CS 392 – Network Security

Nasir Memon – Polytechnic University
Module 5 – Intrusion Detection
Course Logistics

- HW 2 due tonight!!
- HW 2 assigned due next week.
- Lab open. Schedule on web.
Types of intrusion

- External attacks
  - attempted break-ins, denial of service attacks, port-scans
- Masquerading as some other user
- Misuse of privileges, malicious attacks
- Clandestine users: exploiting bugs in privileged programs
Fighting intrusion

- Prevention: isolate from network, strict authentication measures, encryption
- Preemption:
  - “do unto others before they do unto you”
- Deterrence: dire warnings,
  - “we have a bomb too.”
- Deflection: diversionary techniques to lure away
- Detection
- Counter attacks
What is IDS?

An intrusion is defined as the unauthorized use, misuse, or abuse of computer systems by either authorized users or external perpetrators.

Intrusion Detection System (IDS) is the system that attempts to identify these intrusions.

Intrusion detection is the process of identifying and responding to malicious activity targeted at computing and networking resources.
Why Is IDS Necessary?

It’s difficult to ensure that an information system will be free of security flaws.

New bugs keep surfacing.

Even if network is secured from outsiders, how do you guard against malicious insiders?

It’s hard to prevent naive users from opening e-mail viruses and download rigged programs unknowingly.
Examples of IDS in daily life

- Car Alarms
- Fire detectors
- CO detectors
- House Alarms
- Surveillance Systems
- Spy Satellites, and spy planes (U2 and SR-71)
Architecture of a Generic IDS

[Diagram of an IDS architecture with labels for Database, Configuration, Detector, Countermeasure, Alarms, Audits, Actions, and Probes connected to a central system.]
Efficiency of IDS

- **Accuracy**: the proper detection of attacks and the absence of false alarms
- **Performance**: the rate at which audit events are processed
- **Completeness**: to detect all attacks
- **Fault tolerance**: resistance to attacks
- **Timeliness**: time elapsed between intrusion and detection
Taxonomy Elements

- Detection method: the characteristics of the analyzer
  - Behavior based
  - Knowledge based

- Behavior on detection: the response to attacks
  - Active response
  - Passive alerting
Taxonomy Elements (cont’d)

- Audit source location: the kind of input information of the analyzer
  - Host-based: system sources, accounting, syslog, C2 security audit
  - Network-based: SNMP information, network packets
  - Application log files
  - Intrusion-detection alerts
Taxonomy Elements (cont’d)

- Detection paradigm: detection mechanism
  - State-based
  - Transition-based
    - For each of these you could have perturbing or non-perturbing paradigm.

- Usage frequency
  - Continuous monitoring
  - Periodical analysis
Detection Method

- Knowledge-based: to use the information about the attacks and look for evidence of the exploitation of these attacks.
- Behavior-based: to use the information about the normal behavior of the system it monitors and look for deviations from the observed usage.
Knowledge-based Intrusion Detection

**Advantages**
- Low false-alarm rates
- Easy for security officer to understand the problem

**Drawbacks**
- Difficult to gather and maintain the required information on the attacks
- Generalization issue
- Difficult to detect the insider attacks involving an abuse of privileges
Knowledge-based Techniques

- Expert systems
- Signature analysis
- Petri nets
- State-transition analysis
Behavior-based Intrusion Detection

- **Advantages**
  - The ability of detecting new and unforeseen vulnerabilities
  - Less dependent on OS-specific mechanisms
  - “Abuse of privilege” types of attacks detection

- **Drawbacks**
  - High false-alarm rate
Behavior-based Intrusion Detection

- Statistics
- Expert systems
- Neural networks
- User intention identification
- Computer immunology
## Detection Techniques used in IDS

<table>
<thead>
<tr>
<th>Origin</th>
<th>Name</th>
<th>Knowledge-based</th>
<th>Behavior-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Univ. Namur</td>
<td>ASAX</td>
<td>ES X</td>
<td></td>
</tr>
<tr>
<td>AT&amp;T</td>
<td>ComputerWatch</td>
<td></td>
<td>ES X</td>
</tr>
<tr>
<td>USAF</td>
<td>Haystack</td>
<td>SA X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DIDS</td>
<td>PN X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hyperview</td>
<td>STA X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IDES</td>
<td>Stat X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NIDES</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Emerald</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purdue U.</td>
<td>IDIOT</td>
<td></td>
<td>ES X</td>
</tr>
<tr>
<td>UCDavis</td>
<td>NSM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grl DS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W&amp;S</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nadir</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetRanger</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RealSecure</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SecureNet</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CyberCop Server</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>STAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>USTAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stanford U.</td>
<td>Swatch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCNC-NCSU</td>
<td>JiNao</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Behavior on Detection

- Passive vs. Active detection
- Most early IDS’s were passive.
- Newer techniques moving to Active
  - Shut down vulnerable services
  - Reconfigure firewall
  - Restore modified configuration file
Audit source location: Host

- System sources: ps, who etc.
  - Difficult to use in continuous audit collection

- Accounting:
  - Lack of parameterizations
  - Lack of precise time-stamp
  - Lack of precise command identification
  - Absence of system daemon activity
  - Delay of obtaining information

- Syslog: swatch
Audit source location: Network

- SNMP information
- Network packets
  - Advantages:
    - Detection of network specific attacks
    - Usually information collected on separate machine so no impact on rest of system
    - Facilitated cross-platform acquisition
  - Disadvantages
    - Difficult to identify culprit when intrusion detected
    - With switched environment selection of appropriate location for sniffers not obvious
    - Encryption makes it impossible to analyze payload
    - Could be vulnerable to DoS
Audit source location: Network

- Application server
  - Advantages
    - Accuracy
    - Completeness
    - Performance
  - Disadvantages
    - Attacks detected only when application log is written or application maliciously used
    - Many low level attacks will not be discovered: ping of death, teardrop.
Reusability

- CIDF (Common Intrusion Detection Framework) by DARPA
- IDWG (Intrusion Detection exchange format Working Group) by IETF
Characteristics of a normal computer

- Predictable usage pattern
  - A typical user checks mail, checks news, stock prices, bank account when he/she logs in. Its abnormal for a user to run user management utilities once he/she logs in.

- Actions of Users and processes do not subvert security policies of the system
  - Users who try to take advantage of a race condition to gain access to files
Characteristics of a normal computer

- Actions of processes confirms to security policies
  
  Some examples of action that do not confirm to confirms to security policies:
  - sendmail trying to read passwd file!!
  - System logger sending messages to a file in guest users directories
Modeling Techniques for Intrusion Detection

- Look for abnormalities (anomaly detection)
  - Most Intruders use some kind of attack tool, such as rootkits, to break into a system. Root kits modify system configuration file and binaries, which could be detected.

- Look for unusual actions by users (misuse detection)
  - User renaming, copying, or even trying to delete system files
Modeling Techniques (contd)

- Look for inconsistent behavior of privileged programs (Specification-based detection)
  - Ftpd trying to read system configuration files (buffer overflow)
  - passwd trying to write something to a file in user home directory (race condition)
  - Tcp stack overloading the system (DOS attack)
Basic Intrusion Detection

- Main goals of IDS are:
  - Diversity
    - Detect wide verity of intrusions, by learning and adapting to the environment.
  - Timeliness
    - Detect intrusions in a timely fashion, but being timely does not mean real-time.
  - Accuracy
    - Detect attacks accurately by minimizing false positives and negatives
  - Fault Tolerance
    - With stand know IDS attacks such insertion, evasion, and Denial of Service.
Model (Intrusion Detection Techniques)

ID systems are made of one or more models for detecting intrusions. The different kinds of models are:

- Anomaly Modeling
- Misuse Modeling
- Specification-based Modeling
Anomaly Modeling

- This technique basically looks for unexpected usage pattern
- A formal definition

“Anomaly detection analyzes a set of characteristics of the system and compares their behavior to a set of expected values. It reports when the computed statistics do not match the expected measurements.”
Anomaly Modeling

- Anomaly modeling is often done with three different types of statistical models
  - Threshold Metric
  - Statistical moments
  - Marker model
Anomaly Modeling - Threshold Metric

- Based on the cardinality of events that happens over a period of time
  - An alarm is raised if fewer then m or more than n events occur
    Example: Win2k lock a user after n unsuccessful login attempts here lower limit is 0 and upper limit is n
- The challenge in this sub-model is determining m and n
Anomaly Modeling - Statistical moments

- In statistics a moment is mean, standard deviation, or any other correlations
- Analyzer that uses this sub-model knows these moments and any event that falls outside the set interval above or below the moment is said to be anomalous
- Also takes change to system into account by aging data or altering the statistical rule base upon which they make decisions
- Provides more flexibility then threshold model, but more complex.
Anomaly Modeling – Marker Model

- This model identifies intrusion by
  - Examines the system at fixed intervals and keeps track of its state.
  - A probability for each state at a given time interval is computed.
  - When an event occurs, it changes the state of the system, and if the probability for that state to occur at that time interval is low, that event is considered anomalous.
Misuse Modeling

- Misuse refers to an attack by an insider or authorized user.
- This kind of modeling requires knowledge of vulnerabilities that attackers use to exploit.
- ID system incorporates this knowledge into rule set and uses this rule set.
- If an event matches any rule in the set, it reports a possible intrusion is under way.
- Misuse based ID systems often use an expert system to analyze the data and apply the rule set.
Misuse Modeling

- These systems cannot detect attacks unknown to the developers of the rule set, even variations of known attack.
Specification Modeling

- Specification Modeling look states known not to be good.
- When the system enters such a state it reports possible intrusion.

Definition: Specification-based detection determines whether a sequence of instructions violates a specification of how a program, or system, should execute. If so, it reports a potential intrusion.
IDS Architecture

Director – Corresponds to analyzer, it analyzes data sent by the agents. Directors are the heart of the ID systems and they are made using one or more ID models.

Agents – Provides data to directory. Agents collect data from system logs, messages, and application logs.

Notifier – Responds to Intrusions detected by Director.
A Simplified IDS Architecture

- DATABASE
- CONFIGURATION
- DIRECTOR
- NOTIFIER
- ALARMS
- AUDITS
- ACTIONS
- PROBES
- AGENT
- SYSTEM
A More Realistic Architecture

- Inductive Learning Engine
- Audit Data Preprocessor
- Audit Records
- Activity Data
- Detection Engine
- Host-logs: keyboard, application network-logs
- Policies by experts
- Decision Table
- Final Assertion
- Action/Report
Agents - Host-Based Information Gathering

- Uses system and application logs
- Logs could be security or accounting logs, depending upon the goals of the intrusion detection mechanism
- Agents could also be integrated into the kernel for efficiency
  - But these agents are not portable
  - Putting these agents into application make application complex, therefore its best to let application log its output, and the agent simply analyzes that log
Agents - Combining Sources

Agents aggregate sources before presenting it to director. The goal is to provide adequate data so that directory does not have to do much work to understand the data.

Data can be viewed at several levels.

Example:

Consider a BSD UNIX system with two source of information:

- Application level log
- System call level log
## Application call level log looks like:

Feb 12 14:30:00 nob su: root to bishop on /dev/ttyp5

## System call level log looks like:

<table>
<thead>
<tr>
<th>13285 su CALL geteuid</th>
<th>13285 su lseek .....</th>
</tr>
</thead>
<tbody>
<tr>
<td>13285 su RET geteuid 0</td>
<td>13285 su lseek .....</td>
</tr>
<tr>
<td>13285 su CALL stat(...,..)</td>
<td>13285 su CALL read(...)</td>
</tr>
<tr>
<td>13285 su NAMO “..”</td>
<td>13285 su GIO fd 3 read</td>
</tr>
<tr>
<td>13285 su RET stat 0</td>
<td>13285 su RET read ....</td>
</tr>
<tr>
<td>13285 su CALL open(...)</td>
<td>13285 su CALL close(0x3)</td>
</tr>
<tr>
<td>13285 su NAMI “....”</td>
<td>13285 su RET close 0</td>
</tr>
</tbody>
</table>
Agents - Combining Sources

The logs were created by BSD when a user changes privileges by typing su.

Application log provides only the event that a user has elevated his/her privileges, and omits information about the process that elevated the privileges.

System log only provides information about the process, and constructing events from system could be challenging to director.
Agents - Combining Sources

So what kind of information should the agent provide to director? (high level application logs or low level system call logs)

Well it depends upon the ID mechanism used in directors. If the director is made using marker model then system call level logs might be more useful then application level logs. Or lets say director uses Misuse detection model then application level log would be useful to track the users easily and faster.
Director

Director is the heart of the ID system. It also reduces the incoming logs to eliminate unnecessary records. As stated earlier directors are made using more then one model or techniques, because different techniques highlight different aspects of intrusions.

Directors also adapt and alter profiles to changes in the system. Typical adaptive director use machine learning to determine how to alter their behavior.
Notifier

Notifier accepts information from director and takes appropriate action. The simplest notifiers just send message in form of e-mail or alarm to security office, while the advanced ones can reconfigure local firewalls, lock users, or stop services that are misbehaving.
Intrusion Response

How can the system be protected once the intrusion is detected?

The field intrusion response deals with this problem. This field is made of:

- Incident Prevention
- Intrusion handling
- Confinement
- Eradication
Ideally, intrusion attempts will be detected and stopped before they succeed. In order to prevent an attack, the attack has to be identified before it completes.

One efficient way of prevention is to Jail an attacker.
Intrusion Response - Incident Prevention

Jailing an attacker allows the attacker to think the attack has succeeded, but place them in an confined area were their behavior can be monitored and controlled.

Example:
Honey Pots
Intrusion Response – Intrusion Handling

- Handling intrusion means restoring the system to comply with the site policy, and taking action against attackers. Intrusion handling consists of six phases:
  - Preparation for an attack
  - Identification of an attack
  - Confinement of an attack
  - Eradication of the attack
  - Recovery from the attack
  - Follow-up to the attack
Confining an attack means limiting the access to the attacker. There are two approaches to do this:

- Passive monitoring
- Constrain access