# A Web Forensic System Based On Semantic Checking

Jianhui LIN[1,2]
*1 Department of Information Technology, Hubei University of Police, Wuhan, China*
*2 School of Electronic Information, Wuhan University, Wuhan, China*
*linjh_eis@hotmail.com*

## Abstract

*Computer forensics aimed at determining the relevant causes and effects between the present state of computer system and malicious operation through the measures which could be adopted on the court. This paper proposes an intrusion forensics worked on a WEB server. The forensics system monitor the access to the log files and combine it with the timestamp and other clues in the log file, thus comprehensive operation facts are formed and represented by XML. Through analyzing with a decision tree, intrusion behavior evidence can be abstracted. Especially when a hacker tried to wipe his trace, the system can detect it effectively.*

## 1. Introduction

Currently the primary factor limiting the evidence process is the number of qualified technicians. The majority of digital evidence process by law enforcement today is on computers used as instruments of traditional crimes, such as a threatening email rather than a threatening letter. Although a minority of prosecuted cases, crimes in which the computer is the target are important as well. This difference in the number of cases is due to many factors including workload of investigators, familiarity with technology crime and likelihood of a successful conviction.

As is known, there are millions of WEB servers in the world and billions of attacks happen every day [1] [2]. An experienced attacker will attempt to create an illusion to the administrator that a security violation never occurred. Malicious alteration of the operating system's kernel code, through an unauthorized kernel module for example, will corrupt the results of even a trusted, statically-compiled binary used to detect surreptitious activity. The attacker may meticulously delete log files and other evidence of his presence. Yet side effects of his presence may still exist. Consequences of his surreptitious activity, that he doesn't realize were left behind, doesn't have the access privileges to eliminate, or simply doesn't have the time to remove, will remain.

Formalizing the intrusion response process for consistent repeatability purposes, and automating it for practical ones, is critical. While work has been done to frame the problem of digital forensics, most of the advances have been at the evidence collection and preservation stages of an investigation. Concepts and approaches have been invented to manage complexity and arbitrary levels of abstraction in software development. An expert system can do this for the analysis stage of cyber crime investigations. Few projects have focused on recovery from a successful attack. The "Diagnosis, Explanation and Recovery from Break-Ins" (DERBI) project at SRI used a procedural reasoning system to analyze data. In addition, Elsaesser and Tanner[1] present an approach which automatically generates hypotheses of computer attacks, simulates them on the target configuration and applies plan recognition techniques to search for supporting data.

In this paper, we propose a forensic system based on log content semantic checking. The system collects and analyzes the log content and find those item mismatch the logic then abstracts the evidence.

## 2. Approaches

### 2.1 IIS Log File

Common WEB server software includes IIS (Internet Information Server), Apache, Weblogic, Resin etc., for convenient processing, we chose IIS as our experimental environment. IIS has strong log functions, an example log record is showed in fig.1, as is shown, it provides more then twenty kinds of log record attributes [3]. However, these attributes are optional, then we can choose those attribute we need. Within these attributes, we care about time, IP, method and name [4].

978-0-7695-3311-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISCID.2008.76

```
2006-12-26 14:05:32 127.0.0.1 - W3SVC1 VIVIAN
127.0.0.1 8080 GET /iisstart.asp - 302 0 0 479 40 HTTP/1.1
localhost:8080 Mozilla/4.0+(compatible; +MSIE+6.0;
+Windows+NT+5.0; +.NET+CLR+ 1.1.4322)
ASPSESSIONIDQASCQCTD
=PNJDJPODEDBMEJMGPAFLGPCD
http://localhost:8080/ localstart.asp
```

Fig.1 IIS log record format

## 2.2 System Model

In practice, even without proper preparation for a computer attack, relevant evidence of a security violation can be automatically identified. If attackers make their presence clear of course, detection is not difficult. Yet, if they attempt to conceal their activity, they still have changed the state of the system, leaving footprints through the unknown side effects of their activity.

Automated analysis of technical evidence is an obvious approach. With an automated technique, system administrators who identify an anomaly may quickly make a preliminary diagnosis of their system. Beyond the savings of a forensic expert's time, the repeatability of the investigative process at the technical level is important. Instead of an opinion based upon a person's best effort and limited resources, the reasoning process and the evidence upon which the deductions were made are documented, transparent and deterministic.

Since every access request would result in log items, if invaders want to wipe his attack track, the most important thing he has to accomplish is to delete relevant log items. According this assumption, we design a file access probe to monitor the access to log file, once delete operation happened or there were no log items within a considerate small time span, we suspect an attack happened.

The integrated system model is shown in fig.2. Its basic function realizes from intercepting the access request to log file, then probes are triggered. The probe collect the information of file write request and its access subject, then generate a sentence including log access time and operation according predefined semantic, it would be transferred into XML format and saved into database[5]. An expert system, decision tree, according to specified algorithm, processes semantic analysis. While items mismatched logic are deduced, corresponding alert messages were sent to system administrator.

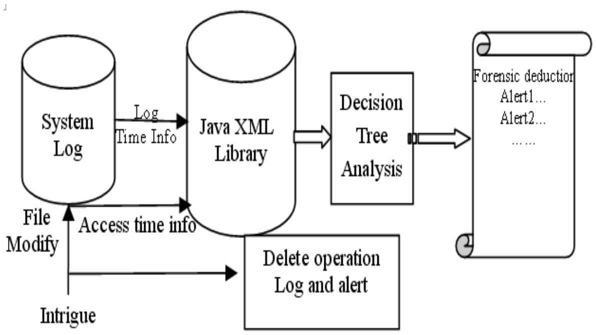


Fig.2 mechanism log forensic system

## 2.3 Expert System

Evidence may be suspicious through analysis, whereby all legitimate reasons for its presence are eliminated and there still not being an explanation. To identify these contradictions, this approach uses a forward chaining, rule-based, expert system. Its working knowledge is the body of evidence [6]. The invariant relationships between digital objects are encoded into the expert system's knowledge base. The ontology is based upon the objects experts use to understand the system in question: memory usage statistics at the kernel level, users, privileges and files for an operating system, network events for a network intrusion detection system, and tables and transactions for a database, for example.

The expert system searches through the data, eliminating those that conform to a known legitimate specification or invariant relationship, and highlights exceptions. These exceptions are semantic contradictions. Fig. 3 is an example of such a decision tree with the goal of determining the set of users who may have deleted the contents of a log file [7],[8].

In this case, the reason to initiate the analysis, or "trigger," was the fact that a file modify request (including delete) in question was captured, but this could be for any reason: manual or automated. Once probe is triggered, the time information and operation is sent to database in XML format, and expert system start working. If any the operation is "IRP-DELETE", we can sure there is an illegal delete operation and system alert is sent. As to file modify operation, taking the time discrepancy between the log time and recorded time from same access operation into account, we add an offset μ onto log time. If access time is within the interval, it is a regular operation. Or else, we judge whether there are file write error occurred. If not, we search other request of the user. If there are some regular operations found, it may be a legal access or there must have some items in the log file was wiped. Meantime, we have to consider that multiple connection request may result in regular items in the log file [9]. Here, we just sent a warning.

In this checking process, expert system tries to eliminate all irrelevant data to deduce the forensic result we need. However, since the forward chaining expert system knowledge base has relatively higher memory consumption, and the deduce computing spend relatively high CPU time.

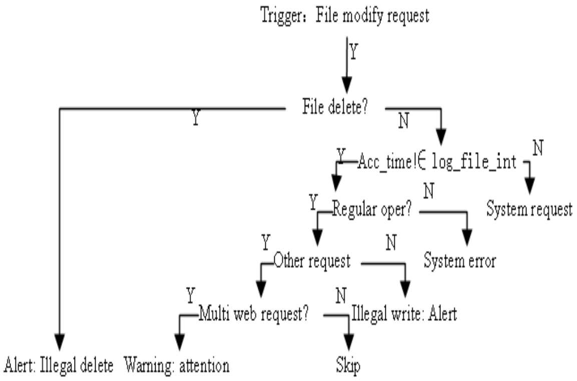


Fig.3 Prototype of the decision tree

## 3. System Realization

### 3.1 Trigger Mechanism

We intercept system file access request from File System Driver layer. Similar to device driver routine, file system driver consist of different routines, which include DriverEntry routine, Dispatch routine, etc.

File System Driver (FSD) is a standard kernel mode driver programs. It is an IO managing subsystem which is in charge of storage information between users and disks and maintains disk structure of various file system. In the process of IO manage, user requests are sent from user mode to kernel mode, IO manager receive the requests. After necessary processing, there requests are dispatched to file system driver layer. FSD send them to device driver program layer then a disk access operation happened.

File filter system driver is an intermediate driver that can capture the file access request. Once set the filter at appropriate position in driver layer, it can capture the access request to log file. We set our filter between the IO manager layer and file system.

Filter module is designed according file system driver. It traces the user access operation request to log file and save them in the buffers. User application loads the captured record from user mode with DeviceIOControl routine and clears the buffers.

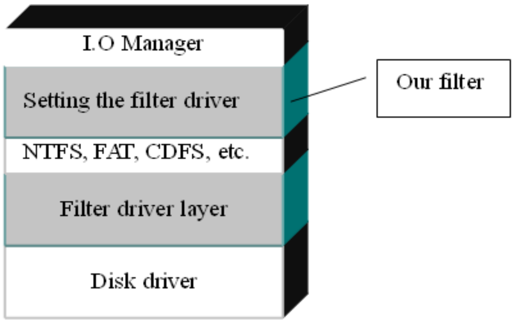


Fig.4 Position of the file filter driver in the system

While user access specified file (system IIS log), I/O Request Packets (IRP) is send to FSD. Since we have set a filter to intercept the IRP_MJ_WRITE and IRP_MJ_DELETE packets, forensic system will be triggered on condition that correlative IRP is captured. Once a IRP_MJ_DELETE request is captured, it will be discard and alert.

### 3.2 Semantic Checking

Experimental system use C++ and Perl script language to collect and standardize the data. C++ programmed module collect evidence include log file information, file access information. After relevant fields abstracted and standardized the time format, Perl script parses and generate XML structure document. Database interface receive and parse XML documents, reduce corresponding event. For example:

User 192.168.0.6 Get in time A;

Log file is modified in Time B.

Subsequently, Rule system (expert system) judges the event according to predefined rules, for example:

Verify B∈[A,μ]

or

B1<B2 (A and B are time format),

And warnings or alerts are sent to user. In general, hints are mainly warning. Fig.3 also showed the decision process.

## 4. Experiment and Conclusion

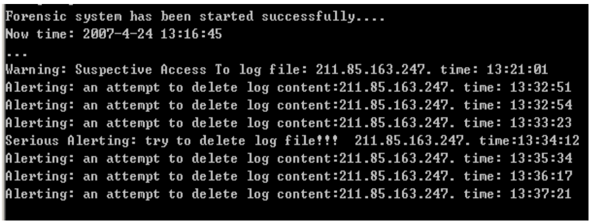
### 4.1 Experiment Results

Experiment environment include a server, a forensic computer and an attacking computer. The server has a Intel Pentium IV 2.93 GHz CPU, 1G memory, windows 2003 server operation system and IIS 6.0, other computers configure Inter Pentium IV 2.0 GHz CPU, 1G memory and windows XP operation system.

The following steps implemented the above scenario:

1. Install experimental system on the server.
2. Simulate normal operation.
3. Assume attacker gained access by using a network sniffer to steal password.
4. Attacker executed unauthorized access.
5. Attacker tried to delete correlative items in log file.
6. Attacker tried deleting the whole log file.
7. Same to step 5.
8. Forensic analyst runs the prototype expert system on the body of evidence.

Attacker tried to delete the items and log file. The output of forensic system is shown in fig. 5. A serial of warnings and alerts were sent to user.



```
Forensic system has been started successfully....
Now time: 2007-4-24 13:16:45
...
Warning: Suspective Access To log file: 211.85.163.247. time: 13:21:01
Alerting: an attempt to delete log content:211.85.163.247. time: 13:32:51
Alerting: an attempt to delete log content:211.85.163.247. time: 13:32:54
Alerting: an attempt to delete log content:211.85.163.247. time: 13:33:23
Serious Alerting: try to delete log file!!!  211.85.163.247. time:13:34:12
Alerting: an attempt to delete log content:211.85.163.247. time: 13:35:34
Alerting: an attempt to delete log content:211.85.163.247. time: 13:36:17
Alerting: an attempt to delete log content:211.85.163.247. time: 13:37:21
```

Fig.5 Result of the experimental forensic system

## 4.2 Conclusions

Since internet and web technology are open to users, its security need closer attention. Expert system is an effective measure to find attack and collect evidence. Experimental results prove the reliability and feasibility of our model. However, we have to solve the problem that system exhausted huge system resource. Meanwhile, because time stamps are key ingredient in deducing process, so the value of μ seems important to improve the accuracy of system.

## Acknowledgment

This work was supported partly by the China Department of Police under Grant #2007yycxhbst065 , China Hubei Province NSF under Grant #2007ABA151 and Hubei Province Department of Education NSF under Grant #2004D002.

## References

[1] Qian Guiqiong. Research and design of computer forensic system. Computer engineering. 2002,28(6).

[2] Chen Aili. Design of a log system supported computer forensic. Computer engineering and application.2003,39(15).

[3] Jiqiang L,Zhen H,Zengwei L.Secure Audit Logs Server to SupportComputer Forensics in Criminal Investigations[C].TENCON'02,Proceedings of IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 2002, 1:180-183.

[4] Abraham T,de Vel O.Investigative Profiling with Computer Forensic Log Data and Association Rules[C]. Proceedings of 2002 IEEE International Conference on Data Mining,2002:8-11.

[5] E. J. Friedman-Hill. Jess, The Rule System for the Java Platform. Technical report, Sandia National Laboratories, Livermore, CA, 2002. Cited 1 June 2003 http://herzberg.ca.sandia.gov/jess.

[6] H. A. Kautz. A formal theory of plan recognition and its implementation. In J. F. Allen, H. A. Kautz, R. Pelavin, and J. Tenenberg, editors, Reasoning About Plans, pages 69–125. Morgan Kaufmann Publishers, San Mateo (CA), USA, 1991.

[7] M. Minsky. A framework for representing knowledge. In The Psychology of Computer Vision, pages 211–277. Mc-Graw Hill, New York, 1975.

[8] D. A. Simovici, D. Cristofor, and L. Cristofor. Impurity measures in databases. Acta Informatica, 28(5):307–324, 2002.

[9] S. J. Templeton and K. Levitt. A requires/provides model for computer attacks. In Proceedings of the New Security Paradigms Workshop, Cork Ireland, Sept. 19-21, 2000.