals

Automatically registering global variables was probably one of the dumbest decisions the developers of PHP made. This directive determines whether or not to register the EGPCS (Environment, GET, POST, Cookie, Server) variables as global variables where they become immediately available to all PHP scripts, and where they can easily overwrite your own variable if you're not careful. Luckily, the PHP developers long since realized the mistake and have depreciated this 'feature'.

If your site is on a shared server with a hosting provider that insists *register_globals* must be on, you should be very worried. Although you can often turn register_globals off for your own site with a local php.ini file, this adds little security as other sites on the same server remain vulnerable to attacks which can then launch attacks against your site from within the server. For more information, see
ZEND Chapter 29. Using Register Globals
.

register_globals = 0

Don't use PHP allow_url_fopen

Don't use PHP *allow_url_fopen*. This option enables the URL-aware fopen wrappers that enable accessing URL object like files. Default wrappers are provided for the access of remote files using the ftp or http protocol, some extensions like zlib may register additional wrappers. Note: This can only be set in php.ini due to security reasons.

allow_url_fopen = 0

Setup a backup and recovery process

The most important rule:

Thou shalt at all time be able to return your site to a previous working state through regular use of a strong, off-site backup and recovery process. Be sure your backup and recovery process is in place and tested BEFORE you go live. This is the single best way (and often the only way) to recover from such inevitable catastrophes as:

1. A compromised/cracked site.
2. Broken site due to a faulty upgrade.
3. Hardware failure, such as dead hard drives, power failures, server theft, etc.
4. Authoritarian government intervention. (More common than some think.)
5. Needing to quickly relocate to a new server or hosting provider.

Site Administration

Use well-formed passwords

Change passwords regularly and keep them unique. A strong password has a random combination of letters, numbers, or symbols. Avoid using single names or words found in a dictionary. Never use the names of your relatives, pets, etc. Search the forums for a script

supplied by Wizzie that automatically changes passwords. This is a great tool for administrators or multiple sites. There are numerous handy websites that have **strong password generators** .

Follow a password leveling scheme

Most users may not need more than three levels of passwords and webmasters no more than five. Each level must be completely unrelated to the others in terms of which usernames and passwords are used. Learn how to do this: **How do you setup a powerful password scheme?**

Maintain a strong site backup process

Never rely on others' backups. Take responsibility for your backup procedures. Many ISPs state in their contract that you can not rely solely on their backups.

Monitor crack attempts

VPS and dedicated server users can run TripWire or SAMHAIN. These applications provide exhaustive file checking and reporting functionality, and can be installed in a stealthy manner to help protect themselves in the event of a serious infiltration. (Note: Users of shared servers can not use this technique.)

Perform automated intrusion detection

Use an Intrusion Prevention/Detection Systems to block/alert on malicious HTTP requests.

- **Google search**

Perform manual intrusion detection

Regularly check raw logs for suspicious activity. Don't rely on summaries and graphs.

Stay current with security patches and upgrades

Apply vendor-released security patches ASAP.

- Review the **vulnerable extensions**

Proactively seek site vulnerabilities

Perform frequent web scanning.

- **Google Search**

Proactively seek SQL injections vulnerabilities

Use tools such as Paros Proxy for conducting automated SQL Injection tests against your PHP applications.

- **Google Search**
- **Wikipedia Article**

Use shell scripts to automate security tasks

Search the forums for these popular scripts:

- Joomla! Version Checking
- Joomla! Component/Module Version Checking
- Exploit Checking

Learn about security software

There is not a single tool that can protect your site. If there were, it would be so heavily targeted that it would probably become a liability.

Don't reinvent every wheel

Every now and then hire a professional Joomla! security consultant to review your configurations. Do you remember the adage, *"Anyone who acts as their own lawyer has a fool for a client."* The same goes for Web development. Don't expect to catch all of your own security mistakes.

Install official versions of Joomla!

To avoid breaking your site, search the forums for reports of incompatible extensions before upgrading to a new version of Joomla.

Upgrade to the **latest stable version of Joomla!** as soon as possible.

Download Joomla! from official sites only, such as **JoomlaCode.org** , and check the **MD5 hash** .

Use **Joomla Diagnostics** to ensure that all files were installed correctly. (Note: the version of Joomla Diagnostics made for the initial release of 1.5 does not work for 1.5.3.)

Change the default administrator username

Change the user name of the default admin user. This simple step effectively increases the security of this critical account 50% by modifying one of the two variables attackers must know to gain access. The password is the other variable. Change it early and often. **(FAQ)**

Protect directories and files

Increase the security of the critical *configuration.php* file by moving it outside of the *public_html* directory. For more information visit
(
FAQ
)

Ensure that all configurable paths to writable or uploadable directories (document repositories, image galleries, caches) are outside of public_html. Check third party extensions such as DOCMan and Gallery2 for editable paths to writable directories.

In the Back-End Global Configuration, change the log path. Some extensions use the built in JLog class. This will, by default write logs to **http://yousite/logs** . Change this to a place that a casual browser cannot find (and don't pick /tmp/), or lock it down with http authentication. Because we are dealing Open Source software, attackers can read the code of third-party extensions and may be able to guess log file names.

In the Back-End Global Configuration, change the temp folder path.

If the log and temp paths are changed and PHP *open_basedir* configuration directive is set, make sure that the new paths fall within the scope of
open_basedir
.

There is currently no easy way to move the Joomla! /image and /media directories. This is because thousands of third party extensions expect to find these important directories at the current location. The best plan is to make sure open_basedir is properly set for all the user accounts on your server. Check with your host if unsure.

Adjust file and directory permissions

This option no longer appears in Joomla. On Older versions of Joomla : Once your site is configured and stable, write-protect critical directories and files by changing directory permissions to 755, and file permissions to 644. There is a feature in Site --> Global Configuration --> Server to set all folder and file permissions at once. Test third party extensions afterwards, and carefully review the code of any extension that has trouble with such settings. Note: Depending on your server's permissions, you may need to temporarily reset to more open permissions when installing more extensions with the Joomla! installer.
This option no longer appears in Joomla.
but is included for historical purposes.

Remove unneeded files

Remove all design templates not needed by your site. Never put security logic into template files.

Disable the XML-RPC server if you don't need it.

Clean up after installs. The installation process will require you to delete the installation directory and all its contents. Do this; do not simply rename it. If you upload files to your site as compressed archives (xxxx.zip for example), don't forget to remove the compressed file. Check the /temp/ directory as temporary files may remain there after a failed installation attempt.

In general, do not leave any unneeded files (compressed or otherwise) on a public server. Each unused (and perhaps long forgotten) file is a potential security hole.

Turn Register Globals Emulation OFF

Turn Joomla's Register Globals Emulation OFF. Although this setting is somewhat safer than PHP register_globals, you are much better off avoiding such settings all together (as well as any applications that require them). On pre-1.0.13 versions of Joomla, this setting is found in the globals.php file. As of version 1.0.13, it can be turned off in the Back-end, under Global Settings.

Joomla 1.5 and greater, does not use register globals, and in fact has smart code to defeat this setting even if it's turned on at the PHP level. Note that although this makes Joomla itself safer, any server with register globals turned on is potentially vulnerable. Any shared server with register globals turned on is more than likely a sitting duck. Any hosting provider that insists register globals should be turned on is ignorant, incompetent, or worse. Was that blunt enough?

For more information on register_globals, please see **Security Checklist: PHP: register_globals** .

Installing Joomla! Extensions

Backup before installing

Before installing extensions, always backup your site's files and database. This follows a very

basic principle:

Thou shalt at all times be able to return your site to a previous working state.

Therefore, it's smart to set up a simple and fast backup script to automate this task. If you don't set up an easy process in advance, you'll be sorely tempted to do a quick upgrade without backing up first. This very understandable tendency is however one of the chief causes of premature hair loss, sudden career changes, and even death.

Check for extension vulnerabilities

Most security vulnerabilities are caused by third party extensions. Before installing extensions, check the Official List of Vulnerable 3rd Party/Non Joomla! Extensions. There's an entire forum dedicated to vulnerable third part extensions. Subscribe to it.

Download from trusted sites

The fully qualified and official definition of a "trusted site" is one that **YOU** trust.

User beware! Check the code quality

Third party extensions come in all flavors of quality and age. Although Joomla! coding standards exist, third party developers are not required to follow them. Extensions listed on the official Joomla! site are not reviewed for compliance, however if verified vulnerabilities are reported, they will be removed from the list until they are fixed.

Test, test, test...

Test all extensions on a development site before installing on a production site. Then test on the production site. Don't forget to check the logs for runtime errors and warnings.

Remove junk files

Remove all unused extensions and double check that related folders and files were actually removed by uninstall scripts. Note that during uninstall, many third party extensions will leave related files on your site, and related database tables complete with data. This is either a feature or a bug depending on your point of view. Any files left on your server remain accessible from the Web via direct URLs, such as **http://yousite.com/modules/bad_module** .

Avoid encrypted code

Joomla is (and dispite disinformation campaigns, always has been) a GNU GPL project. This means that all extensions to Joomla must also be free (as in freedom) and open (as in readable code). Encrypted code may be safe, but you can't determine this for yourself, and so you must trust the developers. Using others' encrypted code puts you back in the world of proprietary software where you must wait for security patches from the developer, hoping that attackers don't find your site first before a fix is released.

You are often not free to modify, improve, or share encrypted code. These restrictions make encrypted code less valuable to the community as a whole, and reduce the overall viability of the Joomla project which depends on open sharing among all participants.

Of course, code that is not distributed to others is exempt from GNU GPL distribution requirements. Thus you can encrypt Joomla-related code your own servers providing you do not share it with others.

Additional Joomla! Hardening Tips and Tricks

Avoid shared servers if possible

For maximum security, avoid a shared server on which you don't know or can't trust all the other users or their code quality.

Use an SSL server

This more to do with secure payments and administration, and is not joomla core or server security, but have been included here for advice

SSL servers are currently the only way to securely process confidential transactions and secure user authentication. SSL works by encrypting all HTTP communications between the Web server and Web clients. Thus, even if a transmission is intercepted, it cannot be read.

Joomla! 1.0.x does not allow you to assign an SSL server to individual sub-directories. Search the forums for "Tommy Hack" for one way to deal with this. Joomla! 1.5 has greatly improved SSL options.

Use Apache's .htaccess

For an additional layer of password protection, you can use .htaccess to password protect critical directories. This is usually adequate for blocking the typical script kiddie, but be aware that .htaccess password protection alone is not a highly secure method. It MUST be combined with an SSL server for maximum protection. An SSL server is required for protecting your site from more sophisticated attacks, such as packet sniffing.

Switch to Joomla! 1.5

The most significant upgrade in Joomla!'s history includes powerful security and performance enhancements.

- **Joomla 1.5 Overview**
- **Joomla Downloads**

Add Joomla! Security Announcements to your site

The Joomla! Security Team supports and RSS feed that provides the latest Joomla security information. The following FAQ explains how to add this feed to your site.

- **How can I add the Joomla! Security Announcements Feed to the Admin Control Panel?**