International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 3; Issue: 2; August - 2014

Enhancement of Audit Tool by Using MD5 Encryption and Cron Jobs

Shantala Giraddi
Department of Computer Science
B.V.B. College of Engineering & Technology
Hubli, India.
Email: shantala@bvb.edu

Ishwari V. Ginimav
Department of Computer Science
B.V.B College of Engineering & Technology
Hubli, India.
Email:ishwari29@gmail.com

Abstract: Audit tool is a tool of control to measure and evaluate the effectiveness of the working of an organization primarily with accounting, financial and operational matters. This paper focuses on saving password using MD5 encryption algorithm and crontab jobs to send background mails. MD5 is one of the most widely used cryptographic hash functions nowadays. The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity. Crontab is a time based job scheduler in Unix like computer operating systems. Cron enables users to schedule jobs (commands or scripts) to run automatically at a certain time or date.

Keywords- MD5 encryption, cron jobs.

I. INTRODUCTION

The job of audit tool is to ensure that the work of the company is going on smoothly, efficiently and economically and that all the laws, rules and regulations governing the operations of the organization are adhered to, besides ensuring that an effective internal control system exists to prevent errors, frauds and misappropriations. Our "Audit Tool" software helps to store the details of any enterprise and the process carried out at that enterprise. It also keeps the tracks of all works done in an organization. It also helps to inform the higher authority people about the works that are not done.

In this paper we will present a new approach of storing the password using MD5 encryption algorithm and sending mails automatically using crontab jobs.

MD5 algorithm was developed by Professor Ronald L. Rivest in 1991. According to RFC 1321, "MD5 message-digest algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA."

The cron utility allows you to schedule a repetitive task to take place at any regular interval desired, and the at command lets you specify a one-time action to take place at some desired time. You might use crontab, for example, to perform a backup each morning at 2 a.m., and use at to remind yourself of an appointment later in the day.

II. Existing System

Audit tool is used for the examination and systematic, structural evaluation of a business organization, the operations therein and the products and processes of production within the business system. In earlier days auditing was by an independent person or body of persons with the help of vouchers, documents, information and explanations received from the authorities, for the purpose of ascertaining whether the works done entered in the

books are genuine and have been entered with proper authority. It was done manually. Its job is also to find out whether they are accurate and that the works are done in accordance with law and rules and regulations of the organization in particular the standards and standard auditing practices.

III. PROPOSED SYSTEM

Our audit tool works in the computerized environment and has become more relevant so as to make the audit personnel very effective in detecting irregularities. It is the examination of all managerial performance.

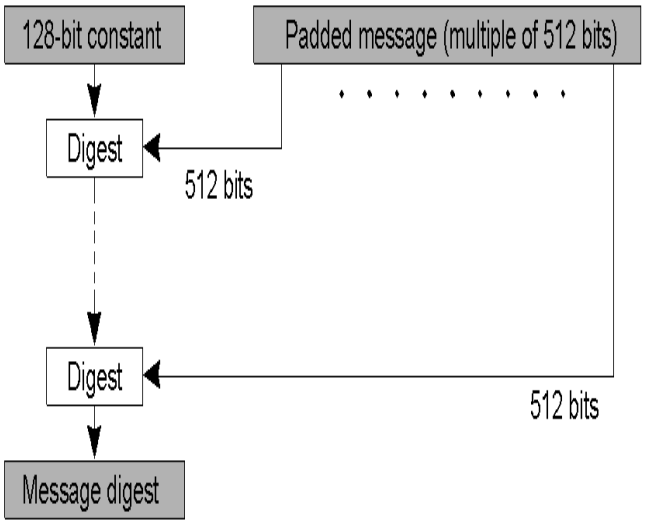
In our audit tool system all the works done or not done can be checked online. In our system any new enterprise can registers to the system. Once the new enterprise registers to the system, the super admin has the authority to activate or delete the newly registered enterprise. The super admin also has the authority to deactivate or reactivate already registered enterprises. Once the super admin activates the newly registered enterprises a mail will be sent to that respective enterprise informing about the url, username and randomly generated password of the enterprise. The password is encrypted using MD5 encryption and stored in the database. Here the enterprise admin can add, view, edit and deletes department, stakeholder, process, checklist online. In our system as soon as the enterprise admin adds a stakeholder a mail will be sent to his/her email id informing that he has been added to so and so department with a particular role assigned to him. In our system when the enterprise admin adds a process he assigns a stakeholder to that process and as soon as the stakeholder is assigned to a process he will get a mail informing to which process he is being added. The supervisor fills the checklist online and the functional admin gets the mail of the works that have not been done along with the reason. The functional admin gets the mail continuously till the works are done. These mails are sent using crontab jobs.

1) MD5 Encryptio :

MD5 is an improved version of MD4. It is similar in design and also produces a 128-bit hash.

After some initial processing, MD5 processes the input text in 512-bit blocks, divided into 16 32- bit sub-blocks. The output of the algorithm is a set of four 32-bit blocks, which concatenate to form a single 128-bit hash value. First, the message is padded so that its length is just 64 bits short of being a multiple of 512. This padding is a single 1 bit added to the end of the message, followed by as many zeros as are required. Then, a 64-bit representation of the message's length (before padding bits were added) is appended to the result. These two steps serve to make the message length an exact multiple of 512 bits in length, while ensuring that different messages will not look the same after padding.

Message Algorithm Structure



MD5 Algorithm

Step 1: Append padded bits

The message is padded so that the length is congruent to 448 modulo 512. A single "1" bit is appended to a message and then "0" bits are appended so that the length in bits equals 448 modulo 512.

Step 2: Append length:

A 64 bit representation of b is appended to the result of the previous step. The resulting message has a length that is an exact multiple of 512 bytes.

Step 3: Initialize MD buffer

The four word buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D, is a 32 bit

register. The registers are initialized to the following values in hexadecimal:

word A: 01 23 45 67

word B: 89 ab c def

word C: fe dc ba 98

word D: 76 54 32 10

Step 4: Process message in 64 word blocks

Four auxiliary function that take as input 3 32-bit words and produce as output 132 bit word.

F(X,Y,Z)=XY v not (X) Z

G(X,Y,Z)=XZ v Y not (Z)

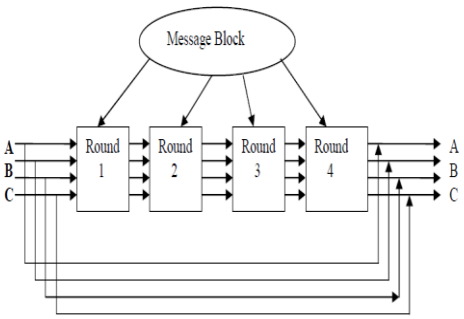
H(X,Y,Z)=X xor Y xor Z

I(X,Y,Z)=Y xor (X v not (Z))

If the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z), H(X,Y,Z) and I(X,Y,Z) will be independent and unbiased.

Step 5: Output

The message digest produced as output is A, B, C, D, i.e., output begins with the low order byte of A and end with the high order byte of D.



2) Crontab Jobs:

Cron is a service that starts up automatically during boot up. It is started from the scripts located in the /etc/rc (and there abouts) directory structure. On my Fedora Core system, the cron daemon (crond) is located in /etc/init.d, and links to it are located in the /etc/rc.d/rcN directory structure. When loaded, cron looks in the /var/spool/cron directory for \ crontab files that map to accounts in /etc/passwd. Crontabs that are found are loaded into memory and kept there.

During system operation, cron examines all stored crontabs once a minute, looking for commands it should execute during that specific minute. Upon successful or unsuccessful completion of a command, cron creates output that is by default emailed to the owner of the crontab. During this minute-by-minute check, cron also checks to see if any of the crontabs have been modified (by examining the modtime of the cron's spool directory, which gets changed by the crontab program whenever a crontab is changed), and reloads them if that is the case. As a result, you don't have to stop and restart cron if you change a crontab. If the computer is down or cron is stopped during a period that there is a task scheduled, that task will not be run once the computer is turned on or cron is restarted. This has implications when your system clock changes, such as daylight savings time. If you have a cronjob scheduled during the hour that is skipped in the spring when daylight savings time causes clocks to be moved from 1 am to 2 am, that job will not run at all. Conversely, if you have a job scheduled during the hour in the fall when clocks are set back, that job will be run twice. In other words, cronjobs do not get placed in a queue that, if the computer is down for a period of time, can be rescanned in order to run jobs that were skipped during the downtime. Instead, cron simply fires off tasks in realtime.

The 'Crontab' Command

Crontab copies the specified file or standard input if no file is specified, into a directory that holds all users' crontabs.

SYNOPSIS:

- crontab [file]
- crontab -e [-u username]
- crontab -r [-u username]
- crontab -l [-u username]

The *-e option* edits a copy of the current users' crontab file or creates an empty file to edit if crontab does not exist.

The *-r option* removes a user's crontab from the crontab directory.

The *-l options* lists the crontab file for the invoking user.

Setting up a Crontab job

A crontab file consists of lines of **six** fields each.The fields are separated by spaces or tabs. The first five are integers that specify the following:

1. minute (0-59),
2. hour (0-23),
3. day of the month (1-31),

4. month of the year (1-12),

5. day of the week (0-6 with 0=Sunday).

Each of these patterns may be either an **asterisk** (meaning all valid values) or a list of elements separated by commas. An element is either a number or two numbers separated by a minus sign (meaning an inclusive range). Notice the time is in 24 hour format, **0** is midnight and **13** is one in the afternoon. The sixth field of a line in a crontab file is a string to be executed by the shell at the specified times by the first fife fields. A percent character in this field (unless escaped by \) is translated to a newline character. Only the first line (up to a % or end of line) of the command field is executed by the shell. The other lines are made available to the command as standard input. Any line beginning with a # is a comment and is ignored.

More graphically they would look like this:

```
* * * * * Command to be executed
- - - - -
| | | | |
| | | | +----- Day of week (0-7)
| | | +------- Month (1 - 12)
| | +--------- Day of month (1 - 31)
| +----------- Hour (0 - 23)
+------------- Min (0 - 59)
```

Example:

To illustrate, 0 0 1,15 1 would run a command on the first and fifteenth of each month, as well as on every Monday at exactly midnight. To specify days by only one field, the other field should be set to *. The entry, 0 23 * * 1 would run a command only on Mondays at eleven PM. A minute specification of 0,30 would indicate the job is to be run on the hour and half hour. Likewise, a day of the month entry of 1,15 would initiate execution on the first and fifteenth of the month. Make sure you include an explicit path to your programs or scripts that you want to run by crontab. Let's assume we want to execute a Perl program, autoclose.cgi, every day at midnight. Additionally assume the full path to the script is /home/www/yourdirectory/cgi-bin. The full crontab command would be:

0 0 * * */home/www/yourdirectory/cgi-bin/autoclose.cgi

Execute at midnight, every day of the month, every month of the year, and every day of the week. If you're not sure of the full path to your script, go to the directory and issue the command:

pwd

This is the "**print working directory**" command. The requirement for explicit paths also apply to any "require" library file used in a Perl script. The line:

001,15*1/home/www/yourdirectory/cgi-bin/autoclose.cgi

runs the command on the first and fifteeth of each month, as well as on every Monday. The line:

0 0 * * 1 /home/www/yourdirectory/cgi-bin/autoclose.cgi

runs a the command only on Mondays.

IV. CONCLUSION & FUTURE ENHANCEMENT

Our audit tool works in the computerized environment and has become more relevant so as to make the audit personnel very effective in detecting irregularities. It is the examination of all managerial performance.

Automated auditing tools can be used to complement manual auditing by subject matter experts and expand the amount of process. In this article, we provide a methodology to estimate the effectiveness of these tools. We showed that the effectiveness depends on both the prevalence of non-compliant cases as well as the performance of the tool. The approach is expected to help businesses make smarter decision on employing subject matter experts and utilize automated audit tools.

Future Scope

Future work will be listed as follows:

- Integration with the widely used Open Source tools. E.g. Moodle & Sakai.
- Making the processes interdependent.
- Making the processes dynamic, i.e., the end date of a particular process should be the beginning date of another process.

REFERENCES

[1] Md. Alam Hossain, Md. Kamrul Islam, Subrata Kumar Das and Md. Asif Nashiryg, "CRYPTANALYZING OF MESSAGE DIGEST ALGORITHMS MD4 AND MD5 ",International Journal on Cryptography and InformationSecurity(IJCIS),Vol.2, No.1,March 2012 .

[2] www.w3schools.com for basic PHP coding

[3] CodeIgniter User Guide Version 2.1.4

[4] http://twitter.github.com/bootstrap/ for AuditTool Design UI

[5] Nicholas C. Zakas, Jeremy McPeak and Joe Fawcett "*Professional Ajax*