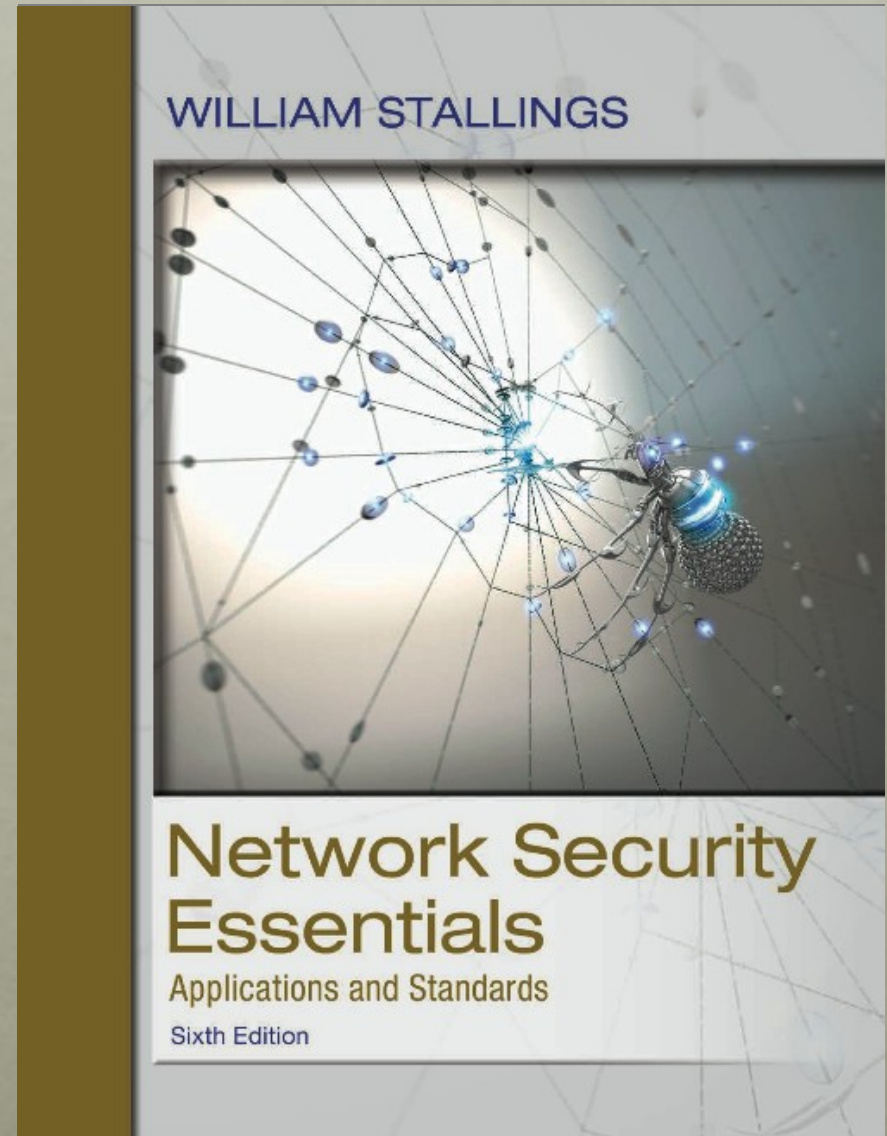


Network Security Essentials

Sixth Edition

by William Stallings



Chapter 11

Intruders

Intruders

Three classes of intruders:

Masquerader: An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account

Misfeisor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection

The masquerader is likely to be an outsider, the misfeisor generally is an insider, and the clandestine user can be either an outsider or an insider.

Examples of Intrusion

- Performing a remote root compromise of an e-mail server
- Defacing a Web server
- Guessing and cracking passwords
- Copying a database containing credit card numbers
- Viewing sensitive data, including payroll records and medical information, without authorization
- Running a packet sniffer on a workstation to capture usernames and passwords
- Using a permission error on an anonymous FTP server to distribute pirated software and music files
- Dialing into an unsecured modem and gaining internal network access
- Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password
- Using an unattended, logged-in workstation without permission

Hackers

- Historically, those who hack into computers did so for the thrill of it or for status. Hackers are now more often motivated by profit.
- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter hacker threats
 - In addition to using such systems, organizations can consider restricting remote logons to specific IP addresses and/or use virtual private network technology
- CERTs
 - Computer emergency response teams
 - These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers
 - Hackers also routinely read CERT reports
 - It is important for system administrators to quickly insert all software patches to discovered vulnerabilities

Criminal hackers

- Organized groups of hackers
- Usually have specific targets, or at least classes of targets in mind
- Once a site is penetrated, the attacker acts quickly, scooping up as much valuable information as possible and exiting
- IDSs and IPSs can be used for these types of attackers, but may be less effective because of the quick in-and-out nature of the attack

Insider Attacks

- Among the most difficult to detect and prevent
- Can be motivated by revenge or simply a feeling of entitlement



Countermeasures:

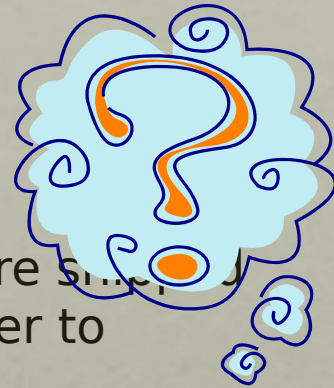
- Enforce least privilege, only allowing access to the resources employees need to do their job.
- Set logs to see what users access and what commands they are entering.
- Protect sensitive resources with strong authentication.
- Upon termination, delete employee's computer and network access.
- Upon termination, make a mirror image of employee's hard drive before reissuing it. That evidence might be needed if your company information turns up at a competitor.

Intrusion Techniques

Objective: Gain access to a system or to increase the range of privileges accessible on a system

- System or software vulnerabilities that allow a user to execute code that opens a backdoor into the system
- Social engineering
- Brute force
- Physical Access

Password Guessing



1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords (those of one to three characters).
3. Try words in the system's online dictionary or a list of likely passwords.
Examples of the latter are readily available on hacker bulletin boards.
4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.
5. Try users' phone numbers, Social Security numbers, and room numbers.

Intrusion Detection

- A system's second line of defense
- Is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified
- Considerations:
 - If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised
 - An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions
 - Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility



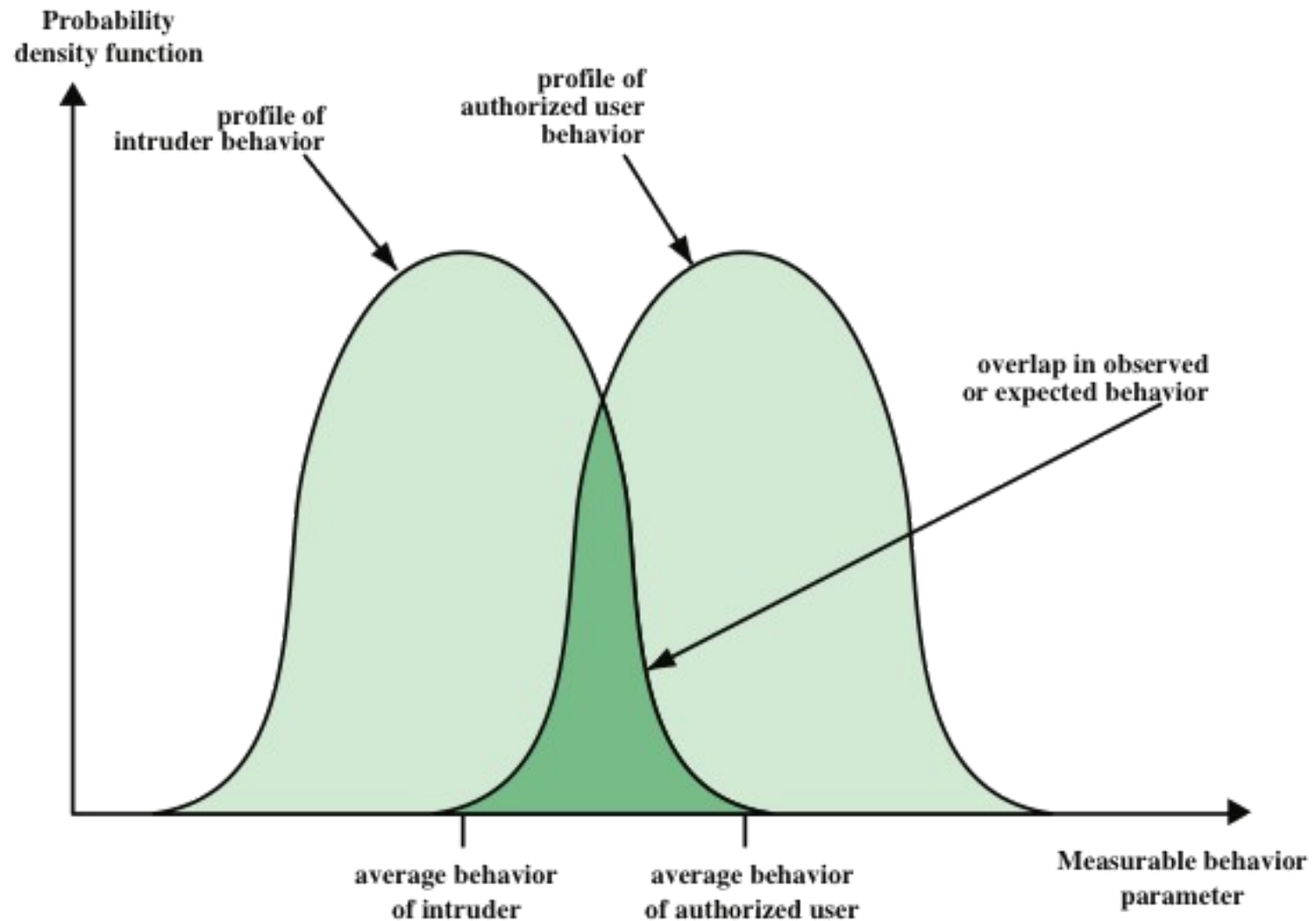


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

Approaches to Intrusion Detection

- Statistical anomaly detection
 - Involves the collection of data relating to the behavior of legitimate users over a period of time
 - Then statistical tests are applied to observed behavior to determine whether that behavior is not legitimate user behavior
 - Threshold detection
 - This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events
 - Profile based
 - A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts
- Rule-based 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
 - Often referred to as *signature detection*

Audit Records

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system.

Basically, two plans are used:

- Native audit records: Virtually all multiuser operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.
- Detection-specific audit records: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine.

Statistical Anomaly Detection

- Threshold detection
 - Involves counting the number of occurrences of a specific event type over an interval of time
 - If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed
 - By itself is a crude and ineffective detector of even moderately sophisticated attacks
- Profile-based
 - Focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations
 - A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert

Table 11.1

Measures That May Be Used For Intrusion Detection

(This table can be found
on page 371 in the
textbook.)

| Measure | Model | Type of Intrusion Detected |
|---|-----------------------------|---|
| Login and Session Activity | | |
| Login frequency by day and time | Mean and standard deviation | Intruders may be likely to log in during off-hours. |
| Frequency of login at different locations | Mean and standard deviation | Intruders may log in from a location that a particular user rarely or never uses. |
| Time since last login | Operational | Break-in on a "dead" account. |
| Elapsed time per session | Mean and standard deviation | Significant deviations might indicate masquerader. |
| Quantity of output to location | Mean and standard deviation | Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data. |
| Session resource utilization | Mean and standard deviation | Unusual processor or I/O levels could signal an intruder. |
| Password failures at login | Operational | Attempted break-in by password guessing. |
| Failures to login from specified terminals | Operational | Attempted break-in. |
| Command or Program Execution Activity | | |
| Execution frequency | Mean and standard deviation | 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. |
| Program resource utilization | Mean and standard deviation | An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization. |
| Execution denials | Operational model | May detect penetration attempt by individual user who seeks higher privileges. |
| File access activity | | |
| Read, write, create, delete frequency | Mean and standard deviation | Abnormalities for read and write access for individual users may signify masquerading or browsing. |
| Records read, written | Mean and standard deviation | Abnormality could signify an attempt to obtain sensitive data by inference and aggregation. |
| Failure count for read, write, create, delete | Operational | May detect users who persistently attempt to access |

Rule-Based Intrusion Detection

- Techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious
- **Rule-based anomaly detection**
 - Is similar in terms of its approach and strengths to statistical anomaly detection
 - Historical audit records are analyzed to identify usage patterns and to automatically generate rules that describe those patterns
 - Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior
 - In order for this approach to be effective, a rather large database of rules will be needed

Base-Rate Fallacy

- To be of practical use, an intrusion detection system should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level
 - If only a modest percentage of actual intrusions are detected, the system provides a false sense of security
 - If the system frequently triggers an alert when there is no intrusion, then either system managers will begin to ignore the alarms or much time will be wasted analyzing the false alarms
- Because of the nature of the probabilities involved, it is very difficult to meet the standard of high rate of detections with a low rate of false alarms
 - If the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating
- See Appendix J for a brief background on the mathematics of this problem

Distributed Intrusion Detection

Traditional systems focused on single-system stand alone facilities

The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork

A more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network

Distributed Intrusion Detection

Design considerations:

- A distributed intrusion detection system may need to deal with different audit record formats. In a heterogeneous environment, different systems will employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records.
- One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. Integrity is required to prevent an intruder from masking his or her activities by altering the transmitted audit information. Confidentiality is required because the transmitted audit information could be valuable.
- Either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single central point of collection and analysis of all audit data. This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure. With a decentralized architecture, there are more than one analysis centers, but these must coordinate their activities and exchange information.

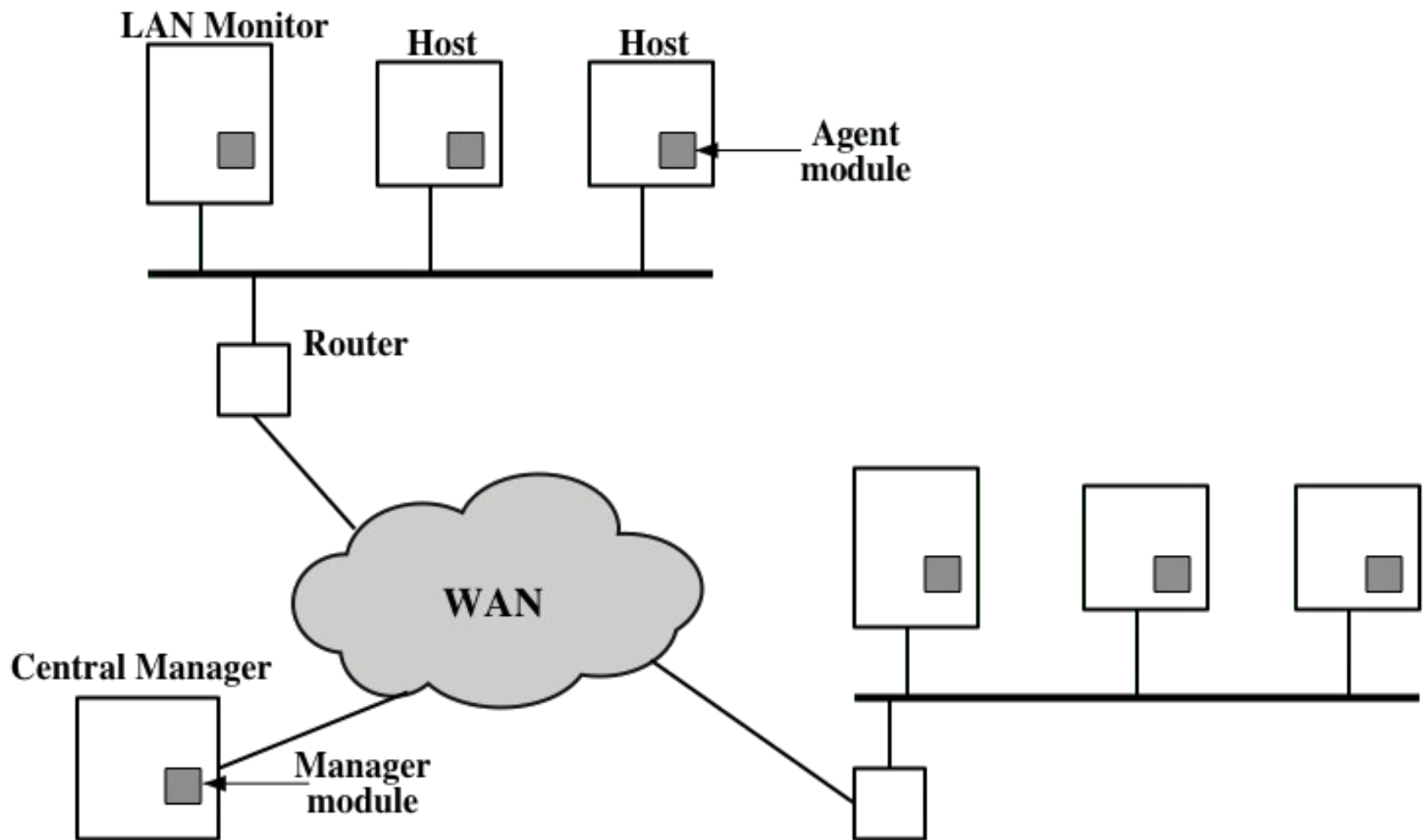


Figure 11.2 Architecture for Distributed Intrusion Detection

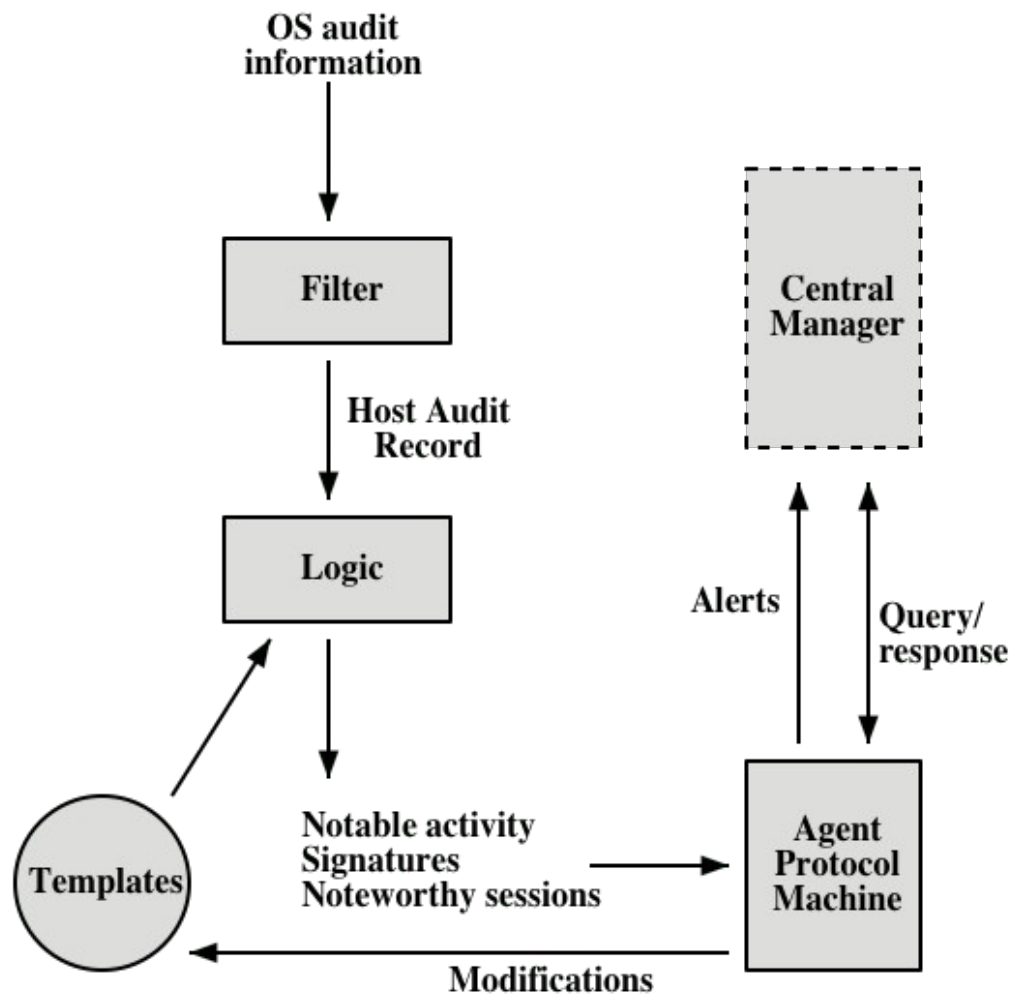


Figure 11.3 Agent Architecture

Honeypots

Honeypots are designed to

- Divert an attacker from accessing critical systems
- Collect information about the attacker's activity
- Encourage the attacker to stay on the system long enough for administrators to respond

The honeypot is a resource that has no production value. There is no legitimate reason for anyone outside the network to interact with a honeypot.

Decoy systems that are designed to lure a potential attacker away from critical systems

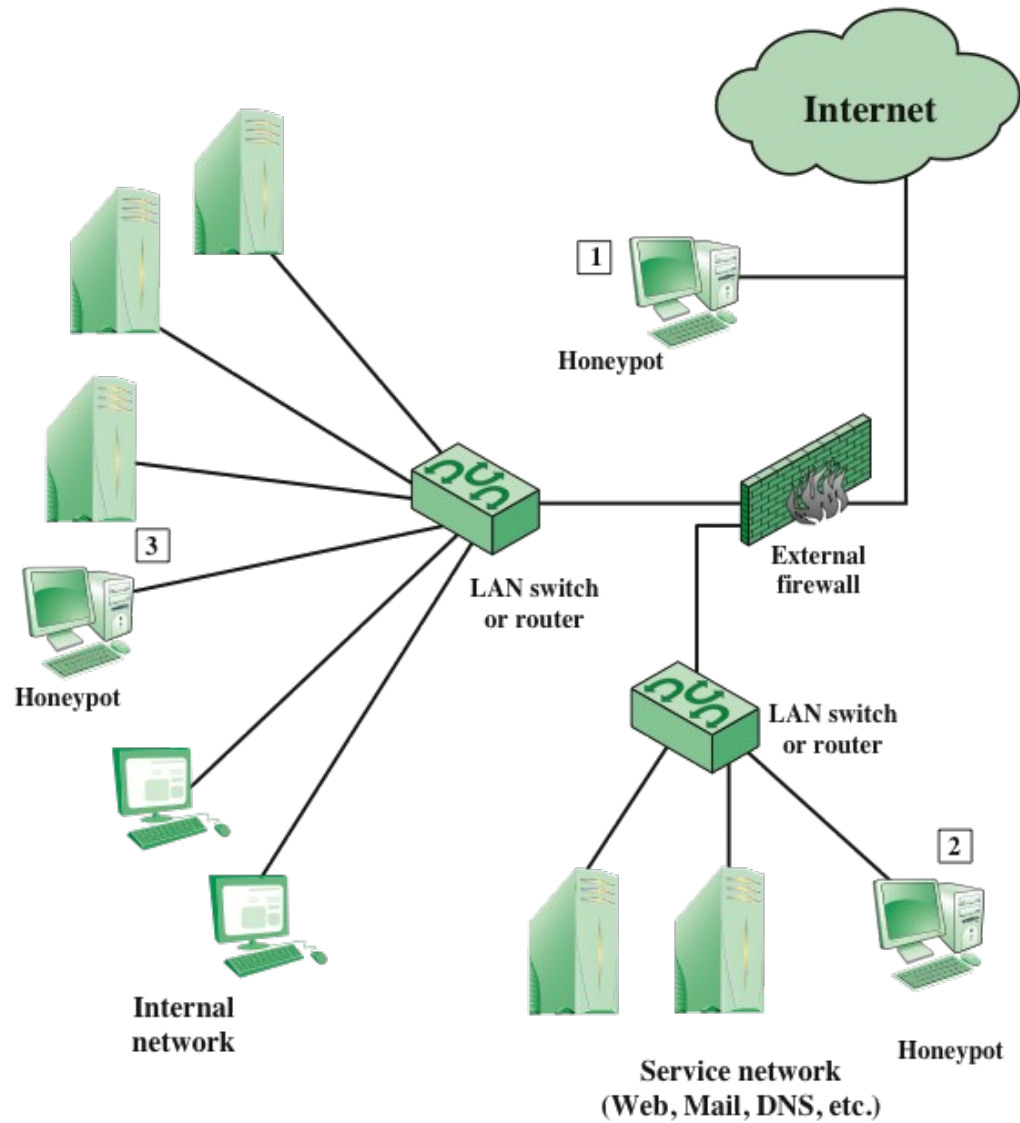


Figure 11.4 Example of Honeypot Deployment

Intrusion detection exchange format

- To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability
- IETF Intrusion Detection Working Group
 - Purpose of the group is to define data formats and exchange procedures for sharing information of interest to intrusion detection with response systems and to management systems that may need to interact with them
 - Have issued the following RFCs:
 - Intrusion Detection Message Exchange Requirements (RFC 4766)
 - The Intrusion Detection Message Exchange Format (RFC 4765)
 - The Intrusion Detection Exchange Protocol (RFC 4767)

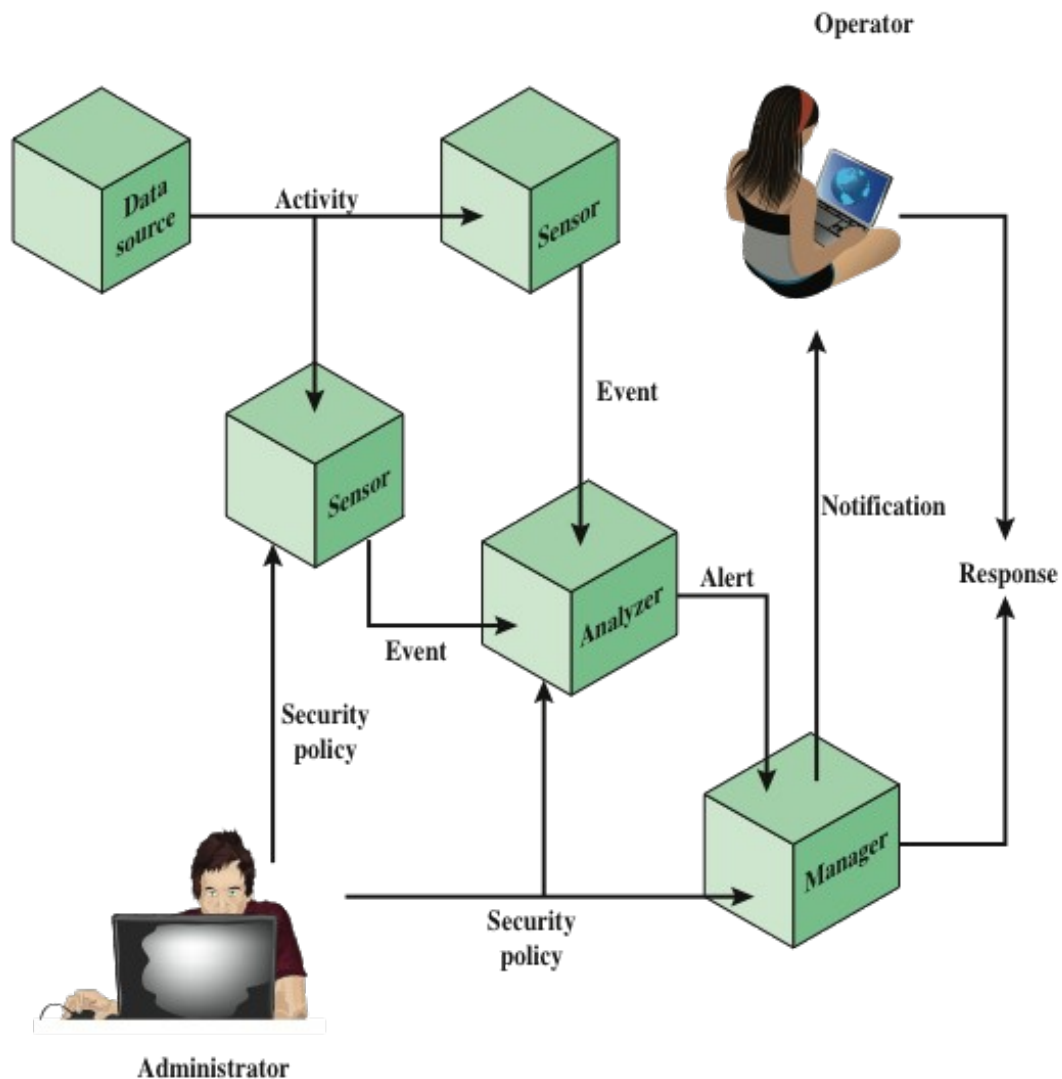


Figure 11.5 Model For Intrusion Detection Message Exchange

Password Management

- Front line of defense against intruders
- Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password
 - Password serves to authenticate the ID of the individual logging on to the system
 - The ID provides security by:
 - Determining whether the user is authorized to gain access to a system
 - Determining the privileges accorded to the user
 - Used in discretionary access control

Attack strategies and countermeasures

Typically, a system that uses password-based authentication maintains a password file indexed by user ID. One technique that is typically used is to store not the user's password but a one-way hash function of the password, as described subsequently.

We can identify the following attack strategies and countermeasures:

- **Offline dictionary attack:** Typically, strong access controls are used to protect the system's password file. However, determined hackers can frequently bypass such controls and gain access to the file. The attacker obtains the system password file and compares the password hashes against hashes of commonly used passwords. If a match is found, the attacker can gain access by that ID/password combination. Countermeasures include controls to prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should the password file be compromised.
- **Specific account attack:** The attacker targets a specific account and submits password guesses until the correct password is discovered. The standard countermeasure is an account lockout mechanism, which locks out access to the account after a number of failed login attempts.
- **Workstation hijacking:** The attacker waits until a logged-in workstation is unattended. The standard countermeasure is automatically logging the workstation out after a period of inactivity. Intrusion detection schemes can be used to detect changes in user behavior.
- **Exploiting user mistakes:** If the system assigns a password, then the user is more likely to write it down. This creates the potential for an adversary to read the written password. A user may intentionally share a password, to enable a colleague to share files. Also, attackers can obtain passwords by using social engineering tactics that trick the user or an account manager into revealing a password. Many computer systems are shipped with preconfigured passwords for system administrators. Unless these preconfigured passwords are changed, they are easily guessed. Countermeasures include user training, intrusion detection, and simpler passwords combined with another authentication mechanism.

Attack strategies and countermeasures

- Exploiting multiple password use: Attacks can also become much more effective or damaging if different network devices share the same or a similar password for a given user. Countermeasures include a policy that forbids the same or similar password on particular network devices.
- Popular password attack: A variation of the preceding attack is to use a popular password and try it against a wide range of user IDs. A user's tendency is to choose a password that is easily remembered; this unfortunately makes the password easy to guess. Countermeasures include policies to inhibit the selection by users of common passwords and scanning the IP addresses of authentication requests and client cookies for submission patterns.
- Electronic monitoring: If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping. Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary.
- Password guessing against single user: The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password. Countermeasures include training in and enforcement of password policies that make passwords difficult to guess. Such policies address the secrecy, minimum length of the password, character set, prohibition against using well-known user identifiers, and length of time before the password must be changed.

Unix implementations

- Crypt(3)
 - Was designed to discourage guessing attacks
 - This particular implementation is now considered inadequate
 - Despite its known weaknesses, this UNIX scheme is still often required for compatibility with existing account management software or in multivendor environments
- MD5 secure hash algorithm
 - The recommended hash function for many UNIX systems, including Linux, Solaris, and FreeBSD
 - Far slower than crypt(3)
- Bcrypt
 - Developed for OpenBSD
 - Probably the most secure version of the UNIX hash/salt scheme
 - Uses a hash function based on the Blowfish symmetric block cipher
 - Slow to execute
 - Includes a cost variable

Table 11.3

Passwords Cracked from a Sample Set of 13,797 Accounts

(This table can be found on
page 586 in the textbook.)

| Type of Password | Search Size | Number of Matches | Percentage of Passwords Matched | Cost/Benefit Ratio ^a |
|----------------------|--------------|----------------------|---------------------------------------|------------------------------------|
| User/account name | 130 | 368 | 2.7% | 2.830 |
| Character sequences | 866 | 22 | 0.2% | 0.025 |
| Numbers | 427 | 9 | 0.1% | 0.021 |
| Chinese | 392 | 56 | 0.4% | 0.143 |
| Place names | 628 | 82 | 0.6% | 0.131 |
| Common names | 2239 | 548 | 4.0% | 0.245 |
| Female names | 4280 | 161 | 1.2% | 0.038 |
| Male names | 2866 | 140 | 1.0% | 0.049 |
| Uncommon names | 4955 | 130 | 0.9% | 0.026 |
| Myths & legends | 1246 | 66 | 0.5% | 0.053 |
| Shakespearean | 473 | 11 | 0.1% | 0.023 |
| Sports terms | 238 | 32 | 0.2% | 0.134 |
| Science fiction | 691 | 59 | 0.4% | 0.085 |
| Movies and actors | 99 | 12 | 0.1% | 0.121 |
| Cartoons | 92 | 9 | 0.1% | 0.098 |
| Famous people | 290 | 55 | 0.4% | 0.190 |
| Phrases and patterns | 933 | 253 | 1.8% | 0.271 |
| Surnames | 33 | 9 | 0.1% | 0.273 |
| Biology | 58 | 1 | 0.0% | 0.017 |
| System dictionary | 19683 | 1027 | 7.4% | 0.052 |
| Machine names | 9018 | 132 | 1.0% | 0.015 |
| Mnemonics | 14 | 2 | 0.0% | 0.143 |
| King James bible | 7525 | 83 | 0.6% | 0.011 |
| Miscellaneous words | 3212 | 54 | 0.4% | 0.017 |
| Yiddish words | 56 | 0 | 0.0% | 0.000 |
| Asteroids | 2407 | 19 | 0.1% | 0.007 |
| TOTAL | 62727 | 3340 | 24.2% | 0.053 |

*** Computed as the number of matches divided by the search size. The more words that needed to be tested for a match, the lower the cost/benefit ratio.**

Password selection strategies

The goal is to eliminate guessable passwords while allowing the user to select a password that is memorable

Four basic techniques are in use:

- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

Summary

- Intruders
 - Behavior patterns
 - Intrusion techniques
- Intrusion detection
 - Audit records
 - Statistical anomaly detection
 - Rule-based intrusion detection
 - The base-rate fallacy
 - Distributed intrusion detection
 - Honeypots
 - Intrusion detection exchange format
- Password management
 - The vulnerability of passwords
 - The use of hashed passwords
 - User password choices
 - Password selection strategies

Rule-Based Intrusion Detection

- **Rule-based penetration identification**
 - Typically, the rules used in these systems are specific to the machine and operating system
 - The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet
 - These rules can be supplemented with rules generated by knowledgeable security personnel
- USTAT
 - A model independent of specific audit records
 - Deals in general actions rather than the detailed specific actions recorded by the UNIX auditing mechanism
 - Implemented on a SunOS system that provides audit records on 239 events

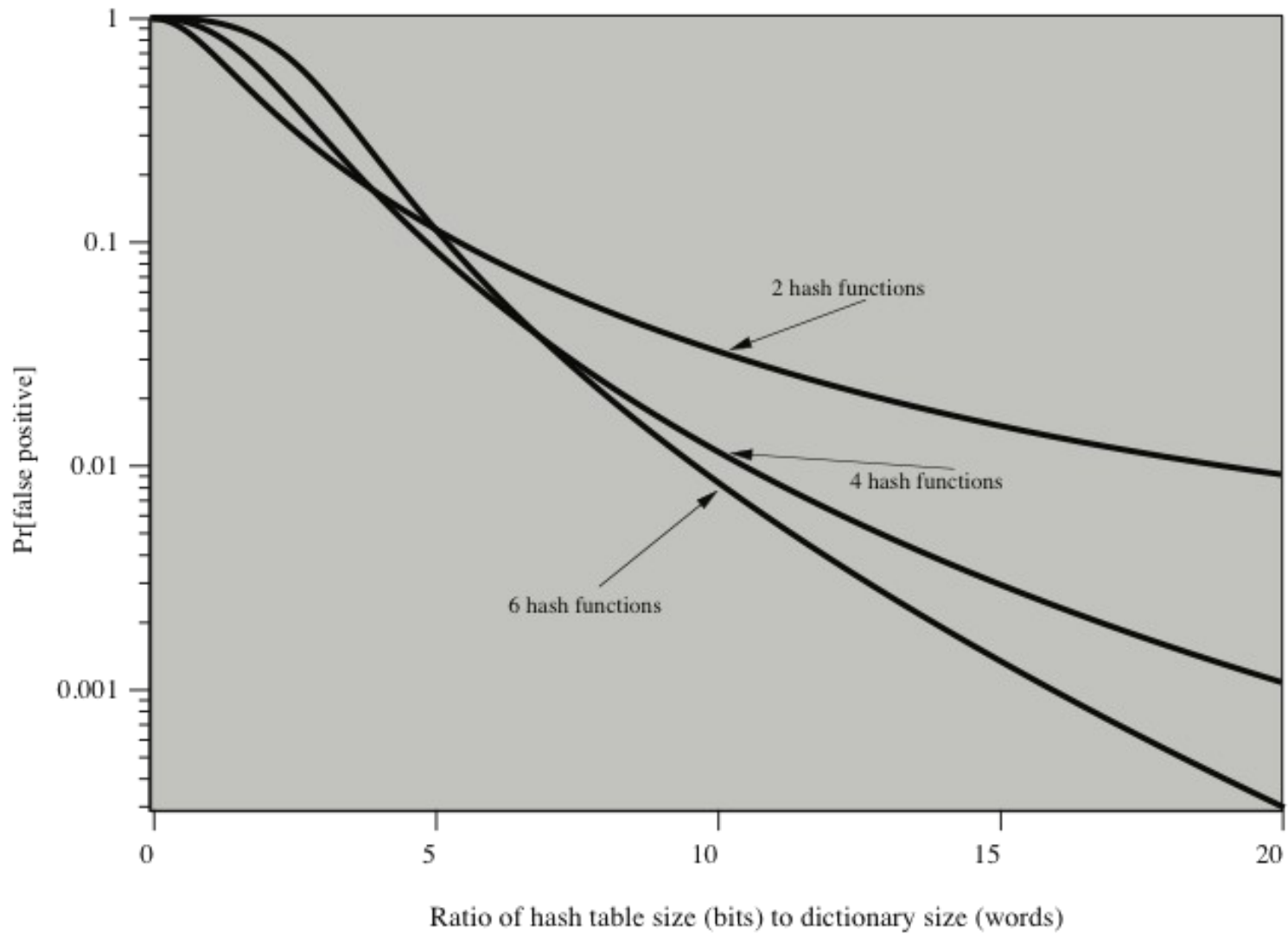


Figure 11.7 Performance of Bloom Filter

Table 11.2

USTAT Actions versus SunOS Event Types

| USTAT Action | SunOS Event Type |
|--------------|---|
| Read | open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt |
| Write | truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct |
| Create | mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod |
| Delete | rmdir, unlink |
| Execute | exec, execve |
| Exit | exit |
| Modify_Owner | chown, fchown |
| Modify_Perm | chmod, fchmod |
| Rename | rename |
| Hardlink | link |