MD5sum Evaluation

Brenden Tyler

The md5sum core utility is used to generate and verify MD5 hashes as defined in RFC 1321. The MD5 hash (or checksum) consists of a 128-bit string which should be unique for each unique file. Due to the constraints place on the size of the output there is a theoretically unlimited number of files that could cause a hash collision (two files that result in the same hash). However, under real-world circumstances two arbitrary files are unlikely to create a hash collision unless they were specifically engineered to do so.

Due to the nature of the underlying hash function, fuzzing was deemed useless for the purposes of evaluating the stability of this code. Hash functions are designed to work on arbitrary input and are often tested for correctness using techniques similar to fuzzing. The code for md5sum was inspected via a manual code review and by Flaw Finder (a static code analysis tool). Overall, the program is well designed and written. There is the possibility that the utility would produce incorrect results if the last string of a file is not null terminated (due to a call to strlen).

While the utility itself has been deemed well written and secure, the underlying hash function is not. It has been shown that a relatively small amount of work can produce files that result in MD5 collisions (See RFC 4270). The two included files are an example of two files that are obviously different but have been engineered to produce the same hash (tested on Ubuntu 12.04).

Fortunately, the code for md5sum also builds several other tools with different underlying hash functions. MD5 and SHA-0 both suffer from significant flaws in their implementation and are not recommended. A theoretical attack exists for SHA-1 and, as such, it has been deemed unsuitable due to potential security concerns. The various versions of SHA-2 and SHA-3 (224,256,384,512) are currently deemed secure and the utilities employing these algorithms are recommended.