Is there a difference between apache module vs cgi (concerning security)?

Asked 16 years, 3 months ago Modified 7 years, 4 months ago

Viewed 22k times Part of PHP Collective

19

E.g. Is it more secure to use `mod_php` instead of `php-cgi`? Or is it more secure to use `mod_perl` instead of traditional `cgi-scripts`?

I'm mainly interested in security concerns, but speed might be an issue if there are significant differences.

PHP php perl apache mod-perl mod-php

Share

Improve this question

Follow

edited Aug 16, 2017 at 11:36

Hinek

9,699 ● 12 ● 54 ● 75

asked Sep 16, 2008 at 22:49

Sarien

6,942 ● 6 ● 38 ● 56

2 I understand this question is old, but shouldn't it now go to **Server Fault**? – Stefan Marinov Mar 4, 2014 at 12:55

@smarinov, It should. But SO has a weird policy when it comes to old threads, which sums up in one word:

"lazyness". SO doesn't care. – Pacerier Mar 26, 2015 at 11:15

6 Answers

Sorted by: Highest score (default)

15 ✓

Security in what sense? Either way it really depends on what script is running and how well it is written. Too many scripts these days are half-assed and do not properly do input validation.

I personally prefer FastCGI to mod_php since if a FastCGI process dies a new one will get spawned, whereas I have seen mod_php kill the entirety of Apache.

As for security, with FastCGI you could technically run the php process under a different user from the default web servers user.

On a seperate note, if you are using Apache's new worker threading support you will want to make sure that you are not using mod_php as some of the extensions are not thread safe and will cause race conditions.

Share Improve this answer Follow

answered Sep 16, 2008 at 23:02

X-Istence

16.6k ● 6 ● 61 ● 75

I have been with mod_php for a year now and I have never seen a child process kill the entirety of Apache. – wlf Oct 12, 2013 at 22:10

1 @wlf, 1 year is barely long. 3 years now. So is it still alive? – Pacerier Mar 26, 2015 at 11:16

8

If you run your own server go the module way, it's somewhat faster. If you're on a shared server the decision has already been taken for you, usually on the CGI side. The reason for this are filesystem permissions. PHP as a module runs with the permissions of the http server (usually 'apache') and unless you can chmod your scripts to that user you have to chmod them to 777 - world readable. This means, alas, that your server neighbour can take a look at them - think of where you store the database access password. Most shared servers have solved this using stuff like phpsuexec and such, which run scripts with the permissions of the script owner, so you can (must) have your code chmoded to 644. Phpsuexec runs only with PHP as CGI - that's more or less all, it's just a local machine thing - makes no difference to the world at large.

Share Improve this answer Follow

answered Sep 16, 2008 at 23:19

djn

3,948 ● 24 ● 21

CGI's setup is more secure,. but slower. But also see stackoverflow.com/a/10439720/632951 – Pacerier Apr 3, 2015 at 13:01

5

Most security holes occur due to lousy programming in the script itself, so it's really kind of moot if they are ran as cgi or in modules. That said, apache modules can potentially crash the whole webserver (especially if using a threaded MPM) and mod_php is kind of famous for it.

cgi will be slower, but nowadays there are solutions to that, mainly FastCGI and friends.

What is your threat model?

Share Improve this answer Follow

answered Sep 16, 2008 at 22:52

Vinko Vrsalovic
340k ● 55 ● 340 ● 373

If the script is weak, it doesn't mean that we stop securing the server. Both needs to be strong. Threat model is to prevent file access to unauthorized folks. Access must only be done via PHP scripts. – Pacerier Mar 26, 2015 at 11:17

4

From the PHP install.txt doc for PHP 5.2.6:

Server modules provide significantly better performance and additional functionality compared to the CGI binary.

For IIS/PWS:

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security section to learn how to defend yourself from those attacks.

Share Improve this answer Follow

answered Sep 16, 2008 at 23:16

Scott Swezey
2,147 ● 2 ● 18 ● 28

3

A module such as mod_php or FastCGI is incredibly faster than plain CGI.. just don't do CGI. As others have said, the PHP program itself is the greatest security threat, but ignoring that there is one other consideration, on shared hosts.

If your script is on a shared host with other php programs and the host is not running in safe mode, then it is likely that all server processes are running as the same user. This could mean that any other php script can read your own, including database passwords. So be sure to investigate the server configuration to be sure your code is not readable to others.

Even if you control your own hosting, keep in mind that another hacked web application on the server could be a conduit into others.

Share Improve this answer Follow

answered Sep 17, 2008 at 2:27

DGM
27k ● 7 ● 64 ● 82

Using a builtin module is definitely going to be faster than using CGI. The security implications depend on the

2

configuration. In the default configuration they are pretty much the same, but cgi allows some more secure configurations that builtin modules can't provide, specially in the context of shared hosting. What exactly do you want to secure yourself against?

Share Improve this answer Follow

answered Sep 16, 2008 at 22:52

Leon Timmermans
30.2k ● 2 ● 64 ● 110

1 I don't have any particular threat in mind. But there are multiple pages hosted on the same machine. – Sarien Sep 16, 2008 at 22:58

You should decide what you want to protect yourself from first. Are those pages owned by the same person/institution? IF not, you should consider any of the various solutions to run different php scripts as different users, for instance (suphp, using virtual hosts and reverse proxying, and others) – Vinko Vrsalovic Sep 16, 2008 at 23:09

@VinkoVrsalovic, The usual threat model. Prevent prying eyes from reading the PHP's source code. – Pacerier Apr 3, 2015 at 13:03