Auditing

Introduction

- Overview
- What is auditing?
- What does an audit system look like?
- Design and mechanisms
- Browsing logs

What is Auditing?

- *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

Uses

- Describe security state
  - Determine if system enters unauthorized state
- Evaluate effectiveness of protection mechanisms
  - Determine which mechanisms are appropriate and working
  - Deter attacks because of presence of record

Problems

- What do you log?
  - Hint: looking for violations of a policy, so record *at least* what will show such violations
- What do you audit?
  - Need not audit everything
  - Key: what is the policy involved?

Three Components

Audit System Structure

- *Logger*: records information, usually controlled by parameters
- *Analyzer*: analyzes logged information looking for something
- *Notifier*: reports results of 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

Example: Windows 10

- Different logs for different types of events
  - *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
  - *Setup event* log records events occurring during application installation
  - *Forwarded event log* records entries forwarded from other systems
- Logs are binary; use *event viewer* to see them
- If log full, can have system shut down, logging disabled, or logs overwritten

Windows 10 Sample Entry

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	03/20/2017 12:02:59 PM
Event ID:	4634	Task Category:	Logoff
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	McLaren
OpCode:	Info		

General:

An account was logged off.

Subject:

Security ID:	MCLAREN\matt
Account Name:	matt
Account Domain:	MCLAREN
Logon ID:	0xACBA30

Details:

+ System

- EventData

TargetUserSID	S-1-5-22-2039872233-608055118-4446661516-2001
TargetUserName	matt
TargetDomainName	MCLAREN
TargetLogonId	Oxacba30

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

Examples

- Using *swatch* to find instances of *telnet* from *tcpd* logs:

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

- Intrusion detection analysis engine (director)
  - Takes data from sensors and determines if an intrusion is occurring

Notifier

- Informs analyst, other entities of results of analysis
- May reconfigure logging and/or analysis on basis of results

Examples

- Three failed logins in a row disable user account
  - Notifier disables account, notifies sysadmin

Two Levels of Logging

Application Logging

- Applications logs made by applications
  - Applications control what is logged
  - Typically use high-level abstractions such as:

    su: bishop to root on /dev/ttyp0

  - Does not include detailed, system call level information such as results, parameters, etc.

System Logging

- Log system events such as kernel actions; typically, low-level events

```
3876 ktrace    CALL   execve(0xbfbff0c0,0xbfbff5cc,0xbfbff5d8)
3876 ktrace    NAMI   "/usr/bin/su"
3876 ktrace    NAMI   "/usr/libexec/ld-elf.so.1"
3876 su                RET    xecve 0
3876 su                CALL   __sysctl(0xbfbff47c,0x2,0x2805c928,0xbfbff478,0,0)
3876 su                RET    __sysctl 0
3876 su                CALL   mmap(0,0x8000,0x3,0x1002,0xffffffff,0,0,0)
3876 su                RET    mmap 671473664/0x2805e000
3876 su                CALL   geteuid
3876 su                RET    geteuid 0
```

Contrast

- Differ in focus
  - Application logging focuses on application events, like failure to supply proper password, and the broad operation (what was the reason for the access attempt?)
  - System logging focuses on system events, like memory mapping or file accesses, and the underlying causes (why did access fail?)
- System logs usually much bigger than application logs
- Can do both, try to correlate them

State-based vs. Transition-based Logging

State-Based Auditing

- Log information about state and determine if state allowed
  - Assumption: you can get a snapshot of system state
  - Snapshot needs to be consistent
  - Non-distributed system and distributed system

Transition-Based Auditing

- Log information about action, and examine current state and proposed transition to determine if new state would be disallowed
  - Note: just analyzing the transition may not be enough; you may need the initial state
  - Tend to use this when specific transitions *always* require analysis (for example, change of privilege)

Browsing Logs

Audit Browsing

- Goal of browser: present log information in a form easy to understand and use
- Several reasons to do this:
  - Audit mechanisms may miss problems that auditors will spot
  - Mechanisms may be unsophisticated or make invalid assumptions about log format or meaning
  - Logs usually not integrated; often different formats, syntax, *etc*.

Browsing Techniques

- Text display
  - Does not indicate relationships between events
- Hypertext display
  - Indicates local relationships between events
- Relational database browsing
  - DBMS performs correlations, so auditor need not know in advance what associations are of interest
- Replay
  - Shows events occurring in order; if multiple logs, intermingles entries
- Graphing
  - Nodes are entities, edges relationships

Key Points

Key Points

- Auditing
- Logger, analyzer, and notifier
- System log and application log
- State-based and transition-based logging
- Browsing logs