

Forum:[Hack Forums](#)

Thread: ["compare" two files?](#)

Caleb™ - February 27, 2017, 11:46 am

I remember some time ago there was some software named something like "cmpare" or some shit, to test the difference between two executables, was mainly made to check if build is unique etc, anybody knows what its named or any similar software?

?lpha - February 27, 2017, 12:36 pm

It's better to create a checksum of the two files and compare those.

Kantos - February 27, 2017, 4:13 pm

You can also use the 'diff' program that's common on unix-like systems. The 'diff' program exists on my Windows installation as well, not sure if that's due to git-bash or not.

Masm4ever - February 28, 2017, 2:54 am

On Windows, it's `fc fileA fileB`

Praet0rian - March 2, 2017, 12:15 am

(Feb 28, 2017 03:54 AM)Masm4ever Wrote: [\[CITE\]](#) On Windows, it's `fc fileA fileB`

Or you could not use the shell and make your own program or function that does i, it's childsplay.

maldev0 - March 2, 2017, 4:34 am

Calculate the md5/sha1 hash of the code section in the PE header of the two specific files and compare them

Praet0rian - March 2, 2017, 10:05 am

(Mar 2, 2017 05:34 AM)maldev0 Wrote: [\[CITE\]](#) Calculate the md5/sha1 hash of the code section in the PE header of the two specific files and compare them

That won't cover the data section though.

oxagast - March 2, 2017, 5:16 pm

Why can't you just use sha256sum or another checksum hash function? Preferably not an MD5 or SHA1 checksum, as collisions have been proven ([recently with SHA1](#)), and there is even code out there to make two C programs have the exact same MD5 sum yet do completely different things (you can check that out [here](#)), I know it wasn't the question, but it is relevant, and damn cool.

There are no proven sha256 or sha512 collision algorithms though that I am aware of (afaik SHA1 doesn't have an algorithm to make two do different things, other than just not work at all... yet).

You can compare the files with

`Code sha256sum file1 file2`Simple, works, and you don't have to reinvent the wheel. Relatively secure.