CS 356 – Lecture 17 and 18
Intrusion Detection

Spring 2013
Review

• Chapter 1: Basic Concepts and Terminology
• Chapter 2: Basic Cryptographic Tools
• Chapter 3 – User Authentication
• Chapter 4 – Access Control Lists
• Chapter 5 – Database Security (skipped)
• Chapter 6 – Malicious Software
• Networking Basics (not in book)
• Chapter 7 – Denial of Service
• Chapter 8 – Intrusion Detection
Chapter 8

Intrusion Detection
Intruders

- two most publicized threats to security are malware and intruders
- generally referred to as a *hacker* or *cracker*

- **classes:**

<table>
<thead>
<tr>
<th>masquerader</th>
<th>misfeasor</th>
<th>clandestine user</th>
</tr>
</thead>
<tbody>
<tr>
<td>• likely to be an outsider</td>
<td>• generally an insider</td>
<td>• can be either insider or outsider</td>
</tr>
<tr>
<td>• an unauthorized individual who penetrates a system to exploit a legitimate user account</td>
<td>• legitimate user who misuses privileges</td>
<td>• individual who seizes supervisory control to evade auditing and access controls or to suppress audit collection</td>
</tr>
</tbody>
</table>
Examples of Intrusion

- remote root compromise
- web server defacement
- guessing / cracking passwords
- copying databases containing credit card numbers
- viewing sensitive data without authorization
- running a packet sniffer
- distributing pirated software
- using an unsecured modem to access internal network
- impersonating an executive to get information
- using an unattended workstation
Hackers

• motivated by thrill of access and/or status
  – hacking community is a strong meritocracy
  – status is determined by level of competence

• benign intruders consume resources and slow performance for legitimate users

• intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to help counter hacker threats
  – can restrict remote logons to specific IP addresses
  – can use virtual private network technology (VPN)

• intruder problem led to establishment of computer emergency response teams (CERTs)
1. select the target using IP lookup tools such as NSLookup, Dig, and others
2. map network for accessible services using tools such as NMAP
3. identify potentially vulnerable services (in this case, pcAnywhere)
4. brute force (guess) pcAnywhere password
5. install remote administration tool called DameWare
6. wait for administrator to log on and capture his password
7. use that password to access remainder of network
Criminals

• organized groups of hackers now a threat
  – corporation / government / loosely affiliated gangs
  – meet in underground forums
  – common target is credit card files on e-commerce servers

• criminal hackers usually have specific targets
  – once penetrated act quickly and get out

• IDS / IPS can be used but less effective

• sensitive data should be encrypted
Criminal Enterprise
Patterns of Behavior

- act quickly and precisely to make their activities harder to detect
- exploit perimeter via vulnerable ports
- use Trojan horses (hidden software) to leave back doors for re-entry
- use sniffers to capture passwords
- do not stick around until noticed
Insider Attacks

• among most difficult to detect and prevent
• employees have access and systems knowledge
• may be motivated by revenge/entitlement
  – employment was terminated
  – taking customer data when moving to a competitor
• IDS / IPS can be useful but also need:
  – enforcement of least privilege, monitor logs, strong authentication, termination process
Internal Threat Patterns of Behavior

- create network accounts for themselves and their friends
- access accounts and applications they wouldn't normally use for their daily jobs
- access the network during off hours
- perform large downloads and file copying
- e-mail former and prospective employers
- visit web sites that cater to disgruntled employees, such as f'dcompany.com
- conduct furtive instant-messaging chats
The following definitions from RFC 2828 (Internet Security Glossary)

• **Security Intrusion**: A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.

• **Intrusion Detection**: A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.
Intrusion Detection Systems (IDSs)

- **host-based IDS**
  - monitors the characteristics of a single host for suspicious activity

- **network-based IDS**
  - monitors network traffic and analyzes network, transport, and application protocols to identify suspicious activity

**comprises three logical components:**

- sensors - collect data
- analyzers - determine if intrusion has occurred
- user interface - view output or control system behavior
IDS Principles

- assume intruder behavior differs from legitimate users
- overlap in behaviors causes problems
  - false positives
  - false negatives

Figure 8.1 Profiles of Behavior of Intruders and Authorized Users
IDS Requirements

- run continually
- be fault tolerant
- resist subversion
- impose a minimal overhead on system
- configured according to system security policies
- adapt to changes in systems and users
- scale to monitor large numbers of systems
- provide graceful degradation of service
- allow dynamic reconfiguration
Host-Based IDS

- adds a specialized layer of security software to vulnerable or sensitive systems
- monitors activity to detect suspicious behavior
  - primary purpose is to detect intrusions, log suspicious events, and send alerts
  - can detect both external and internal intrusions
Host-Based IDS Approaches to Intrusion Detection

**anomaly detection**

- threshold detection
  - involves counting the number of occurrences of a specific event type over an interval of time
  - profile based
    - profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts

**signature detection**

- involves an attempt to define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder
Audit Records

**native audit records**

- multiuser operating systems include accounting software that collects information on user activity
- advantage is that no additional collection software is needed
- disadvantage is that records may not contain the needed information or in a convenient form

**detection-specific audit record**

- collection facility that generates records containing only information required by the IDS
- advantage is that it could be made vendor independent and ported to a variety of systems
- disadvantage is the extra overhead of having, in effect, two accounting packages running on a machine
### Measures That May Be Used For Intrusion Detection

<table>
<thead>
<tr>
<th>Measure</th>
<th>Model</th>
<th>Type of Intrusion Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login frequency by day and time</td>
<td>Mean and standard deviation</td>
<td>Intruders may be likely to log in during off-hours.</td>
</tr>
<tr>
<td>Frequency of login at different locations</td>
<td>Mean and standard deviation</td>
<td>Intruders may log in from a location that a particular user rarely or never uses.</td>
</tr>
<tr>
<td>Time since last login</td>
<td>Operational</td>
<td>Break-in on a &quot;dead&quot; account. Significant deviations might indicate masquerader.</td>
</tr>
<tr>
<td>Elapsed time per session</td>
<td>Mean and standard deviation</td>
<td>Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.</td>
</tr>
<tr>
<td>Quantity of output to location</td>
<td>Mean and standard deviation</td>
<td></td>
</tr>
<tr>
<td>Session resource utilization</td>
<td>Mean and standard deviation</td>
<td>Unusual processor or I/O levels could signal an intruder.</td>
</tr>
<tr>
<td>Password failures at login</td>
<td>Operational</td>
<td>Attempted break-in by password guessing.</td>
</tr>
<tr>
<td>Failures to login from specified terminals</td>
<td>Operational</td>
<td>Attempted break-in.</td>
</tr>
</tbody>
</table>

#### Command or Program Execution Activity

<table>
<thead>
<tr>
<th>Measure</th>
<th>Model</th>
<th>Type of Intrusion Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution frequency</td>
<td>Mean and standard deviation</td>
<td>May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.</td>
</tr>
<tr>
<td>Program resource utilization</td>
<td>Mean and standard deviation</td>
<td>An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.</td>
</tr>
<tr>
<td>Execution denials</td>
<td>Operational model</td>
<td>May detect penetration attempt by individual user who seeks higher privileges.</td>
</tr>
</tbody>
</table>

#### File Access Activity

<table>
<thead>
<tr>
<th>Measure</th>
<th>Model</th>
<th>Type of Intrusion Detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read, write, create, delete frequency</td>
<td>Mean and standard deviation</td>
<td>Abnormalities for read and write access for individual users may signify masquerading or browsing. Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.</td>
</tr>
<tr>
<td>Records read, written</td>
<td>Mean and standard deviation</td>
<td></td>
</tr>
<tr>
<td>Failure count for read, write, create, delete</td>
<td>Operational</td>
<td>May detect users who persistently attempt to access unauthorized files.</td>
</tr>
</tbody>
</table>
Signature Detection

- rule-based anomaly detection
  - historical audit records are analyzed to identify usage patterns
  - rules are generated that describe those patterns
  - current behavior is matched against the set of rules
  - does not require knowledge of security vulnerabilities within the system
  - a large database of rules is needed

- rule-based penetration identification
  - key feature is the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses
  - rules can also be defined that identify suspicious behavior
  - typically rules are specific to the machine and operating system
# USTAT Actions vs. SunOS Event Types

<table>
<thead>
<tr>
<th>USTAT Action</th>
<th>SunOS Event Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt</td>
</tr>
<tr>
<td>Write</td>
<td>truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct</td>
</tr>
<tr>
<td>Create</td>
<td>mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_w, open_wt, open_wc, open_wct, mknod</td>
</tr>
<tr>
<td>Delete</td>
<td>rmdir, unlink</td>
</tr>
<tr>
<td>Execute</td>
<td>exec, execve</td>
</tr>
<tr>
<td>Exit</td>
<td>exit</td>
</tr>
<tr>
<td>Modify_Owner</td>
<td>chown, fchown</td>
</tr>
<tr>
<td>Modify_Perm</td>
<td>chmod, fchmod</td>
</tr>
<tr>
<td>Rename</td>
<td>rename</td>
</tr>
<tr>
<td>Hardlink</td>
<td>link</td>
</tr>
</tbody>
</table>
Figure 8.2 Architecture for Distributed Intrusion Detection
Distributed Host-Based IDS

Figure 8.3 Agent Architecture
Network-Based IDS (NIDS)

- Monitors traffic at selected points on a network.
- Examines traffic packet by packet in real or close to real time.
- May examine network, transport, and/or application-level protocol activity.
- Comprised of a number of sensors, one or more servers for NIDS management functions, and one or more management consoles for the human interface.
- Analysis of traffic patterns may be done at the sensor, the management server or a combination of the two.
NIDS Sensor Deployment

- inline sensor
  - inserted into a network segment so that the traffic that it is monitoring must pass through the sensor
- passive sensors
  - monitors a copy of network traffic

Figure 8.4 Passive NIDS Sensor (based on [CREM06])
Figure 8.5 Example of NIDS Sensor Deployment
Intrusion Detection Techniques

- **signature detection**
  - at application, transport, network layers; unexpected application services, policy violations
- **anomaly detection**
  - denial of service attacks, scanning, worms
- **when a sensor detects a potential violation it sends an alert and logs information related to the event**
  - used by analysis module to refine intrusion detection parameters and algorithms
  - security administration can use this information to design prevention techniques
Figure 8.6  Overall Architecture of an Autonomic Enterprise Security System

PEP = policy enforcement point
DDI = distributed detection and inference
Intrusion Detection Exchange Format

Figure 8.7 Model For Intrusion Detection Message Exchange
Honeypot

- decoy systems designed to:
  - lure a potential attacker away from critical systems
  - collect information about the attacker’s activity
  - encourage the attacker to stay on the system long enough for administrators to respond
  - filled with fabricated information that a legitimate user of the system wouldn’t access
  - resource that has no production value
    - incoming communication is most likely a probe, scan, or attack
    - outbound communication suggests that the system has probably been compromised
  - once hackers are within the network, administrators can observe their behavior to figure out defenses
Honeypot Deployment

Figure 8.8 Example of Honeypot Deployment
SNORT

- lightweight IDS
  - real-time packet capture and rule analysis
  - easily deployed on nodes
  - uses small amount of memory and processor time
  - easily configured

Figure 8.9  Snort Architecture
SNORT Rules

• use a simple, flexible rule definition language
• each rule consists of a fixed header and zero or more options

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>Generate an alert using the selected alert method, and then log the packet.</td>
</tr>
<tr>
<td>log</td>
<td>Log the packet.</td>
</tr>
<tr>
<td>pass</td>
<td>Ignore the packet.</td>
</tr>
<tr>
<td>activate</td>
<td>Alert and then turn on another dynamic rule.</td>
</tr>
<tr>
<td>dynamic</td>
<td>Remain idle until activated by an activate rule, then act as a log rule.</td>
</tr>
<tr>
<td>drop</td>
<td>Make iptables drop the packet and log the packet.</td>
</tr>
<tr>
<td>reject</td>
<td>Make iptables drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.</td>
</tr>
<tr>
<td>sdrop</td>
<td>Make iptables drop the packet but does not log it.</td>
</tr>
</tbody>
</table>
Examples of SNORT Rule Options

| meta-data   | msg | Defines the message to be sent when a packet generates an event. |
| reference   |     | Defines a link to an external attack identification system, which provides additional information. |
| classtype   |     | Indicates what type of attack the packet attempted. |
| payload     | content | Enables Snort to perform a case-sensitive search for specific content (text and/or binary) in the packet payload. |
|             | depth | Specifies how far into a packet Snort should search for the specified pattern. Depth modifies the previous content keyword in the rule. |
|             | offset | Specifies where to start searching for a pattern within a packet. Offset modifies the previous content keyword in the rule. |
|             | nocase | Snort should look for the specific pattern, ignoring case. Nocase modifies the previous content keyword in the rule. |
| non-payload | ttl  | Check the IP time-to-live value. This option was intended for use in the detection of traceroute attempts. |
|             | id   | Check the IP ID field for a specific value. Some tools (exploits, scanners and other odd programs) set this field specifically for various purposes, for example, the value 31337 is very popular with some hackers. |
|             | dsize | Test the packet payload size. This may be used to check for abnormally sized packets. In many cases, it is useful for detecting buffer overflows. |
|             | flags | Test the TCP flags for specified settings. |
|             | seq  | Look for a specific TCP header sequence number. |
|             | icmp-id | Check for a specific ICMP ID value. This is useful because some covert channel programs use static ICMP fields when they communicate. This option was developed to detect the stacheidraht DDoS agent. |
| post-detection | logto | Log packets matching the rule to the specified filename. |
|             | session | Extract user data from TCP Sessions. There are many cases where seeing what users are typing in telnet, rlogin, ftp, or even web sessions is very useful. |
Summary

• intruders
  – masquerader
  – misfeasor
  – clandestine user
– intruder behavior patterns
  – hacker
  – criminal enterprise
  – internal threat
– security intrusion/intrusion detection
– intrusion detection systems
  – host-based
  – network-based
  – sensors, analyzers, user interface

• host-based
  – anomaly detection
  – signature detection
– audit records
– distributed host-based intrusion detection
– network-based
  – sensors: inline/passive
– distributed adaptive intrusion detection
– intrusion detection exchange format
– honeypot
– SNORT