

Chapter 22 INTRUSION DETECTION

- **System managers must protect computer systems from attack.**
- **The mechanisms and techniques discussed in this book help protect systems, data, and resources.**
- **Yet, nothing is perfect.**
- **Even the best protected systems must be monitored to detect successful (and unsuccessful) attempts to breach security.**
- **This chapter discusses automated systems for detecting intrusions and looks at responses to attacks.**

22.1 Definitions

- **Computer systems that are not under attack exhibit several characteristics:**
 1. **The actions of users and processes generally conform to a statistically predictable pattern. A user who does only word processing when using the computer is unlikely to perform a system maintenance function.**
 2. **The actions of users and processes do not include sequences of commands to subvert the security policy of the system.**
 - **In theory, any such sequence is excluded.**
 - **In practice, only sequences known to subvert the system can be detected.**
 3. **The actions of processes conform to a set of specifications describing actions that the processes are allowed to do (or not allowed to do).**
- **Denning (An Intrusion-Detection Model, IEEE Transactions on Software Engineering, Vol. 13, No. 2, February 1987) hypothesized that systems under attack fail to meet at least one of these characteristics.**

Chapter 22 INTRUSION DETECTION

- **EXAMPLE:**
 - If the goal is to put in a back door, the intrusion may modify a system configuration file or program.
 - If the attacker enters the system as a nonprivileged user, he/she must acquire system privileges to change the files.
 - The nonprivileged user may not be a user who normally acquires system privileges (violates characteristic #1).
 - The techniques used to acquire those privileges may involve sequences of commands designed to violate the security policy of the systems (violates characteristic #2).
 - If they do not, the alterations in the system files may introduce elements that cause processes to act in ways that violate specifications (violates characteristic #3).
 - If the attacker modifies a user file, processes executing on behalf of that user can now behave in abnormal ways. Examples are given below (for characteristic #1):
 - Allowing network connections from sites not able to connect earlier.
 - Executing commands that the user did not execute before.
 - The commands may subvert the security policy, gaining system privileges for the user and the attacker (for characteristic #2).

Chapter 22 INTRUSION DETECTION

22.2 Basic Intrusion Detection

- The characteristics listed above guide the detection of intrusions.
- Once the province of technologically sophisticated, attacks against systems have been automated.
- Therefore, a sophisticated attack does not need to be the work of a sophisticated attacker.
- **DEFINITION:** An attack tool is an automated script designed to violate a security policy.
- **EXAMPLE:** rootkit
 - Exists for many versions of the UNIX operating system.
 - Designed to sniff passwords from the network and to conceal its presence.
 - Assumes that the installer has acquires root privileges.
 - Comes with tools to automate the installation procedure.
 - In addition to the network sniffing program, it comes with modified versions of system utilities:

Chapter 22 INTRUSION DETECTION

- The modified version of netstat, which lists network connections, uses a control file to determine which network connections to conceal.
 - The modified version of ps, which lists executing processes, uses another control file to determine which processes to conceal.
 - The modified versions of ls and du, which lists files and disk space used, use a control file to determine which files to conceal.
 - The network configuration program ifconfig, which reports network device configuration, claims that the network device is not in promiscuous mode, as it must be to sniff the network.
 - The login program accepts a “magic password” as authenticating any user. This enables attackers to return to obtain the sniffed passwords.
- All the replacement programs are modified so that they and the originals will produce the same checksum, as computing by a simple checksumming program.
 - Rootkit contains several other programs for concealing the attacker.
 - The program zapper deletes the user’s entry from the *utmp* file. This means that the user will not show up when logged in.
 - Fixer installs the programs and adjusts their permissions to match those of the replaced programs.
 - Attack tools do not change the nature of the intrusion detection fundamentally.
 - They do eliminate many errors arising from incorrect installation and perform routine steps to clean up detrius of the attacks.
 - They cannot eliminate all traces.

Chapter 22 INTRUSION DETECTION

- **Denning suggests automation of the intrusion detection process.**
 - Her specific hypothesis is that exploiting vulnerabilities requires an abnormal use of normal commands or instructions, so security violations can be detected by looking for abnormalities.
 - Her model is very general and includes abnormalities such as deviations from usual actions (anomaly detection), execution of actions that lead to break-ins (misuse detection), and actions inconsistent with the specifications of privileged programs (specification-based detection).
 - Systems that do this are called intrusion detection systems (IDS).
 - Their goals are fourfold:
 - Detect a wide variety of intrusions.
 - Intrusions from within the site, as well as those from outside the site, are of interest.
 - Furthermore, both known and previously known attacks should be detected.
 - This suggests a mechanism for learning or adapting to new types of attacks or to changes in normal user activity.
 - Detect intrusions in a timely fashion.
 - Timely does not need to be in real time.
 - Often, it suffices to discover an intrusion within a short period of time.
 - Real-time intrusion detection raises issues of responsiveness.
 - If every command and action must be analyzed before it can be executed, only a very simple analysis can be done before the computer or network being monitored becomes unusable.
 - On the other hand, in all but a few rare cases, determining that an intrusion took place a year ago is probably useless.

Chapter 22 INTRUSION DETECTION

- **Present the analysis in a simple, easy-to-understand format.**
 - **Ideally, this should be a binary indicator.**
 - It glows green for no detection intrusions and changes to red when an attack is detected.
 - Unfortunately, intrusions are rarely this clear-cut, so intrusion detection mechanisms must present more complex data to a site security officer.
 - The security officer determines what action (if any) to take.
 - Because intrusion detection mechanisms may monitor many systems (not just one), the user interface is a critical issue.
- **Be accurate.**
 - A false positive occurs when an intrusion detection system reports an attack, but no attack is underway.
 - False positives reduce confidence in the correctness of the results as well as increase the amount of work involved.
 - A false negative occurs when an intrusion detection system fails to report an ongoing attack.
 - False negatives are worse because the purpose of an intrusion detection system is to report attacks.
 - The goal of an intrusion detection system is to minimize both types of attacks.

Chapter 22 INTRUSION DETECTION

- **22.3 Models**

- Intrusion detection systems determine if actions constitute intrusions on the basis of one or more models of intrusion.
- A model classifies a sequence of states or actions, or a characterization of states or actions, as “good” (no intrusion) or “bad” (possible intrusions).
- Anomaly models use a statistical characterization, and actions or states that are statistically unusual are classified as “bad.”
- Misuse models compare actions or states with sequences known to indicate intrusions, or sequences believed to indicate intrusions, and classify those sequences as “bad.”
- Specification-based models classify states that violate the specifications as “bad.”
- The models may be adaptive or static.
 - Adaptive models alter their behavior on the basis of system states and actions.
 - Static models are initialized from collected data and do not change as the system runs.

Chapter 22 INTRUSION DETECTION

- **Anomaly Models**
 - Anomaly detection analyzes a set of characteristics of the system, and compares their behavior with a set of expected values. It reports when the computed statistics do not match the expected measurements.
 - Denning identifies three different statistical models:
 - The first model uses a threshold metric.
 - A minimum of m and a maximum of n events are expected to occur (for some event and some values m and n).
 - If, over a specific period of time, fewer than m or more than n events occur, the behavior is deemed anomalous.
 - **Example: Microsoft NT 4.0**
 - Allows the system to lock a user out after some number n of failed login attempts.
 - This is an intrusion detection system using the threshold metric with the lower limit 0 and the upper limit n .
 - The attempted logins are deemed anomalous after n failed attempts to login in.
 - Determining the threshold complicates use of this model.
 - The threshold must take into account differing levels of sophistication and characteristics of the users.
 - For example, if n were set to 3 in the example above for a system in France, and the primary users of that system were in the US, the difference in the keyboards would result in a large number of false alarms.
 - If the system were located in the US, setting n to 3 would be more reasonable.
 - One approach would be to combine this approach with the other two models to adapt the thresholds to observed or predicted behavior.

Chapter 22 INTRUSION DETECTION

- The second model uses statistical moments.
- The analyzer knows the mean and standard deviation (first two moments) and possibly other measures of correlation (higher moments).
- If measured values fall outside the expected interval for that moment, the behavior that the values represent is deemed anomalous.
- Because the profile (description of the system) may evolve over time, anomaly-based intrusion detection systems take these changes into account by aging (or weighing) data or altering the statistical rule base on which they make decisions.
- **EXAMPLE: Intrusion Detection Expert System (IDES)**
 - Developed at SRI International based on Denning's original model.
 - Represents subjects, which can include a user, a login session, applications, routers, and other entities as an ordered sequence of statistics $\langle q_{0,j}, \dots, q_{n,j} \rangle$, where $q_{i,j}$ is the i th statistic on day j . The metrics are counts or time intervals.
 - The profile for each subject is updated every day on the basis of observed behavior.
 - IDES weights its statistics to favor recent behavior over past behavior.
 - Let $A_{k,j}$ be the summation of counts making up the metric for the k th statistic on j th day.
 - The statistics $q_{k,t+1} = A_{k,t+1} - A_{k,t} + 2^{-rt}q_{k,t}$ where t is the number of log entries or the total time elapsed since time 0, and r is a half-life determined through experience.
 - This is an exponential decay of previous values and is quite sensitive to changes in behavior over a short period of time.
- **The statistical moments model provides more flexibility than the threshold model.**
 - Administrators can tune it to discriminate better than the threshold model.
 - However, with flexibility comes complexity.
 - An explicit assumption is that the behavior of processes and users can be statistically modeled.
 - An additional problem is the difficulty of computing these moments in real time.

Chapter 22 INTRUSION DETECTION

- The third model is a Markov model.
- Examine a system at some particular state.
- Events preceding that time have put the system into a particular state.
- When the next event occurs, the system transitions into a new state.
- Over time, a set of probabilities of transition can be developed.
- When an event occurs that causes a transition that has a low probability, the event is deemed anomalous.
- This model suggests that a notion of state (or past history) can be used to detect anomalies.
- The anomalies are no longer based on statistics of the occurrence of individual states, but on the sequences of events.
- Teng, Chen, and Lu used this approach in Digital Equipment Corporation's TIM research system.
 - Their scheme used an artificial intelligence technique called time-based inductive learning.
 - The system is given a type of event to be predicted.
 - It develops a set of temporally related conditions that predict the time that the event will occur with respect to the set.

Chapter 22 INTRUSION DETECTION

- **EXAMPLE:** Consider the sequence of events *abcdedeabcabc*.
- The following rules are examples that that TIM might derive.
 $R_1: ab \rightarrow c (1.0)$ $R_2: c \rightarrow d (0.5)$ $R_3: c \rightarrow e (0.5)$
 $R_4: d \rightarrow e (1.0)$ $R_5: e \rightarrow a (0.5)$ $R_6: e \rightarrow d (0.5)$
- The left side of each rule is the antecedent, and the right side is the event being predicted.
- The number in parentheses is the probability that the antecedent event(s) is(are) followed by the event on the right side of the rule.
- Rules 1 and 4 are good indicators of expected behavior. The other rules are not particularly good.
- They will either be dropped (should the probability decrease over time) or become better (should the probability increase over time).
- Anomalies are detected when a sequence of events matches the left side of a rule but the succeeding event differs from the expected right side.
- Using the rules above, if the sequence *abd* occurs, an alert will be triggered because *c* should always come after *ab*.
- The sequence *acf* will not cause an alert because multiple events may follow *c*.
- This sequence could cause a new rule to be added, namely $R_7: c \rightarrow f (0.33)$.
- The probabilities for rules R_2 and R_3 would change to 0.33.
- The effectiveness of Markov-based models depends on the adequacy of the data used to establish the model.
- This data is called training data, and is obtained experimentally, usually from populations that are believed to be normal (not anomalous).

Chapter 22 INTRUSION DETECTION

- **Misuse Modeling**

- **DEFINITION:** Misuse detection determines whether a sequence of instructions being executed is known to violate the site security policy being executed. If so, it reports a potential intrusion.
- Modeling of misuse requires a knowledge of system vulnerabilities or potential vulnerabilities that attackers attempt to exploit.
- The intrusion detection system incorporates this knowledge into a rule set.
- When data is passed to the intrusion detection system, it applies the rule set to the data to determine if any sequences of data match any of the rules. If so, it reports that a possible intrusion is underway.
- Misuse-based intrusion detection systems often use expert systems to analyze the data and apply the rule set.
- These systems cannot detect attacks that are unknown to the developers of the rule set.
- Previously unknown attacks, or even variations of known attacks, can be difficult to detect.
- Later intrusion detection systems used adaptive methods involving neural networks and Petri nets to improve their detection abilities.
- One important feature for intrusion detection systems is an interface into which new users and/or maintainers can add new rules or data.

Chapter 22 INTRUSION DETECTION

- **EXAMPLE:** The intrusion detection tool Network Flight Recorder (NFR) has three components.
 - A packet sucker reads packets off the network.
 - The packets are passed to a decision engine, which uses filters in a language called N-node to extract information.
 - The backend writes the data generated by the filters to disk; the packet itself is discarded.
 - A query backend allows administrators to extract both raw and postprocessed data from the disk file.
 - Although some filters are supplied, users can write their own filters using the N-code language.
 - This language is a stack-oriented language with an interpretive engine built into NFR.
 - It includes all usual high-level language features (such as loops and conditionals), as well as a set of data types for counters and IP addresses.
 - Packets are considered structures, and the fields are built into the language.
 - For example, to have the filter ignore all traffic that is not intended for a set of Web servers, we have the following code.

Chapter 22 INTRUSION DETECTION

```
# list of my web servers  
my_web_servers = [ 10.237.100.189 10.237.55.93 ] ;  
# we assume all HTTP traffic is on port 80  
filter watch tcp ( client, dport:80 )  
{  
    if (ip.dest != my_web_servers)  
        return;  
# now process the packet; we just write out packet info  
    record system.time, ip.src, ip.dest to www._list;  
}  
www_list = recorder("log")
```

Chapter 22 INTRUSION DETECTION

- **Specification Modeling**
 - **DEFINITION:** Specification-based detection determines whether or not a sequence of instructions violates a specification of how a program, or system, should execute. If so, it reports a possible intrusion.
 - For security purposes, only those programs that in some way change the protection state of the system need to be specified and checked.
 - **EXAMPLE:**
 - Ko, Ruschitzka, and Levitt developed a specification-based intrusion detection system for the UNIX environment.
 - The UNIX program *rdist* (for remote distribution) updates programs on remote systems.
 - It first creates a temporary file */tmp/rdistxxxxx*.
 - It then copies the contents of the new file into the temporary file, changes the protection mask as required, and copies the temporary file over the file to be replaced.
 - The problem is that *rdist* modifies protection modes by acting on the file name.
 - If an attacker can replace the file by a symbolic link, he can force *rdist* to modify the protection modes of any file in the system.
 - For example, he can turn on the setuid bit for the program */bin/sh*, which would give him superuser privileges instantly.
 - Specification-based intrusion detection is in its infancy.

Chapter 22 INTRUSION DETECTION

22.4 Architecture

- An intrusion detection system is also an automated auditing mechanism.
- Like auditing systems, it consists of three parts:
 - The agent corresponds to the logger. It acquires information from a target (such as a computer system).
 - The director corresponds to the analyzer. It analyzes the data from the agents as required (usually to determine if an attack is in progress or has occurred).
 - The director then passes this information to the notifier, which determines whether, and how, to notify the requisite entity.
 - The notifier may communicate with the agents to adjust the logging, if appropriate.
- Agent
 - An agent obtains information from a data source or a set of data sources.
 - The source may be a log file, another process, or a network.
 - Once acquired, the information may be sent directly to the director.
 - Usually, it is preprocessed into a specific format to save the director from having to do this.
 - The agent may also discard information that it deems irrelevant.
 - **EXAMPLE:** If the agent is to transmit the time and location of a failed attempt, it will scan the appropriate log file, discard any records of successful logins, and send the remainder information to the director.
 - The director may determine that it needs more information from a particular information source. In that case, the director can instruct the agent to collect additional data, or to process the data it collects differently.

Chapter 22 INTRUSION DETECTION

– Director

- The director itself reduces the incoming log entries to eliminate unnecessary and redundant records.
- It then uses an analysis engine to determine if an attack (or the precursor to an attack) is underway.
- The analysis engine may use any of, or a mixture of, several techniques to perform its analysis.
- Because the functioning of the director is critical to the effectiveness of the intrusion detection system, it is usually run on a separate system.
- This allows the system to be dedicated to the director's activity.
- It has the side effect of keeping the specific rules and profiles unavailable to ordinary users.
- So, the attackers lack the knowledge needed to evade the intrusion detection system by conforming to known profiles or using only techniques that the rules do not include.
- The director must correlate information from multiple logs.
- **EXAMPLE:**
 - A particular user logs in during the day to perform system maintenance functions.
 - Occasionally, she logs in during the late evening to write reports.
 - One day, she apparently logs in during the late evening and begins altering the kernel (a system maintenance procedure).
 - Agents provide information from both the log of login times and the log of commands executed.
 - Neither set of data by itself will give an indication of a security problem.
 - However, if the director correlates the two sets of data, the anomaly will be apparent.

Chapter 22 INTRUSION DETECTION

- Many types of directors alter the set of rules that they use to make decisions.
- These adaptive directors alter the profiles, add or delete rules, and otherwise adapt to changes in the system being monitored.
- Typical adaptive directors use aspects of machine learning or planning to determine how to alter their behavior.
- **EXAMPLE:**
 - Debar, Becker, and Siboni proposed the use of a neural network to analyze logs.
 - Their goal was to reduce the complexity of analyzing the data from the agent.
 - They constructed a neural network that adapted to the user's behavior over time, enabling them to discard data and simplify the analysis.
 - This also enabled them to use several learning techniques to improve the classification of events as anomalous, thereby reducing the number of false alarms.
- Directors rarely use only one analysis techniques because different techniques highlight different aspects of intrusions.
- The results of each are combined, analyzed and reduced, and then used.

Chapter 22 INTRUSION DETECTION

– Notifier

- The notifier accepts information from the director and takes the appropriate action.
- In some cases, this is simply a notification to the system security officer that an attack is believed to be underway.
- In other cases, the notifier may take some action to respond to the attack.
- Many intrusion detection systems use graphical interfaces.
- A well-designed graphics display allows the intrusion detection system to convey information in an easy-to-grasp image or set of images.
 - It must allow users to determine what attacks are underway.
 - This requires the Graphical User Interface (GUI) be designed with a lack of clutter and unnecessary information.
- **EXAMPLE:**
 - The Graphical Intrusion Detection System (GrIDS) uses a graph-oriented user interface to show the progress of attacks across multiple systems.
 - The hosts are represented as nodes, and as an attack from one system to another is identified, the nodes are connected with edges labeled to show the progress of the attack.

Chapter 22 INTRUSION DETECTION

22.5 Organization of Intrusion Detection Systems

- An intrusion detection system can be organized in several ways.
 - The first system examines network traffic only.
 - The second explores how to combine network and host sources.
 - The third system distributes the director among multiple systems to enhance security and reliability.
- **Monitoring Network Traffic for Intrusions: NSM**
 - The Network Security Monitor (NSM) develops a profile of expected usage of a network and compares current usage with that profile.
 - It also allows the definition of a set of signatures to look for specific sequences of network traffic that indicate attacks.
 - It runs on a local area network and assumes a broadcast medium
 - The monitor measures network utilization and other characteristics and can be instructed to look at activity based on a user, a group of users, or a service.
 - It reports anomalous behavior.
 - The NSM monitors the source, destination, and service of network traffic.
 - It assigns a unique connection ID to each connection.
 - The source, destination, and service are used as axes for a matrix.
 - Each element of the matrix contains the number of packets sent over that connection for a specified period of time, and the sum of the data of those packets.
 - NSM also generates expected connection data from the network.
 - The data in the array is masked by the expected connection data, and any data not within the expected range is reported as an anomaly.

Chapter 22 INTRUSION DETECTION

- The developers of the NSM quickly found that too much data was being generated during the network analysis.
- To reduce the overhead, they constructed a hierarchy of elements of the matrix and generated expected connection data for those elements.
- If any group in the hierarchy showed anomalous data, the system security officer could ask the NSM to break it down into the underlying elements.
- The groups were constructed by folding the axes of the matrix.
- For example, one group would be the set of traffic between two hosts for each service.
- It would have the elements $\{(A,B,SMTP), (A,B,FTP), \dots\}$, where A and B are host names.
- The next group would collapse the service names and simply group all traffic into source-destination pairs.
- At the highest level, traffic would be grouped into its source.
- The NSM would analyze the data at the source level.
- If it flagged an anomaly, the system security officer could have the NSM examine each component of the underlying group and determine which specific source-destination pair had the anomaly.
- From there, it could be broken into the specific service or services involved.

Chapter 22 INTRUSION DETECTION

- The NSM is important for two reasons.
- First, it served as the basis for a large number of intrusion detection systems.
- Eleven years after its creation, it was still in use at many sites although with an augmented set of signatures.
- Second, it proved that performing intrusion detection on networks was practical.
- As network traffic becomes enciphered, the ability to analyze the contents of the packets diminishes, but NSM did not look at the contents of the traffic.
- The NSM would analyze the data at the source level.
- It performed traffic analysis.
- Hence, its methodology will continue to be effective even after widespread deployment of network encryption.

Chapter 22 INTRUSION DETECTION

- **Combining Host and Network Monitoring: DIDS**
 - The Distributed Intrusion Detection System (DIDS) combined the abilities of the NSM with intrusion detection monitoring of individual hosts.
 - It sprang from the observation that neither network-based monitoring nor host-based monitoring was sufficient.
 - An intruder attempting to log into a system through an account without a password would not be detected as malicious logic by a network monitor.
 - Subsequent actions, however, might make a host-based monitor report that an intruder is present.
 - Similarly, if an attacker tries to telnet to a system a few times, using a different login name each time, the host-based intrusion detection mechanism would not report a problem, but the network-based monitor could detect failed login attempts.
 - DIDS used a centralized analysis engine (the DIDS director) and required that agents be placed on the system being monitored as well as in a place to monitor the network traffic.
 - The agents scanned logs for events of interest and reported them to the DIDS director.
 - The DIDS director invoked an expert system that performed the analysis of the data.
 - The expert system was a rule-based system that could make inferences about individual hosts and about the entire system (hosts and networks).
 - It would then pass results to the user interface, which displayed them in a simple, easy-to-grasp manner for the system security officer.

Chapter 22 INTRUSION DETECTION

- One problem is the changing of the identity as an intruder moves from host to host.
- An intruder might gain access to the first system as user *alice*, and then to the second system as user *bob*.
- The host-based mechanisms cannot know that *alice* and *bob* are the same user, so they cannot correlate the actions of those two user names.
- However, the DIDS director would note that *alice* connected to the remote host and that *bob* logged in through that connection.
- The expert system would infer that they were the same user.
- To enable this type of correlation, each user was identified by a network identification number (NID).
- In the example above, because *alice* and *bob* are the same user, both would share a common NID.
- The host agents and network agent provide insight into the problems distributed intrusion detection faces.
- The host logs are analyzed to extract entries of interest.
- In some cases, simple reduction is performed to determine if the records should be forwarded; for example, the host agents monitor the system for attacks using signatures.
- Summaries of these results go to the director.
- Other events are forwarded directly.

Chapter 22 INTRUSION DETECTION

– Autonomous Agents: AAFID

- **DEFINITION:** An autonomous agent is a process that can act independently of the system of which it is a part.
- In 1995, Crosbie and Spafford examined intrusion detection systems in light of fault tolerance.
- They noted that an intrusion detection system that obtains information by monitoring systems and networks is a single point of failure.
- If the director fails, the IDS will not function.
- They suggested developing autonomous agents each of which performed one particular monitoring function.
- Each agent would have its internal model, and when the agent detected a deviation from expected behavior, a match with a particular rule, or a violation of a specification, it would notify other agents.
- The agents would jointly determine whether the set of notifications were sufficient to constitute a reportable intrusion.
- The beauty of this organization lies in the cooperation of the agents.
- No longer is there a single point of failure.
- If one agent is compromised, the others can continue to function.
- If an attacker compromises one agent, she has learned nothing about the other agents in the system or monitoring the network.
- The director itself is distributed among the agents, so it cannot be attacked in the same way that an intrusion detection system with a director on a single host can be.
- This approach appears to be scalable to larger networks because of the distributed nature of the director.
- The drawback of autonomous agents lie in the overhead of the communications needed.

Chapter 22 INTRUSION DETECTION

- **EXAMPLE:** The Autonomous Agents for Intrusion Detection (AAFID) system
- Each host has a set of agents and a transceiver, which controls the execution of the agents, collates the information, and forwards it to a monitor (director).
- If the transceiver's host does not have a monitor, the transceiver simply transmits the information to a monitor on another host.
- In theory, each agent obtains its own data.
- This approach causes unnecessary duplication of work and leads to agents that are highly system dependent.
- Transceivers collect data from the local agents, process it, and forward it to other agents or to monitors as appropriate.
- Monitors are the distributed components of the AAFID director. They accept information from transceivers and can communicate with the transceivers and other monitors.
- The user interface plays one of the roles of a notifier. This interface interacts with the monitors. It may be graphical (for human interaction) or textual (for command scripts).
- The implemented AAFID prototype runs on Linux and Solaris systems.
- It was implemented in the Perl language for easy of programming, portability, and modification.
- Because the prototype was a proof of concept and not a production system, the loss of performance was considered acceptable.
- The prototype validated the architecture and demonstrated that autonomous agents were a practical method for intrusion detection systems.

Chapter 22 INTRUSION DETECTION

22.6 Intrusion Response

- Once the intrusion is detected, how can the system be protected?
- The field of intrusion response deals with this problem.
- Its goal is to handle the (attempted) attack in such a way that damage is minimized.
- Some intrusion detection mechanisms may be augmented to thwart intrusions.
- Otherwise, the security officers must respond to the attack and attempt to repair any damage.
- **Incident Prevention**
 - Ideally, intrusion attempts will be detected and stopped before they succeed.
 - This typically involves closely monitoring the system (usually with an intrusion detection mechanism) and taking action to defeat the attack.
 - In the context of response, prevention requires that the attack be identified before it completes.
 - The defenders then take measures to prevent the attack from completing. This may be done manually or automatically.

Chapter 22 INTRUSION DETECTION

- **Intrusion Handling**
 - When an intrusion occurs, the security policy of the site has been violated.
 - Handling the intrusion means restoring the system to comply with the site security policy and taking any actions against the attacker that the policy specifies.
 - Intrusion handling consists of six phases:
 - Preparation for an attack: This step occurs before any attacks are detected. It establishes procedures and mechanisms for detecting and responding to attacks.
 - Identification of an attack: This triggers the remaining phases.
 - Containment (or confinement) of the attack: This step limits the damage as much as possible.
 - Eradication of the attacks: This step stops the attack and blocks further similar attacks.
 - Recovery from the attack: This step restores the system to a secure state (with respect to the site security policy).
 - Follow-up to the attack: This step involves taking action against the attacker, identifying problems in the handling of the incident, and recording lessons learned (or lessons not learned that should be learned).