These Slides are prepared from
Matt Bishop slides and book "Introduction to Computer Security"
Benefiting from the Slides posted by Ahmad Al-Mulhem

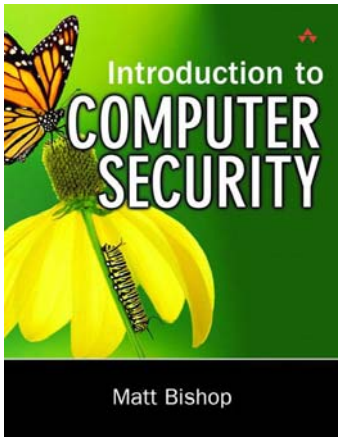
Auditing

Chapter 21

Adnan Gutub

gutub@kfupm.edu.sa

Computer Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia



Chapter 21: Auditing

Overview
- What is auditing?
- What does an audit system look like?
- When & where is it needed?
- How do you design an auditing system?

Structure
- Logger
- Analyzer
- Notifier

Auditing Mechansims

What is Auditing?

- Tracing security violations
- Security-relevant actions should be logged
- Accountability
- Logs should be protected

Logging

– Recording events or statistics to provide information about system use and performance

Auditing

– Analysis of log records to present information about the system in a clear, understandable manner

Auditing Uses

Describe security state (security policy)

– Determine if system enters unauthorized state

Evaluate effectiveness of protection mechanisms

– Determine which mechanisms are appropriate and working

– Deter (prevent) attacks because of presence of record

- Abuses are being watched!

Auditing Challenges & Problems

What do you log?

- Hint: looking for violations of a policy, so record *at least* what will show such violations
  - Examples: packets, system calls
- How about logging everything?

What do you audit?

- Need not audit everything
- Key: what is the policy involved?

Audit System Structure

Logger

- Records information, usually controlled by parameters
  - Store logger record as: binary or human-readable form
  - Or transmit record directly to analysis mechanism

Analyzer

- Analyzes logged information looking for something
  - May change data being recorded
  - May detect some event (Unexpected activity)
  - May detect some problem (activity to compromise the system)

Notifier

- Reports results of analysis
  - May take action in response to analysis

Logger

Type, quantity of information recorded controlled by system or program configuration parameters

May be human readable or not

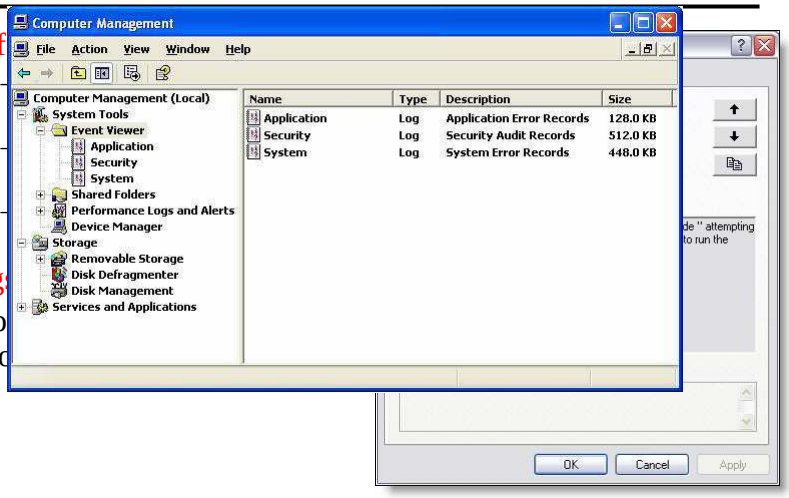
- If not, usually viewing tools supplied
- Space available, portability influence storage format

Logger Example: Windows

Diff

Log

If lo
lo



Analyzer

Analyzes one or more logs

- Logs may come from multiple systems, or a single system
- May lead to changes in logging
- May lead to a report of an event

Analyzer Examples

Text-based: Using *swatc*

`/telnet`

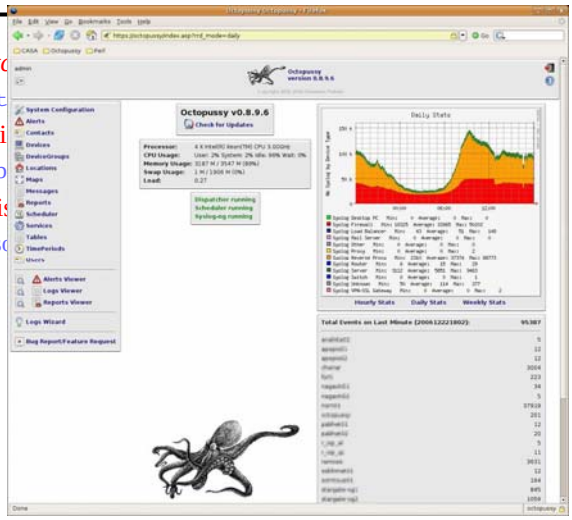
Query set overlap control i

- If too much overlap b

Intrusion detection analysis

- Takes data from senso

GUI: Octopussy



Notifier

Informs analyst, other entities of results of analysis

- email, pager, sms message, . . . etc.

May reconfigure logging and/or analysis on basis of results

Notifier Examples

Using *swatch* to notify of *telnets*

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

Query set overlap control in databases

- Prevents response from being given if too much overlap occurs

Example: Three failed logins attempts:

- logger records the attempts
- Analyzer checks the number of failed attempts
- Notifier disables account, notifies sys admin (when the number reaches in 3 a row disable user account)

Designing an Audit System

Essential component of security mechanisms

Goals: determine what information is logged

- Idea: auditors want to detect violations of policy
  - which provides a set of constraints that the set of possible actions must satisfy
- So, audit functions that may violate the constraints

Considerations:

- Implementation Considerations
- Syntax Issues
- Log Sanitization
- Application & System Logging

- Auditing Mechanisms
  - Secure Systems
  - Non-secure systems

Implementation Issues

Assume beginning with a secure state

- Show non-security or find violations?
- Requires logging initial state as well as changes

Defining violations

- Does "write" include "append" and "create directory"?

Multiple names for one object

- Logging goes by *object* and not name
- Representations can affect this (if you read raw disks, you're reading files; can your auditing system determine which file?)

Syntax Issues

- How to log?
- What data should be in the log file?
- How should it be expressed?

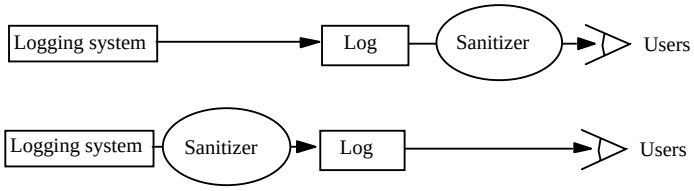
Data that is logged may be ambiguous

- Example: two optional text fields followed by two mandatory text fields
- If three fields, which of the optional fields is omitted?
  - Possible Solution: use grammar to ensure well-defined syntax of log files

Log Sanitization (purification)

- *Logs may contain confidential information*
  - *if to be used for auditing, they need to be removed*
- Only based on valid reasoning, *the system administration can monitor users*
  - *When attacking the system*
  - *When engaging in illegal activities.*
- Sites must protect the privacy of other users so that the investigators cannot determine

Logging Organization



Top prevents information from leaving site

- Users' privacy not protected from system administrators, other administrative personnel

Bottom prevents information from leaving system

- Data simply not recorded, or data scrambled before recording

Application Logging

Applications logs consists of entries made by applications

- Applications control what is logged
- Typically use high-level abstractions
- Does not include detailed, system call level information such as results, parameters, etc.

Auditing Mechanisms

Systems use different mechanisms

- Most common is to log *all* events by default, allow system administrator to disable logging that is unnecessary

Two examples

- One audit system designed for a secure system
- One audit system designed for non-secure system

Secure Systems

Auditing mechanisms integrated into system design and implementation

Security officer can configure reporting and logging:

- To report specific events
- To monitor accesses by a subject
- To monitor accesses to an object

Controlled at audit subsystem

- Irrelevant accesses, actions not logged

Non-Secure Systems

Have some limited logging capabilities
- Log accounting data, or data for non-security purposes
- Possibly limited security data like failed logins

Auditing subsystems focusing on security usually added after system completed
- May not be able to log all events, especially if limited kernel modifications to support audit subsystem

Key Points

- Logging is collection and recording
- Auditing is analysis
  - Need to have clear goals when designing an audit system
- Auditing should be designed into system
  - not patched into system after it is implemented
- Browsing through logs helps auditors determine completeness of audit
  - determine effectiveness of audit mechanisms!