

# Deny from all in subdirectory htaccess not overriding file rules in root htaccess

Asked 8 years, 9 months ago Modified 9 months ago Viewed 45k times



12



1



I've got a situation where I'm trying to deny access to all files in a subdirectory of my website. I have added an htaccess file to this subdirectory and added a deny from all directive to it. However, I also have an htaccess file in the site root, which allows specific file types, and it seems like these file types are still accessible in the subdirectory even though I no longer want them to be. I have a workaround for this (see below), but I feel like there must be a better way. Here are my two htaccess files:

Root .htaccess

```
# Deny access to everything by default
Order Deny,Allow
deny from all

# Allow access to html files
<Files *.html>
    allow from all
</Files>
```

Subdirectory .htaccess

```
# Deny access to everything
deny from all
```

**Workaround:**

Subdirectory .htaccess

```
# Deny access to everything
Order Allow,Deny
deny from all
<Files *.*>
    deny from all
</Files>
```

This does what I want, but I feel like there should be a way to make the deny from all statement work by itself. Does anyone know how?

[apache](#) [.htaccess](#)

**Join Stack Overflow** to find the best answer to your technical question, help others answer theirs.

Sign up





Uwe Keim

38.1k 55 168 279



FrolickingFerret

281 1 2 12

3 Answers

Sorted by:

Highest score (default)



You can have your Root .htaccess like this

17



```
# Deny access to everything by default
Order Deny,Allow
deny from all
```



```
# Allow access to html files
<Files *.html>
    allow from all
</Files>
```



```
# Deny access to sub directory
<Files subdirectory/*>
    deny from all
</Files>
```

There is no need for a separate .htaccess in the sub directory.

You are allowing access to all html files in your .htaccess in the Root Directory and not denying it anywhere in the sub directory in the first case. Apache parses all your rules and uses the last matching rule, unlike firewalls (which uses the first rule match hit). The global rules are read first and the the specific rules later.

Share Improve this answer Follow

edited Jul 12, 2013 at 9:57

answered Jul 12, 2013 at 9:47



bansi

52.3k 6 38 48

Ah, thank you. I see now where I went wrong. I had mistakenly believed that all the rules in the root htaccess would be evaluated followed by all of the rules in the subdirectory htaccess. I did not realize that in fact there is a merge of the components in the two files, the result of which is applied in a single pass.

– FrolickingFerret Jul 12, 2013 at 15:56



Order of directives in apache is really not obvious.

7

You have a complete description of it on the documentation "[How the sections are merged](#)".

Here is an excerpt:

**Join Stack Overflow** to find the best answer to your technical question, help others answer theirs.

Sign up



- <Directory> (except regular expressions) and .htaccess done simultaneously (with .htaccess, if allowed, overriding <Directory> )
- <DirectoryMatch> (and <Directory ~> )
- <Files> and <FilesMatch> done simultaneously
- <Location> and <LocationMatch> done simultaneously
- <If>

So what happens is that your <File> directive is handled **after** the Directory ones (as a .htaccess is in fact a Directory directive for the current directory).

It works in your given exemple as theses files directives are in fact nested in the .htaccess Directory directives, and the second File directive is applied after the parent directory one.

You cannot use a <FileMatch> directive in the parent, where files from the subdirectory would be excluded, as fileMatch is only working on the file name, and not on the file path. But you could maybe try with a LocationMatch, but it would maybe end in a quite complex one, to forbid also location hacks with dots.

In fact the solution I would use here is a RedirectMatch in the parent folder:

```
RedirectMatch 403 ^.*\.html$
```

Share Improve this answer Follow

answered Jul 12, 2013 at 13:38



regilero

28.7k 6 58 97

Thanks for the answer! I took a look at the link you supplied, and along with your answer I believe that I now understand the problem. However, I'm not really sure how the RedirectMatch you provide will deny access to all files in that particular subfolder... – FrolickingFerret Jul 12, 2013 at 15:59

2 The actual form would prevent access to all html files. I wrote that a long term ago, but I think the right formula is more something like: in all subdirectories, but something like RedirectMatch 403 ^.\*subdirectory/\*\.html\$ – regilero Aug 20, 2013 at 8:04



0

After finding this thread I couldn't get it to work for a while, then I found out why, and thus hence my answer, to (hopefully) help people with the same problem in the future.

I copied @bansi's answer, like this:



#Deny access to everything by default.

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up X

```
# Allow access to the root index.php.  
<Files index.php>  
    allow from all  
</Files>  
  
#allow access to the other .PHP files  
<Files pages/*.php>  
    allow from all  
</Files>
```

And it did work partly, I couldn't access any javascript files in my js folder, or .PHP files that are not in the pages folder. But I couldn't access my index.php file in the root folder, even though it is in the rules. the problem turned out to be because I also rewrote /index.php to -> /home like this:

```
RewriteRule ^home$ index.php [L]
```

the solution is to add this:

```
<Files home>  
    allow from all  
</Files>
```

Share Improve this answer Follow

answered Jul 7, 2021 at 14:10



Niels

107 7