

# Security Topics in Networking and Distributed Systems

## CS 590D

### **Lecture 7**

Department of Computer Sciences  
Purdue University

# Outline

- An Intrusion-detection Model. D. E. Denning. IEEE Trans. on Software Engg., SE-13, pp.222-232, February 1987.
- Detecting Intruders in Computer Systems T. F. Lunt Conference on Auditing and Computer Technology, 1993.



# Why Intrusion Detection?

- Systems have many security flaws; assuming that we can fix all of them and have perfect secure system is unrealistic.
- Replacing existing systems is difficult.
- Even if we do have secure systems, user can still abuse their privileges.

# Detect ...

- Attempted break-in
- Masquerading or successful break-in
- Penetration by legitimate users
- Leakage by legitimate user
- Inference by legitimate user
- Trojan horse (remember the rootkits?)
- Virus
- Denial-of-service

# Approaches

- **Knowledge-based approach:**
  - Accumulates knowledge about specific attacks and system vulnerabilities
  - Any action that is not explicitly recognized as an attack is considered acceptable
  - Potential for low false alarm rates
- Drawbacks:
  - difficult to gather/update information about known attacks
  - Very focused on attack/platform/environment
  - Does not work very well to detect detect insider attacks

# Approaches

- **Behavior-based approach:**
  - Detect intrusion by observing a deviation from normal or expected behavior of the system or the users.
  - Behavior model compared with current activity, alarm generated
  - Can detect attempts to exploit new and unforeseen vulnerabilities.
  - Can detect even with partial information more general, less dependent of platform and attack specifics
- Drawbacks:
  - High false alarm rate
  - Requires periodic update of the profile

# Host-based/Network-based

- **Host-based intrusion detection**
  - uses log files and/or the system's auditing agents as sources of data
  - checks the integrity of your system files and watch for suspicious processes
- **Network-based intrusion detection**
  - monitors the traffic on its network segment as a data source.
- Effective intrusion detection: use a combination of network- and host-based intrusion detection.

# An Intrusion Detection Model

- **Subjects**: usually users, can be a process
- **Objects**: files, commands, devices
- **Audit records**: user login, command execution, file access
- **Profiles**: behavior of subjects with respect to objects based on metrics and observed activity
- **Anomaly records**
- **Activity rules**: action taken when condition satisfied, results in updating profile, detect abnormal behavior, produce reports.

# Audit Records

- What level to do the auditing?
  - OS system calls
  - command line
  - network data (e.g., from router and firewall logs or MIBs)
- Processes
- Keystrokes
- Subject, action, object, exception-condition, resource-usage, timestamp
- **AUDITING MUST BE DONE AT LOWEST LEVEL**

# Observed Behavior

- Observed behavior: metric and model
- Methods to detect abnormal behavior
  - Statistical methods
  - Neural networks
  - Machine learning
- Metric: a random variable  $x$  representing a quantitative measure accumulated over a period
- Metrics:
  - Event counter: example # of logins
  - Interval timer: length of time between the timestamps in the respective audit records
  - Resource measure: quantity of resources consumed by an action during a period

# Abnormal Behavior

- Detecting abnormal behavior depends significantly on how much the user behavior fluctuates from normal
- Difficult to detect users that abuse their privileges

# Examples of Statistical Models

- **Operational**: compare fixed limits
- **Mean and standard deviation**: abnormal behavior if it falls outside the interval of confidence
- **Multivariate model**: correlations among several metrics
- **Markov process model**: apply only to event counters' abnormal behavior if the probability determined by the previous state and transition matrix is very low
- **Time series model**: abnormal behavior if probability that an event occurs at that time is too low

# Possible Profiles

- Login and session activity
  - Login frequency
  - Location frequency
  - Last login
  - Session elapsed time
  - Session output
  - Session CPU, IO, Pages
  - Password Fails
  - Location Fails
- Command or Program execution
  - Execution frequency
  - Program CPU, IO
  - Execution Denied
  - ProgramResourceExhaustion



# Possible Profiles

- File-Access activity
  - Read Frequency, Write Frequency, Create Frequency, Delete Frequency
  - Records Read, Records Written
  - Read Fails, Write Fails, Create Fails, Delete Fails
  - File Resource Exhaustion



# Anomaly Records and Activity Rules

- If abnormal behavior detected, an anomaly record is generated
  - **Event**
  - **Timestamp**
  - **Profile**
- Activity rule: specifies an action to be taken when an audit record or an anomaly record is detected
  - **Audit-record rule**
  - **Periodic-activity-update**
  - **Anomaly-record rules**
  - **Periodic-anomaly-analysis rules**

# Machine Learning

- Builds a tree of statistical “rules”
- Branches are labeled with conditional probabilities
- Low-occurrence branches are combined
- Tree is “trained” from a few days of data
- Tree cannot be updated to “learn” as usage patterns change
- Activity is considered abnormal if it does not “match” a branch in the tree or if it matches a branch with low conditional probability last node

# Use of Expert Systems

- Expert system
- Rule-based approach
- List is not comprehensive
- Drawback: the system is looking for known threats (expressed in the rules), not-known attacks can not be detect and represent a high danger.

# Traps

- Set up trap doors for intruders: fake accounts, with magic passwords
- Once these accounts are broken, an alarm is triggered
- Honeypots:
  - programs that simulate one or more network services on a computer's port
  - attacker assumes vulnerable services are run and he gets tricked to attack the machine
  - can be used to log access attempts to those ports including the attacker's keystrokes.

# Integrity Checker

- Compromise of a system usually affects the file systems
- Main idea: compute a check sum of the file system, build an external database of signatures
- Use CRC-32 or hash functions as MD5
- Challenges
  - Maintain the database (update when the file system updates)
  - Make sure that the initial database is created on a non-compromised system
- Example: Tripwire

# Still to Be Addressed/Improved

- Discriminate between suspicious and normal behaviors
- High number of false alarms that required additional investigation
- Too much data must be logged
- Audit logs do not scale well
- Cannot deal with missing, incomplete, untimely, or otherwise faulty data
- Detect a wide variety of intrusion types and unanticipated attacks

# Still to Be Addressed/Improved ...

- Detection performance in realistic settings with single methods and combinations of methods
- Distinction from common failure modes
- What data to collect/observe
- What data to collect for maximal effectiveness; network instrumentation
- Automated response
- Real-time detection