A Rule-based Normal-form Model for Security Logs Analysis

Qian WANG, Sheng LU, Jian GONG
Computer Department, Southeast University
Nanjing 210096, China

Abstract
Based on the analysis of system logs, we can keep track of the behavior of systems that are under management. However, as the information contained in the original system logs is too much and too complex to be analyzed directly, efficient methods are required to strengthen the system maintenance and management. A new analysis method for security logs is proposed in this paper, which uses a set of normal-form rules for modeling the analysis. An example is arisen also in detail for explaining how the model works.

Keywords
Network security, Security management, System log analysis, Normal-form model.

1. Introduction
Over the past decade, significant progress has been made toward the improvement of computer system security. Unfortunately, since operating systems together with the related networking software and hardware utilities still have some kinds of security flaws, the reality remains that all computers are vulnerable to attacks from both non-authorized users (outsider attacks) as well as authorized users who abuse their privileges (insider attacks). So there is an obvious need for mechanisms that can detect both outsider and insider attacks.

Operating systems generate descriptions of their behavior in so-called system logs. With the contents of system logs, we can keep track of the behavior of systems that are under management. Therefore, analyzing system logs is an effective means of detecting attacks. However, because of the huge amount of log data generated, the traditional approach of manual analysis is impossible in practice. Within the past few years, there has been a steadily growing interest in the research and development of automated log analysis tools, referred to as intrusion detection systems. These tools try to provide one of the few practical methods of analyzing log data efficiently in order to aid the security administrators in identifying attacks.

Two important factors that make the direct analysis of original system logs difficult are related to the volume and diversity of logs. As operating systems produce mountains of logs, the analysis of them becomes problematic because overall system performance may be adversely affected. Moreover, each kind of system logs is in different formats, and modifications are demanded when expanding an analysis tool to process multiple system logs.

A new analysis method for security logs is proposed in this paper, which uses a set of normal-form rules to unify the analysis work. The basic idea is given in section 2. In section 3, we establish a rule-based normal-form analysis model of security logs. And an example is arisen in detail in

section 4 for explaining how the model works. Section 5 presents some concluding remarks.

2. Normal-form Analysis of System Logs

Generally speaking, the semantics contained in system logs can be classified as of statistic and event. That is, any item in a system log will describe either facts about an action, or facts about the result of the action. To a specific application, only certain information contained in the original system logs is concerned. For example, network management systems mainly emphasize on the statistic information (i.e., the result of an action), while intrusion detection systems primarily on the event information (i.e., the facts about the action itself). Therefore, when working in one specific field, we can predefine a relatively fixed normal-form log format, and preprocess the original system logs, viz. filtering out the irrespective content, selecting the needed items and translating them into the uniform normal-form format. Then the analysis is simplified by just processing the normal-form logs with a common analysis engine. Using this method, efficiency and portability are both achieved. The reduction of the amount of log data makes the analysis feasible. And all the modifications are limited to the preprocess function.

Security logs refer to the system logs that are security relevant, such as login log, ftp log, smtp log, etc. The analysis of security logs mainly concerns the event information, viz. when, where, and who do which action to whom. As according to ftp log, the required event information includes when, who, from where, to where, download/upload which file, while the statistic information like file number and file length transferred is of little interest. So the normal-form log format of security logs is a syntax and semantic definition of the event information contained in the logs.

The design of the normal-form log format ought to meet the following goals: completeness, extensibility, and simplicity. Completeness means that it must include all the needed information, or the logs are unusable. Extensibility implies that it must allow representing any other type, so the appearance of new logs doesn't force a change in the format. Simplicity means that it must be easy to be processed by the analysis engine.

In the automated analysis of security logs, the analysis engine acts as an expert system. The knowledge of intrusion detection, for example, known attack methods and signatures, known system flaws, expected system behavior, and the site-specific security policies, are encapsulated in rules. From the perspective of the rule-based analysis engine, the log data are viewed as facts, which map to the rules. A binding analysis is performed to determine if the fact/rule binding is consistent. The rules may recognize single events that represent attacks by themselves, or they may recognize a sequence of events that represent an entire attack scenario.

3. A Rule-based Normal-form Analysis Model

This model has three major operation components: data collector/preprocessor, rule-based analysis engine, and information archiver, which may reside on different systems. Since logs have their own data formats, each of them employs a different data collector/preprocessor. All the normal-form logs are analyzed with a central analysis engine based on a set of normal-form rules. The logs and results are archived in a repository for later investigating. Figure 1 puts the pieces of the entire model together in one diagram. New functions must be added to support the data

transmission, but it is beyond the scope of this paper.

Because the analysis of security logs primarily emphasizes on the event information, the normal-form log format commonly comprises the following elements: subject, time, location, and activity. Further, the time includes arrive time and leave time; location consists of source location and destination location; and activity is made up of object and action. Thus the format is as follows:

(subject, (arrive time, leave time), (source, destination), (object, action)).

Logically the rules have the form:

Antecedent ⇒ consequence,

where the antecedent is either an event characterized in logs, or a consequence of some previously satisfied rule, or any conjunction of these. Because the interested aspects of an event are expressed as elements in the uniform normal-form log format, the antecedent is a conjunction of canonical predicates, each of which corresponds to either an element of the log format or a consequence. Since conditional predicates can always be conversed into standardized prenex normal conjunctive form, the rules can be translated into the form of:

(set of conditional predicate clauses) ⇒ consequence predicates.

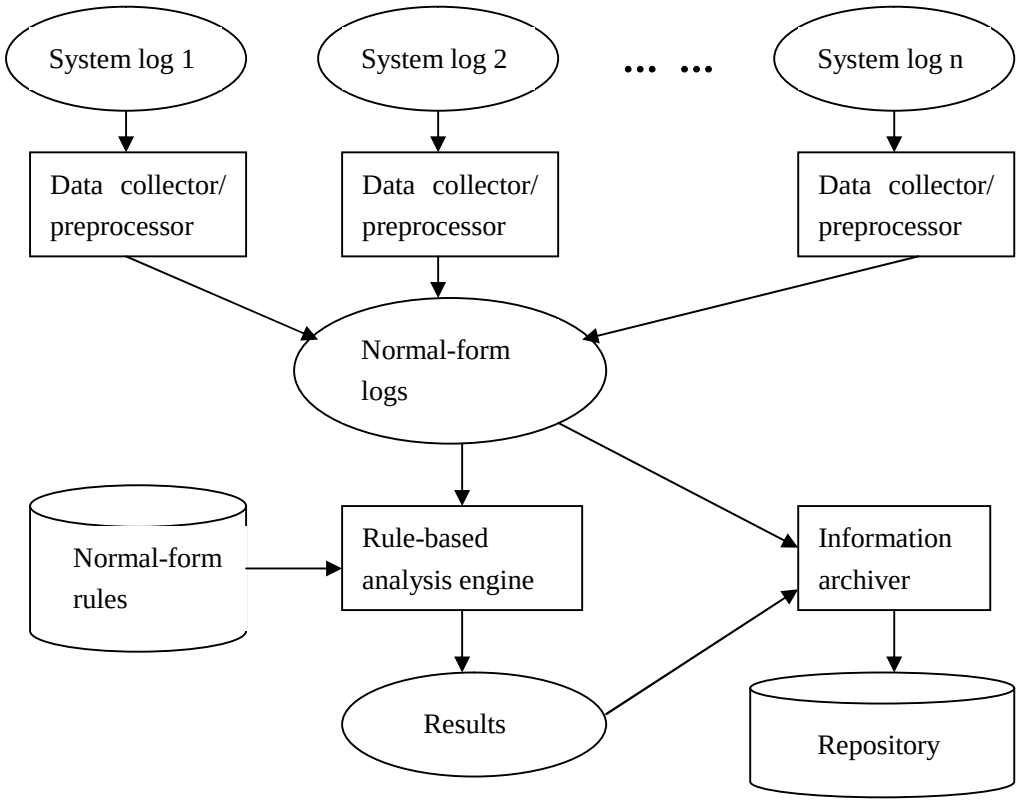

Figure 1. The Normal-form Analysis Model

Not all the elements defined in the normal-form log format are recorded in all the original system logs, so the normal-form log with some empty elements may appear, which makes the boolean values of some if-clauses uncertain. Hence, ternary logic with: T(true), F(false), U(undefined), is

used, and its truth value calculation rules are supplemented as following :

① U∧T=T∧U=U ② U∧F=F∧U=F ③ U∧U=U

④ U∨T=T∨U=T ⑤ U∨F=F∨U=U ⑥ U∨U=U ⑦¬U=U.

The analysis engine is responsible for reading new events described in the normal-form logs, attempting to apply the rules to the events, reporting suspected intrusions, and maintaining the various dynamic values associated with the rules. The deduction of one rule is transformed to the judgement of the satisfiability of if-clause set. When the input events and the previous consequences satisfy the antecedent, the consequence is deduced. Priority order criteria is applied to resolve the conflict among multiple rules.

4. An implementation of the model

A prototype of the rule-based normal-form analysis model has been implemented as a part of the VULCAN (VULnerability Capturer for Auditing in Network) system, a multihost-based security monitor developed in Southeast University in 1998. It employs a client/server model, where each monitored hosts collects and preprocesses security logs, and transmits the normal-form logs to a central VULCAN server via a secure channel. The VULCAN server analyzes the data for signs of attacks, produces reports or alarms as required, and archives the related information. Here an example is arisen in detail for explaining how it works.

The following rules are applied to two kinds of unix logs (wtmp and sulog):

Rule1: if an account, except "billy", successfully su to root, then the root is compromised by an insider user;

Rule2: if an account successfully login from an IP address beyond 202.112.23.0 – 202.112.23.255, then the account is compromised by an outsider user;

Rule3: if an account successfully login from an IP address beyond 202.112.23.0 – 202.112.23.255 and su to root, then the system is compromised by an outsider user.

The priorities are assigned in increasing order.

The normal-form log format is defined in section 3. Suppose there are two pieces of logs on the host "hanan":

① wtmp

billy pts/22 202.112.25.213 Thu Jul 2 11:44 - 12:16 (00:31)

② sulog

SU 07/02 12:01 + pts/22 billy-root.

They are translated into the following normal-form logs:

Log1:

(billy,(07.02.11:44,07.02.12:16),(202.112.25.213,hanan:22),(,login))

Log2:

(billy,（07.02.12:01,）,（, hanan:22),（root, su)).

There is a subset of the canonical predicates:

U(x,L) – x is the subject element of log L

AT(x,L) – x is the arrive time element of log L
LT(x,L) – x is the leave time element of log L
S(x,L) – x is the source element of log L
D(x,L) – x is the destination element of log L
O(x,L) – x is the object element of log L
A(x,L) – x is the action element of log L
Is(x,y) – x is y
Between(x,x1,x2) – x is between x1 and x2
P(L) – the root is compromised by an insider user
Q(L) – the account is compromised by an outsider user
R(L1,L2) – the system is compromised by an outsider user.

Using these predicates, the above rules can be expressed as follows:

Rule1':

$$\exists L \forall x((U(x,L) \supset \neg Is(x,billy)) \wedge (O(x,L) \supset Is(x,root)) \wedge (A(x,L) \supset Is(x,su))) \Rightarrow P(L)$$

Rule2':

$$\exists L \forall x((S(x,L) \supset \neg Between(x,202.112.23.0,202.112.23.255)) \wedge (A(x,L) \supset Is(x,login))) \Rightarrow Q(L)$$

Rule3':

$$\exists L1 \exists L2 \forall x \forall y \forall z(((U(x,L1) \wedge U(y,L2)) \supset Is(x,y)) \wedge ((S(x,L1) \wedge S(y,L2)) \supset Is(x,y)) \wedge ((D(x,L1) \wedge D(y,L2)) \supset Is(x,y)) \wedge ((AT(x,L1) \wedge LT(z,L1) \wedge AT(y,L2)) \supset Between(y,x,z)) \wedge (S(x,L1) \supset \neg Between(x,202.112.23.0,202.112.23.255)) \wedge (O(y,L2) \supset Is(y,root)) \wedge (A(x,L1) \supset Is(x,login)) \wedge (A(y,L2) \supset Is(y,su))) \Rightarrow R(L1,L2).$$

They are conversed into the following if-clause set form:

Rule1'':

$$(\neg U(x,L) \vee \neg Is(x,billy), \neg O(x,L) \vee Is(x,root), \neg A(x,L) \vee Is(x,su)) \Rightarrow P(L)$$

Rule2'':

$$(\neg S(x,L) \vee \neg Between(x,202.112.23.0,202.112.23.255), \neg A(x,L) \vee Is(x,login)) \Rightarrow Q(L)$$

Rule3'':

$$(\neg U(x,L1) \vee \neg U(y,L2) \vee Is(x,y), \neg S(x,L1) \vee \neg S(y,L2) \vee Is(x,y), \neg D(x,L1) \vee \neg D(y,L2) \vee Is(x,y), \neg AT(x,L1) \vee \neg LT(z,L1) \vee \neg AT(y,L2) \vee Between(y,x,z), \neg S(x,L1) \vee \neg Between(x,202.112.23.0,202.112.23.255), \neg O(y,L2) \vee Is(y,root), \neg A(x,L1) \vee Is(x,login), \neg A(y,L2) \vee Is(y,su)) \Rightarrow R(L1,L2).$$

Using the method of predicate logical decision, Q(Log1) and R(Log1,Log2) are deduced. Because the priority of rule3 is higher than that of rule2, R (Log1,Log2) is archived as the final result. The contents of Log1 and Log2 are also archived as the parameters of it.

5. Conclusion

This paper proposes a rule-based normal-form model for security logs analysis, which turns the intuitive judgement of system logs by human into automatic processing by machine. As the original system logs are reduced and unified into a predefined normal-form format, frequently periodic monitoring huge log information becomes possible, and reusability of the analysis tool is achieved at the physical level. The analysis engine acts as an expert system that codes the knowledge of intrusion detection in a set of normal-form rules. The first phase of our work

produces a prototype of the model, which performs basic functions of it. Future work continues on the design, development and refinement of rules, particular of those that can take advantage of the knowledge about particular kind of attacks. We try to consummate the analysis engine to make it a more intelligent and sensitive expert system. The control schedule of the analysis also can be optimized to justifying the monitoring period by security policies as well as the current status of the monitored system.

References

[1] Shaokui MO, *Symbolic Logic Tutorial*, Publishing House of Huazhong University of Science and Technology

[2] Xiangjin YANG, Qingsheng CAI, *Artificial Intelligence*, Science and Technology Literature Publishing House

[3] N. Habra, B. Le Charlier, A. Mounji, I. Mathieu, *ASAX: Software architecture and rule-based language for universal audit trail analysis*, Proc. of the 2nd European Symposium on Research in Computer Security (ESORICS' 92), Toulouse, France, Nov. 1992, 435 – 450

[4] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, D. Mansur, *DIDS (Distributed Intrusion Detection System) - Motivation, architecture and an early prototype*, Proc. of the 14th National Computer Security Conference, Washington, D. C., Oct. 1991, 167 – 176

[5] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, T. Grance, L. T. Heberlein, C.-L. Ho, K. N. Levit, B. Mukherjee, D. L. Mansur, K. L. Pon, S. E. Smaha, *A System for Distributed Intrusion Detection*, Proc. of the COMPCON, Feb./Mar. 1991, San Francisco, CA, 170 – 176

[6] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, P. G. Neumann, H. S. Javitz, A. Valdes, T. D. Garvey, *A real time Intrusion Detection Expert System (IDES) – Final Report*, SRI International, Menlo Park, CA, Feb. 1992

[7] D. Anderson, T. Frivold, A. Valdes, *Next-generation Intrusion Detection Expert System (NIDES): A Summary*, SRI-CSL-95-07, SRI International, Menlo Park, CA, May 1995

[8] G. G. Christoph, K. A. Jackson, M. C. Neumann, C. L. B. Siciliano, D. D. Simmonds, C. A. Stallings, J. L. Thompson, *UNICORN: Misuse Detection for UNICOS*, Proc. of the Supercomputing '95, San Diego, CA

一种基于规则的安全日志范式分析模型

王倩　陆晟　龚俭

(东南大学计算机科学与工程系　　南京　　210096)

【摘要】通过日志分析，能够有效地掌握系统运行情况、加强系统维护与管理。但由于原始日志数据量大，格式多变，难以对其进行直接处理，需要寻求更为有效的分析方法。本文面向安全日志，介绍了范式分析的思想，提出了一种基于规则的安全日志范式分析模型，并给出了实现示例。

【关键词】网络安全，安全管理，系统日志分析，范式模型

作者简介
王倩，女，生于 1974 年 8 月，主要研究方向网络安全