PURDUE UNIVERSITY

CS52600: Information Security

Audit

5 November 2010

Prof. Chris Clifton

What is Auditing?

- Webster: a methodical examination and review
- Information Security: An a-posteriori technique to identify security violations
  - How does this help maintain security?

Issues

- What information do we need?
  - After the fact – current state of system isn't enough
  - *logging*
- How do we perform an Audit?
  - Audit methodology
- What do we do with the results?

CS52600

Logging

- Goal: Record all information that might be needed for an audit
  - Authentication attempts
    - Failed only?
  - Access to trusted resources
    - All? Just failed attempts?
- Log must enable detection of security violations
  - Is this enough?

CS52600

Example: Bell-LaPadula

- What must be logged?
  - Action (read/write)
  - Level of subject
  - Level of object
- Can now check
  - Read: $S \geq O$
  - Write: $O \geq S$
- Is this necessary?
  - What if system validated as not allowing illegal read/write?
- What about change of security level?

Logging *Trusted* Operations

- Secure system *prevents* security violations
- Trusted components: those that *can* violate security
  - Assumptions made to justify system secure
- Log actions by trusted components
  - Change in security level
  - Writes performed when not at maximum level
  - *All* reads (why?)

Logging: Implementation

- Log Format
  - Standard
  - Machine readable
  - Transform to human readable
- Wrong:
  - Connection blocked from 128.10.3.4 to cs.purdue.edu
  - Right: Structured format, standards

Logging: Implementation

- Log must be protected
  - Doesn't do any good if security violations erased from log
- Sanitization
  - Remove sensitive information from log
    - Why?
  - Before or after logging?

Audit

- Detect security violation
  - State-based auditing: identify if state at prior time is valid
  - Transition-based auditing: Identify if prior transition would lead to unauthorized state
- Detect attempts to breach security
  - Not necessarily violations

CS52600

Using Audit Results

- Repair
  - Recover critical information
  - Risk mitigation
  - Restore integrity
- Punish
  - Identify violator

Both may demand additional logged information

CS52600

Why Not Design a Secure System Up Front?

- Audit catches security violations
  - Why allow them in the first place?
- Possible reasons:
  - (un)trusted components
  - Changes in security policy

CS52600

Other types of Audit

- Logging done for many reasons
  - System tuning
  - Backup / failure recovery
  - ?
- Can this be used for security audit?
  - Example: Basic Security Module add-on to SunOS
  - Defines audit events
  - Captures identity, action

CS52600

Example / Reading

- Network File System logging
  - What is the policy?
  - What requests logged?
  - What information logged?
- Read Bishop 24.6

CS52600