Data Gathering and User Behavior Analysis System

Igor V. Mashechkin, Mikhail I. Petrovskiy, Sergey V. Troshin, Andrew A. Shestimerov

Computer Science Department of Lomonosov Moscow State University

Abstract — **In this paper we investigate the problem of development effective program security systems that are used for defense against internal intrusions. We offer technology for constructing systems that protect against the internal intrusions based on non signature methods. These systems possess autonomy, adaptability and self-teach properties. We propose methods for information gathering from log files, for data consolidation, for data presentation, transmitting and storing. And the security analyst console and consolidations subsystem architectures are proposed. We show the applicability of OLAP technology for collected data analysis. Also we offer data mining algorithms for user behavior model construction based on association rules. These models can be applied for users' activity description and visualization or for users' behavior abnormality detection, including potential threat from each user. An experimental system was verified using DARPA Intrusion Detection Evaluation Program methodic. The results are shown in the paper.**

Index Terms — **Intrusion detection systems, Insider, User behavior modeling, Security data consolidation, OLAP, Data Mining.**

I. Introduction

Recently the actuality of a computer systems security problem does not express doubts. Using of standard information protection tools, based on the control of access rights, the control of integrity and authorization of users appears ineffective against specially planned intrusions. These intrusions can be organized from outside or by those users which have access to the computer system. Specialized intrusion detection and prevention systems (IDS) become more popular.

Computer system intrusion (or attack) [1] is any activity which compromises integrity or confidentiality or availability of the data being stored or processes by the computer system. The basic categories of intrusions are: R2L - unauthorized remote login to machine); not authorized reception of exclusive access rights (U2R - unauthorized access to root privileges); DoS - Denial of Service; scanning, probe - the configuration analysis and determining of weak places in protection of system.

U2R intrusions are the most dangerous and the most difficult attacks to detect. This type of intrusions is realized by the malefactor who already has access to the computer system or probably even appears to be its legal user. Potential damage from internal attacks is much more considerable than from the external ones. At the same time, an ultimate goal of many external attacks is to access the system with the purpose of the further lead attack "from inside".

«National Survey on Managing the Insider Threat» researches results published by Computer World magazine (www.computerworld.com) and joints reports of FBI and Computer Security Institute (www.gocsi.com) cited that:

- More than 90% of the companies and the organizations receive damage from internal intrusions, and 60% are actually suffering from it;
- More than 30% of an IT departments operating time is spending on detection of already occurred internal intrusions traces;
- Average Hi-tech USA Company spends approximately 1 million dollars per year for internal intrusions protection and is ready to increase these expenses.

Creation of a new system construction technology for effective protection against internal intrusions is very important today. The basic task of such systems is user's activity monitoring for the purpose of detection of not authorized or ill-intentioned actions. Computer system analyst is being informed about these actions.

Manuscript received April 10, 2007. This work was supported in part by RFFI Grants No. 05-01-00744, 06-01-00691 and 06-07-08035-ofi, Russian Federation President Grant MK-4264.2007.9.

S. V. Troshin is post-graduate student with Computer Science Department of Lomonosov Moscow State University, Building 2, MSU, Vorobjovy Gory, Moscow, 119992, Russia Phone: +7 495 9391789, Fax: +7 495 9391988 (e-mail: troshin@mlab.cs.msu.su).

M. I. Petrovsliy is associated professor with Computer Science Department of Lomonosov Moscow State University, Building 2, MSU, Vorobjovy Gory, Moscow, 119992, Russia Phone: +7 495 9391789, Fax: +7 495 9391988 (e-mail: michael@cs.msu.su).

I. V. Mashechkin is professor with Computer Science Department of Lomonosov Moscow State University, Building 2, MSU, Vorobjovy Gory, Moscow, 119992, Russia Phone: +7 495 9391789, Fax: +7 495 9391988 (e-mail: mash@cs.msu.su).

A. A. Shestimerov is specialist student with Computer Science Department of Lomonosov Moscow State University, Building 2, MSU, Vorobjovy Gory, Moscow, 119992, Russia Phone: +7 495 9391789, Fax: +7 495 9391988 (e-mail: vrand@mail.ru).

II. Related Work

Traditionally, intrusions detection systems are based on so-called signature methods [2, 3]. These methods usually consist of applications that are the set of rules declared by an expert or a security officer. This knowledge base describes the characteristics of possible intrusions scenarios or the rigid users' behavior. However the existing signature approach has a number of serious lacks:

1. Traditional signature systems can't detect new types of attacks before corresponding update occurs in the knowledge base. This fact leads to essential decrease in a level of safety for protecting computer system.
2. Internal intrusions detection systems constructed on the basis of signature methods are in need of "manual" adjustment. An administrator makes them under the concrete configuration of protected computer system. Such adjustment demands, both serious time expenses and high qualification of administrator.
3. There are the methods, allowing "to masquerade" intrusion in such a way, that by means of a set of rules it cannot be defined as not authorized action. The professional malefactor, having access to the knowledge base of protection system (it can receive it always, in particular, by having it as a legal user), almost can "deceive" traditional detection system of the intrusions, based on signature methods. Program tools (such as root kit), intended for fast development, updating and "masking" malicious code exist in «hacker community».
4. There is a class of potential users of intrusions detecting systems, that can't use external knowledge bases (and also their automatic updating contained by the third parties) because of safety reasons. The systems constructed on the basis of external knowledge bases, essentially depends on quality and efficiency of work of the companies that support knowledge bases updates, thus, activity of these companies becomes a vulnerable part in maintenance system of computer security.
5. The majority of signature systems work only within the limits of one source of the information on activity of the user (for example, carry out monitoring of file system usage or network traffic, etc.), using the rules specific to this source and policy of safety. But researches showed that one information source is not enough for internal intrusions detection. So it is necessary to carry out monitoring of many diverse parameters simultaneously, to consolidate received information in a uniform stream of events and to trace dependencies and correlations for polytypic security events.

Intrusions detection systems constructed on the basis of non signature methods use for analysis algorithms methods of mathematical statistics, probability theory, neuronet, genetic algorithms and trees of decisions. The majority of these methods possess one serious lack: they are not intended for work with the heterogeneously structured data of great volume, and also are very sensitive to presence of noise and outliers in a training set. As the result accuracy of the constructed models turns out to be low that leads to poor rate of recognition of intrusions and a high level of false positive operations.

To avoid these problems, in developed system it is supposed to use hybrid methods, in particular, on the basis of potential functions and indistinct sets. Such methods allow to process effectively great volumes of the diverse structured data, and also to consider the expert information of a subject domain.

Thus, it is necessary to develop technology for constructing systems that protect against the internal intrusions based on **non signature methods.** These systems must possess the following properties:

- **Autonomy** - independence of external knowledge bases and experts
- **Adaptability** and **learning** - ability to reveal new or obfuscated computer attacks and also automatically adjust to the changes in configuration of protected computer system and its users behavior.
- **Information fusion -** complex approach to the analysis of users' activity data due to simultaneous monitoring and the uniform analysis of diverse sources of protected computer system security information [4].

The listed requirements cause steady increase of computer security experts attention to the application of data mining and machine learning methods in intrusions reveal systems.

Object of research of the present work is the computer security systems intended for detection and prevention of internal intrusions and confidential information outflow.

Subject of research is the development of construction technology of program systems based on the monitoring and modeling of protected system users' behavior by the means of learning machine methods, intellectual data analysis and mathematical statistics.

The main idea of methods usage is based on the assumption that users' activity can be monitored and its mathematical model can be constructed. This will let us to find out infringements of security policy and anomaly in users' behavior. Many experts on computer security think that such approach is especially perspective in tasks of internal intrusions problems detecting because it allows creating so-called «early warning systems».

Internal intrusions detecting systems' operating experience shows, that in most cases direct internal intrusion is preceded by some abnormal (though probably legal) users' behavior , i.e. before attack or information thefts the user starts to perform some unusual actions. Detection of such abnormal actions allows to find suspicious users, to check them up, toughen and to correct security policies options for them and establish more detailed supervision over their activity even before any possible ill-intentioned action from their party. This allows to prevent many internal intrusions.

III. Our Approach

Not only development of algorithms that detect intrusions is interesting. The development of system's architecture is also important as it should meet the requirements of scalability and expansibility. In this connection, the multi-agent approach is offered to take as a base. In this case sensor controls and detectors should be realized as the external agents cooperating with the central analytical module, data warehouse and other components of system under the special report.

It will make possible to solve a productivity problem by duplicating of main system components, and to solve a problem of scalability and expansibility by creation of appropriate agents for new platforms supporting.

We offer a new technology for construction of monitoring and user behavior analysis systems. It provides data gathering, statistical reports formation and intellectual data analysis (Data Mining), namely association rules and behavior model construction, search for anomalies in user and software behavior. These systems are multi-agent software. The information sources are software and OS log files.

A. *General architecture*

The system consists of consolidation server, information gathering agents and analyst console (Figure 1). The gathering agents collect data from OS log files and send it to a consolidation server. Analyst console performs analysis on the collected data. Statistical reports, found associations, constructed models and revealed anomalies are transferred to analyst in the form of interactive dynamic reports.

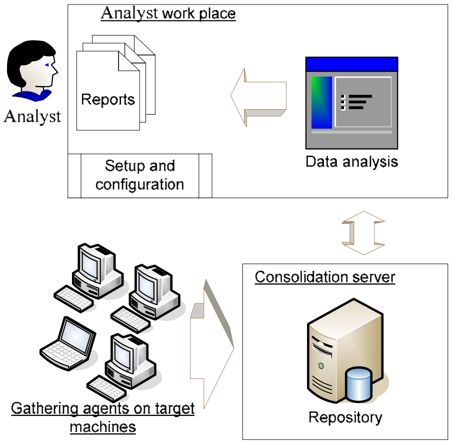

Fig. 1. System architecture.

Analyst work place is a program module which provides interfaces between system and user. The given functionalities are

1. Consolidation management (agent and consolidation server setup and configuration).
2. Data analysis management (data filtering, models training and so on).
3. Reporting and visualization..

B. *Consolidation subsystem*

Log records consolidation subsystem is a system for gathering and preprocessing information from software and OS log files. Consolidation subsystem solves problem of association in repository of all log records with granting the unified access interface, independent on structure of consolidated data. Consolidation subsystem architecture is shown on Figure 2.

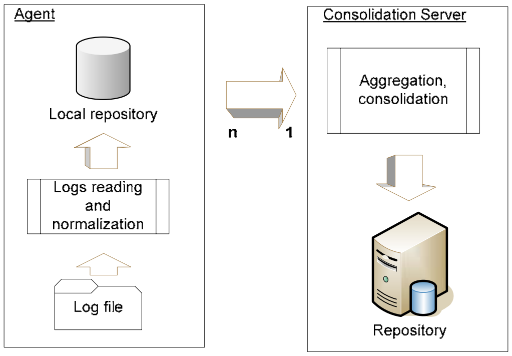

Fig. 2. Consolidation subsystem.

It consists of two parts: consolidation server and agents. Agent must be installed on each target machine. Theoretically agent can be realized for any operating system and/or log file. Also, there is no need to restart a consolidation server when installing new or updating existed agent. Agent contains a local repository for buffering the collected data before transmitting.

Consolidation subsystem is scalable, see Figure 3. In case of a large data flow additional consolidation servers can be used.

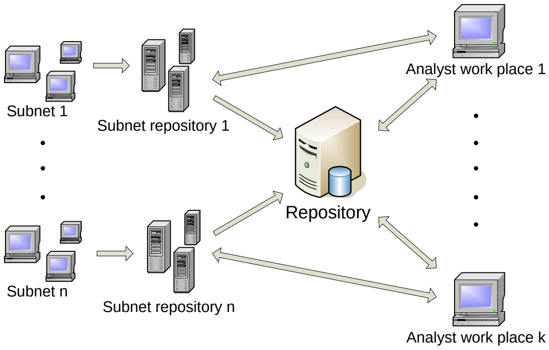

Fig. 3. Consolidation subsystem is scalable and distensible.

1) *Agent*

Agent collects, normalizes, buffers and then transmits the data events from log files to consolidation server. Agent configures standard log files and additional information collection modules. For realization of these functions the following architecture has been offered: configuration module, transfer module, local repository and a set of readers for different log types.

a) *Additional data collection*

Operation system auditing tools are enough neither for detailed user activity information construction nor for anomaly user behavior revelation [7].

For example, analysis system uses only the following

Windows auditing facts:

- Logon/logoff (528-540, 682-683 event types in event log);
- Account management events (624-644);
- Object access events (560-565);
- Privilege use;
- Process tracking events (592-595);
- System events;
- Policy change events;

Additional information data gathering modules are necessary. Consolidation subsystem must provide an interface for interaction with this kind of modules. The set of modules must be easy to extend.

Suggested technology doesn't restrict a potential set of data sources and data collection modules. A specific interface between agent and additional data collection modules, which uses log files, was developed.. Every gathering module writes discovered information (activity facts) into a log file. At the same time agent reads log file records as usual. Agent also controls additional modules. Today the following data gathering modules has been developed:

1. Network activity, detailed for each user, process, address, local and remote ports and so on.
2. User activity in applications – mouse and key activity in applications.
3. PnP and removable devices usage (CD-ROM, flash drive and so on)

2) *Consolidation Server*

Consolidation server receives data from agent, converts data to internal binary format and adds it to repository. Consolidation server consists of data receiving module, data import queue and data inserting module and configuration module. See Figure 4.

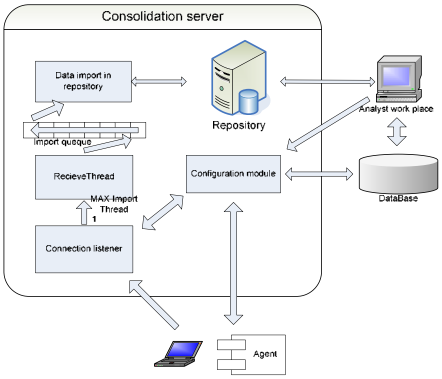

Fig. 4. Consolidation server architecture.

In the case of wide data flow from agents to consolidation server (usually, then agents count is greater than 1000) it is possible to use a few consolidation servers at the same time. Consolidation server uses only SSL secured connections for agent identification.

3) *Repository and Data Notion*

Repository must provide efficient data insertion and selection tools. Each log file record defines a set of attribute (attribute name, attribute value and attribute value data type). Entire log record structure consists of common for all log types attributes and additional attributes – peculiar for current log type. Usually an XML format [5] is used for semi structured data representation. But in this case it is unacceptable because of addition overhead charges. A new specific data format was developed.

Each log record is separated between seven files: a file for common parameters, a file for additional parameters and references files. Using this structure it is easy to find log records by common parameters such as date, time, user name, computer name and event id. Use of common references reduces disk usage quota.

Consolidation server repository structure is organized like tree of folders. Leaves are files for common and additional parameters (attributes). See Figure 5. Common and additional attributes files are placed in computer folders. They can be separated, for example, by days. (for data selection acceleration needs). References are common files and can be arranged in any folder.

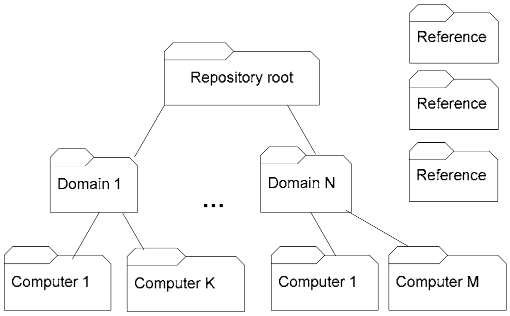

Fig. 5. Consolidation server architecture.

References are organized as follows: for each attribute hash function is calculated. In the case of collision it is calculated once more. Each hash function value is associated with file pointer shift. This association is supported by binary search tree. Such technique provides an effective value adding and searching by value operations. Also using this approach there is no need to determine the size of reference file. It is limited only by file system.

For the case of abnormal or emergency situations shadow copies mechanisms were implemented.

C. *Analysis subsystem*

Today we integrated two data analysis technologies: OLAP and Data Mining.

a) OLAP

OLAP [6] technology can be applied to each log type. OLAP cubes provide detailed and flexible statistical information visualization. We created OLAP cubes for each user activity facts (See Figure 6). We use pivot tables and pivot charts for OLAP cubes visualization.

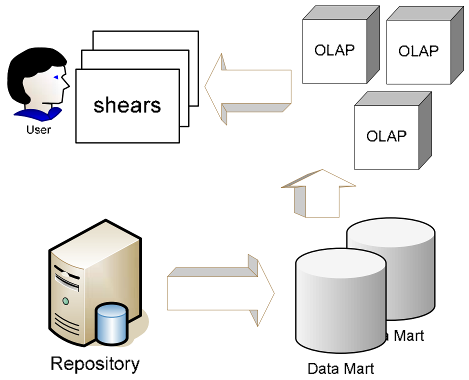

Fig. 6. Use of OLAP technology.

Each OLAP cube has dimensions, hierarchies and measures

(1) Dimensions, hierarchies, measures

The following OLAP cube parameters (dimensions and hierarchies) are common for all fact types:

1. Computer name (hierarchy: domain -> computer);
2. Event time (hierarchy: Year > Month > Week > Day > Hour > …);
3. Process (hierarchy: computer > image file name);
4. Parent process (hierarchy: computer> image file name);
5. Source (hierarchy: computer> path > file);
6. Source type (no hierarchy: file extensions and so on);
7. Login type (no hierarchy);
8. Operations (no hierarchy: for example, descriptors open/close operations, processes start/stop, changes of access right and so on);
9. User (hierarchy: domain > user);
10. Completion status (no hierarchy).

The common OLAP measures are:

1. Operations count;
2. Active time (duration): duration between start and stop process operations, for example;
3. Operations count in unit time

Also each fact type has its own set of measures. For example, network activity facts have the following measures: bytes sent and bytes received.

b) *Data Mining*

Using data mining approach on information from log files allows finding the latent, substantial and potentially useful laws, to build associative rules, to find exceptions and anomalies [8, 9, 10].

The basic idea is based on user behavior modelling. Users are combined into groups according to their roles in operation system (administrators, common users and so on). In that case user behavior model typical for a group (common group profile) can be constructed. Then, an accordance of current user activity and common group profile can be analyzed. Using this approach we solve both the abnormality and intrusion detection problems (See Figure 7). If current user behavior doesn't correspond to his/her group profile, it should be referred to anomaly activity. However, if current user activity corresponds to any other group profile (for example, ordinary user activity correspond to administrators group profile), it should be referred to violations.

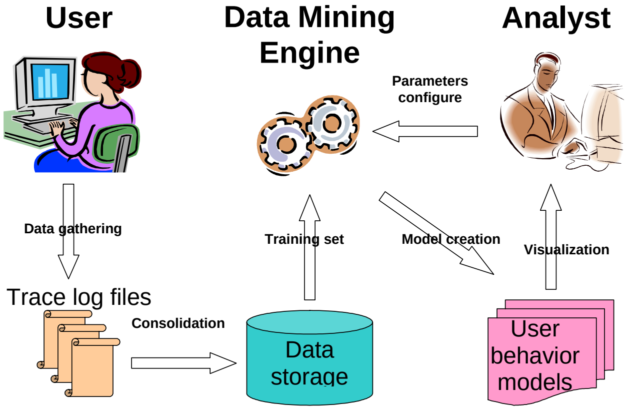

Fig. 7. Data mining techniques implementation scheme.

System uses the following behavior models [11]:

- Fact's attribute correlation retrieval for the purpose of common users activity describing. Association rules algorithm is used.
- Statistical anomalies retrieval (from OLAP shears).

IV. Experiments

The Information Systems Technology Group (IST) of MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA ITO) and Air Force Research Laboratory (AFRL/SNHS) sponsorship has collected and distributed the first standard corpora for evaluation of computer network intrusion detection systems.

These evaluations measure probability of detection and probability of false-alarm for each system under test.

Realized system anomaly detection method was tested on DARPA dataset. You can find the results in Receiver Operating Characteristic curves format on Figure 8.

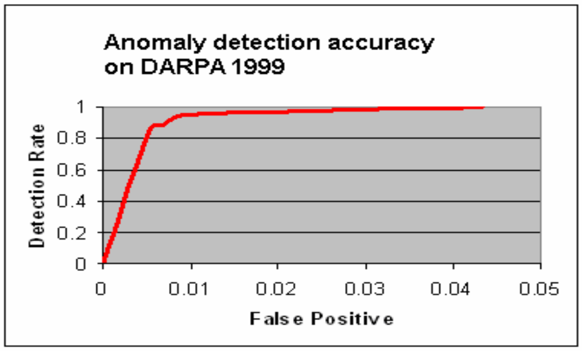

Fig. 8. Receiver Operating Characteristic curves.

Detection rate 87% and false positive 0.5% corresponds to optimal tuning. Detection rate 100%, false positive 4% corresponds extreme tuning (all attacks were discovered).

V. Conclusion

The new program technology based on non-signature methods of internal intrusions revealing is created. This new program technology is capable to find out types of intrusions that are already known (but "obfuscated") and also types of internal intrusions unknown before. Intrusions retrieval is made by using machine learning algorithms and methods. These methods do not depend on external knowledge bases, they dynamically adapt for changes in functioning of protected computer system and to features of each separate system user. The developed technology is based on principles of modeling behaviors of protected computer system users.

We developed the architecture of intrusions detection system. This architecture uses multi-agent approach and the centralized repository of data. It allows to accumulate security events from diverse sources in one general data warehouse, and lets systems constructed on the given decision basis, be scaled, supporting various operational systems and platforms, due to development of new agents for these platforms.

The developed architecture, functioning algorithms and the program prototype of system that detects intrusions, possesses the following competitive advantages in comparison with existing domestic and foreign analogues:

- Maintenance of complex protection from internal intrusions due to gathering and consolidation of security information from various sources in one data warehouse;
- Detecting the "obfuscated" and previously unknown types of intrusions and also the independence from external knowledge bases;
- Application of analytical methods to estimate the potential threat proceeding level from each system user with the purpose of duly adjustments and individual security policies updating.·
- Dynamic adaptation to changes of functioning of protected computer system and to features of each individual system user behavior.

References

[1] A. Lukatskiy: Intrusion detection. Saint Peterburg.: BHV-Peterburg, 2001: pp. 50 - 200. (in Russian)

[2] Theuns Verwoerd, Ray Hunt: Intrusion Detection Techniques and Approaches. Department of Computer Science University of Canterbury, New Zealand, 2002, pp. 2-14.

[3] Kathleen A. Jackson: Intrusion detection system (ids) product survey. Distributed Knowledge Systems Team Computer Research and Applications Group Computing, Information, and Communications Division Los Alamos National Laboratory Los Alamos, New Mexico USA, 1999: pp. 6-22.

[4] Cristina Abadyz, Jed Taylory, Cigdem Senguly, William Yurcik: Log Correlation for Intrusion Detection: A Proof of Concept. Department of Computer Science, University of Illinois at Urbana-Champaign, 2003: pp. 3-6.

[5] Lee Dolz: XML and databases? Trust your intuition. [HTML] (http://www.iso.ru/journal/articles/206.html) (in Russian)

[6] A. Fedotov, H. Elmanova: OLAP introdution. [HTML] (http://olap.ru/basic/OLAP_intro1.asp) (in Russian)

[7] Lincoln Laboratory Massachusetts Institute of Technology. The Detectability of Attacks in NT Audit Logs [HTML] (http://www.ll.mit.edu/IST/ideval/docs/2000/ntaudit-table.html)

[8] SQL Server 2005 Books Online. Microsoft Association Algorithm [HTML](http://msdn2.microsoft.com/en-us/library/ms174916.aspx)

[9] Han J., Kamber M. - Data mining: concepts and techniques., 2000: pp. 279-310.

[10] Mikhail Petrovskiy: A data mining approach to learning probabilistic user behavior models from database access log. Faculty of Computational Mathematics and Cybernetics, Moscow State University.

[11] I. Mashechkin, M. Petrovskiy, S. Troshin: Data Gathering and User Behavior Analysis System. Faculty of Computational Mathematics and Cybernetics, Moscow State University.