Design and implementation for MD5-based data integrity checking system

Danyang Cao
College of Information Engineering
North China University of Technology, NCUT
Beijing 100144, China
ufocdy@163.com

Bingru Yang
College of Information Engineering
Beijing University of Science and Technology, BUST
Beijing 100083, China
bryang@yahoo.com

Abstract—**With the increase of the amount of data and users in information system, the requirements of data integrity in system need to be improved as well. This article presents a data integrity checking method based on MD5 in order to ensure that the data of information system is in correct state. It also elaborates the principle of MD5 algorithm, researches a way to utilize data integrity checking method on the protection of information system. Finally, the paper designs and implements a data integrity checking system based on MD5 algorithm. Practice has shown that the IT departments of the enterprise achieve the system management and security auditing with the data integrity checking system.**

Keywords- MD5; data integrity; information system

I. INTRODUCTION

The designers of information security were devoted to take many measures to protect private data and systems in early days, for example they used strictly controls of access authorization and accesses to the audit and encrypted data. The traditional security model tends to strength the protection layer of the protected resource. Designers believe that strengthening the protection layer is enough without caring about the integrity of the protected data.

However, with the development of network technology, network security tasks extend from protection of confidential data to protection of information systems which provide network services to institution employees, customs and partners. Right now as things stand, a strong protective layer is bound to weaken the flexibility of information systems and service capabilities to provide users. Moreover, it is not realistic to construct 99%of the protective layer. In the face of the inevitable network security threats, data integrity checking is the inevitable choice to solve current defect of traditional model [1].

For it is uncertain to invade information system, system administrators need to find a way to convince that the state of information system maintains in a safe way, integrity checking provides system administrators with a way to fully understand the security level of information system, administrators can ensure that data and networks of the system are in the right state at any time, that is to say, administrators can know whether current system is complete, the locations and moments damaged probably. This paper presents a data and network integrity method based on MD5 message digest algorithm [2], studies how to use data integrity checking to protect the safe of information systems, and implements a data integrity detection system based on MD5.

978-1-4244-5265-1/10/$26.00 ©2010 IEEE

II. MD5 ALGORITHM

MD5 stands for Message-Digest Algorithm 5. MD5 algorithm is co-invented by Rivest in MIT Computer Science Laboratory and RSA Data Security Company. MD5 is a non-reversible encryption algorithm [3]. It is widely applied in many aspects, including digital signature, encryption of information in a database and encryption of communication information. It makes large amounts of information to be compressed into a confidential format before signing the private key by digital signature soft (that is, any length byte string is transformed into a certain length of big integer).

A brief description of MD5 algorithm as follows: MD5 algorithm divides plaintext input into blocks each which has 512-bit, and each block is again divided into sixteen 32-bit message words, after a series of processing, the outputs of the algorithm consist of four 32-bit message words. After these four 32-bit message words are cascaded, the algorithm generates a 128-bit hash value which is the required ciphertext. Specific steps are as follows [3, 4, 5]:

(1) Padding-bit: Without loss of generality, supposes that the original data at the source has k bits ($m_{k-1}, m_{k-2} \ldots m_0$), where $m_i \in \{0, 1\}$. For MD5 algorithm, its k bits data must be processed in 512-bit message block, so if the length of source is less than that length, padding is always added until its length in bits is congruent to 448 modulo 512 (length≡448 mod 512). The padding consists of a single 1-bit followed by the necessary number of 0-bits.

(2) Padding the length of data: a 64-bit representation of the length on bits of the original message is appended to the result of above step. It is present by two 32-bit digits. At this time, the length of message is filled to a multiple of 512.

(3) Initialize MD5 parameters: four 32-bit integers *A, B, C, D* are called chaining variables, used to calculate the message digest, are initialized by hexadecimal number *A*=0x01234567, *B*=0x89abcdef, *C*=0xfedcba98, *D*=0x76543210.

(4) Bit operation functions: We define four bit operation functions, *F, G, H* and *I* respectively, in which *x, y, z* are three 32-bit integers. The operation is as follows:

$$F(x, y, z) = (x \wedge y) \vee ((\neg x) \wedge z) \quad (1)$$

$$G(x, y, z) = (x \wedge z) \vee (y \wedge (\neg z)) \quad (2)$$

$$H(x,y,z) = x \oplus y \oplus z \quad (3)$$

$$I(x,y,z) = y \oplus (x \vee (\neg z)) \quad (4)$$

In four functions, if the corresponding bits of *x, y* and *z* are independent and uniform, then each bit of the results should be independent and uniform as well. For $x = \sum_{i=1}^{32} x_i 2^{i-1} \in Z/(2^{32}), x_i \in \{0,1\}$, we call x_i the *i*-th bit of *x*.

(5) Main transformation process: the number of main loop in this algorithm is the number of 512-bit information groups. The main loop have four rounds, each round carries out 16 operations, so the total of operations are 64 steps. The above four chaining variables are assigned to another four chaining values: $a_0=A$, $b_0=B$, $c_0=C$, $d_0=D$. One of the chaining values is updated in each step and computation is continued in sequence. Here we have defined four rounds composite functions of main loop *FF*, *GG*, *HH*, and *II* respectively, which change from *F*, *G*, *H* and *I*. the operation is as follows:

$$FF \rightarrow a = b + ((a + F(b,c,d) + M_i + t_i) \lll s) \quad (5)$$

$$GG \rightarrow a = b + ((a + H(b,c,d) + M_i + t_i) \lll s) \quad (6)$$

$$HH \rightarrow a = b + ((a + H(b,c,d) + M_i + t_i) \lll s) \quad (7)$$

$$II \rightarrow a = b + ((a + I(b,c,d) + M_i + t_i) \lll s) \quad (8)$$

Where, + is addition modulo 2^{32}, M_i $(0 \leqslant i \leqslant 15)$ is a 32-bit message word and the 512-bit message block is divided into 16 32-bit message words. $x \lll s$ is the left shift rotation of *x* by *s* bits. The t_i and *s* are step-dependent constants, t_i has the following options: in *i*-th step, t_i is the integer part of $4294967296 \times \text{abs}(\sin(i))$, $4294967296=2^{32}$.

After all of these steps, *A ,B ,C ,D* add a , b, c, d respectively, then the algorithm is continued to run the next 512-bit message block, the final output is *A , B, C , D* of cascading .

Application of MD5 algorithm is to generate a message digest of information in order to prevent tampering. We view the entire file as a large text message, and result in a unique MD5 message digest by the irreversible string transform method. In the future, if the contents of file are changed, we only recalculate MD5 message digest of this file, and will find the difference from the original message digest. Thereby, we can sure the checked file is incorrect.

III. Data Integrity Checking

Data integrity is a fundamental aspect of storage security and reliability. With the advent of network storage and new technology trends that result in new failure modes for storage, interesting challenges arise in ensuring data integrity.

Integrity violations could be caused by malicious intrusions. Security advisory boards over the last few years have noticed a steep rise in the number of intrusion attacks on systems. Large classes of these attacks are caused by malicious modifications of disk data. An attacker that has gained administrator privileges could potentially make changes to the system, like modifying system utilities, adding back-doors or Trojans, changing file contents and attributes, accessing unauthorized files, etc. Such file system inconsistencies and intrusions can be detected using utilities like Tripwire [6, 7].

Data integrity refers to a data state which relative to the damage and loss of data, it usually indicates that the reliability and accuracy of the data is reliable, simultaneously, in bad cases, means that the data may be invalid or incomplete. Integrity checking is to verify whether the existing data and the original data is completely consistent [1].

The major views of data integrity are as follows:

(1) The data in memory must be the same as it first input or its latest change.
(2) Computers used to establish information, peripherals or accessories have to be run correctly.
(3) Data can not be illegal usage of other people.

Data integrity checking technology has became an essential component of institutional and enterprise on information security and network management. Administrators use data integrity checking system to ease the burden of system management, improve consistency and manageability of information systems, speed up the repair efficiency of post-invasion.

The paper implements a data integrity checking system. The system aids system administrators to monitor their file systems for unauthorized modifications. The main goal of the system is to detect and prevent malicious replacement of key files in the system by Trojans or other malicious programs.

IV. Design and Implementation of Data Integrity Checking System

A. System Design

Design ideas of data integrity checking system are as follows: when information system is in steady state, the system reads specified file or network configuration according to administrator's requirements, and generates the corresponding baseline state value (file attributes, visited date, MD5 value, etc.). Administrator customizes the integrity of checking tasks, and then obtains the current state values of monitored data, compares with baseline state value. If data deviation is discovered, the system will inform system administrator such deviation.

Data integrity checking system is divided into two parts: Client-side and Server-console. Client-side tests the data integrity in monitored computers, reports any changes in these data. In addition to detecting the file content's changes based on MD5 value, it also detect the file attributes in monitored system, such as file size, access token, update time, the register changes. Client-side can generate to read easily reports and sent it to Server-console. Client-side is

installed on the monitored computers. Server-console is a console soft which control all Client-sides in the same network. It is responsible for level-set audit report to administrators, and allows administrators to centralized manage all information systems within the same network. System architecture diagram is shown in Fig. 1.

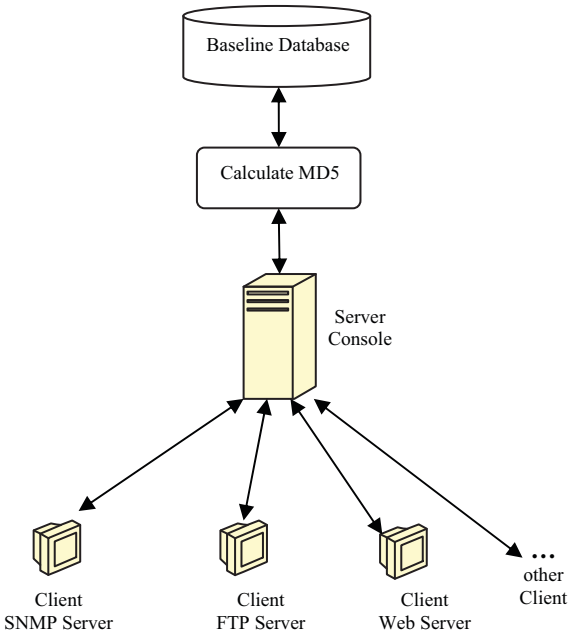

Figure 1. System architecture

When the checked computer is in a steady state, the system gets important file attributes by API interface of operation system, and calculates the MD5 value of the file according to MD5 algorithm. Then the system preserves these data in baseline database. Finally, by comparing the difference between baseline database and current file attributes, the system can provide administrators with a broad and rapid detection capability on changes. Different files can be given different severity levels by the system. Integrity checking reports can be sent to corresponding administrators based on severity levels. When changes are happened, administrators can arrange the dealing order based on severity levels.

This system has the capability of remote management. System administrators manage all Client-sides remotely by Server-console, including the implementation of the integrity checking, viewing the integrity of checking report, remote file configuration. Client-side communicates with Server-console by means of standard TCP/IP protocol, which ensures the security of communication process.

B. System Implementation

This system makes use of C/S (Client/Server) structure, which represents request-response relationship. That is to say, the Client requests the server for certain information or data, and the Server processes the request, then returns the results to the Client as response to the Client. As C++ is an efficient programming language, we adopt C++ language to implement the system [8].

System flow chart is shown in Fig. 2. When the monitored system is in a stable state, the data integrity checking system establishes a baseline database for all files. The baselines of these files represent current correct state of known. After the baseline database is established, administrators can check differences between the state of monitored file and the baseline by means of periodic checking or commands. When this is happened, Server-console sends checking command to Client-sides. Client-sides receive the command from administrators, and then check the file content's changes based on MD5 values. They also detect the file attributes in monitored system, such as file size, access token, modification time, and the register changes, and then generate reports to read easily and sent them to Server-console. Administrators view these differences and distinguish which changes are permitted, and report the information to the system. The system will update the files' baseline with the current state for the licensed changes, and take appropriate security measures to restore the files based on baseline database for unauthorized changes.

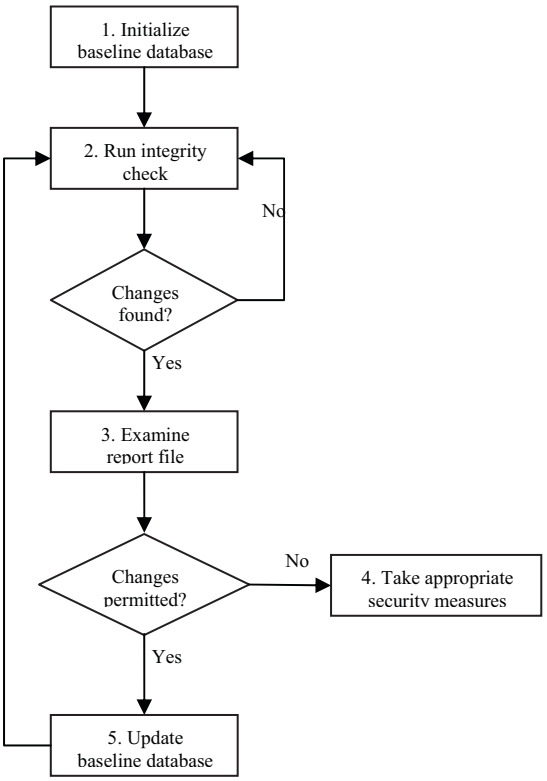

Figure 2. System flow chart

The main interface of Server-console is shown in Fig. 3. From Fig. 3, we can see that the MD5 value of file 'sn.txt' is 072F486C8585BE2212856D95D4EBB200 in baseline database according to formula (1)-(8). After 'sn.txt' is

modified, we implement integrity checking to ‘sn.txt’, and get the MD5 value of ‘sn.txt’ to be 310E4DE986C62A3AE8005E86E9C5CA33. This shows that the file has been modified, administrators should determine whether the change is legal, if it is legal, administrators should update baseline of the file with the modified MD5 value, otherwise the old MD5 value of the file can be used to restore the file ‘sn.txt’. In addition to, the file attributes, such as read only, hidden, archive, file size etc, are shown in Fig. 3 as well.

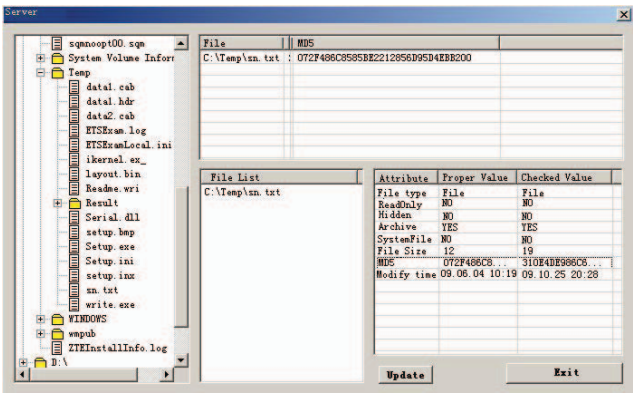

Figure 3. 3 Server console

V. CONCLUSIONS

The message digest to be generated by MD5 algorithm has the irreversible and non-counterfeit features, so MD5 algorithm is superior in anti-tamper capability. This paper implements a data integrity checking system based on MD5 algorithm. The system aids system administrators to monitor their file systems for unauthorized modifications. The main goal of the system is to detect and prevent malicious replacement of key files in the system by Trojans or other malicious programs. It plays a protective role which prevents the hacker and the virus from invading. Practice has shown that the IT departments of the enterprise achieve the system management and security auditing with the data integrity checking system, it brings visible benefits to the enterprise.

ACKNOWLEDGMENT

This paper is supported by Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality (No.PHR20100509), China National Science & Technology Pillar Program (No.2009BAE85B00), and Research Fund of North China University of Technology.

REFERENCES

[1] QUE Xi-rong. Information Security Principles and Applications [M]. Beijing: Tsinghua University Press, 2003.

[2] Willian Stallings, Cryptography and Network Security: principles and Practice. Beijing: Tsinghua University Press, 2002.

[3] Rivest R L. The MD5 message digest algorithm [EB/OL]. http://www.faqs.org/rfcs/rfcl320.html., 2005.

[4] Wang Xiaoyun, Feng Dengguo, Lai Xuejia,et a1. Collisions for hash functions MD4, MD5, Haval-l28 and RIPEMD [EB/OL]. http://eprint.iacr.org/2004/199.pdf, 2004.

[5] Lai Xue Jia, An objective look at MD, SHA-l has been "cracked" [J]. National Information Security Evaluation and Certification, 2005, No. 3 pp. 6-7.

[6] G. Kim and E. Spafford. The Design and Implementation of Tripwire: A File System Integrity Checker. In Proceedings of the 2nd ACM Conference on Computer Communications and Society (CCS), November 1994.

[7] Tripwire Inc. Tripwire Software. http://www.tripwire.com.

[8] Wang Hong Tao. In-depth analysis of Visual C++ programming techniques and Application [M]. Beijing: Posts & Telecom Press, 2003.