Implementation of File Interpolation Detection System

Fujimura, Naomi
Faculty of Design, Kyushu University : Professor : Information Technology

Mei, Jin
Graduate School of Design, Kyushu University : Student : Design

出版情報：Proc. of SIGUCCS 2007, pp.118-121, 2007-10. ACM SIGUCCS
バージョン：

Implementation of File Interpolation Detection System

Naomi Fujimura
Faculty of Design, Kyushu University
4-9-1, Shiobaru, Minami-ku
Fukuoka, Japan
+81-92-553-4434
fujimura@design.kyushu-u.ac.jp

Mei Jin
Graduate School of Design, Kyushu University
4-9-1, Shiobaru, Minami-ku
Fukuoka, Japan
himula@gsd.design.kyushu-u.ac.jp

ABSTRACT
Recently we have found a high possibility to encounter file interpolation and Web defacements by vicious crackers and software. It is not easy for us to find such interpolated files because of the numbers and volumes of files are great in computer systems. We need a good tool such as "Tripwire" for that purpose. However, such a system is only for system administrators and not for users. It is also difficult for administrators to set up the configuration file to do the suitable file check.

We implemented the file interpolation detection system for both administrators and users. The system detects insertion, deletion, and modification (interpolation) of files. Both administrators and users can check the files concerned to themselves and get the result. Users can update the file specification information in the database by command, then it makes the system possible to avoid finding of the error interpolation. The system can be periodically executed by CRON or on demand by users, and then compares the value of MD5 for each file to detect file interpolation. The system has the command line interface and Web interface.

The system first creates the database that contains full path file name, last update time, and values of MD5 according to the information set-upped by users that specifies the location to check for each user. It judges the insertion and deletion of files by the existence and no existence of records in the database. It also judges the file interpolation by the comparison with the value of MD5 for every file. It reports the result by e-mail, in command line interface, or in Web interface.

Categories and Subject Descriptors
K.6.5[**MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS**] : Security and Protection - *Unauthorized access*; D.4.3 [**OPERATING SYSTEM**] : File Systems Management - *Maintenance*

General Terms
Management, Measurement, Performance, Design, Reliability, Experimentation, Security

SIGUCCS'07, October 7-10, 2007, Orlando, Florida, USA.

Keywords
Web server, File interpolation, Security, Management.

1. INTRODUCTION
Recently we have found a high possibility to encounter file interpolation and Web defacements by vicious crackers and software [1, 2]. Many users modify the Web contents respectively in a Web server. Many files such as commands and system configuration files do not appear apparently to administrators and users. As a result, it is difficult even for administrators to find the various changes in the file systems and to judge whether it is the file interpolation or not.

It is necessary for administrators and users only to get messages against the file interpolation but not for proper updates of contents modified by the ordinary users. We need a good tool such as "Tripwire" for that purpose [3]. It detects all changes in files including normal modification. Such kind of system is only for system administrators and not for users. It is also difficult even for administrators to set up the configuration files to do the suitable file check. User interface is not so good from the view point of casual users.

We implemented the file interpolation detection system for both administrators and users. It makes us possible to detect file interpolation and Web defacement with easy and convenient interface. This is the report of the function, user interface, and some experience with the system.

2. SYSTEM FACILITIES
The system provides not only administrators but also ordinary users with the facilities to detect insertion, deletion, and modification (interpolation) of files. It has the following facilities.

1) It keeps the full path location and the values of MD5 of file, user names concerned, and last updated date & time as characteristic information in the database.
2) The file check program is periodically executed by CRON or on demand by users, and then it compares the value of MD5 for each file to detect file interpolation
3) Users can update the characteristic information in the database by command when they modify the files. Therefore, it makes the system possible to avoid finding of the error interpolation.
4) Both administrators and users can check the files concerned

to themselves. The information which files and directories should be checked is kept in the database corresponding to each user.

5) The system has the command line interface and Web interface for users.

6) All users can use this system because the system uses the ordinary user account information in the server. Administrators have their own password kept in the administrator table of database.

Figure 1 shows the system configuration. It consists of "Files", "Database", and users such as administrators and ordinary users to get the "Results".

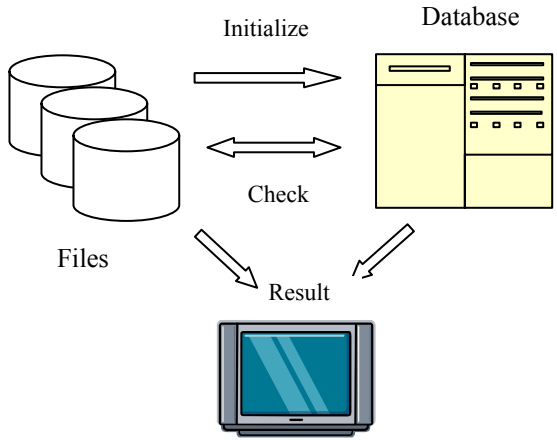

Figure 1 System Configuration

3. THE WAY TO DETECT

The system checks the file interpolation according to the following method.

1) It reads the all files under the directory specified by users and computes the values of MD5 for the first time. It keeps the information such as full path name, last checked date, and the value of MD5 in the database.

2) It computes the values of MD5 of files periodically, and compares them with the values of MD5 in the database as characteristics information to check the file interpolation.

3) When the value of MD5 is the same to the original value, the file does not change. If the value of MD5 is different from the original value, the contents of the file must have changed since last check.

4) The system judges the possibility of insertion and deletion of files according to the existence and no existence of the real file and characteristic information in the database.

5) When the ordinary user modified the file, they are expected to update the characteristic information in the database by command. It makes the system possible to avoid finding of the error detection of file interpolation.

4. USER INTERFACE

The system has the user interface of command line and Web. The command line interface is as follows.

- Initialize the database (initfile.php)
- Checking of files (check.php)
- Update the characteristic information for each file modified normally (update.php)

Figure 2 shows the execution sample for command line interface. In this example, the first command (php check.php) checks the files to find file modification (interpolation) for user "fujimura". One insertion of a file (/home/fujimura/public_html/INDEX.HTML.old) is detected, and the unconfirmed modification of a file (/home/fujimura/public_html/INDEX.HTML) is reported by this check. User updated the characteristic information by hand as a command (php update.php /home/fujimura/public_html/INDEX.HTML). The system does not report the modification for the next check.

```
Example % php check.php
User：fujimura
/home/fujimura/public_html
New Files :
/home/fujimura/public_html/INDEX.HTML.old
Unconfirmed Files :
/home/fujimura/public_html/INDEX.HTML
Deleted Files :
No file is deleted.
 The report is sent to fujimura@design.kyushu-u.ac.jp.
Example % php update.php /home/fujimura/public_
html/INDEX.HTML
Update this file
[/home/fujimura/public_html/INDEX.HTML] to the data-
base?(Y/n):Y
file [/home/fujimura/public_html/INDEX.HTML] was
updated.
Example % php check.php
User：fujimura
/home/fujimura/public_html
New Files :
/home/fujimura/public_html/INDEX.HTML.old
Unconfirmed Files :
No unconfirmed file is found.
Deleted Files :
No file is deleted.
 The report is sent to fujimura@design.kyushu-u.ac.jp.
Example %
```

Figure 2 Sample for command line interface

The Web interface is as follows.

- Set up the directory for each user
- Show the result

Figure 3 shows the sample screen to set up the directory to check. Users can specify the directory by selecting the check box and click "ADD" at the lower part of the screen. Users can select the lower level of directories by clicking the directory itself. Users can remove the selection of directory by selecting the check box and clicking "DEL" in the upper part of the screen.

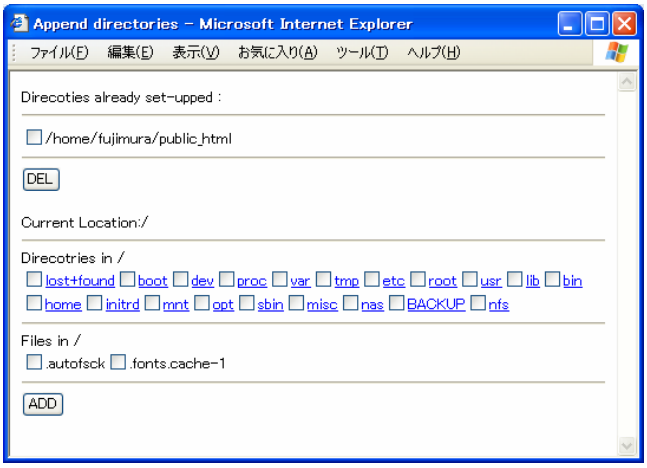

Figure 3 Sample screen for directory set-up

Figure 4 shows the sample screen of file check result in Web interface. It shows that INDEX.HTML was modified and not yet confirmed by the right user. It also shows that INDEX.HTML.old is inserted. The field of information shows the old value of MD5 for the file.

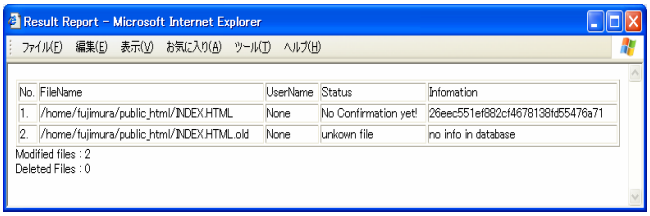

Figure 4 Sample screen for check result

5. DATABASE

The system uses the tables of "admininfo", "dirinfo", and "USER_info". "USER" means individual user such as "fujimura" here, then it becomes "fujimura_info" for user "fujimura".

Table "admininfo" contains the list of administrative users. The first member should be inserted into this table by hand, but additional administrator is appended via Web interface. Table 1 shows the detailed fields for "admininfo". The administrators do not have the important meaning actually now.

Table "dirinfo" contains the information about the users and locations to check. Table 2 shows the detailed fields. Each record contains the username such as "fujimura" and pathname for directory such as "/home/fujimura/public_html". The system checks the status of file specified in this table.

Table "USER_info" contains the characteristic information of files. The filed "md5str" contains the value of MD5 for the file. The check program compares the value kept in this table with the value calculated now to check the file modification.

The filed "new" is used to check the new file, and the field "delete" is used to check the deleted file. The field "new" is set to "0" at the beginning, and set to "1" when new record is appended into this table. After all files are checked, the file which has the value "1" for "new" field is newly created files.

The filed "deleted" is used to detect the deleted files. The value "1" is set for this field at the beginning. When the value of MD5 is checked, this value is set to "0". It means that the file exists. After all check is done, the records which have the value "1" for "deleted" field meads that file does not exist now, i.e. it is deleted.

Table 1 Admininfo

Field	Type	Purpose
username	varchar(16)	Account name
passwd	varchar(32)	Password for username
mailaddr	varchar(100)	Mail address for username

Table 2 Dirinfo

Field	Type	Purpose
username	varchar(16)	User name
pathname	varchar(250)	Full path for directory to check

Table 3 USER_info

Field	Type	Purpose
pathname	varchar(250)	Full path name for each file
username	varchar(16)	Account name
md5str	varchar(32)	The value of MD5 for each file
lastdate	varchar(20)	Last date checked
updated	tinyint(1)	Last date updated
new	tinyint(1)	Flag to identify new or not
deleted	tinyint(1)	Flag to identify deleted or not

6. PERFORMANCE

1) Database Initialization

The system has to initialize the database for the first time. We measured the time to initialize the database. Before that, we copied the original contents in our Web server to other location because it is not good for the measurement if the content is changed by other users during the daily operation. The number of target files is 34259; the volume of files is 3.34GB.

It took 4 minutes 14 seconds for elapse time and 60 seconds for CPU time to initialize the database for the first time. The hardware specification was Celeron 2.00GHz for CPU, 512MB for main memory, and RedHat Enterprise Linux ES V4 for OS.

2) File Check

It is important to know the precise time required to check files because we want to check files periodically as often as possible. If the necessary elapsed and CPU time are so heavy, it is not useful. If we execute the file check so often, we can detect the file interpolation quickly, but the system may fall into the overloaded

status. We should know the time to check the files for our convenience.

It took 4 minutes 38 seconds for elapse time and 63 seconds for CPU time to check the whole files prepared for the target. It is almost the same time to initialize the database because the system reads files and calculates the value of MD5 for the files, and register them in the database for initialization and compare the value in file check.

3) Comparison with Tripwire

We measured the same time to check the same files with Tripwire. It took 5 minutes 44 seconds for elapse time and 1 minute 23 seconds for CPU time to prepare the initial set-up. It also took 5 minutes 33 seconds for elapse time and 1 minute 50 seconds for CPU time to check files. It shows that our system has the slightly higher performance than "Tripwire" in CPU and elapsed time. It is because our system only reads files and calculates the value of MD5 to compare with it in the database.

7. PROBLEMS

We implemented the system with PHP and MySQL. We intended to use Web interface as a whole for the first time. However, we soon found that the apache server works as a user of "apache". As a result, it cannot read the files which have no read permission for "others". It has no problems for files opened to others, but is not good for private files under user's home directory except public_html because most files do not have the read permission for "others". When user specifies the directory for the check, error occurs when "apache" tries to read the files and directories that are not opened to "others".

When users modified the file properly, they are expected to update the characteristic information in the database as described above. User can update it with the command "php update.php FULL_ PATH_NAME". They are forced to update the information one by one so far. It may be good for users to be able to update all information at one time for their convenience. However, it might cause the problem if a cracker happened to interpolate the files just at that time.

It may require much CPU time to check so many files. When we used CGI SAPI PHP version, the program was cancelled because of the time over for CPU time upper limitation. We replaced the PHP from CGI to CLI SAPI PHP. According to the document, this version has the unlimited "max_execution_time", but we are not sure that it works well. It may be a problem to use proper parameter in the execution.

8. CONCUSION

We implemented the system to check file interpolation. It makes administrators and ordinary users possible to notice that a file is modified among so many files. It is possible for them to notice the file interpolation in not only Web contents but also ordinary files under the user's home directory. Every user can check their files respectively by the system with ease of use. We are planning to operate it daily in our Web server, and find the problems to improve the problems and user interface.

9. REFERENCES

[1] Web defacement news in Japan：

http://izumino.jp/Security/def_jp.html

[2] Web defacement news in the world: http://zone-h.org/

[3] Tripwire: http://Web.tripwire.co.jp/