DATA SECURITY MANAGEMENT

SOFTWARE TOOLS FOR DETECTING MISUSE ON UNIX SYSTEMS

Stephen E. Smaha and Jessica Winslow

INTRODUCTION
Past experience has shown that most computer misuse is perpetrated by authorized insiders, and it may be difficult to prevent abuse of authorized privilege. In addition, increased network connectivity over the Internet and other public networks greatly magnifies the risk of penetration by highly skilled outsiders.

In networks of UNIX machines it is especially easy for a user to move from machine to machine and use different user identifiers on each one. Because each machine recognizes only the functions it sees, the individual audit trails or security logs are not able to tie together all these activities across machines as belonging to a single user. The sheer volume of network communications traffic and the evanescent nature of network connections make it difficult to ensure accountability without continuous monitoring.

To detect computer and network misuse, it is essential to increase accountability for user activities on computers and networks. Accountability requires reliable and consistent authentication of users plus the trustworthy association of user identities with user activities. Monitoring without user-level accountability is

PAYOFF IDEA
The prevention of computer misuse is a major concern to many public and private sector organizations that need to provide assurance of data confidentiality, integrity, and availability. Improved access controls, better password mechanisms, and encryption of data can prevent misuse on networks and hosts. However, these controls do not guarantee detection of all problems that actually occur. This article discusses how to select software tools for detecting misuse on UNIX computers, reviews important features of misuse detection tools, and addresses issues that may arise when implementing these tools.
inadequate for most environments; in only a few situations would an organization not care if an employee were responsible for a breach of system security. Throughout this article, we use “misuse detection” to refer to security violations by insiders. “Intrusion detection,” on the other hand, refers to security violations by outsiders.

A new generation of misuse detection tools is becoming available. These allow systems managers to discover the source, time, and type of misuse shortly after its occurrence. When used in conjunction with improved access controls, better password mechanisms, and encryption of data, these misuse detection tools can provide a higher degree of assurance regarding data confidentiality, integrity, and availability. They can also provide a mechanism by which to respond to incidents of misuse.

This article discusses how to select software tools for detecting misuse on UNIX computers. Advantages and disadvantages of each of the five types of tools are presented. The article describes the important features system managers should look for when selecting a misuse detection tool and discusses issues that must be addressed in implementing these security software tools.

INFORMATION SECURITY GOALS
An organization’s computer and network security goals can be viewed in terms that are analogous to the law enforcement model of prevention, deterrence, investigation, and interception. Given these security goals, it is useful to examine if and how particular security tools satisfy these objectives.

Prevention
Implementation of appropriate measures to increase the difficulty of performing unacceptable activities can prevent criminal behavior.

Deterrence
The credible threat of detection is a major deterrent to criminal behavior.

Investigation
After a crime has been committed, the law enforcement goal is to identify the criminal, collect all relevant evidence, and manage the investigatory process.

Interception
A criminal identified in the act of committing a crime is more likely to be prosecuted and convicted.
DATA SOURCES FOR MISUSE DETECTION

There are four fundamental data sources used for detecting misuse in UNIX systems:

• Audit trails and other security logs
• Network traffic data
• Specialized network devices
• Keystroke capture

The data available in any particular situation varies substantially among user organizations and UNIX platforms. A major technical problem in designing misuse detection tools is the difficulty in integrating more than one type of data into one product because of the vast differences in their source, format, and availability.

AUDIT TRAILS AND OTHER SECURITY LOGS

The richest information available is from operating system audit trails, which are designed to provide accountability. Most versions of the UNIX operating system are capable of extensive event logging, in addition to the much weaker system accounting files. These audit trails were typically introduced to meet U.S. Department of Defense C2 security rating, a standard required by many U.S. government customers.

The old rationalizations for disabling audit mechanisms (e.g., excessive processing and storage overhead) are mostly fictions in this era of cheap processors, large disks, and high-volume tape backups. In addition to audit trail data generated by the operating system, some applications programs and database management systems generate log files with audit trail characteristics, including timestamps and user identity information.

NETWORK TRAFFIC DATA

Network traffic logs, especially from multiple points, are as valuable as audit trails, but they may become incredibly voluminous if they must be stored. Depending on whether only network packet addressing information is captured (i.e., traffic capture) or if everything on the network is captured and stored (i.e., traffic and content capture), a busy network can generate large quantities of data every second, and require far more storage hardware than most sites could afford. However, if network data can be analyzed for important events in real time and does not have to be stored, this is a feasible data source for misuse detection.

It is also tricky to combine network data with host-based information because network events are described in very different terms from host audit trail events. For example, network events usually contain informa-
tion about hosts (e.g., network addresses), whereas audit events usually describe actions by authenticated users. When logging is done at the network level, network events typically report the source and destination network addresses, protocol and service types, and similar information recorded at the network packet level. At the network level, nothing is known about individual users at a particular host. Host audit events, on the other hand, have access to all the information that the host has about its users. This typically includes user IDs, group memberships, changes in privilege, and where the login originated. Another problem is the reliability of data in network logs, because spoofing a network address is relatively simple.

Eventually, it will become very difficult to utilize network data, because the increasing bandwidth of most corporate networks already exceeds the ability of network sniffers to collect a fully utilized network backbone. And as encryption technologies become more widespread, network sniffers will not be able to look inside encrypted packets to see requested actions of the destination by the source address. Because the technologies required to implement this monitoring are widely available, this has been a busy area of product development. Comparisons of these products are outside the scope of this paper.

Commercial products that look at network data include:

- Internet Security Systems’ RealSecure
- Netect’s Netection
- Network Associates’ CyberCop
- Network Flight Recorder
- Secure Network’s Ballista
- WheelGroups’s NetRanger

SPECIALIZED NETWORK DEVICES
Two examples of specialized network devices are firewalls and authentication servers. Internetwork firewalls, designed to keep unauthorized external users from penetrating a private network domain, may be designed to report unusual traffic or repeated attempts to break into the domain. A central authentication system such as a smart-card system or a Kerberos authentication server can report which users have provided appropriate credentials, as well as failed login attempts.

KEYSTROKE CAPTURE
Few environments are able to support full-scale keystroke capture logs for all users. These logs can make an impressive evidence base for a legal case if the data has been appropriately protected. Aside from the obvious privacy problems, the biggest problem is the difficulty in performing au-
Automated analysis of keystroke logs. We do not yet have analysis engines that can understand both natural language and the contextual meaning of a large number of computer commands. Keystroke information can be ambiguous from a security point of view. For example, most modern operating systems allow simple aliasing of commands, so the command to change a user’s privileges could be invoked with an innocuous print command, which would be hard to flag as suspicious without detailed knowledge of the behavior of all the commands.

FIVE TYPES OF SECURITY TOOLS

Five types of security tools are available to address an organization’s computer and network security goals. These tools provide the following services:

- Identification and authentication
- Access control
- Static analysis
- Policy administration
- Dynamic analysis, with subtypes of anomaly detection and misuse detection

Exhibit 1 shows how each type of security tool supports the four security goals. As shown in the exhibit, no individual security tool satisfies all four security goals.

IDENTIFICATION AND AUTHENTICATION

Current UNIX operating systems typically provide simple identification and authentication of users with user-generated passwords. Third-party identification and authentication products can improve the quality of passwords by preventing use of easily guessed passwords and by enforcing password aging, for example. Or they may replace entirely the standard password mechanism. Although identification and authentication tools help prevent unauthorized access and may deter misuse by making

<table>
<thead>
<tr>
<th>Security Goal</th>
<th>Identification and Authentication</th>
<th>Access Control</th>
<th>Static Analysis</th>
<th>Policy Administration</th>
<th>Anomaly Detection</th>
<th>Misuse Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prevention</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterrence</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Investigation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Interception</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
the login process more rigorous, they usually do not provide information for investigating and intercepting incidents of misuse.

Available identification and authentication products include password checkers, Kerberos authentication systems, public key-based authentication systems, and token-based authentication schemes.

ACCESS CONTROL
UNIX operating systems check permissions attached to all files and devices and allow access only to authorized user IDs. This prevents unauthorized access as well as deterring attempts to gain access.

Firewalls provide access controls for networks and network hosts. They protect networks from intrusions by providing a single point of entry into the network by means of hardware, software, or a combination of hardware and software controls. File-level access controls are built into UNIX.

Access controls may deter unsophisticated intruders after access is denied. However, these controls do not provide independent assurance that access controls are not bypassed and are therefore of limited usefulness for interception. In addition, access controls do not provide data for investigating misuse, and require a great deal of planning and ongoing system administration.

There are few add-on file-level access controls for UNIX, although Tivoli’s Tivoli Management Framework and Computer Associates’ CA-Unicenter provide convenient interfaces to existing access control features in UNIX. Firewall software products include Raptor Systems’ Eagle and Netgate, CheckPoint’s Firewall-1, and Trusted Information Systems’ Gauntlet; free software packages are also available on the Internet. Hardware-based firewalls such as Storage Technologies’ BorderGuard are often based on commercially available routers. Memco’s SeOS is another commercially access control software package.

STATIC ANALYSIS
Static analysis software tools look for vulnerabilities in the security-relevant parts of a computer’s file system at a single point in time. These tools ensure that the system’s security state and configuration files have not changed in undesirable ways since the prior check. They also verify access controls on system files, check passwords, and conduct limited searches for known viruses or Trojan-horse programs.

Static audit tools may be further broken into local versus remote static audit tools. A local tool must execute as a process on the device under test, like COPS, though it may send the results of its tests to a central point for consolidation or reporting. A remote tool can test multiple remote devices; it operates by sending TCP/IP packets to the remote device and studies the network-level behavior of the device-under-test without having any processes or agents installed on the remote device.
Static analysis software prevents problems by alerting system managers to weaknesses in system configuration. They are usually simple to operate. Periodic use of static analysis software is recommended to identify changes in security-relevant files.

There are also limitations to the effectiveness of these tools. First, static analysis tools do not provide accountability, that is, they cannot determine who made the changes or how the changes were made. For example, static analysis tools cannot differentiate between authorized actions of a system manager and unauthorized changes resulting from misuse. These tools also do not deter or intercept misuse, and they usually do not provide evidence for investigating misuse. Their effectiveness is somewhat limited in more secure, better administered sites, and generally needs to be complemented with tests that occur inside the device.

Several free products for static analysis are available on the Internet, including:

- Purdue University’s COPS
- Purdue University’s Tripwire
- SATAN

Commercially available products include:

- AXENT’s Security Toolkit
- Intrusion Detection’s Kane Security Analyzer
- ISS’s Internet Security Scanner and related products
- Lucent’s PingWare
- Qualix’s NetProbe

POLICY ADMINISTRATION

Policy administration tools allow a centralized authority to define and propagate the access control or authentication information required by network or computing devices. They may enhance the underlying security mechanisms of the preexisting devices. Two tasks associated with policy administration are setting up existing policy/ACL mechanisms and creating and configuring new policy mechanisms.

Commercially available products include:

- Datalynx’s Fortress
- AXENT’s OmniGuard/Enterprise Resource Manager
- Single sign-on packages from various vendors

DYNAMIC ANALYSIS

In contrast to static analysis tools, which analyze at a snapshot of the system, dynamic analysis tools use an ongoing data source, such as audit
trails and network traffic logs, which provide a connection between logged events and user IDs or other authentication information. As a result, these tools provide increased accountability for user actions on computers and networks. Two approaches to dynamic analysis are used to identify security problems: anomaly detection and misuse detection.

Anomaly Detection
Anomaly detection looks for statistically anomalous behavior; it assumes that intrusions and other security problems are rare and that they appear unusual when compared to other user behavior. Anomaly detection tools are generally used to check events against predetermined thresholds; for example, they may post an alarm when three failed logins are noted.

The typical way to perform anomaly detection is to compare actual user behavior with a behavioral profile, usually based on historically expected events. Anomaly detection assumes that most people behave in a standard way. Empirical studies of computer use show that most users employ a very limited set of system commands and applications; as would be expected, programmers and system administrators are major exceptions to this finding.

Anomaly detection tools can intercept some forms of misuse, particularly masqueraders (i.e., individuals using someone else's login ID and password), by noticing changes in the victim's behavior in real time. It should be noted that the masquerader problem is mostly resolved through the use of common technologies for identification and authentication, like hardware-based tokens or software-based key certificates.

Anomaly detection has been most effective when applied to homogeneous transactional data, such as records of credit card transactions, securities trading, or telephone calls. It can also provide useful management reports on use of system resources. In general, most known attacks are not detected with anomaly detection techniques.

In addition, not all activities identified as anomalous are necessarily unauthorized or criminal. Users' activities often change, so that profiles of user behavior must constantly change for anomaly detection techniques to be effective. Without constant updating, anomaly detection tools tend to generate false alarms. Because of false alarms, such tools may not be amenable to automated operation, and may require a significant amount of operator time.

Anomaly detection research prototypes are computationally intensive, using complex statistics. The operating system audit trails required to provide user accountability are not transactional in nature and are difficult to process with the statistical techniques used in anomaly detection tools.

Commercially available anomaly detection tools include:
Misuse Detection

Misuse detection looks for patterns known to reflect misuse of the system (e.g., known attacks and attempts to exploit vulnerabilities) and also for known outcomes of misuse.

Misuse detection tools are more sophisticated than the other types of security tools, and they meet three of the four security goals. These tools intercept unauthorized activities, investigate suspected problems, and deter misuse by providing accountability checks for users activities. They assume other control mechanisms attempt to prevent problems, but do not assume those mechanisms are effective.

The mechanisms for detection can use pattern recognition techniques or expert system rules.

There are two steps to misuse detection:

- Defining the sequences of observable events that accompany known bad behavior; and
- Detecting instances of those signatures in the available data streams.

There are a number of advantages and disadvantages to misuse detection tools. Among the advantages, these tools provide user accountability and misuse detection by determining who did what when and how it was done. This may reduce an organization’s liability in lawsuits by providing accountability for user actions. Misuse detection is generally more efficient than anomaly detection because it scans for particular items, while anomaly detection constructs aggregates of items and matches them against statistical profiles.

In addition, misuse detection techniques generally result in fewer false alarms than anomaly detection techniques because every alarm from a misuse detection tool is a report of a declared misuse, not just an unusual occurrence. Misuse detection tools have been successfully tested for effectiveness using audit trail data of known break-ins. (In a May 1993 challenge to developers of intrusion and misuse detection software issued by the Federal Bureau of Investigation and the National Security Agency, Trusted Information Systems’ Stalker succeeded in detecting all staged attacks.) Misuse detection generates reports with evidence that can be used in investigating misuse and prosecuting intruders.

There are disadvantages to these tools as well. Currently available misuse detection tools are designed to process audit data files. Only recently have they become able to operate in real time. While they do not prevent misuse, and must depend on other mechanisms for preventing intrusions.
and attacks, more recent products allow automated reconfiguration and response to the detected incidents.

Deployment of misuse detection tools requires an organizational commitment to collect, store, and manage audit trails, which may add administrative and hardware overhead to the cost of the tool. And although the availability of attack and vulnerability information is improving, vendors, government agencies, and CERT have strongly resisted disseminating this information for a variety of reasons, including fear of lawsuits and data classification concerns.

Commercially available products include:

- Centrax’s Entrax
- Intrusion Detection’s Kane Security Monitor
- SAIC’s CMDS
- Trusted Information Systems’ Stalker, WebStalker, and ProxyStalker

SELECTION AND IMPLEMENTATION ISSUES

An organization’s needs for misuse detection vary depending on the system configuration and types of data processed and applications used. Misuse detection tools can provide an increased level of assurance of data confidentiality, integrity, and availability.

There are several important features that system managers should consider when acquiring a tool for misuse detection to ensure that it will work effectively in their environment. When evaluating a misuse detection tool, systems managers should ask the following questions:

- Does the tool work in the typical distributed UNIX environment (multiple-segment networks with many attached computers)?
- Does the tool operate automatically after it has been configured?
- Can the tool be easily updated to incorporate new information about attacks and misuses? When a new UNIX system vulnerability is discovered and exploited on one computer, a hacker usually attempts the same attack on multiple computers. In addition, as knowledge of these attacks spread throughout the hacker community, similar attacks begin to propagate across the Internet.
- Does the misuse detection tool violate user privacy by looking at keystrokes, electronic mail, or user files? These activities alienate users and may be legally problematic, thus making the tool ineffective.
- Can the tool simplify the management of system audit trails? Management of system audit trails includes the collection, storage, and routine disposition of the files for disposal or long-term retention. These management activities can be time consuming, depending on the number of audit trail files generated daily.
• Is the reporting capability of the tool flexible enough to allow connections with such event notification methods as pagers and network management control centers?

System managers should recognize that misuse detection tools are a very important part of their overall computer and network security program. But no one element of a security program is sufficient by itself. Systems managers should continue to rely on external sources of information, including direct observation of human and system behavior. Because misuse detection tools are extremely powerful, systems managers must ensure they are installed properly and used appropriately. The following section addresses these implementation issues.

USING EXTERNAL SOURCES OF INFORMATION
Information from external sources such as informants and administrative reports indicating possible computer misuse should prompt system managers to investigate. For example, a local user or system manager may report that a machine is operating erratically or may report an apparent attack by an unknown user.

In some cases, reports of misuse may come from a third party that uses more sophisticated detection techniques than are available locally. For example, in cases involving the Internet in which multiple sites are affected by a given incident, a site with sophisticated monitoring tools is most likely to discover the attack; it would then notify other sites involved in the same chain of attacks.

Internal administrative information about employee status can alert system managers to computer misuse. Disgruntled employees or former employees can cause substantial damage to computer systems, especially by masquerading as other employees who are absent from work. Information from risk analyses, analysis of network topologies and systems configurations, and assessment of vulnerabilities of a site's hardware and software may indicate possible sources of abuse.

Regardless of its source, relevant information always requires further investigation and validation to establish accountability. This investigation depends on details, provided by misuse detection tools, describing who did what and when it was done.

CONFIGURING THE DETECTION TOOL
Proper configuration and control of the misuse detection tool is essential. Because of the sensitive information produced, only authorized personnel should access the tool. In addition, the tool should be configured to discover misuse by using all available data. Finally, periodic adjustments
to the configuration, such as revising the types of audit trail events collected, can reveal possible deficiencies in the configuration.

**REPORTING RESULTS**

Systems managers should review reports frequently for evidence of misuse perpetrated by individual users and to investigate the possible misuse by establishing a schedule for detailed review of the identified users' activities on the system. In addition, the systems manager can establish an automated schedule for prompt notification of suspicious activities. Timely reporting of the results is essential to prevent the escalation of ongoing misuse and to obtain information for a criminal investigation. For maximum effectiveness, these reports and notifications should be limited to authorized personnel. Setting up a hierarchical reporting system can facilitate communication to the proper administrative officials and minimize the impact on human resources.

**CONCLUSION**

A wide range of computer security tools is available today for the UNIX operating system. The choice among tools that provide security goals of identification and authentication, access control, static analysis, or dynamic analysis depends on a particular organization’s needs and budgetary restrictions. Examination of each security tool’s benefits and drawbacks will provide a clearer view of available options. Exhibit 2 summarizes the features of each type of tool.

Both identification and authentication tools and access controls are more efficient in prevention and deterrence than in investigation and interception of system misuse. Identification and authentication tools prevent unauthorized access (thereby deterring misuse) by making a more user-specific login process. Access control tools protect networks by allowing access to the system by only one point of entry. Implementation of either requires considerable planning and system administration.

Static analysis tools alert systems managers to vulnerabilities in a computer system at a certain point in time by comparing present results to previous results. While they are helpful in preventing misuse, the efficiency of these tools for deterrence, investigation, and interception is limited because they cannot provide accountability for changes within a system.

Dynamic analysis can be divided into two approaches: anomaly detection and misuse detection. Anomaly detection provides interception and deterrence of misuse by comparing actual user behavior with a statistical, behavioral profile. Because activities identified as anomalous may not necessarily be unauthorized, their value in detecting system attacks is limited. Misuse detection tools seek patterns that can represent attacks on a system and compare data to typical results of misuse. They also pro-
vide accountability for user actions. Although these tools can intercept unauthorized activities and investigate possible attacks, they do not operate in real time and therefore are not effective in preventing intrusions.

BIBLIOGRAPHY


Stephen E. Smaha is vice president for technology of Trusted Information Systems in Glenwood, MD. He was founder and president of Haystack Labs, Inc., which TIS acquired in 1997. A pioneer in the research and development of intrusion detection systems for sensitive government users since 1987, he has more than 20 years of experience in designing and developing sophisticated computer systems.

Jessica Winslow is the author of numerous publications on information systems including computer security and policy for information resources management. A former vice president at Haystack Labs, she has over 10 years industry experience in the private and public sectors.

Stalker and WebStalker are trademarks of Trusted Information Systems, Inc. All other trademarks belong to their respective owners.