

Forum:[Hack Forums](#)

Thread: [\[tutorial\] path traversal attack \(directory traversal\)](#)

go for it - December 11, 2016, 5:54 pm

disclaimer: i'm a white hat security researcher, this tutorial is for educational purposes, Any actions and or activities related to the material contained within this Website is solely your responsibility, pla pla i'm board

path traversal attack aims to gain access to files and directories you should otherwise not have access to by manipulating file references either by using ../ or an absolute path, and is called dot dot slash, directory traversal, directory climbing and backtracking

lets give an example

a website with url of

example(dot)com/something/download.php?filename={param}

a legit use for ths url would be

example(dot)com/something/download.php?filename=report.txt

but by exploiting the engine used to parse those strings which is the same used to reference css sheets and such (ex. ../../public/css/main.css) we execute an attack using the same exploit, for the url to be

example(dot)com/something/download.php?filename=../../index.php

which will result into either a check validating our input and throwing an error or it will download the index.php we asked for

and when you can download the code source, you can search for the db credentials (usually stored in another file but you know how to download any file path you can find) and access to db or at least to get the code source and look for more exploits

now we've only scratched the surface, but the rabbit hole goes much deeper to include and combine url encoding, null bytes to bypass server's validation, inject it with cookie, absolute urls

how to test for that vulnerability? you can simply try any illegal character or imaginary path and see how the website behaves

dorks would include download.php?somestr={param} where "somestr" could be "filename, file, id, number, name, etc.."

how to protect yourself against such attack (do i really have to do this ?)

- clean paths in queries, cookies, anything the user can touch
 - if you're only delivering static files then use complete path white lists, if using dynamic paths use partial white lists (dictionary white lists if you may)
 - use asp.net core ?
- etc..

further reading i would suggest (and by that i mean they include full lists of possible attacks and far more advanced techniques, so read it dum dum) :

https://www.owasp.org/index.php/Path_Traversal

https://en.wikipedia.org/wiki/Directory_traversal_attack

watch it in action

Spoiler [\(Click to View\)](#)

[Video: <https://www.youtube.com/watch?v=L95M0F55Fp0>]

1 video per post, can't embed the rest

<https://www.youtube.com/watch?v=gJOcEYGFrw4>

<https://www.youtube.com/watch?v=yVtIA82ZJuA>

<https://www.youtube.com/watch?v=oU9pN-1LhMQ>

<https://www.youtube.com/watch?v=FYPodlehFfo>

<https://www.youtube.com/watch?v=etFpAvjV0CA>

with good enough feed back ill make an advanced tutorial, give away vulnerable websites for labs,
do the rest of owasp top 10