

Chapter 21 AUDITING

- Auditing is an a posteriori technique for determining security violations.
- This chapter presents two notions:
 - Logging: recording of system events and actions
 - Auditing: analysis of these records

21.1 Definitions

- Logging is the recording of events or statistics to provide information about system use and performance.
- Auditing is the analysis of log records to present information about the system in a clear and understandable manner.
- Logging
 - Logs provide a mechanism for analyzing the system security state.
 - If the log records all events that cause state transitions, the system can be reconstructed at any time.
 - Even if only a subset of this information is recorded, it may be possible to eliminate some possible causes of a security problem.

Chapter 21 AUDITING

- **Two distinct but related problems arise:**
 - Which information to log?
 - Which information to audit?
- **The decision of which events and actions should be audited requires:**
 - A knowledge of the security policy of the system
 - What attempts to violate that policy involve
 - How such attempts can be detected
- **The question of how such attempts can be detected raises the question of what should be logged:**
 - What commands must an attacker use to violate the security policy
 - What system calls must be made
 - Who must issue the commands or system calls
 - In what order, what objects must be altered
- **Logging of all events implicitly provides all this information.**
- **The problem is how to discern which parts of the information are relevant, which is the problem of determining what to audit.**

Chapter 21 AUDITING

21.2 Anatomy of an Auditing System

- An auditing system is consisted of 3 components:
 - The logger
 - The analyzer
 - The notifier
- **Logger**: Records information, usually controlled by parameters.
 - Type and quantity of information recorded controlled by system or program configuration parameters
 - The information may be recorded in binary or human-readable form.
 - If the logs are recorded in binary form, a log-viewing tool is usually provided.
- **Analyzer**: Takes a log as input and analyzes it.
 - The results of the analysis may lead to:
 - Changes in logging
 - Detection of some event or problem
 - Both changes and detection
- **Notifier**: The analyzer passes the analysis results to the notifier.
 - Informs the analyst, and other entities, of the results of the audit.
 - The entities may take some action in response to these results.
 - For example, they may reconfigure logging and/or analysis based on the results.

Chapter 21 AUDITING

- **EXAMPLE (logger): RACF**
 - A security enhancement package for the IBM MVS operating system and VM environment.
 - Logs failed access attempts and the use of privileges o change security levels.
 - Can be set to log RACF interactions.
 - RACF can also log its interactions with users, so that if a user attampts to modify it any way, a log entry will be made.
 - The command LISTUSER lists information about RACF users.

Chapter 21 AUDITING

RACF: Sample Entry

```
USER=EW125004 NAME=S.J.TURNER OWNER=SECADM CREATED=88.004
  DEFAULT-GROUP=HUMRES PASSDATE=88.004 PASS-INTERVAL=30
  ATTRIBUTES=ADSP
  REVOKE DATE=NONE RESUME-DATE=NONE
  LAST-ACCESS=88.020/14:15:10
  CLASS AUTHORIZATIONS=NONE
  NO-INSTALLATION-DATA
  NO-MODEL-NAME
  LOGON ALLOWED (DAYS) (TIME)
  -----
  ANYDAY ANYTIME
GROUP=HUMRES AUTH=JOIN CONNECT-OWNER=SECADM
  CONNECT-DATE=88.004
  CONNECTS= 15 UACC=READ LAST-CONNECT=88.018/16:45:06
  CONNECT ATTRIBUTES=NONE
  REVOKE DATE=NONE RESUME DATE=NONE
GROUP=PERSNL AUTH=JOIN CONNECT-OWNER=SECADM CONNECT-DATE:88.004
  CONNECTS= 25 UACC=READ LAST-CONNECT=88.020/14:15:10
  CONNECT ATTRIBUTES=NONE
  REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY AUTHORIZATION
  NONE SPECIFIED
```

Chapter 21 AUDITING

- **EXAMPLE (logger): MS Windows NT**
 - Has three different sets of logs
 - System event logs record system crashes, component failures, and other system events.
 - Application event logs record events that applications request be recorded.
 - Security event log records security-critical events such as logging in and out, system file accesses, and other events.
 - The Windows NT logger defines a record as a header followed by a description and possibly an additional data field.
 - The header contains:
 - An event identifier
 - User identity information
 - The date and time
 - The source that caused the record to be generated
 - The specific policy setting that triggered the record
 - The computer involved
 - All records are kept in binary form.
 - A tool called the event viewer translates the records into readable form.
 - Only administrators can access the security event log.
 - The designers of Windows NT allowed the system administrator to specify what should happen if the log should get full.
 - The Administrator can do 3 things when the log is full:
 - Have the system shut down.
 - Disable logging completely
 - Cause the oldest entries to be overwritten or discarded

Chapter 21 AUDITING

Date: 2/12/2000 Source: Security
Time: 13:03 Category: Detailed Tracking
Type: Success EventID: 592
User: WINDSOR\Administrator
Computer: WINDSOR

Description:

A new process has been created:

New Process ID: 2216594592
Image File Name:
\Program Files\Internet Explorer\IEXPLORE.EXE
Creator Process ID: 2217918496
User Name: Administrator
FDomain: WINDSOR
Logon ID: (0x0,0x14B4c4)

**An example
Windows NT
security event log**

**This event arose
from the
Administrator
successfully
executing the
Internet Explorer.**

Chapter 21 AUDITING

- **EXAMPLE (analyzer):**
 - A system administrator wants to list all systems from which users have connected using the login or telnet program, excluding systems at the site.
 - System event logs record system crashes, component failures, and other system events.
 - The following swatch patterns match the lines generated by these remote connections.

```
/rlogin/&!/localhost/&!/*.site.com/
```

This line matches all log file entries containing the word “rlogin” and not containing either “localhost” or any string ending in “.site.com.”

```
/telnet/&!/localhost/&!/*.site.com/
```

This line matches all log file entries containing the word “telnet” and not containing either “localhost” or any string ending in “.site.com.”

Chapter 21 AUDITING

- **EXAMPLE (notifier):**
 - The swatch program mentioned in the previous example provides a notification facility.
 - The configuration file to make swatch report rlogin and telnet connections is:

```
/rlogin/&!/localhost/&!/*.site.com/mail staff
```

```
/telnet/&!/localhost/&!/*.site.com/mail staff
```

Chapter 21 AUDITING

21.3 Designing an Auditing System

- A single, well-unified logging process is an essential component of computer security mechanisms.
- The auditing mechanism analyzes information related to the security state of the system.
 - It determines
 - if specific actions have occurred
 - If certain states have been entered
- The goals of the auditing process determine what information is logged.
- The auditors, in general, want to detect violations of policy, which provides a set of constraints p_i that the set of possible actions A_i must satisfy.
 - Represent the constraints p_i as “action \Rightarrow condition”
 - If, for example, the record’s action is a “read” and the constraint’s action is a “write,” then the constraint holds.
 - As the goal of the auditing is to determine if the policy has been violated, the result of the operation (success or failure) should match the satisfaction of the constraint.
 - If the constraint is true, the result is irrelevant.
 - If the constraint is false and the operation is successful, a security violation has occurred.

Chapter 21 AUDITING

- **EXAMPLE: Bell-LaPadula**
 - A subject S has the level $L(S)$ and the object O has the level $L(O)$.
 - Simple security condition and *-property
 - S reads $O \Rightarrow L(S) \geq L(O)$
 - S writes $O \Rightarrow L(S) \leq L(O)$
 - To check for violations, on each read and write, logs must contain:
 - $L(S), L(O)$
 - action (read, write)
 - result (success, failure)
 - The names of the subject and object do not need to be recorded.
 - However, in practice the site security policy would require the security analyst to identify both the objects of the violation and the user who attempted the violation.
 - With this modification of the policy, the names of the subject and object would also be recorded.
 - In this limited case, auditing of reads and writes in a Bell-LaPadula-based systems requires logging of three records:
 - The subject's security level
 - The object's security level
 - The result of the action

Chapter 21 AUDITING

21.4 A Posteriori Design

- The design of an effective auditing subsystem is straightforward when one is aware of all possible policy violations and can detect them.
- Unfortunately, this is rarely the case!
- There is need to design an auditing mechanism for systems not built with security in mind.
- **Goals of auditing**
 - The first goal is to detect any violations of a stated policy.
 - The second goal is to detect actions known to be part of an attempt to breach security.
 - The difference is subtle but important.
 - The first goal focuses on the policy, and records attempted actions that violate the policy.
 - The set of such actions may not be known in advance.
 - The second goal focuses on specific actions that the managers of the system have determined indicate behavior that poses a threat to system security.
 - Hence, there are 2 approaches:
 - One approaches the first goal by examining the desired policy.
 - One approaches the second goal by examining the actions (attacks) that pose the treat.

Chapter 21 AUDITING

21.5 Auditing Mechanisms

- Different systems approach logging in different ways.
- Most systems log all events by default, and allow the system administrator to disable logging that is unnecessary.
- Two examples
 - Secure systems: systems designed with security in mind have auditing mechanisms integrated with the system design and implementation.
 - These systems typically provide a language or interface that allows system managers to configure reporting and logging:
 - » To report specific events
 - » To monitor accesses by a subject
 - » To monitor accesses to an object
 - This is controlled at the audit subsystem.
 - Nonsecure systems: auditing subsystems for systems not designed with security in mind are generally for purposes of accounting.
 - These systems have some limited logging capabilities.
 - » Possibly limited security data like failed logins.
 - » Auditing subsystems focusing on security are usually added after the system is completed.

Chapter 21 AUDITING

21.6 Examples: Auditing File Systems

- **Network File System (NFS)**
 - Many sites allow computers and users to share file systems.
 - One computer (client host) requests access to the file system of another computer (server host).
 - The server host responds by exporting a directory of its file system.
 - The client host imports this information and arranges its own file system.
 - The imported directory (server host's mount point) appears as a directory in the client host's file system (client host's mount point).
- **NFS Version 2 Protocol**
- **Logging and Auditing File System (LAFS)**

Chapter 21 AUDITING

21.6 Examples: Auditing File Systems

- **Network File System (NFS)**
 - Many sites allow computers and users to share file systems.
 - One computer (client host) requests access to the file system of another computer (server host).
 - The server host responds by exporting a directory of its file system.
 - The client host imports this information and arranges its own file system.
 - The imported directory (server host's mount point) appears as a directory in the client host's file system (client host's mount point).
- **NFS Version 2 Protocol**
- **Logging and Auditing File System (LAFS)**

Chapter 21 AUDITING

21.7 Audit Browsing

- Auditors run audit mechanisms to analyze log files.
- They also look through the log files themselves.
 - The audit mechanisms may miss information or irregularities in the log that a knowledgeable auditor can detect.
 - If the audit mechanisms are unsophisticated, the auditors may examine the logs directly to uncover evidence of previously unknown patterns of misuse and attack.
 - Finally, few systems provide a fully integrated suite of logs.
- The goal of an audit browsing tool is to present log information in a form that is easy to understand and use.
- Six basic browsing techniques
 - Text display: does not indicate relationships among events, entries, and entities.
 - Hypertext display: indicates local relationships between entries and entities.
 - Relational database browsing: the logs are stored in a relational database.
 - Replay: shows events occurring in temporal order.
 - Graphing: provides a visual representation of the contents of logs; often too cluttered to show everything, so graphing selects subsets of events.
 - Slicing: shows minimum set of log events affecting object.

Chapter 21 AUDITING

- **EXAMPLE: Visual Audit Browser**
 - Designed for general purpose audit browsing.
 - It is consisted of 4 tools:
 - **Frame Visualizer**
 - Generates graphical representation of logs
 - **Movie Maker**
 - Generates sequence of graphs, each event creating a new graph suitably modified
 - **Hypertext Generator**
 - Produces page per user, page per modified file, summary and index pages
 - **Focused Audit Browser**
 - Enter node name, displays node, incident edges, and nodes at end of edges