A Web Service for File Fingerprints:

The Goods, the Bads, and the Unknowns

Abstract:
Databases of cryptographic hash codes are increasingly being used by computer security professionals and forensic investigators. Nevertheless, today's databases are hampered by the way that they are distributed and accessed. After surveying systems currently in use, this paper presents the framework for a web service that would allow for the collection, searching, and replication of a cryptographic hash code database. Several applications that could be built using this database are explored.

Garfinkel, Simson L. < simsong@mit.edu>

1507 words

1 File Fingerprints, Fingerprint Databases, and Current Services

Cryptographic hash functions such as MD5[1] and SHA-1[2] have the property of condensing a large blocks of data into compact representations of 128 bits (in the case of MD5) and 160 bits respectively. These condensed representations have the property that they cannot be predicted from a given input file, and changing any bit in the input file results in a bitwise inversion of approximately half of the bits in the resulting cryptographic residue. Because the numbers 2^{128} and 2^{160} are vast when compared to the total number of electronic documents that have been created during the course of human history, hash representations can be treated as "fingerprints" or "signatures" for files: no two files have ever been found that have the same MD5 or SHA-1 code.

Because even a single bit changed in a file results in the creation of a radically different cryptographic hash, many software vendors separately publish the MD5 or SHA-1 codes of their software distributions, allowinf end users to verify the integrity and authenticity of downloaded software.[3]

Cryptographic hash codes can also be used to determine the identity of a file. By computing the hash of a suspect file and then looking up that hash in a database, it is possible to determine if that suspect file is a copy of a file that has previously been evaluated, characterized and registered. CDROMs of hash codes are now available from a variety of sources. For example, the *National Software Reference Library Reference Data Set*[4] is a collection of more than 7 million hash codes of executable files, Dynamic Linked Libraries (DLLs), and auxiliary files gathered from commercially distributed software packages sold over the past two decades. Various law enforcement agencies have distributed data sets containing the hash codes of "known child pornography.[5]" Hash sets can be imported into hard drive forensic analysis tools such as EnCase[6] and Forensic Toolkit,[7] allowing these tools to automatically identify "known goods" (e.g. application files) and "known bads" (e.g. exploits).

There is growing interest in accessing databases of "known goods" and "known bads" over the Internet using web-based interfaces. For example, Brian Wotring's website www.knowngoods.com allows a user to search by MD5 code through one of 40 different software releases for a matching file. Similar websites are under development by Filip Maertens and Chris Loper, both participants in the online forum forensics@securityfocus.com. Although inactive now, the website http://www.newshog.com/viewblock.cgi apparently once allowed searching for reported child pornography by MD5 code, filename and a variety of other search criteria.

2 A File Fingerprint Web Service

Today's state-of-the-art could be considerably advanced by the creation of a file fingerprint web service. This service would allow both searching the database of hash codes for matches and the uploading of new hash codes for incorporation into the

database. Other queries would allow for database replication.

Searching. The database could be searched with two modalities. An interactive search could be performed through either a query entered interactively on a CGI form or else through an XML query uploaded using SOAP. In either case, the query would consist of one or more MD5 or SHA-1 hashes coded in hexdecimal. Larger searches, for hundreds or thousands of hashes, could be performed by sending an XML query by email to the web service; the search would be executed and the results returned by email. In either case, the result would be an XML document consisting file records for hash matches, followed by a list of file sets for each set mentioned in the file records. Unlike the www.knowngoods.com website, there is no need for the client to specify the file set to search, since hash codes are globally unique.

Uploading. Although the web service's database will be seeded with hash codes from existing services, the service will be most useful if users are provided with a means for uploading their own hash sets. The easiest way to do this will be to provide agents that can run on Windows and Unix computers. These agents will scan the file system, compute the MD5 and/or SHA-1 of every file, and upload the results. The web service will assign a unique hash set ID for each upload session and will track uploads by date and time and source IP. To secure the service against vandalism, uploads could be in the form of XML objects signed with XML Signatures[8]; the web service could then implement a voting system allowing users to rate the credibility of various signing keys. Data from keys that is rated poorly by the community could be expunged from the database at a later time, or simply ignored.

Replication. This web service needs to be replicated to protect against hardware failure and to allow the database contents to be geographically distributed. Fortunately, there is no need to create a separate database replication system. Instead, a slave database can query the master database for all hash codes uploaded since the last update; the resulting set of hash codes, files and file sets can then be incorporated into the replicate. This web services replication approach has the advantage that replicas need not be running the same back-end database.

3 Applications

The "known goods and bads" web service will allow for a variety of applications, including:

* Rapid identification and removal from consideration of "known goods" and "known bads" on a computer that is under forensic investigation, giving an investigator more time to analyze the material on the computer that is actually unique.

* Automated identification and removal of unlicensed copyrighted software from computers in a corporate setting (assuming that the hash codes for the copyrighted software has been registered).

* Automated identification of “known bads” on production computers. Such “bads” might include known Trojan horses, “root kits[9],” and other hacker tools. Such identification techniques would rely, in part, on the fact that most computer attackers lack the sophistication to create their own attack tools, but instead simply re-use tools that they have downloaded from other attackers.

* Automatic reasoning about a system. There is no reason that the web service should be limited responding to simple database queries: a second-generation semantic service should be able to draw conclusions about a system by simply analyzing the hash codes that are uploaded. For example, a web service could determine the operating system in use and whether or not particular patches had been applied.

4 Implementation Issues

I propose building such a web service on a system running the FreeBSD operating system using the Apache web server as a front-end, SOAP::Lite for Perl as the web services gateway, and MySQL as the back-end database.

Whereas previous efforts have stored hash codes as hex strings, this effort will store codes as binary BLOBS, as binary strings will consume half the space on disk and will index better.

Although fields will be provided for SHA-1 entries, it is expected that most users will only employ MD5 codes. SHA-1 is increasingly preferred because it is the current NIST standard and is more appropriate for high-security applications given the larger hash size. In this application, however, the 128 bits of MD5 provide adequate security. MD5 codes can also be computed in a third the time as SHA-1 codes—an important factor given the high computational overheads for forensic analysis.

Whereas previous efforts have allowed only a single description for each MD5 or SHA-1 code, this web service will necessarily need to provide for a single hash code to reside in multiple file sets. This is important, for example, because some DLLs are included in two different programs—one good (like Microsoft word) and one bad (like Back Orifice). Without significant manual intervention, it will not be possible to evaluate individual hash codes or files to determine if they are “good” or “bad.” Instead, the semantic database will need to be able to tolerate ambiguity and apparent contradiction.

Although it is tempting to create a simple schema for representing file sets, more power might be had by representing them with RDF.[10] It may even be valuable to allow RDF to describe individual files: this would allow the service to understand that one file had been superseded by another file—for example, in the case of a security update.

Finally, it may be desirable to create a gateway that would allow the hash code database to be quickly queried using the DNS protocol. Although non-standard, such a gateway would allowing querying the database from behind most firewalls with a minimal amount of configuration. This might be important during some forensic investigations.

5 Conclusion and Acknowledgements

Databases of MD5 and SHA-1 codes are increasingly important in the computer forensics and security community. I have presented a proposal for web service that would greatly simply the creation, maintenance, and access of such databases. I currently plan to create this service during the spring of 2003.

I would like to give special thanks to Professor Benjamin Grosof for suggesting that I take his IAP course on Web Services and the Semantic Web.

---

[1] Rivest, R. "The MD5 Message-Digest Algorithm," Network Working Group Request for Comments: 1321, April 1992. http://www.ietf.org/rfc/rfc1321.txt

[2] "Secure Hash Standard," Federal Information Processing Standards Publication 180-1, 17 April 1995. http://www.itl.nist.gov/fipspubs/fip180-1.htm

[3] For example, the Solaris Package Archive at http://www.ibiblio.org/ distributes the MD5 "checksums" for SPARC and i386 Solaris releases of popular GNU software packages. See http://www.ibiblio.org/pub/packages/solaris/sparc/.

[4] "National Software Reference Library Reference Data Set," National Institutes of Standards and Technology. http://www.nsrl.nist.gov/

[5] McCreight, Shawn, and Patzakis, John, "Hash sets and their proper construction," Guidance Software, 2001. http://www.guidancesoftware.com/support/downloads/hashsets_wp.pdf

[6] Guidance Software, "EnCase," http://www.guidancesoftware.com/

[7] Access Data, "Forensic Toolkit," http://www.accessdata.com/

[8] D. Eastlake 3rd, J. Reagle, D. Solo, "XML-Signature Syntax and Processing," network Working Group Request for Comments: 3275, March 2002. http://www.ietf.org/rfc/rfc3275.txt

[9] Oktay Altunergil, "Understanding Rootkits," O'Reilly Network, 14 December 2001. http://linux.oreillynet.com/pub/a/linux/2001/12/14/rootkit.html

[10] "Resource Description Framework (RDF)," World Wide Web Consortium. http://www.w3.org/RDF/