

BLM5102

Computer Systems and Network Security

Prof. Dr. Hasan Hüseyin BALIK

(6th Week)

Outline

- 2. Management issues
 - 2.1. IT Security Management and Risk Assessment
 - 2.2. IT Security Controls, Plans and Procedures
 - 2.3. Physical and Infrastructure Security
 - 2.4. Human Resources Security
 - 2.5. Security Auditing
 - 2.6. Legal and Ethical Aspects

2.5. Security Auditing

2.5. Outline

- Security Auditing Architecture
- The Security Audit Trail
- Implementing the Logging Function
- Audit Trail Analysis
- Security Information and Event Management

Security Audit Terminology (RFC 4949)

Security audit An independent review and examination of a system's records and activities to determine the adequacy of system controls, ensure compliance with established security policy and procedures, detect breaches in security services, and recommend any changes that are indicated for countermeasures.

The basic audit objective is to establish accountability for system entities that initiate or participate in security-relevant events and actions. Thus, means are needed to generate and record a security audit trail and to review and analyze the audit trail to discover and investigate attacks and security compromises.

Security Audit Trail A chronological record of system activities that is sufficient to enable the reconstruction and examination of the sequence of environments and activities surrounding or leading to an operation, procedure, or event in a security-relevant transaction from inception to final results.

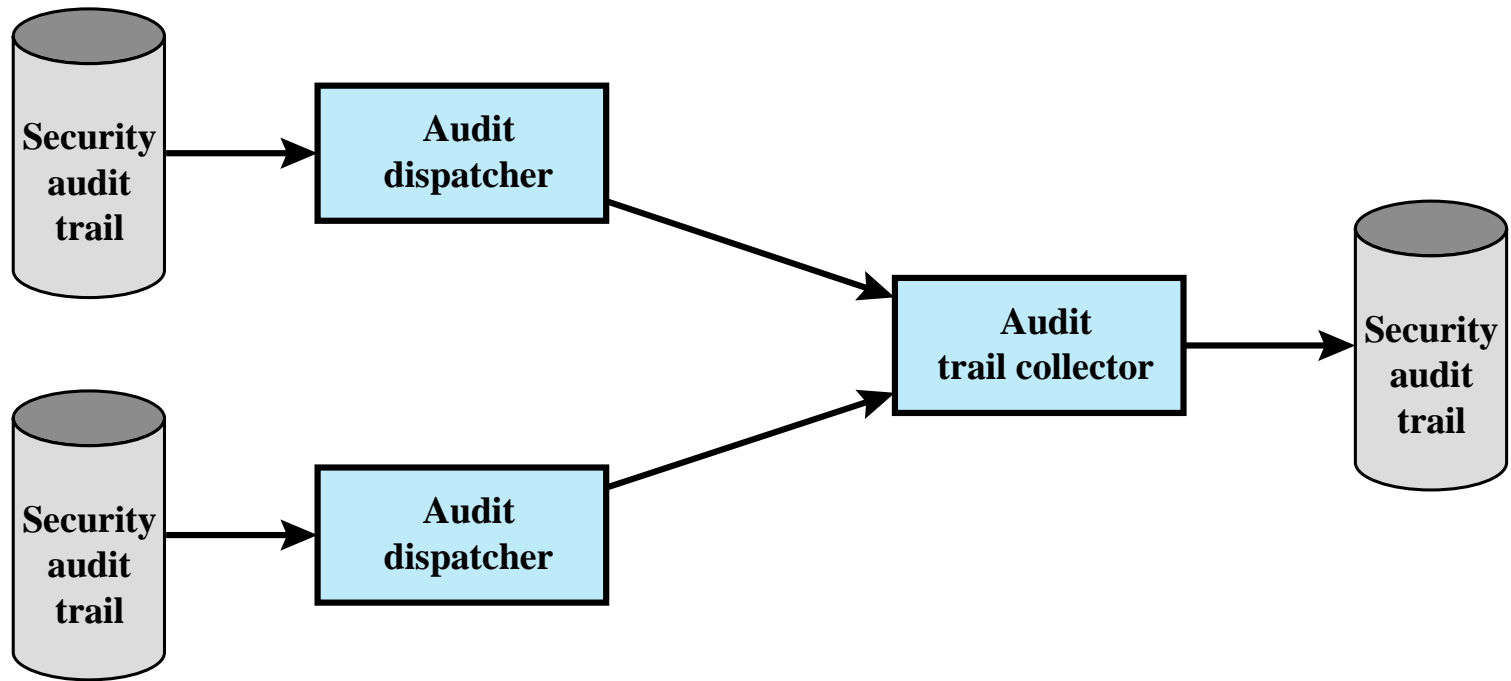


Figure 18.2 Distributed Audit Trail Model (X.816)

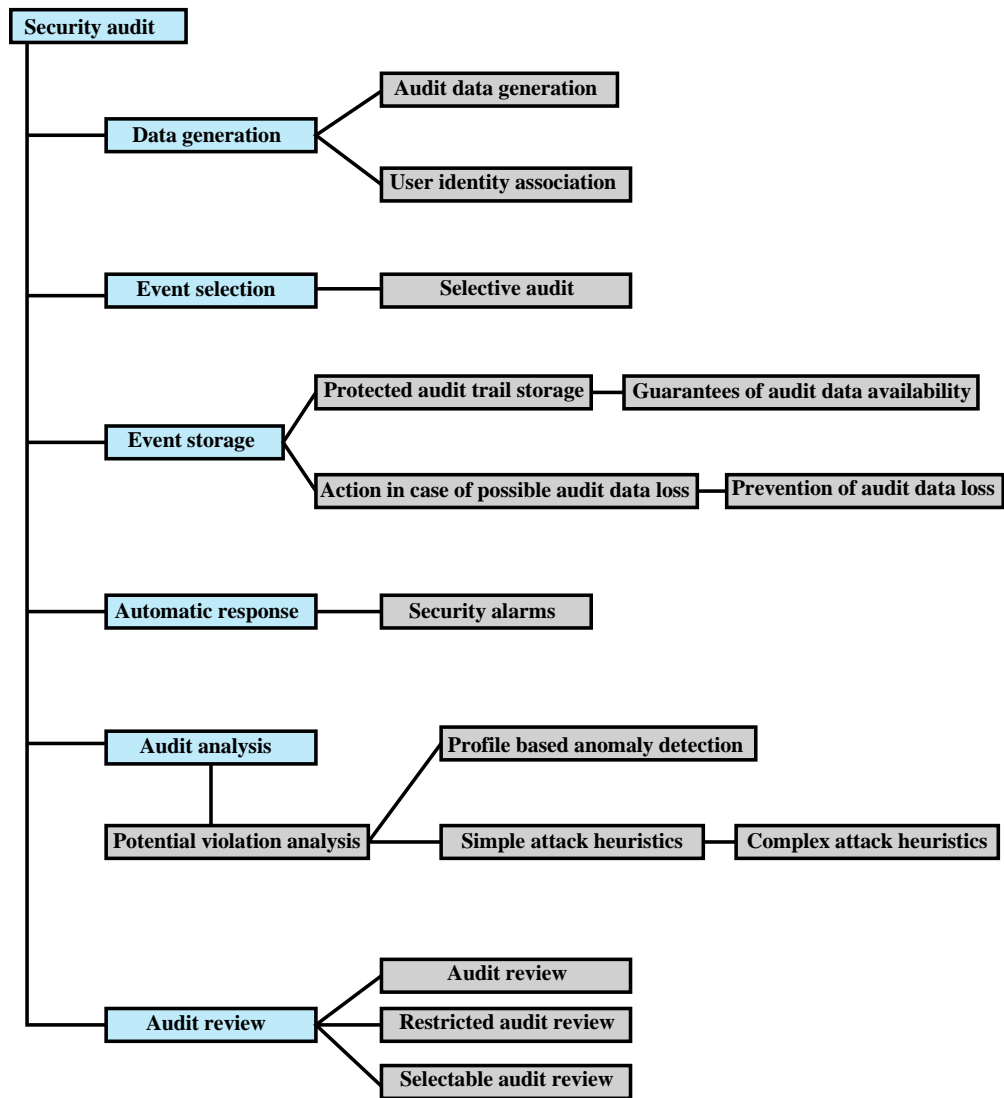


Figure 18.3 Common Criteria Security Audit Class Decomposition

Event Definition

- Must define the set of events that are subject to audit

Common criteria suggests:

- Introduction of objects
- Deletion of objects
- Distribution or revocation of access rights or capabilities
- Changes to subject or object security attributes
- Policy checks performed by the security software
- Use of access rights to bypass a policy check
- Use of identification and authentication functions
- Security-related actions taken by an operator/user
- Import/export of data from/to removable media

Event Detection

- Appropriate hooks must be available in the application and system software to enable event detection
- Monitoring software needs to be added to the system and to appropriate places to capture relevant activity
- An event recording function is needed, which includes the need to provide for a secure storage resistant to tampering or deletion
- Event and audit trail analysis software, tools, and interfaces may be used to analyze collected data as well as for investigating data trends and anomalies
- There is an additional requirement for the security of the auditing function
- Auditing system should have a minimal effect on functionality

Implementation Guidelines

Agree on audit requirements with appropriate management

Scope of technical audit tests should be agreed and controlled

Audit tests should be limited to read-only access to software and data

Audit tests that could affect system availability should be run outside business hours

Requirements for special or additional processing should be identified and agreed

Access other than read-only should only be allowed for isolated copies of system files

All access should be monitored and logged to produce a reference trail

What to Collect

- Events related to the use of the auditing software
- Events related to the security mechanisms on the system
- Events that are collected for use by the various security detection and prevention mechanisms
- Events related to system management and operation
- Operating system access
- Application access for selected applications
- Remote access

Security related events related to a specific connection

- Connection requests
- Connection confirmed
- Disconnection requests
- Disconnection confirmed
- Statistics appertaining to the connection

Security related events related to the use of security services

- Security service requests
- Security mechanisms usage
- Security alarms

Security related events related to management

- Management operations
- Management notifications

The list of auditable events should include at least

- Deny access
- Authenticate
- Change attribute
- Create object
- Delete object
- Modify object
- Use privilege

In terms of the individual security services, the following security-related events are important

- Authentication: verify success
- Authentication: verify fail
- Access control: decide access success
- Access control: decide access fail
- Non-repudiation: non-repudiable origination of message
- Non-repudiation: non-repudiable receipt of message
- Non-repudiation: unsuccessful repudiation of event
- Non-repudiation: successful repudiation of event
- Integrity: use of shield
- Integrity: use of unshield
- Integrity: validate success
- Integrity: validate fail
- Confidentiality: use of hide
- Confidentiality: use of reveal
- Audit: select event for auditing
- Audit: deselect event for auditing
- Audit: change audit event selection criteria

Auditable
Items
Suggested
in X.816

Monitoring Areas Suggested in ISO 27002

- | | |
|--|--|
| <ul style="list-style-type: none">a) user IDsb) system activitiesc) dates, times and details of key events, e.g. log-on and log-offd) device identity or location if possible and system identifiere) records of successful and rejected system access attemptsf) records of successful and rejected data and other resource access attemptsg) changes to system configuration | <ul style="list-style-type: none">h) use of privilegesi) use of system utilities and applicationsj) files accessed and the kind of accessk) network addressees and protocolsl) alarms raised by the access control systemm) activation and de-activation of protection systems, such as anti-virus systems and intrusion detection systemsn) records of transactions executed by users in applications |
|--|--|

```

Jan 27 17:14:04 host1 login: ROOT LOGIN console
Jan 27 17:15:04 host1 shutdown: reboot by root
Jan 27 17:18:38 host1 login: ROOT LOGIN console
Jan 27 17:19:37 host1 reboot: rebooted by root
Jan 28 09:46:53 host1 su: 'su root' succeeded for user1 on /dev/ttyp0
Jan 28 09:47:35 host1 shutdown: reboot by user1
Jan 28 09:53:24 host1 su: 'su root' succeeded for user1 on /dev/ttyp1
Feb 12 08:53:22 host1 su: 'su root' succeeded for user1 on /dev/ttyp1
Feb 17 08:57:50 host1 date: set by user1
Feb 17 13:22:52 host1 su: 'su root' succeeded for user1 on /dev/ttyp0

```

(a) Sample system log file showing authentication messages

```

Apr 9 11:20:22 host1 AA06370: from=<user2@host2>, size=3355, class=0
Apr 9 11:20:23 host1 AA06370: to=<user1@host1>, delay=00:00:02,stat=Sent
Apr 9 11:59:51 host1 AA06436: from=<user4@host3>, size=1424, class=0
Apr 9 11:59:52 host1 AA06436: to=<user1@host1>, delay=00:00:02, stat=Sent
Apr 9 12:43:52 host1 AA06441: from=<user2@host2>, size=2077, class=0
Apr 9 12:43:53 host1 AA06441: to=<user1@host1>, delay=00:00:01, stat=Sent

```

(b) Application-level audit record for a mail delivery system

```

rcp      user1  tty0  0.02 secs Fri Apr 8 16:02
ls       user1  tty0  0.14 secs Fri Apr 8 16:01
clear   user1  tty0  0.05 secs Fri Apr 8 16:01
rpcinfo user1  tty0  0.20 secs Fri Apr 8 16:01
nroff   user2  tty2  0.75 secs Fri Apr 8 16:00
sh       user2  tty2  0.02 secs Fri Apr 8 16:00
mv       user2  tty2  0.02 secs Fri Apr 8 16:00
sh       user2  tty2  0.03 secs Fri Apr 8 16:00
col      user2  tty2  0.09 secs Fri Apr 8 16:00
man      user2  tty2  0.14 secs Fri Apr 8 15:57

```

(c) User log showing a chronological list of commands executed by users

Figure 18.4 Examples of Audit Trails

Physical Access Audit Trails

- Generated by equipment that controls physical access
 - Card-key systems, alarm systems
- Sent to central host for analysis and storage
- Data of interest:
 - Date/time/location/user of access attempt
 - Both valid and invalid access attempts
 - Attempts to add/modify/delete physical access privileges
 - May send violation messages to personnel

Protecting Audit Trail Data

Read/write file on host

- Easy, least resource intensive, instant access
- Vulnerable to attack by intruder

Write-once/read-many device

- More secure but less convenient
- Need steady supply of recordable media
- Access may be delayed and not available immediately

Write-only device

- Provides paper trail
- Impractical for capturing detailed audit data on large or networked systems
- Useful when a permanent, immediately available log is required

Must protect both integrity and confidentiality

- Encryption, digital signatures, access controls

Implementing Logging

- Foundation of security auditing facility is the initial capture of the audit data
- Software must include hooks (capture points) that trigger data collection and storage as preselected events occur
- Dependent on the nature of the software
 - Varies depending on operating system and applications involved

Windows Event Log

- Event is an entity that describes some interesting occurrence
 - Contains:
 - A numeric identification code
 - A set of attributes
 - Optional user-supplied data
- Three types of event logs:
 - System: system related apps and drivers
 - Application: user-level apps
 - Security: Windows LSA

Windows Event Schema Elements

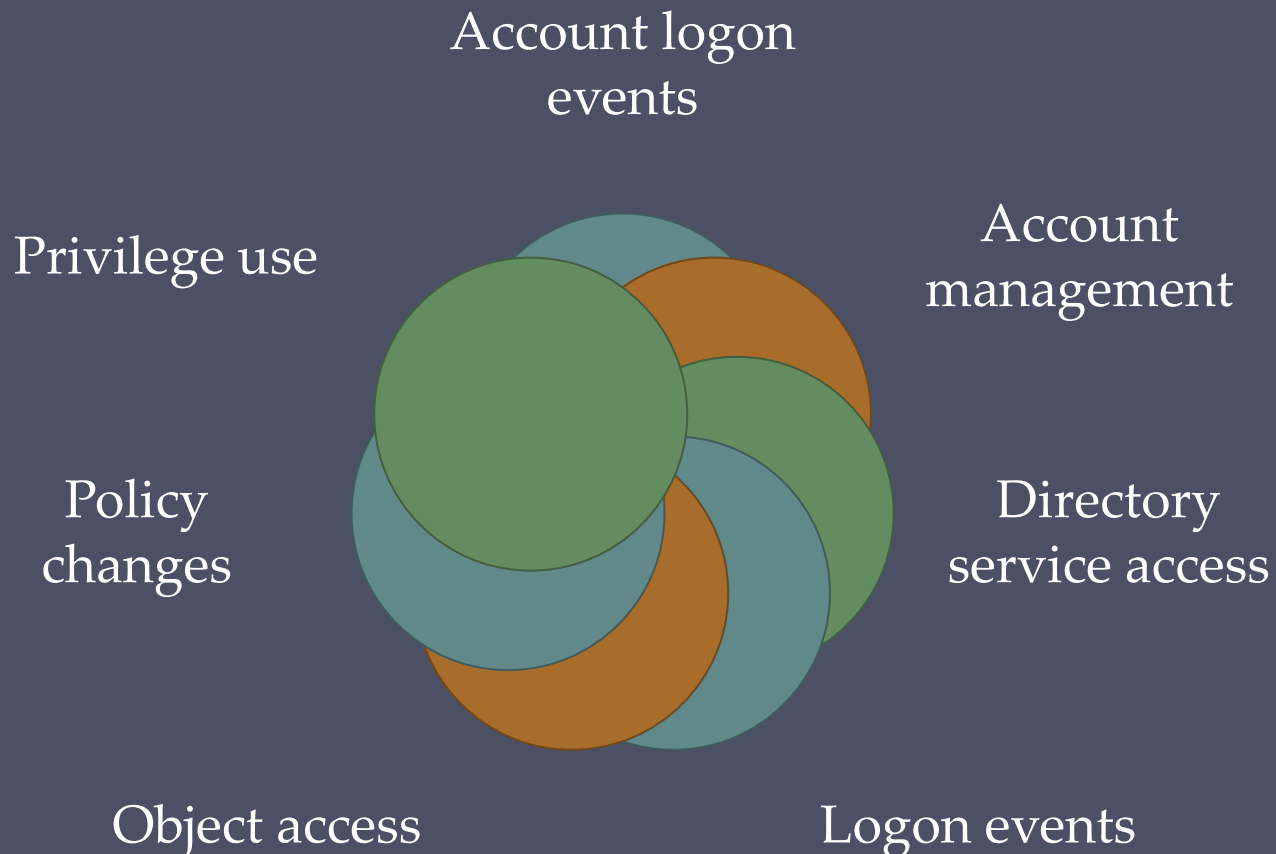
Property values of an event that contains binary data
Binary data supplied by Windows Event Log
Channel into which the rendered event is published
Complex data for a parameter supplied by the event provider
ComponentName WPP debug tracing field used in debug events
Computer that the event occurred on
Two 128-bit values that can be used to find related events
Name of the event data item that caused an error when the event data was processed
Data that makes up one part of the complex data type supplied by the event provider
Data for a parameter supplied by the event provider
Property values of Windows software trace preprocessor (WPP) events
Error code that was raised when there was an error processing event data
A structured piece of information that describes some interesting occurrence in the system
Event identification number
Information about the process and thread in which the event occurred
Binary event data for the event that caused an error when the event data was processed
Information about the process and thread the event occurred in
FileLine WPP debug tracing field used in debug events in debug channels
FlagsName WPP debug tracing field used in debug events in debug channels
KernelTime WPP debug tracing field used in debug events in debug channels
Keywords that will be rendered for an event
Keywords used by the event

The LevelName Windows software trace preprocessor (WPP) debug tracing field used in debug events in debug channels
Level that will be rendered for an event
Level of severity for an event
FormattedString WPP debug tracing field used in debug events in debug channels
Event message rendered for an event
Opcode that will be rendered for an event
The activity or a point within an activity that the application was performing when it raised the event
Elements that define an instrumentation event
Information about the event provider that published the event
Event publisher that published the rendered event
Information that will be rendered for an event
The user security identifier
SequenceNum WPP debug tracing field used in debug events in debug channels
SubComponentName WPP debug tracing field used in debug events in debug channels
Information automatically populated by the system when the event is raised or when it is saved into the log file
Task that will be rendered for an event
Task with a symbolic value
Information about the time the event occurred
Provider-defined portion that may consist of any valid XML content that communicates event information
UserTime WPP debug tracing field used in debug events in debug channels
Event version

```
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 18.5 Windows System Log Entry Example

Windows Event Categories



UNIX Syslog

- UNIX's general-purpose logging mechanism
 - Found on all UNIX / Linux variants

Elements:

syslog()

API referenced by several standard system utilities and available to application programs

logger

Command used to add single-line entries to the system log

/etc/syslog.conf

Configuration file used to control the logging and routing of system log events

syslogd

Daemon to receive/route log events

Syslog Service

Basic service provides:

A means of capturing relevant events

A storage facility

A protocol for transmitting syslog messages from other machines to a central machine that acts as a syslog server

Extra add-on features may include:

Robust filtering

Log analysis

Event response

Alternative message formats

Log file encryption

Database storage

Rate limiting

Syslog Protocol

- A transport allowing hosts to send IP event notification messages to syslog servers
 - Provides a very general message format
 - Allowing processes and applications to use suitable conventions for their logged events
 - Common version of the syslog protocol was originally developed on the University of California Berkeley Software Distribution (BSD) UNIX/TCP/IP system implementations
 - Messages in the BSD syslog format consist of:
 - PRI - facilities/severity code
 - Header – timestamp and hostname/IP address
 - Msg - program name and content

```
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 18.6 Examples of Syslog Messages

(a) syslog Facilities

Facility	Message Description (generated by)
kern	System kernel
user	User process
mail	e-mail system
daemon	System daemon, such as ftpd
auth	Authorization programs <code>login</code> , <code>su</code> , and <code>getty</code>
Syslogd	Messages generated internally by syslogd
lpr	Printing system
news	UseNet News system
uucp	UUCP subsystem
clock	Clock daemon
ftp	FTP daemon
ntp	NTP subsystem
log audit	Reserved for system use
log alert	Reserved for system use
Local use 0–7	Up to 8 locally defined categories

(b) syslog Severity Levels

Severity	Description
emerg	Most severe messages, such as immediate system shutdown
alert	System conditions requiring immediate attention
crit	Critical system conditions, such as failing hardware or software
err	Other system errors; recoverable
warning	Warning messages; recoverable
notice	unusual situation that merits investigation; a significant event that is typically part of normal day-to-day operation
info	Informational messages
debug	Messages for debugging purposes

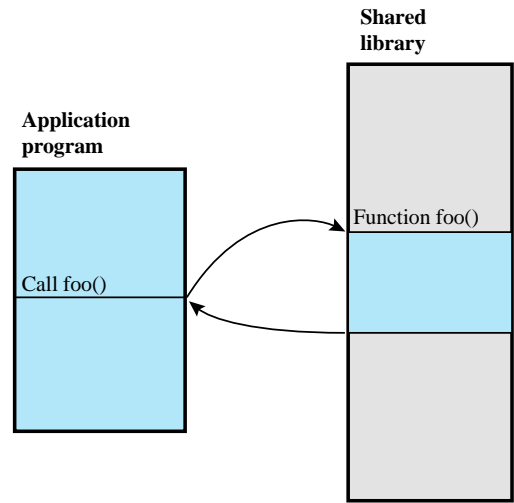
UNIX syslog Facilities and Severity Levels

Logging at Application Level

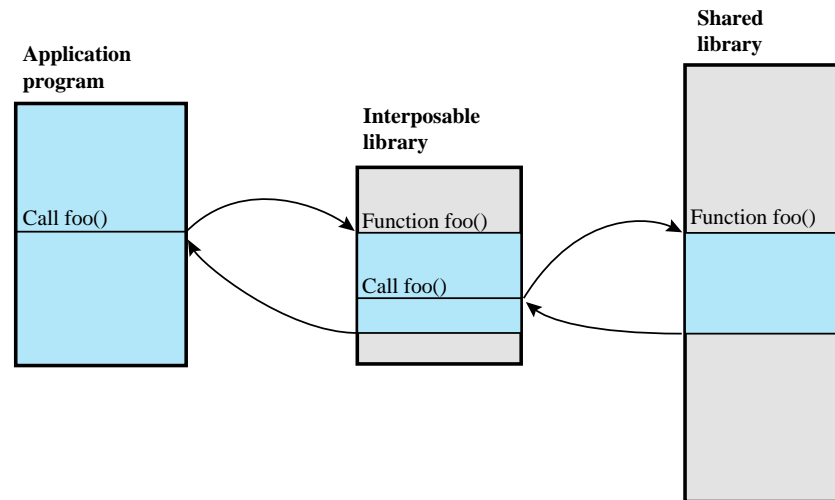
- Privileged applications present security issues
 - May not be captured by system/user-level audit data
 - Constitute a large percentage of reported vulnerabilities
- Vulnerabilities exploited:
 - Lack of dynamic checks on input data
 - Errors in application logic
- May be necessary to capture behavior of application beyond its access to system services and file systems
- Two approaches to collecting audit data:
 - Interposable libraries
 - Dynamic binary rewriting

Interposable Libraries

- Allows the generation of audit data without needing to recompile either the system libraries or the application
 - Audit data can be generated without changing the system's shared libraries or needing access to the source code for the executable
 - Exploits the use of dynamic libraries in UNIX
- Statically linked libraries
 - A separate copy of the linked library function is loaded into the program's virtual memory
 - Statically linked shared libraries
 - Referenced shared object is incorporated into the target executable at link time by the link loader
 - Each object is assigned a fixed virtual address
 - Link loader connects external referenced objects by assigning their virtual addresses when the executable is created
 - Dynamically linked shared libraries
 - The linking to shared library routines is deferred until load time
 - If changes are made to the library prior to load time any program that references the library is unaffected



(a) Normal library call technique



(b) Library call with interposition

Figure 18.7 The Use of an Interposable Library

```

1 /*****
2 * Logging the use of certain functions *
3 *****/
4 char *strcpy(char *dst, const char *src) {
5     char *(*fptr)(char *,const char *); /* pointer to the real function */
6     char *retval; /* the return value of the call */
7
8     AUDIT_CALL_START;
9
10    AUDIT_LOOKUP_COMMAND(char *(*) (char *,const char *),"strcpy",fptr,NULL);
11
12    AUDIT_USAGE_WARNING("strcpy");
13
14    retval=(*fptr)(dst,src);
15
16    return(retval);
17 }

```

(a) Function definition (items in all caps represent macros defined elsewhere)

```

1 #define AUDIT_LOOKUP_COMMAND(t,n,p,e)
2     p=(t)dlsym(RTLD_NEXT,n);
3     if (p==NULL) {
4         perror("looking up command");
5         syslog(LOG_INFO,"could not find %s in library: %m",n);
6         return(e);
7     }

```

(b) Macro used in function

Figure 18.8 Example of Function in the Interposed Library

Dynamic Binary Rewriting

- Can be used with both statically and dynamically linked programs
- Postcompilation technique that directly changes the binary code of executables
 - Change is made at load time and modifies only the memory image of a program
 - Does not require recompilation of the application binary
- Implemented on Linux using two modules:
 - Loadable kernel module
 - Monitoring daemon
- Loadable modules
 - Can be automatically loaded and unloaded on demand

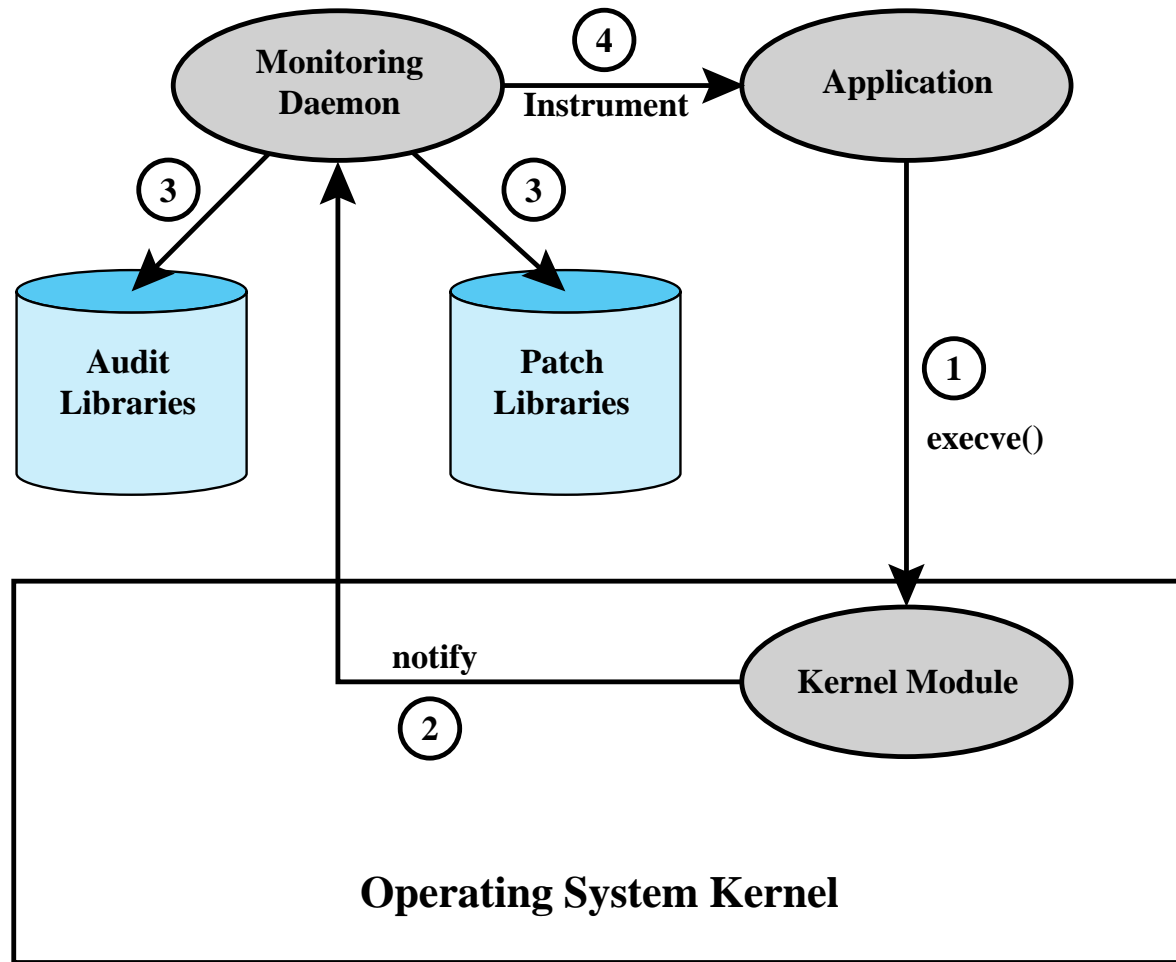


Figure 18.9 Runtime Environment for Application Auditing

Audit Trail Analysis

- Analysis programs and procedures vary widely
- Must understand context of log entries
 - Relevant information may reside in other entries in the same logs, other logs, and nonlog sources
- Audit file formats contain mix of plain text and codes
 - Must decipher manually/automatically
- Ideally regularly review entries to gain understanding of baseline

Types of Audit Trail Analysis

Audit trails can be used in multiple ways

- This depends in part on when done

Possibilities include:

- Audit trail review after an event
 - Triggered by event to diagnose cause and remediate
 - Focuses on the audit trail entries that are relevant to the specific event
- Periodic review of audit trail data
 - Review bulk data to identify problems and behavior
- Real-time audit analysis
 - Part of an intrusion detection function

Audit Review

- Audit review capability provides administrator with information from selected audit records
 - Actions of one or more users
 - Actions on a specific object or resource
 - All or a specified set of audited exceptions
 - Actions on a specific system/security attribute
- May be filtered by time/source/frequency
- Used to provide system activity baseline
- Level of security related activity

Approaches to Data Analysis

Basic alerting

- Indicate interesting type of event has occurred

Baselining

- Define normal versus unusual events/patterns
- Compare with new data to detect changes
- Thresholding is the identification of data that exceed a particular baseline value

Windowing

- Detection of events within a given set of parameters

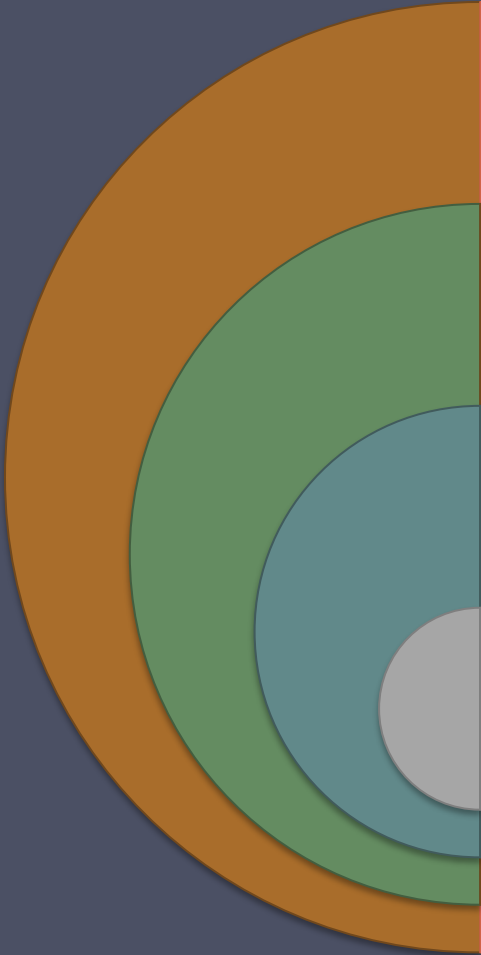
Correlation

- Seeks relationships among events

SIEM Systems

- Software is a centralized logging software package similar to, but much more complex than, syslog
- Provide a centralized, uniform audit trail storage facility and a suite of audit data analysis programs
- There are two general configuration approaches:
 - Agentless
 - SIEM server receives data from the individual log generating hosts without needing to have any special software installed on those hosts
 - Agent-based
 - An agent program is installed on the log generating host to perform event filtering and aggregation and log normalization for a particular type of log, and then transmit the normalized log data to a SIEM server, usually on a real-time or near-real-time basis for analysis and storage

SIEM Software



SIEM software is able to recognize a variety of log formats, including those from a variety of OSs, security software, application servers, and even physical security control devices such as badge readers

Software normalizes these various log entries so that the same format is used for the same data item in all entries

Software can delete fields in log entries that are not needed for the security function and log entries that are not relevant

SIEM server analyzes the combined data from the multiple log sources, correlates events among the log entries, identifies and prioritizes significant events, and initiates responses to events if desired