BMC Software Inc.

Technical Disclosure Publication Document

Author: David A. Solin

Method to Circumvent Hashing Algorithm Collision-Generation Attacks

Posted: November 4th, 2009

Overview

This document describes a method of using hashing algorithms to construct digital signatures in such a way as to minimize the signature's susceptibility to collision attacks.

Background

Digital signatures are commonly employed to verify the authenticity and integrity of electronic file contents. Creating such a signature requires the use of a cryptographic hash function.[1] Until recently, MD-5 and SHA-1 were the most commonly-used hashing algorithms for this function. However, fast numerical methods for generating collisions were discovered for MD-5 in 2004, and SHA-1 in 2005. These methods make it possible to create files with different contents, whose computed checksums are the same. In effect, this makes a digital signature based on these algorithms untrustworthy, since it would be possible for an attacker to substitute the contents of a signed file while the signature itself would remain valid – defeating the signature. Specialized attacks make it possible to generate a collision incorporating a pre-determined prefix, giving an attacker the ability to produce a signature for two largely pre-determined sets of data – one authentic and the other willfully counterfeit.

NIST has previously advised generating checksums using multiple algorithms as a means of maintaining the security of the signature.[2] Currently, creators of digital signatures commonly use SHA-2 variants (e.g., SHA-224, SHA-256, SHA-512), however according to Wikipedia "Although no attacks have yet been reported on the SHA-2 variants, they are algorithmically similar to SHA-1 and so efforts are underway to develop improved alternatives".[3]

Solution

Instead of requiring the use of new (as yet undeveloped) hashing algorithms to generate digital signatures, or employing multiple hashing algorithms, the following method defeats the methodologies of known collision-generation attacks by applying existing hashing algorithms to

[1] See http://en.wikipedia.org/wiki/Cryptographic_hash_function
[2] See http://www.nsrl.nist.gov/collision.html
[3] http://en.wikipedia.org/wiki/SHA_hash_functions

generate a pair of checksums. The first is generated normally by processing the contents of the file. The second is generated by passing the file contents through the hashing algorithm backwards.

It is claimed (without proof) that it should be impossible to numerically generate a stream of data whose checksum values in both directions could be pre-determined, as there is no *determinate* algorithm from which a reverse data stream can be numerically computed. (In this way the method differs from simply transforming the data, for example, by computing a checksum using only every n^{th} byte, or by applying any other numerical transform.) It is further claimed that an attack using a library of files with known checksums would require an impractically large library – using this method in conjunction with the MD5 algorithm would require a library of size $O(2^{256})$, which is 2^{128} times larger than required to attack conventional uses of the algorithm.

The method should thus afford secure use of any given cryptographic hashing function, even well after a numerical collision-generation method is devised.