System Security

Daniele Cono D'Elia – delia@diag.uniroma1.it
Riccardo Lazzeretti – lazzeretti@diag.uniroma1.it

Chapters 14,15 of William Stallings: Operating Systems: Internals and Design Principles

Operating Systems: Internals and Design Principles

The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.

— THE ART OF WAR,
Sun Tzu

Computer Security

- The NIST Computer Security Handbook defines ***computer security*** as:

  The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

Key Objectives of Computer Security

- **Confidentiality**
  - ***Data confidentiality*** assures that private or confidential information is not made available or disclosed to unauthorized individuals
  - ***Privacy*** assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed
- **Integrity**
  - ***Data integrity*** assures that information and programs are changed only in a specified and authorized manner
  - ***System integrity*** assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system
- **Availability**
  - assures that systems work promptly and service is not denied to authorized users

CIA Triad

- Security Objectives:
  - Confidentiality
    - a loss of confidentiality is the unauthorized disclosure of information
  - Integrity
    - a loss of integrity is the unauthorized modification or destruction of information
  - Availability
    - a loss of availability is the disruption of access to or use of information or an information system

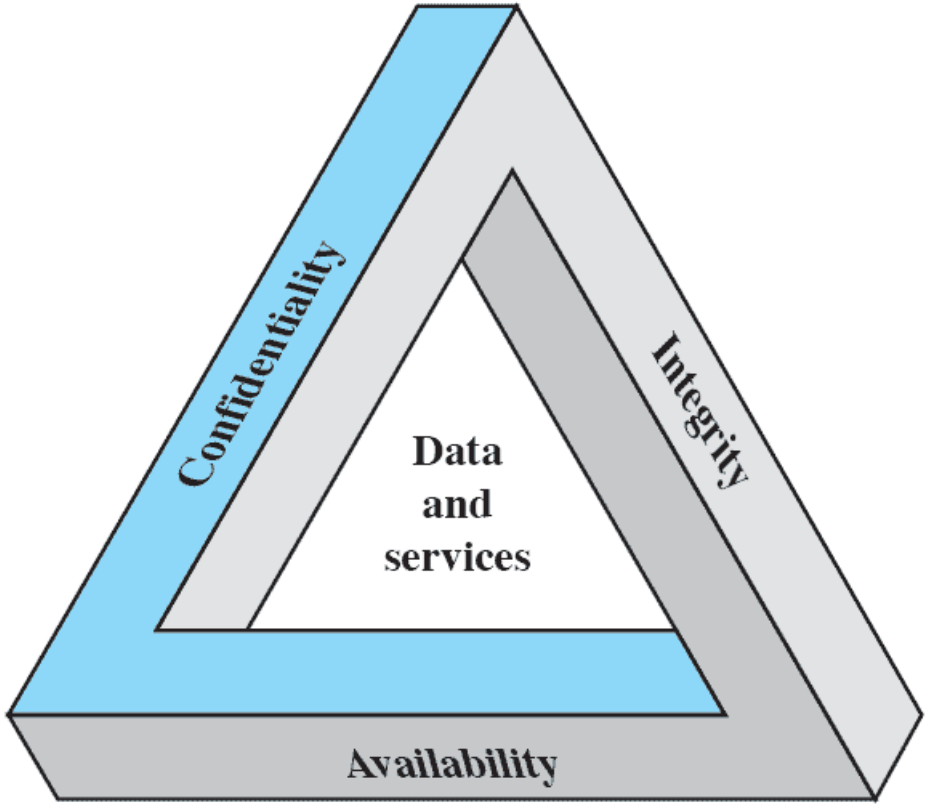


Figure 14.1 The Security Requirements Triad

Additional Concepts

- Two further concepts are often added to the core of computer security:

Authenticity

- The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator
- Verifying that users are who they say they are and that each input arriving at the system came from a trusted source

Accountability

- The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity
- We must be able to trace a security breach to a responsible party
- Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes

Table 14.1 ----- Unauthorized Disclosure

Threat Consequences, and the Types of Threat Actions
That Cause Each Consequence

(Based on RFC 2828)

Threat Consequence	Threat Action (Attack)
Unauthorized Disclosure A circumstance or event whereby an entity gains access to data for which the entity is not authorized	**Exposure**: Sensitive data are directly released to an unauthorized entity **Interception**: An unauthorized entity directly accesses sensitive data, travelling between authorized sources and destinations **Inference**: A threat action whereby an unauthorized entity indirectly accesses sensitive data (but not necessarly the data contained in the communication) by reasoning from characteristics or by products of communications **Intrusion**: An unauthorized entity gains access to sensitive data by circumventing a system's security protection

Table 14.1 ----- Deception

Threat Consequences, and the Types of Threat Actions That Cause Each Consequence

(Based on RFC 2828)

Threat Consequence	Threat Action (Attack)
Deception A circumstance or event that may result in an authorized entity receiving false data and believing it to be true	**Masquerade**: An unauthorized entity gains access to a system or performs a malicious act by posing as an authorized entity **Falsification**: False data deceive an authorized entity **Repudation**: An entity deceives another by falsely denying responsibility for an act

Table 14.1 ----- Disruption

Threat Consequences, and the Types of Threat Actions That Cause Each Consequence

(Based on RFC 2828)

Threat Consequence	Threat Action (Attack)
Disruption A circumstance or event that interrupt or prevents the correct operation of system services and functions	**Incapacitation**: Prevents or interrupts system operation by disabling a system component **Corruption**: Undesirably alters system operation by adversely modifying system functions or data **Obstruction**: A threat action that interrupts delivery of system services by hindering system operation

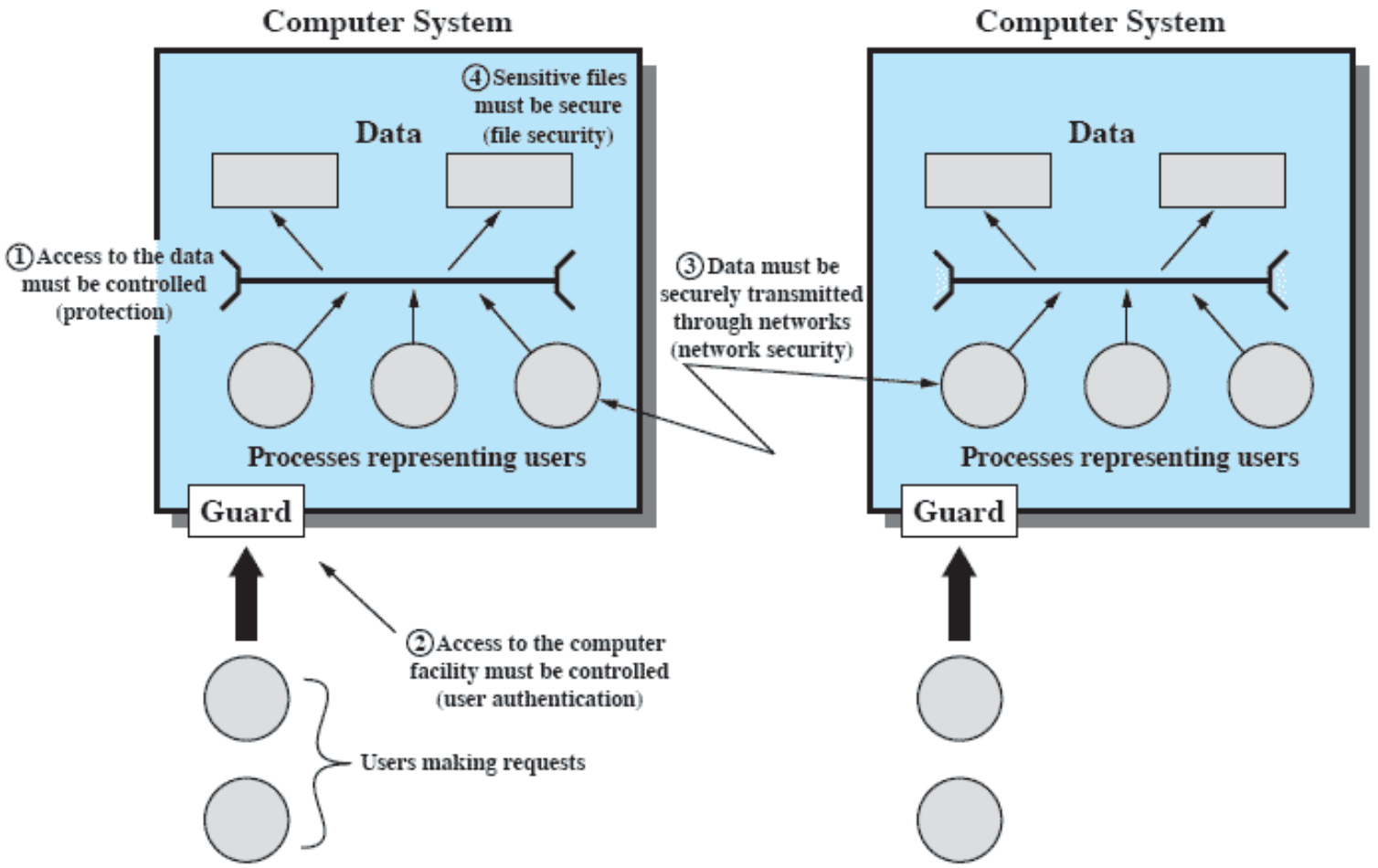
Table 14.1 ----- Usurpation

Threat Consequences, and the Types of Threat Actions That Cause Each Consequence

(Based on RFC 2828)

Threat Consequence	Threat Action (Attack)
Usurpation A circumstance or event that results in control of system services or functions by an unauthorized entity	**Misappropriation**: An entity assumes unauthorized logical or physical control of a system resource **Misuse**: Causes a system component to perform a function or service that is detrimental to system security

Scope of System Security



Examples of Threats

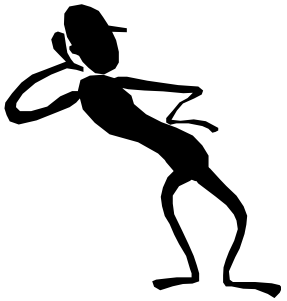
	Availability	Confidentiality	Integrity
Hardware	Equipment is stolen or disabled, thus denying service.		
Software	Programs are deleted, denying access to users.	An unauthorized copy of software is made.	A working program is modified, either to cause it to fail during execution or to cause it to do some unintended task.
Data	Files are deleted, denying access to users.	An unauthorized read of data is performed. An analysis of statistical data reveals underlying data.	Existing files are modified or new files are fabricated.
Communication Lines	Messages are destroyed or deleted. Communication lines or networks are rendered unavailable.	Messages are read. The traffic pattern of messages is observed.	Messages are modified, delayed, reordered, or duplicated. False messages are fabricated.

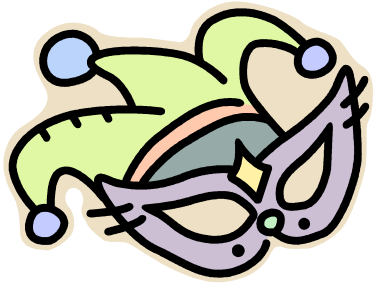
Passive Attacks

- Attempts to learn or make use of information from the system but does not affect system resources
- Are in the nature of eavesdropping on, or monitoring of, transmissions
- Goal of the attacker is to obtain information that is being transmitted
- Difficult to detect because they do not involve any alteration of the data
  - is feasible to prevent the success of these attacks by means of encryption
- Emphasis in dealing with passive attacks is on prevention rather than detection

Types:

- release of message contents
- traffic analysis





Active Attacks

- Involve some modification of the data stream or the creation of a false stream
- Four categories:
  1. **Replay**
     - involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect
  2. **Masquerade**
     - takes place when one entity pretends to be a different entity
  3. **Modification of messages**
     - some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect
  4. **Denial of service**
     - prevents or inhibits the normal use or management of communications facilities
     - disruption of an entire network either by disabling the network or by overloading it with messages so as to degrade performance

Intruder Patterns of Behavior: Hackers

- Traditionally those who hack do so for the thrill of it or for status
- Attackers often look for targets of opportunity and then share the information with others
- Benign intruders consume resources and may slow performance for legitimate users
- Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are designed to counter this type of hacker threat
- Computer emergency response teams (CERTs) are cooperative ventures who collect information about system vulnerabilities and disseminate it to systems managers

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.

(a) Hacker

Intruder Patterns of Behavior: Criminal Enterprise

- Organized groups of hackers
- They meet in underground forums to trade tips and data and coordinate attacks
- A common target is a credit card file at an e-commerce server
- Usually have specific targets, or at least classes of targets in mind
- Quick in and quick out attacks

1. Act quickly and precisely to make their activities harder to detect.
2. Exploit perimeter through vulnerable ports.
3. Use Trojan horses (hidden software) to leave backdoors for reentry.
4. Use sniffers to capture passwords.
5. Do not stick around until noticed.
6. Make few or no mistakes.

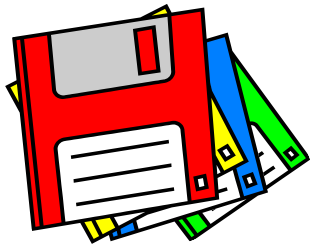
(b) Criminal Enterprise

Intruder Patterns of Behavior: Insider Attacks

- Among the most difficult to detect and prevent
- Can be motivated by revenge or simply a feeling of entitlement
- Employees already have access to and knowledge of the structure and content of corporate databases

1. Create network accounts for themselves and their friends.
2. Access accounts and applications they wouldn't normally use for their daily jobs.
3. E-mail former and prospective employers.
4. Conduct furtive instant-messaging chats.
5. Visit Web sites that cater to disgruntled employees, such as f'dcompany.com.
6. Perform large downloads and file copying.
7. Access the network during off hours.

(c) Internal Threat



Malware

General term for any malicious software

Software designed to cause damage to or use up the resources of a target computer

Frequently concealed within or masquerades as legitimate software

In some cases it spreads itself to other computers via e-mail or infected discs

Terminology of Malicious Programs

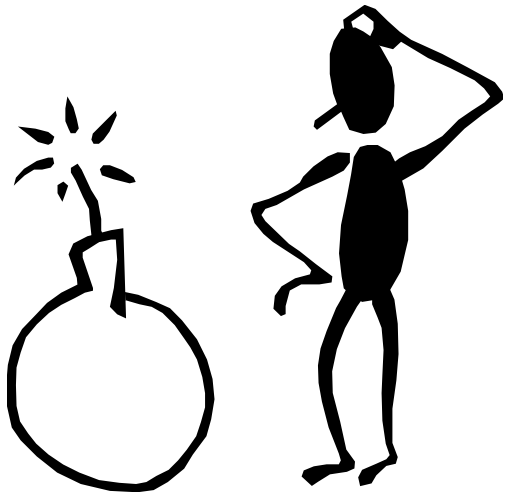
Name	Description
Virus	Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.
Logic bomb	A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.
Backdoor (trapdoor)	Any mechanisms that bypasses a normal security check; it may allow unauthorized access to functionality.
Platform independent code	Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Kit (virus generator)	Set of tools for generating new viruses automatically.
Spammer programs	Used to send large volumes of unwanted e-mail.
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.
Spyware	Software that collects information from a computer and transmits it to another system.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Backdoor

- Also known as a trapdoor
- A secret entry point into a program that allows someone to gain access without going through the usual security access procedures
- A ***maintenance hook*** is a backdoor that programmers use to debug and test programs, e.g., that need a long setup
- Become threats when unscrupulous programmers use them to gain unauthorized access
- It is difficult to implement operating system controls for backdoors

Logic Bomb

- One of the oldest types of program threat
- Code embedded in some legitimate program that is set to “explode” when certain conditions are met
- Once triggered a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage



Trojan Horse

- Useful, or apparently useful, program or command procedure that contains hidden code that, when invoked, performs some unwanted or harmful function
- Trojan horses fit into one of three models:
    1) continuing to perform the function of the original program and additionally performing a separate malicious activity
    2) continuing to perform the function of the original program but modifying the function to perform malicious activity or to disguise other malicious activity
    3) performing a malicious function that completely replaces the function of the original program

Platform independent Code

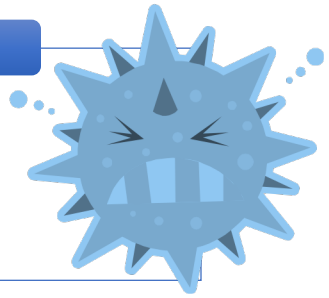
- Sometimes referred to as *mobile code*
- Programs that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics
- Transmitted from a remote system to a local system and then executed on the local system without the user's explicit instruction
- Often acts as a mechanism for a virus, worm, or Trojan horse to be transmitted to the user's workstation
- Takes advantages of vulnerabilities
- Popular vehicles for mobile code include Java applets, ActiveX, JavaScript, and VBScript

Multiple-Threat Malware

- Infects in multiple ways
- A multipartite virus is capable of infecting multiple types of files
- A blended attack uses multiple methods of infection or transmission to maximize the speed of contagion and the severity of the attack
- An example of a blended attack is the Stuxnet attack

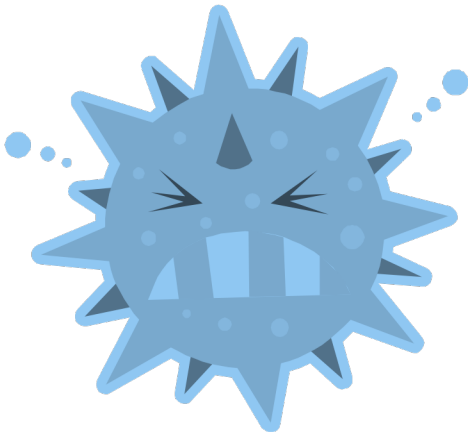
Stuxnet uses many distribution methods:

- Usb infection
- Windows vulnerabilities
- Network analysis and Privilege escalation
- Specific applications for PLA programming

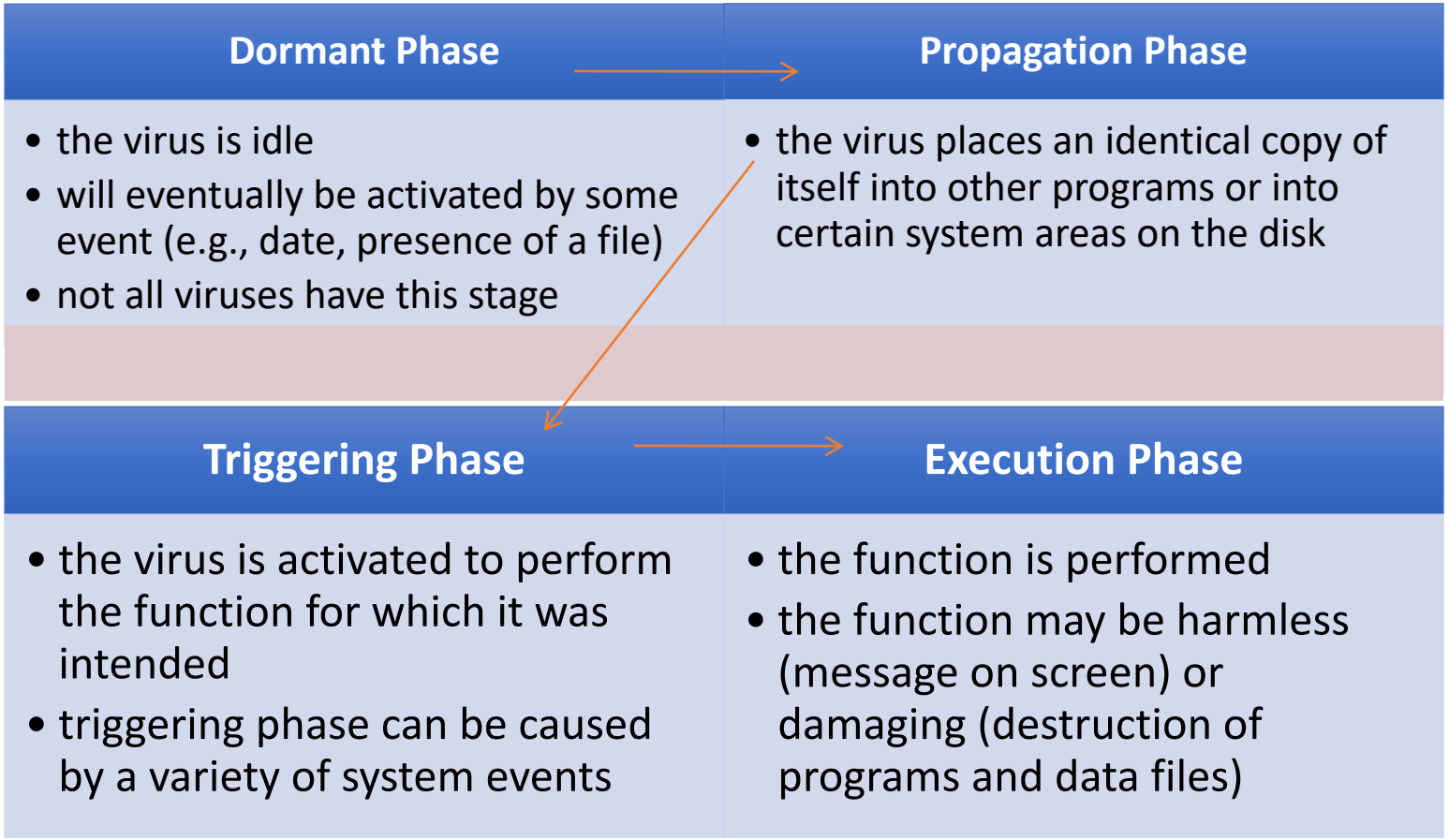


Viruses

- Software that “infects” other programs by modifying them
  - carries instructional code to self duplicate
  - becomes embedded in a program on a computer
  - when the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program
  - infection can be spread by swapping disks from computer to computer or through a network (a perfect culture for the spread of a virus!)
- A computer virus has three parts:
  - an infection mechanism
  - trigger
  - payload
    - may involve damage
    - or may involve benign but noticeable activity



Virus Phases



Virus Classification

- There is no universally agreed upon classification scheme for viruses
- **Classification by target** includes the following categories:

Boot sector infector

- infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus

File infector

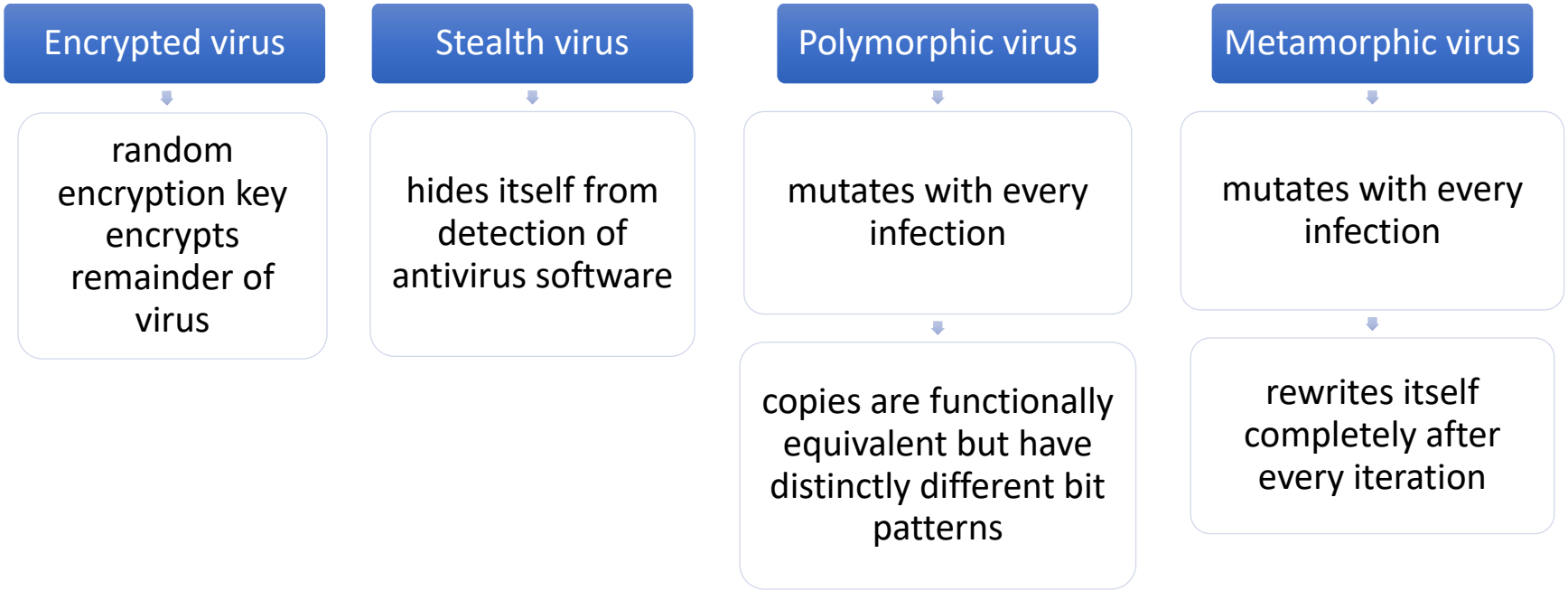
- infects files that the operating system or shell consider to be executable

Macro virus

- infects files with macro code that is interpreted by an application

Concealment Strategy

- A virus **classification by concealment strategy** includes:



Macro Viruses

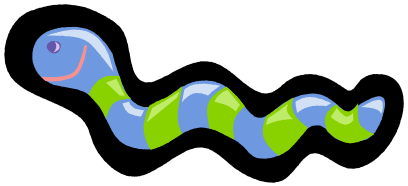
- In the mid 1990's macro viruses became by far the most prevalent type of virus
- Macro viruses are particularly threatening because:
    - they are platform independent; many macro viruses infect Microsoft Word documents or other Microsoft Office documents
    - they infect **documents**, not executable portions of code
    - they are easily spread; a very common method is by e-mail
    - file system access controls are of limited use in preventing their spread

E-Mail Viruses

- The first rapidly spreading e-mail viruses made use of a Microsoft Word macro embedded in an attachment

- In 1999 a newer, more powerful version of the e-mail virus appeared
  - can be activated merely by opening an e-mail that contains the virus rather than opening an attachment
  - the virus uses the Visual Basic scripting language supported by the e-mail package

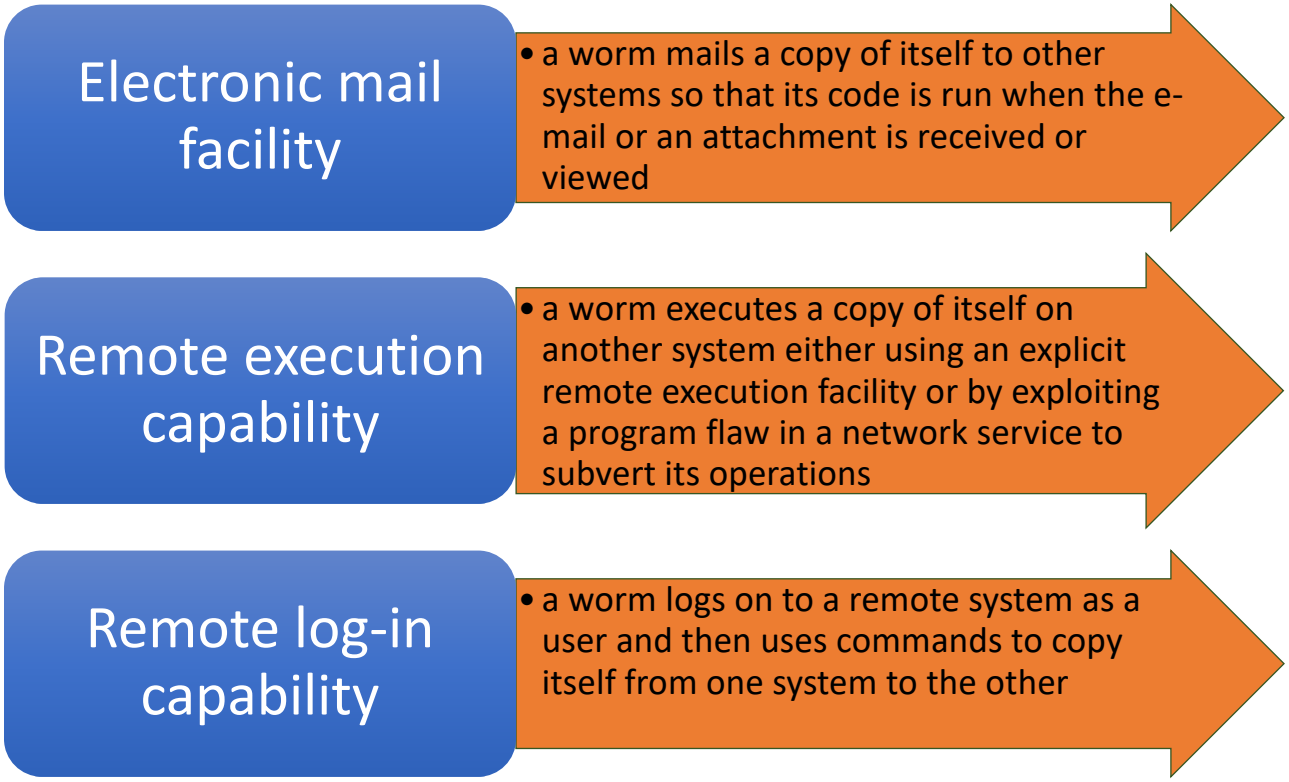
Worms

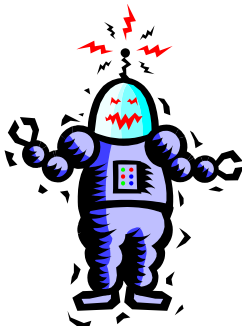


- A program that can replicate itself and send copies from computer to computer across network connections
- Upon arrival the worm may be activated to replicate and propagate again
- In addition to propagation the worm usually performs some unwanted function
- Actively seeks out more machines to infect and each machine that is infected serves as an automate launching pad for attacks on other machines

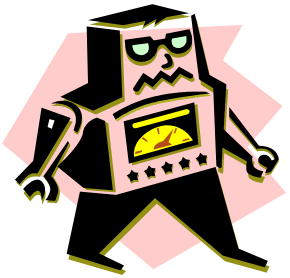
Worm Propagation

- To replicate itself a network worm uses some sort of network vehicle





Bots



- A program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the bot's creator
    - also known as a Zombie or drone
- Typically planted on hundreds or thousands of computers belonging to unsuspecting third parties
- Collection of bots acting in a coordinated manner is a **botnet**
- A botnet exhibits three characteristics:
    1) the bot functionality
    2) a remote control facility
    3) a spreading mechanism to propagate the bots and construct the botnet

Uses of Bots

Distributed denial-of-service (DDoS) attacks

- causes a loss of service to users

Spamming

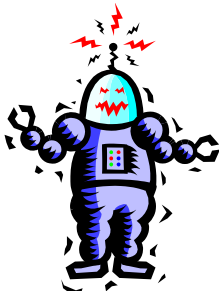
- sending massive amounts of bulk e-mail (spam)

Sniffing traffic

- a packet sniffer is used to retrieve sensitive information like user names and passwords

Keylogging

- captures keystrokes



Spreading new malware

- botnets are used to spread new bots

Installing advertisement add-ons and browser helper objects (BHOs)

- set up a fake Web site and negotiate a deal with hosting companies that pay for clicks on ads

Attacking Internet Relay chat (IRC) chat networks

- victim is flooded with requests, bringing down the IRC network; similar to a DDoS attack

Manipulating online polls/games

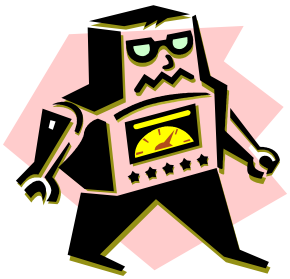
- every bot has a distinct IP address so it appears to be a real person

Remote Control Facility

- Distinguishes a bot from a worm
    - a worm propagates and activates itself, whereas a bot is controlled from some central facility
- A typical means of implementing the remote control facility is on an IRC server
    - all bots join a specific channel on this server and treat incoming messages as commands
- More recent botnets tend to use covert communication channels via protocols such as HTTP
- Distributed control mechanisms are also used to avoid a single point of failure

Constructing the Attack Network

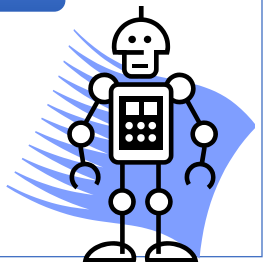
- The first step in a botnet attack is for the attacker to infect a number of machines with bot software that will ultimately be used to carry out the attack
- Essential ingredients:
  1) software that **can** carry out the attack
  2) a vulnerability in a large number of systems
  3) strategy for locating and identifying vulnerable machines (a process known as scanning or fingerprinting)



- In the scanning process the attacker first seeks out a number of vulnerable machines and infects them
  - the bot software in the infected machines repeats the same scanning process until a large distributed network of infected machines is created

Scanning strategies:

- Random
- Hit list
- Topological
- Local subnet

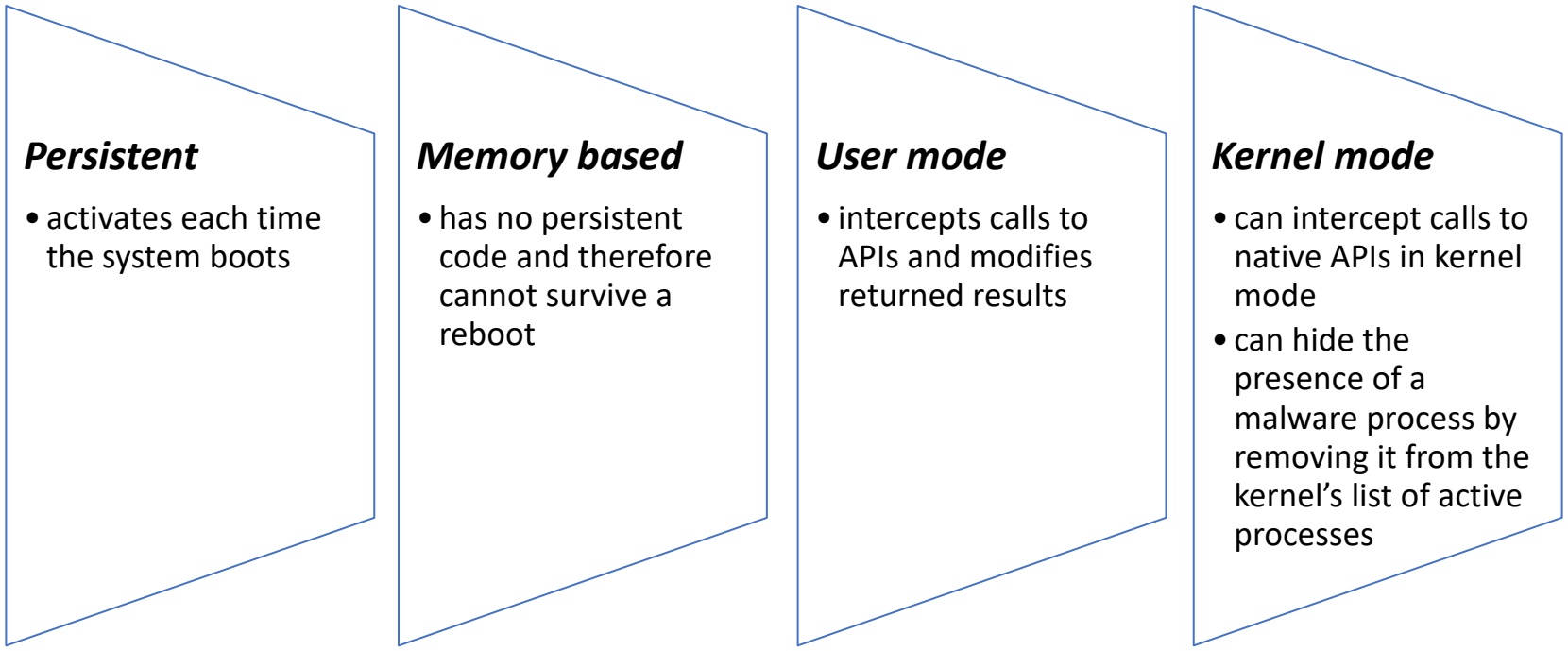


Rootkit

- Set of programs installed on a system to maintain administrator (or root) access to that system
- Root access provides access to all the functions and services of the operating system
- The rootkit alters the host's standard functionality in a malicious and stealthy way
    - with root access an attacker has complete control of the system and can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand
- A rootkit hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer

Rootkit Classification

- Rootkits can be classified based on whether they can survive a reboot and execution mode
- A rootkit may be:



Rootkit Installation

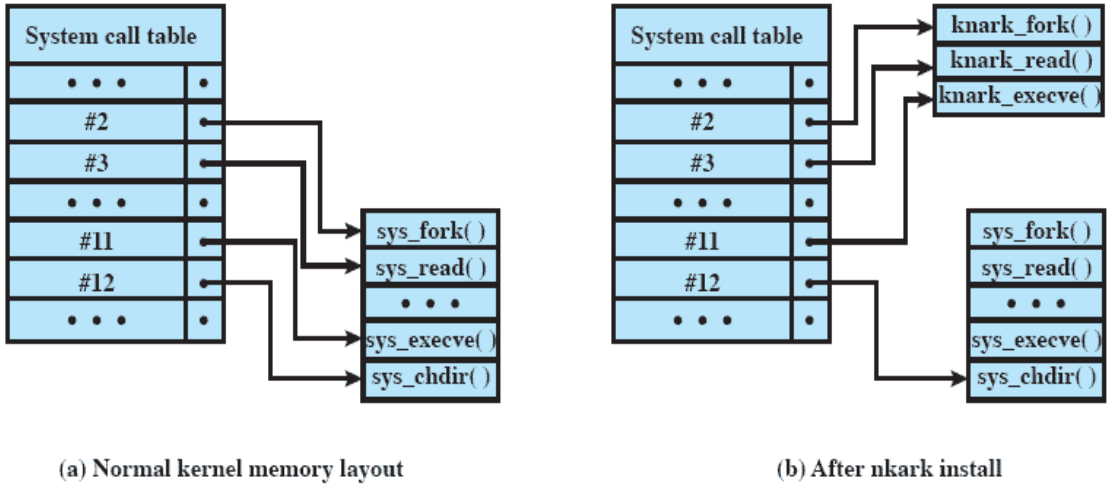
- Rootkits do not directly rely on vulnerabilities or exploits to get on a computer

- One method of installation is via a Trojan horse program

- Another means of installation is by hacker activity

System-Level Call Attacks

- Programs operating at the user level interact with the kernel through system calls
- In Linux each system call is assigned a unique syscall number
- Three techniques that can be used to change system calls:
  - modify the system call table to point to rootkit's code
  - modify system call table targets
  - redirect the entire system call table



(a) Normal kernel memory layout

(b) After nkark install

Operating Systems Defense: Internals and Design Principles

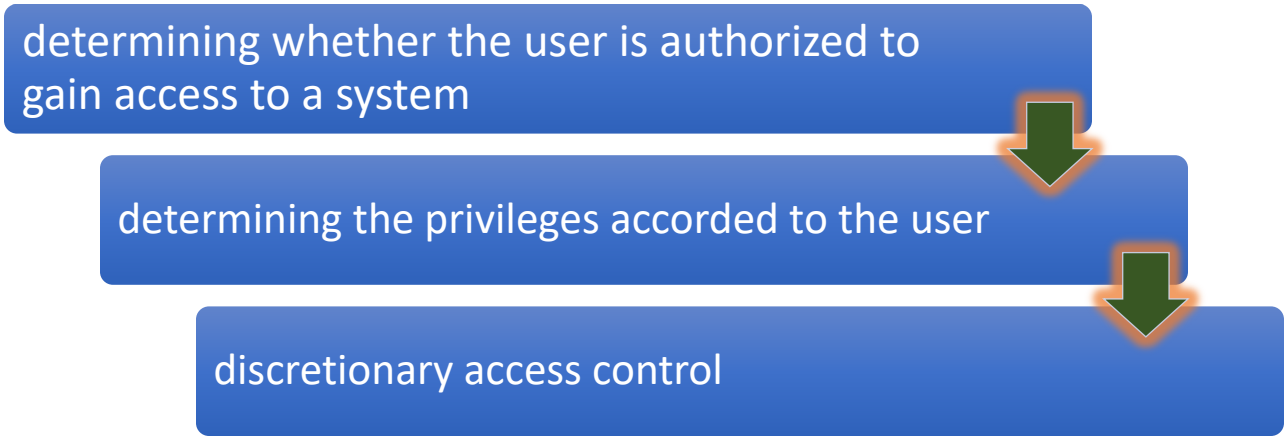
To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.

— THE GOLDEN BOUGH,

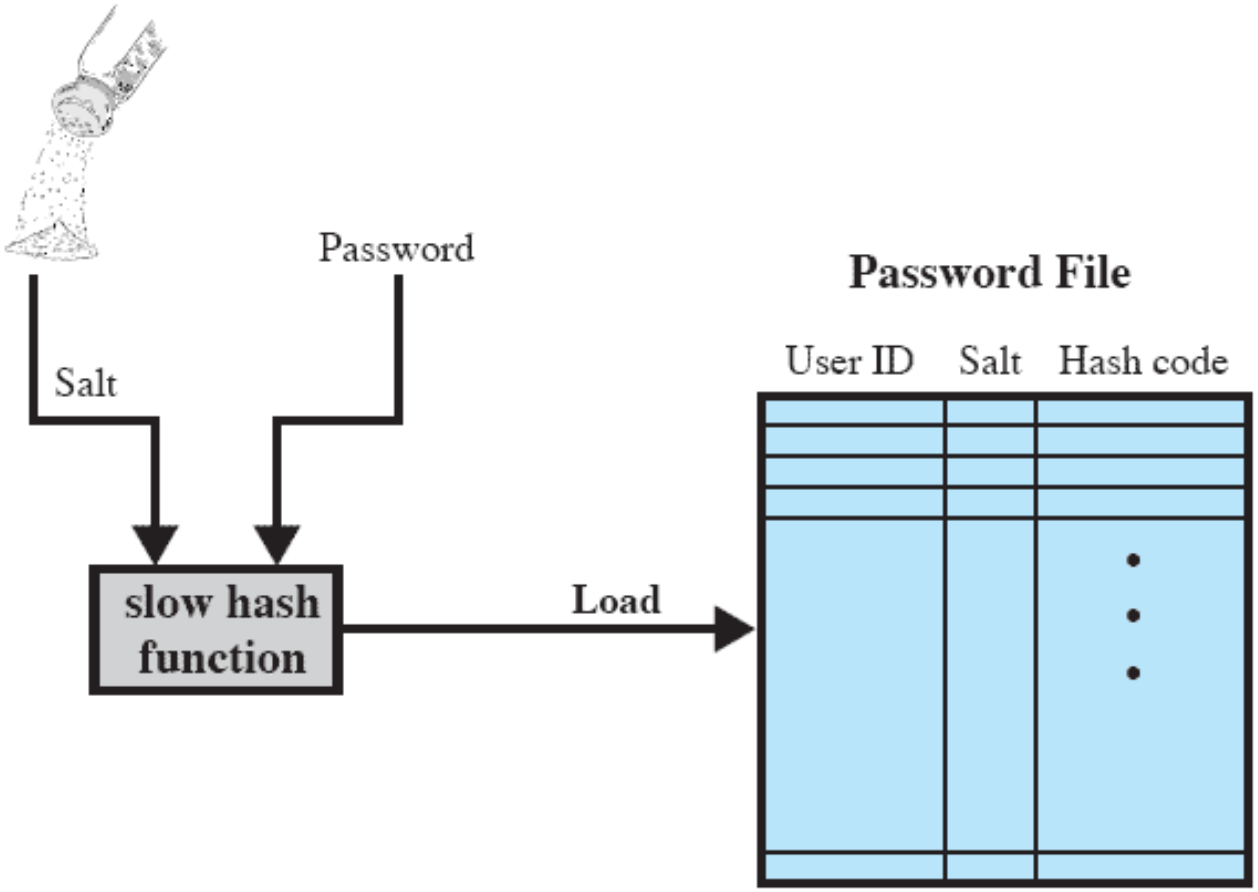
Sir James George Frazer

Password-Based Authentication

- A widely used line of defense against intruders is the password system
- The password serves to authenticate the ID of the individual logging on to the system
- The ID provides security by:

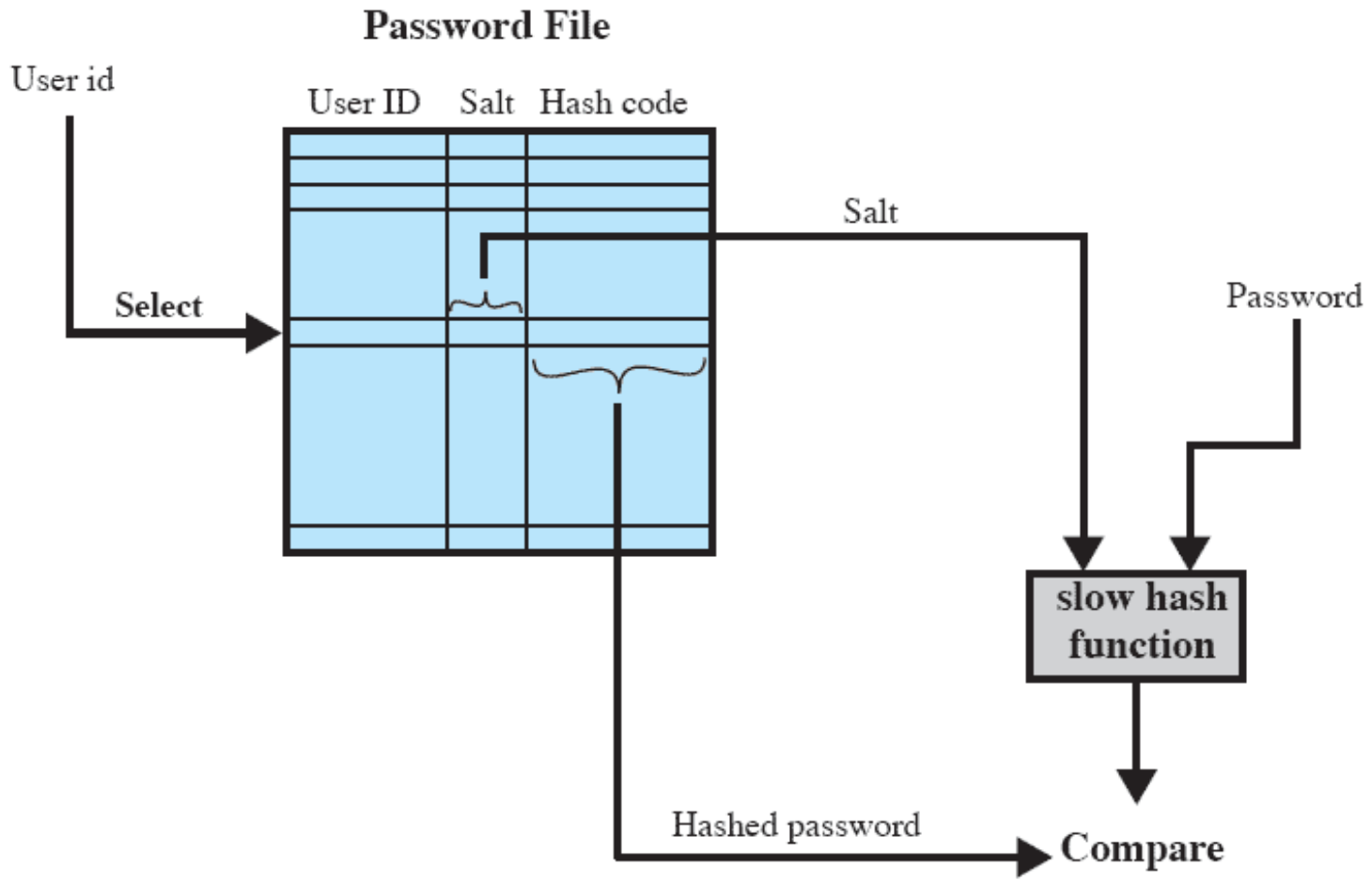


Hashed Passwords/Salt Value



(a) Loading a new password

UNIX Password Scheme



(b) Verifying a password

Salt

- Serves three purposes:
  1. prevents duplicate passwords from being visible in the password file
     - even if two users choose the same password, the passwords will be assigned different salt values
  2. greatly increases the difficulty of offline dictionary attacks
  3. it becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them

UNIX Password Scheme

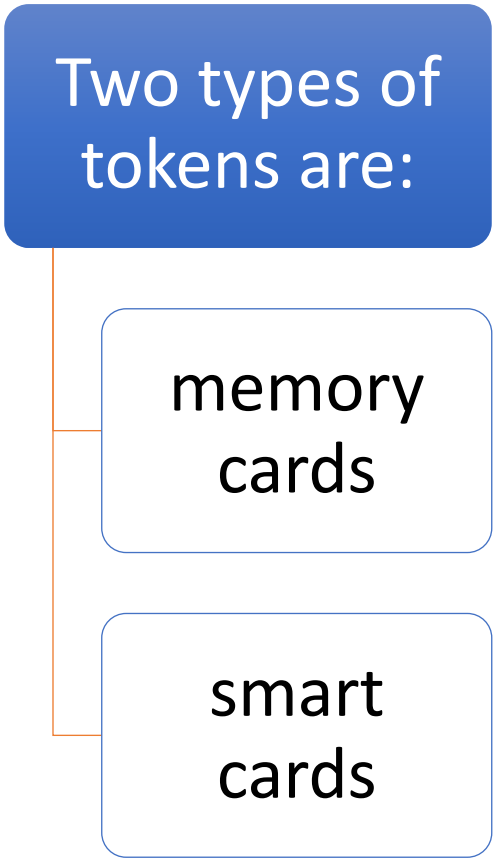
- There are two threats to the UNIX password scheme:
  1. A user can gain access on a machine using a guest account, then they run a ***password cracker*** (password guessing program) on it
  2. If an opponent is able to obtain a copy of the password file, a cracker program can be run on another machine at leisure
     - this enables the opponent to run through millions of possible passwords in a reasonable period

UNIX Implementations

- Most implementations have relied on a password scheme where each user selects a password of up to eight printable characters in length which is converted into a 56-bit value (using 7-bit ASCII) that serves as the key input to an encryption routine
- The hash routine, known as crypt(3) is based on DES
- The crypt(3) routine is designed to discourage guessing attacks
- Software implementations of DES are slow compared to hardware versions
- The recommended hash function for many UNIX systems, including Linux, Solaris, and FreeBSD is based on the MD5 hash algorithm
- The most secure version of the UNIX hash/salt scheme was developed for OpenBSD and uses a hash function (Bcrypt) based on the Blowfish symmetric block cipher

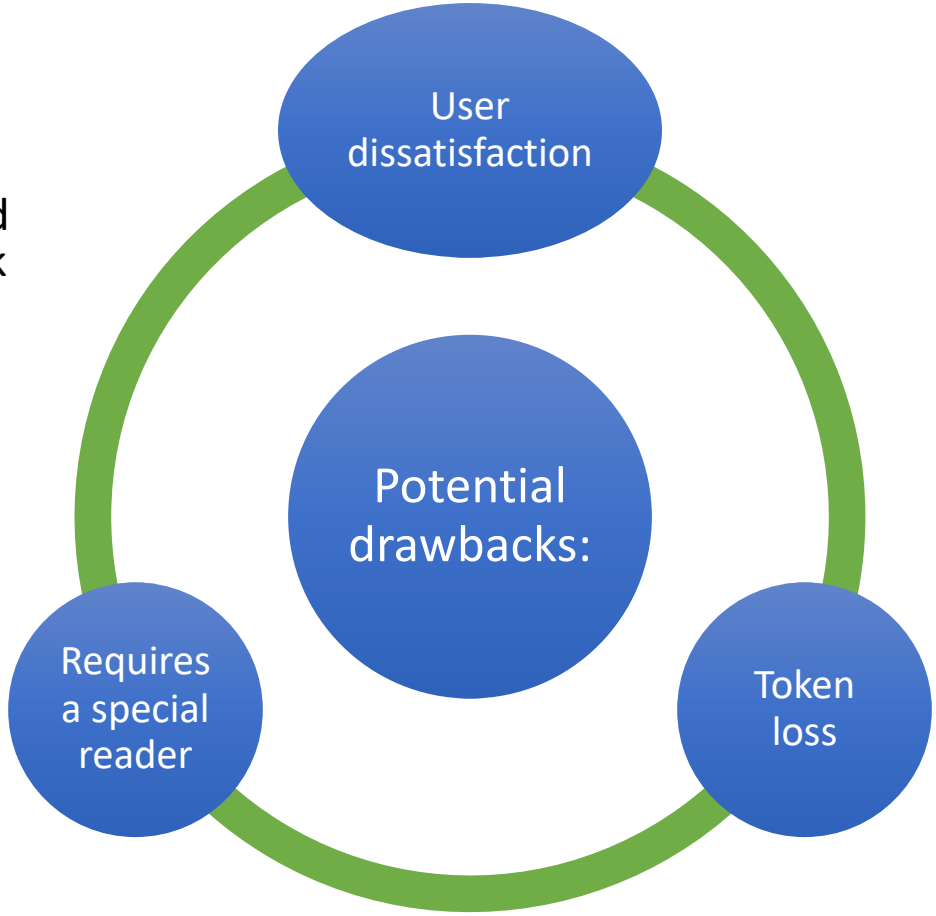
Token-Based Authentication

- Objects that a user possesses for the purpose of user authentication are called tokens



Memory Cards

- Memory cards can store but not process data
  - the most common is the bank card with a magnetic stripe on the back
  - a magnetic stripe can store only a simple security code which can be read and reprogrammed by an inexpensive card reader
  - there are also memory cards that include an internal electronic memory
- Can be used alone for physical access or with some form of password or personal identification number (PIN)



Smart Cards

Physical characteristics:

- include an embedded processor
- a smart token that looks like a bank card is called a smart card
- smart tokens can look like calculators, keys, or other small portable objects

Interface

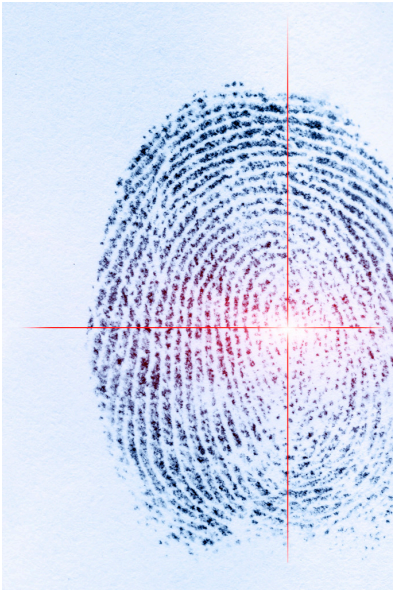
- manual interfaces include a keypad and display for human/token interaction
- smart tokens with an electronic interface communicate with a compatible reader/writer

Authentication protocol

- static
- dynamic password generator
- challenge-response

Static Biometric Authentication

- Attempts to authenticate an individual based on his or her unique physical characteristics
- Includes static characteristics such as:
  - fingerprints
  - hand geometry
  - facial characteristics
  - retinal and iris patterns
- Dynamic characteristics such as:
  - voiceprint
  - signature



Physical Characteristics

Facial characteristics

- most common means of human-to-human identification
- examples: eyes, eyebrows, nose, lips, and chin shape
- alternative approach is to use an infrared camera to produce a face thermogram

Fingerprints

- pattern of ridges and furrows on the surface of the fingertip
- unique across the entire human population

Hand geometry

- identify features of the hand, including shape, and lengths and widths of fingers

Retinal pattern

- the pattern formed by veins beneath the retinal surface is unique
- a retinal biometric system obtains a digital image of the retinal pattern by projecting a low-intensity beam of visual or infrared light into the eye

Iris

- the detailed structure of the iris is unique

Signature

- each individual has a unique style of handwriting

Voice

- voice patterns are closely tied to the physical and anatomical characteristics of the speaker

Cost versus Accuracy

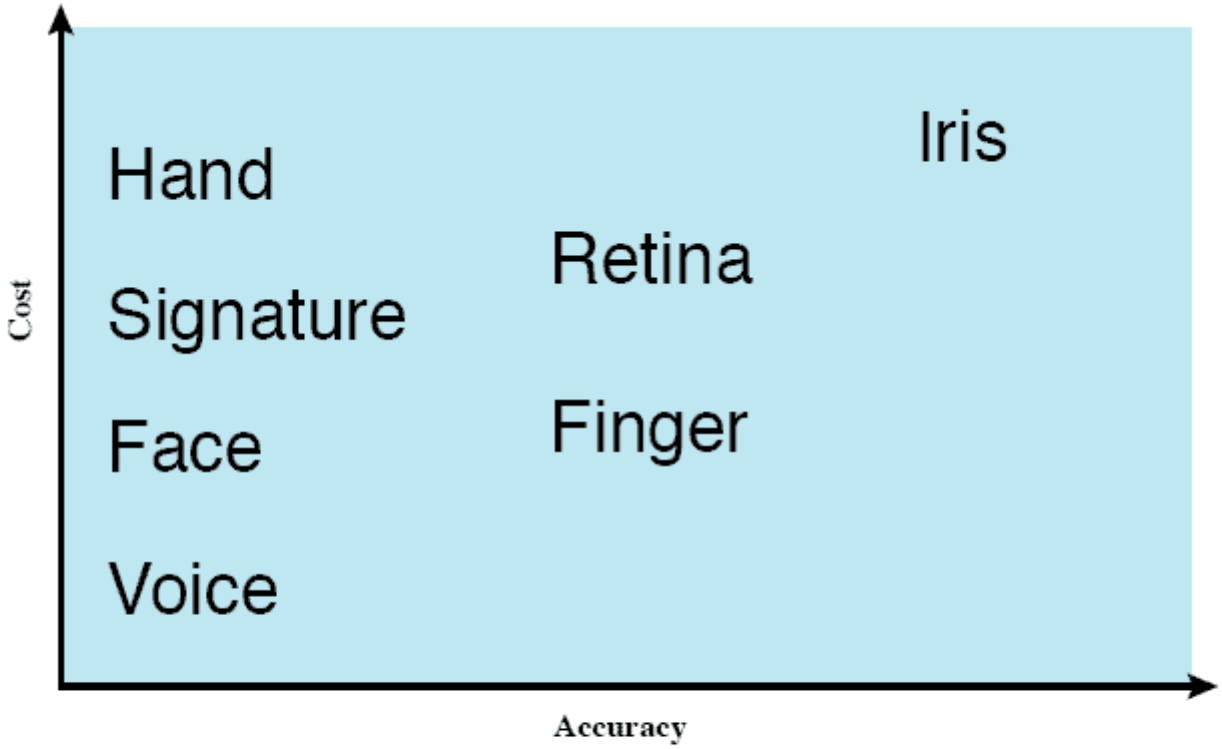


Figure 15.2 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

Access Control

- An *access controly policy* dictates what types of access are permitted, under what circumstances, and by whom
- Access control policies are generally grouped into the following categories:
    - Discretionary access control (DAC)
        - controls access based on the identity of the requestor and on access rules stating what requestors are (or are not) allowed to do
    - Mandatory access control (MAC)
        - controls access based on comparing security labels with security clearances
    - Role-based access control (RBAC)
        - controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles

Access Control Policies

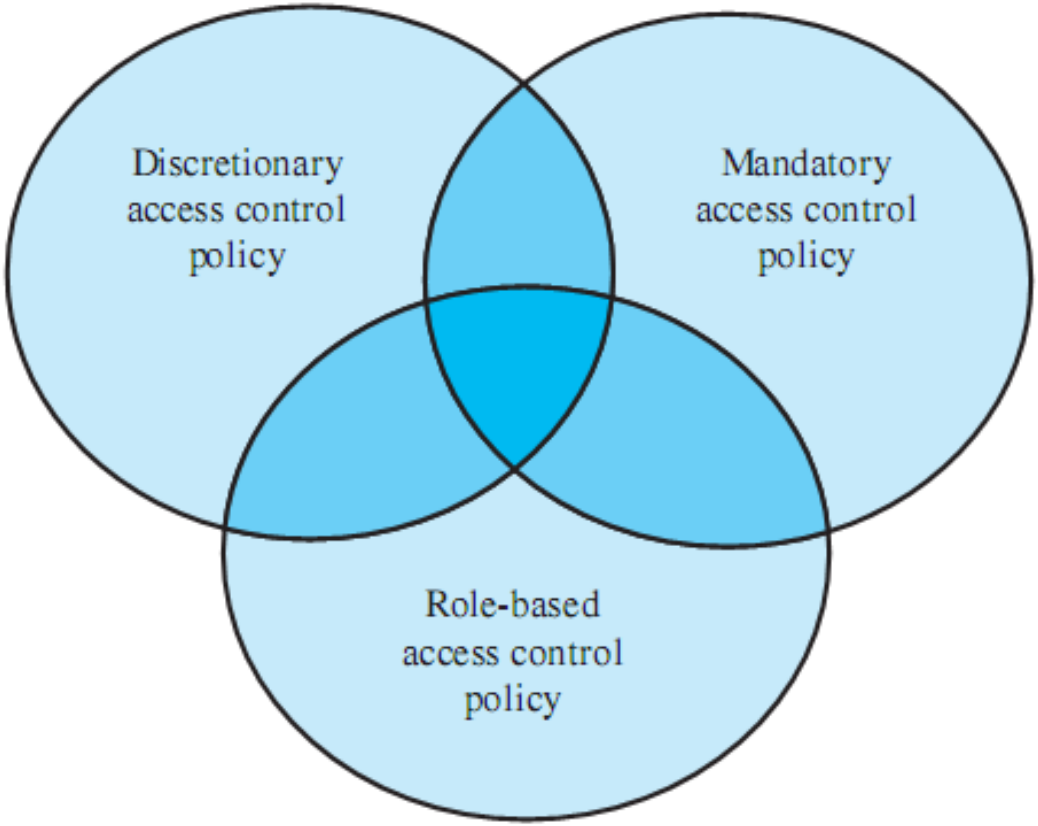
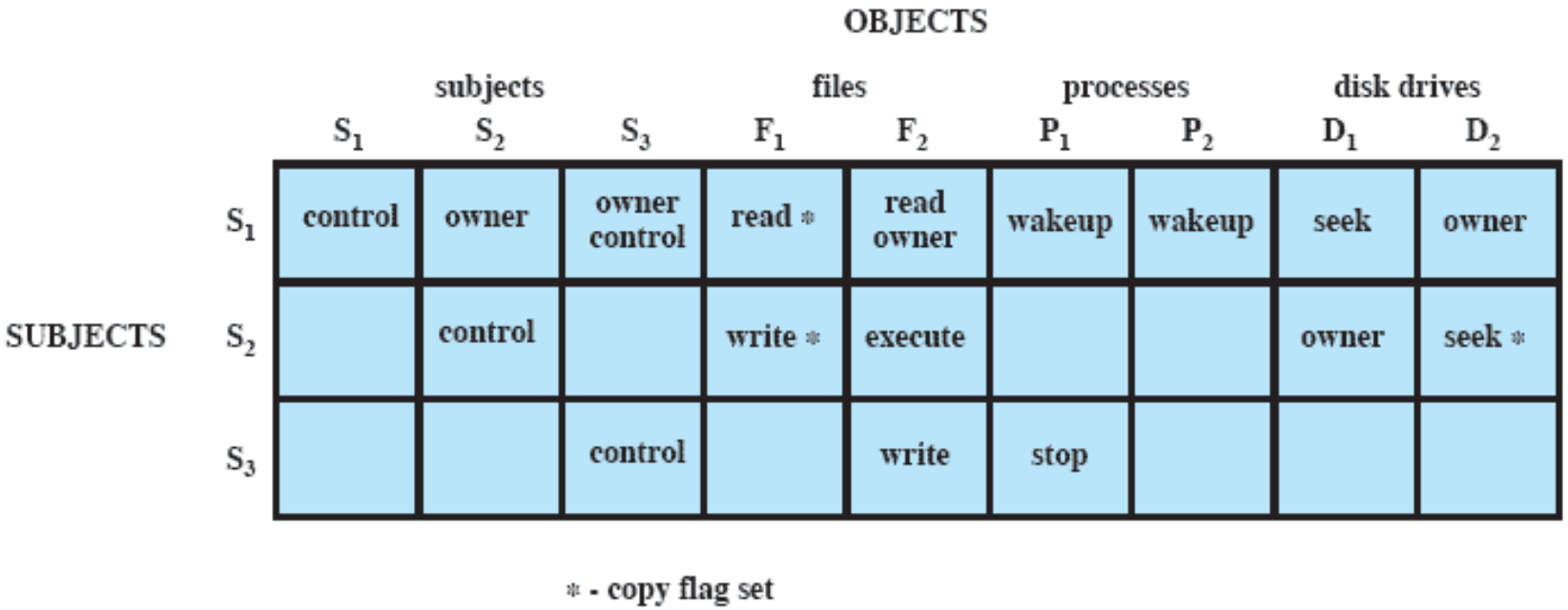


Figure 15.3 Access Control Policies

Extended Access Control Matrix



		OBJECTS								
		subjects			files		processes		disk drives	
		S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
SUBJECTS	S_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	S_2		control		write *	execute			owner	seek *
	S_3			control		write	stop			

* - copy flag set

Figure 15.4 Extended Access Control Matrix

Organization of the Access Control Function

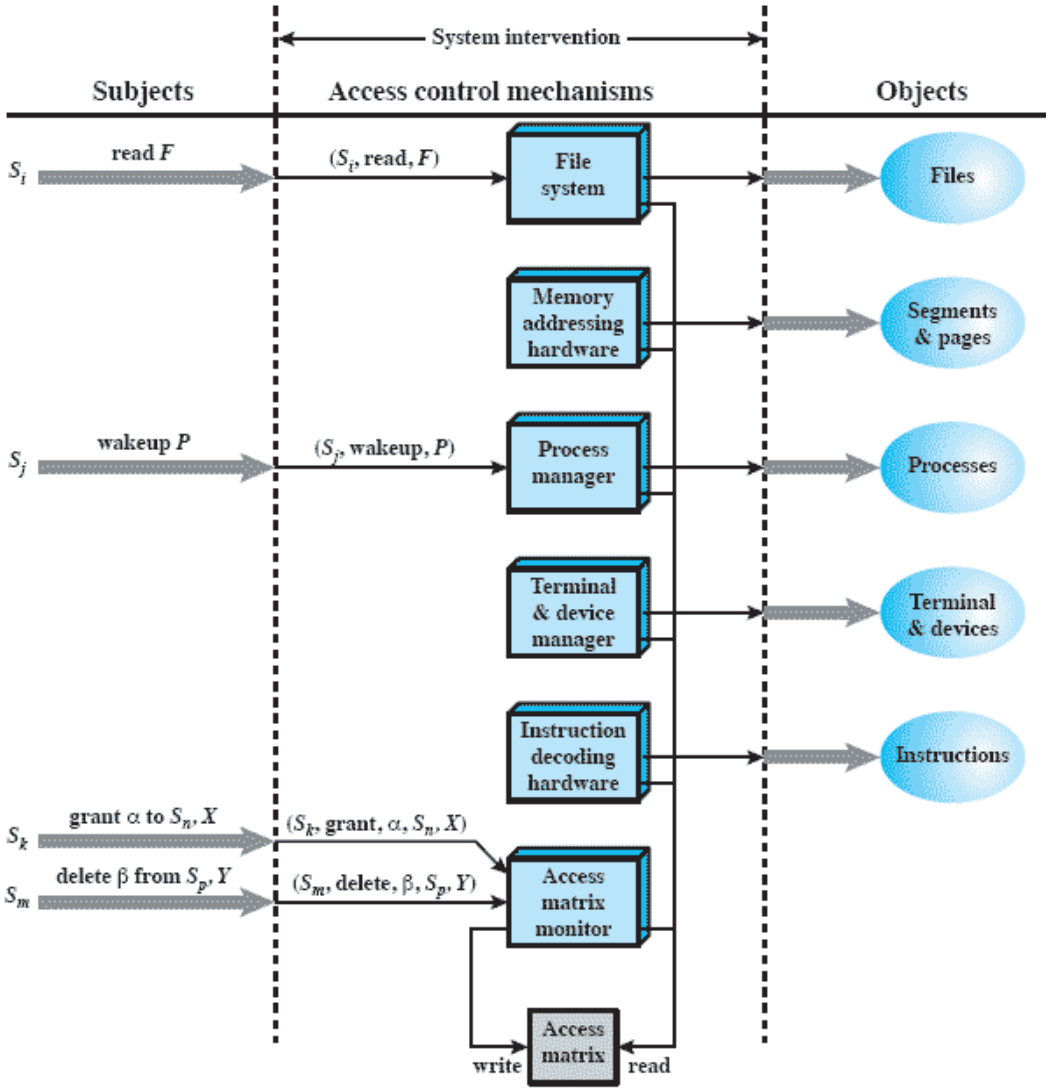


Figure 15.5 An Organization of the Access Control Function

Access Control System Commands

Rule	Command (by S_0)	Authorization	Operation
R1	**transfer** $\left\{\begin{matrix}\alpha^* \\ \alpha\end{matrix}\right\}$ **to** S, X	' *' in $A[So, X]$	store $\left\{\begin{matrix}\alpha^* \\ \alpha\end{matrix}\right\}$ in $A[S, X]$
R2	**grant** $\left\{\begin{matrix}\alpha^* \\ \alpha\end{matrix}\right\}$ **to** S, X	'owner' in $A[So, X]$	store $\left\{\begin{matrix}\alpha^* \\ \alpha\end{matrix}\right\}$ in $A[S, X]$
R3	**delete** α **from** S, X	'control' in $A[So, S]$ or 'owner' in $A[So, X]$	delete α from $A[S, X]$
R4	w α **read** S, X	'control' in $A[So, S]$ or 'owner' in $A[So, X]$	copy $A[S, X]$ into w
R5	**create object** X	None	add column for X to A; store 'owner' in $A[So, X]$
R6	**destroy object** X	'owner' in $A[So, X]$	delete column for X from A
R7	**create subject** S	None	add row for S to A; execute **create object** S; store 'control' in $A[S, S]$
R8	**destroy subject** S	'owner' in $A[So, S]$	delete row for S from A; execute **destroy object** S

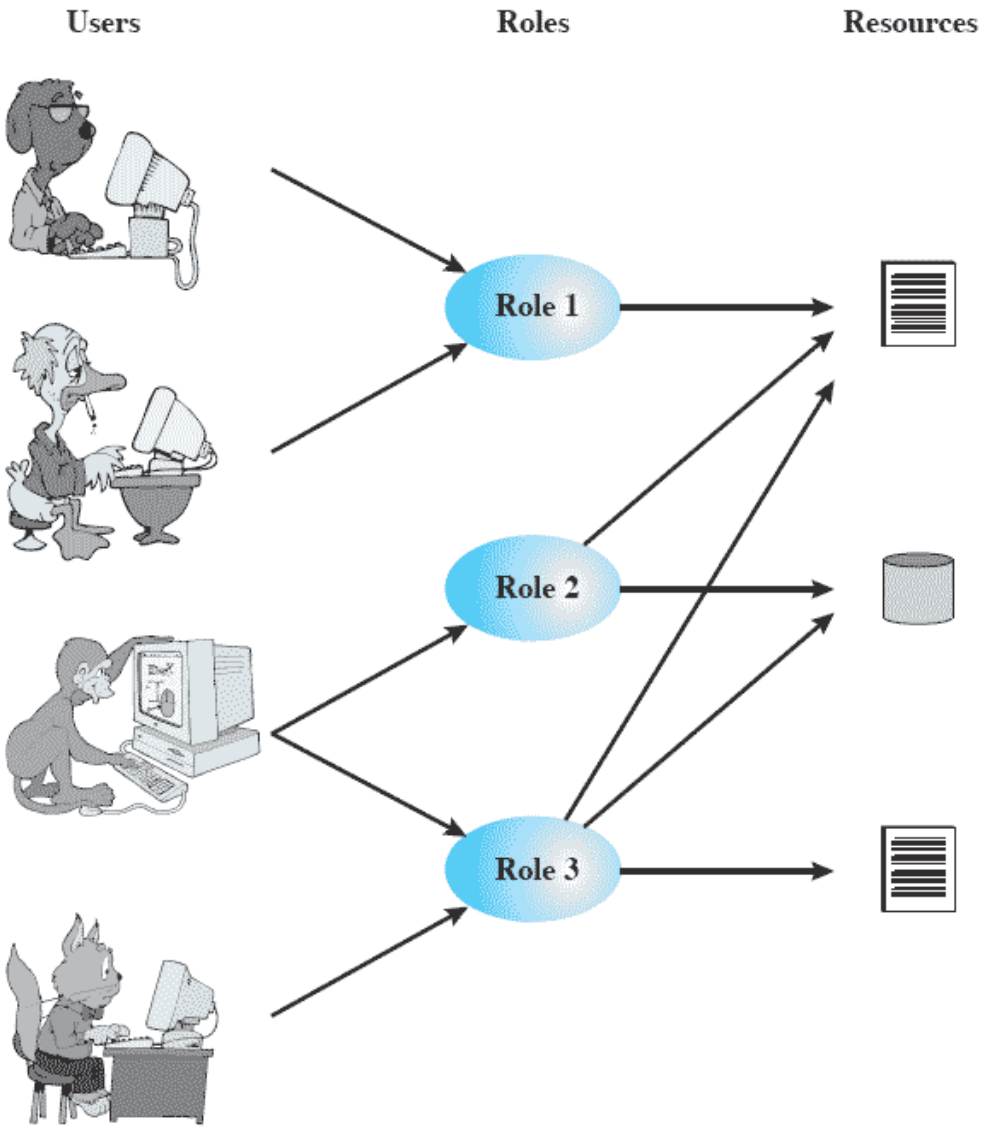
Role-Based Access Control

- Based on the roles that users assume in a system rather than the user's identity
- Models define a role as a job function within an organization
- Systems assign access rights to roles instead of individual users
    - in turn, users are assigned to different roles, either statically or dynamically, according to their responsibilities
- NIST has issued a standard that requires support for access control and administration through roles

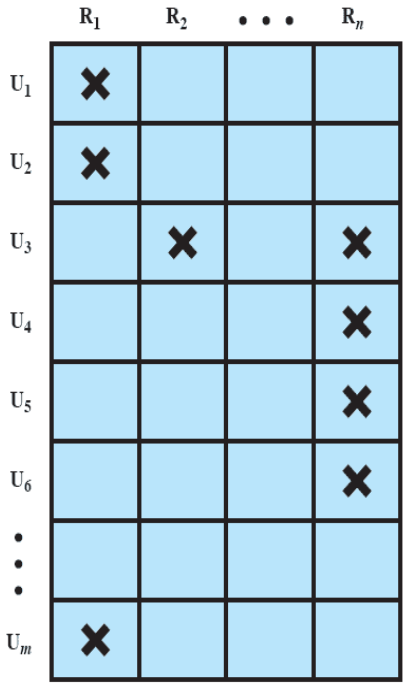
Users

Roles

Resources



Access Control Matrix Representation of RBAC



OBJECTS

ROLES	R_1	R_2	R_n	F_1	F_1	P_1	P_2	D_1	D_2
R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R_2		control		write *	execute			owner	seek *
• • •									
R_n			control		write	stop			

Intrusion Detection - Basic Principles

- Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified
- 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 IDS can serve as a deterrent, thus acting to prevent intrusions
- Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen instrusion prevention measures

Profiles of Behavior

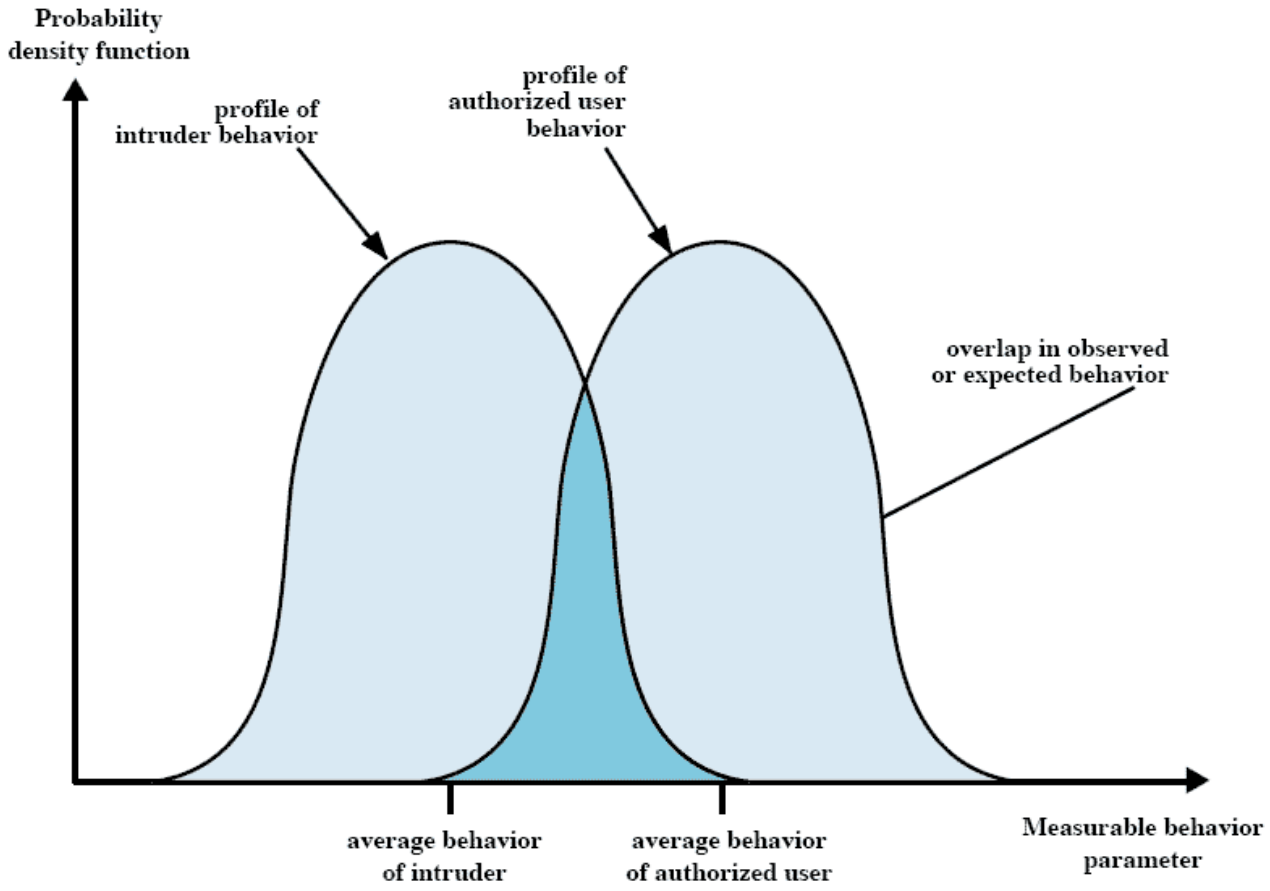
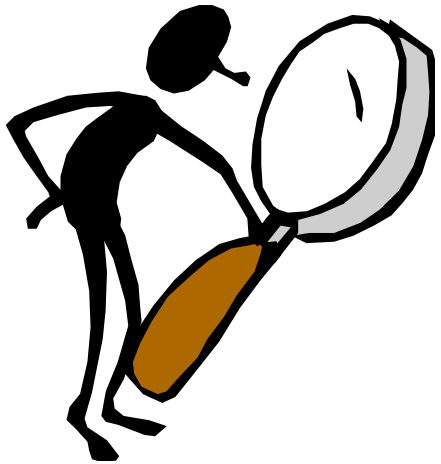


Figure 15.8 Profiles of Behavior of Intruders and Authorized Users

Host-Based Intrusion Detection System (IDS)

- Monitors activity on the system in a variety of ways to detect suspicious behavior
- Primary purpose is to detect intrusions, log suspicious events, and send alerts
- Can detect both external and internal intrusions

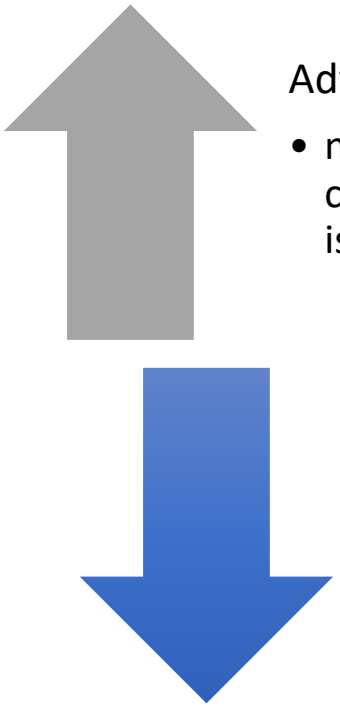


- Anomaly detection
  - collection of data relating to behavior of legitimate users over time
  - threshold detection
  - profile based detection
- Signature detection
  - define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder

Audit Records

Some record of ongoing user activity is provided to the IDS

Native



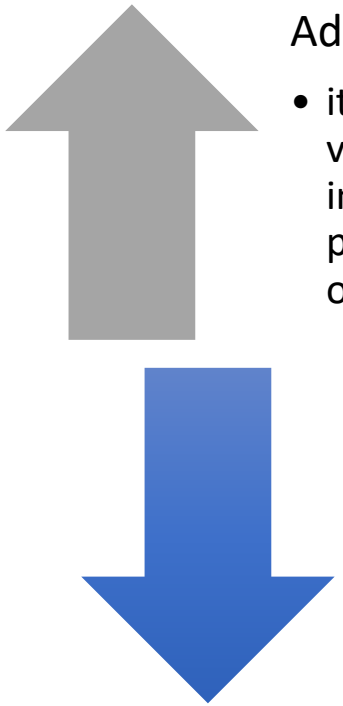
Advantage

- no additional collection software is needed

Disadvantage

- the native audit records may not contain the needed information or may not contain it in a convenient form

Detection-specific



Advantage

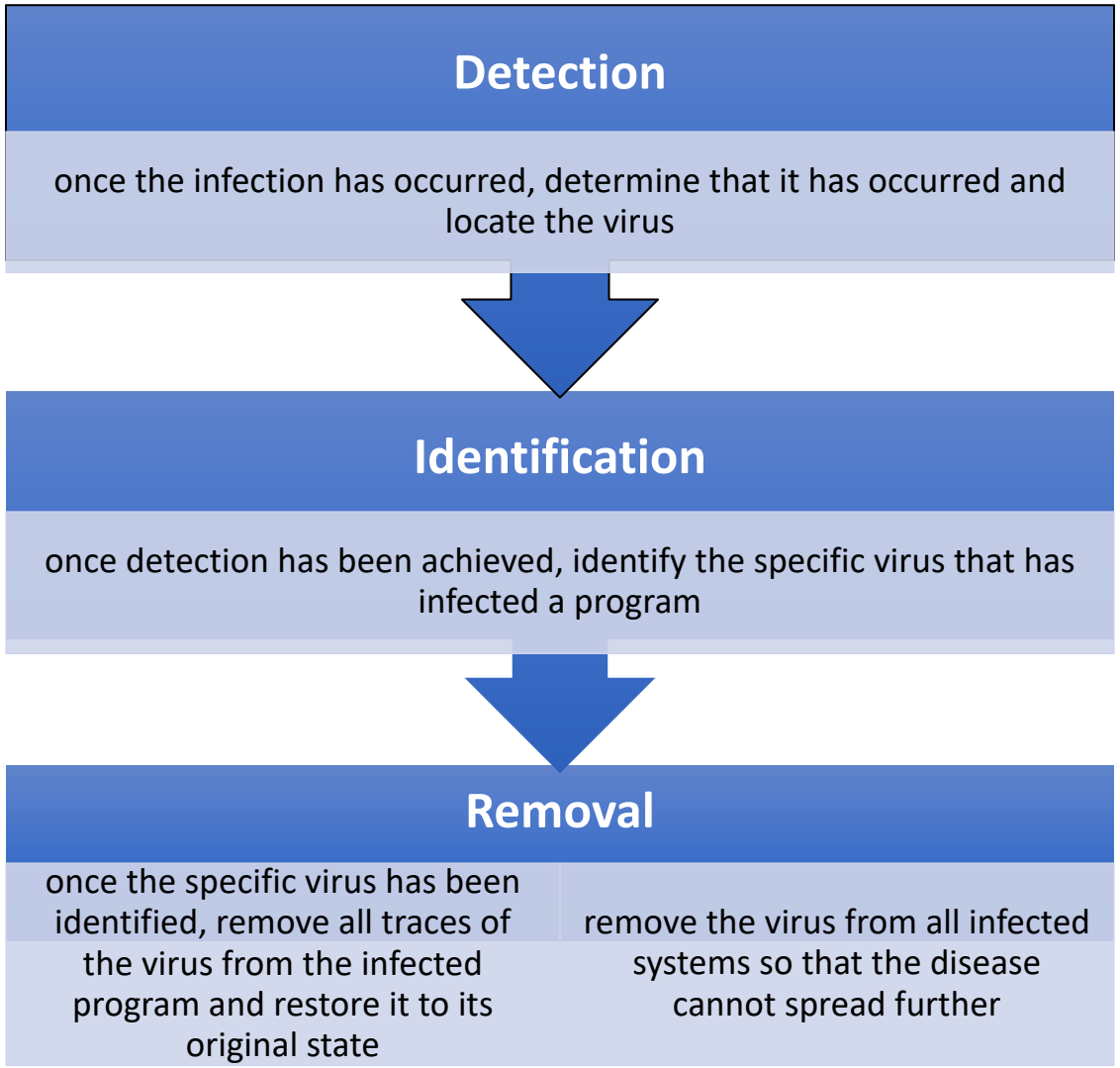
- it could be made vendor independent and ported to a variety of systems

Disadvantage

- the extra overhead involved in having, in effect, two accounting packages running on a machine

Antivirus Approaches

- Ideal solution to the threat of viruses is prevention, don't allow a virus onto the system in the first place!
- That goal is, in general, impossible to achieve, although prevention can reduce the number of successful viral attacks
  - If detection succeeds but either identification or removal is not possible, then the alternative is to discard the infected program and reload a clean backup version



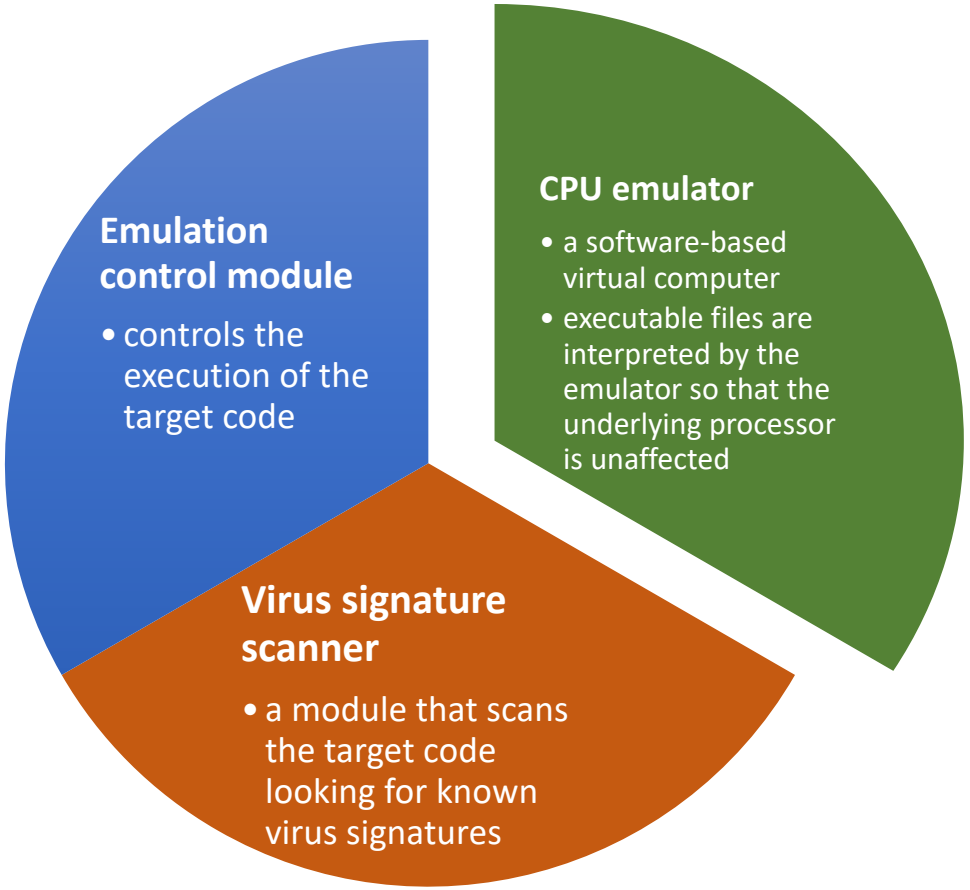
Generic Decryption (GD)

- Enables the antivirus program to easily detect even the most complex polymorphic viruses while maintaining fast scanning speeds
- When a file containing a polymorphic virus is executed, the virus must decrypt itself to activate
- Executable files are run through a GD scanner

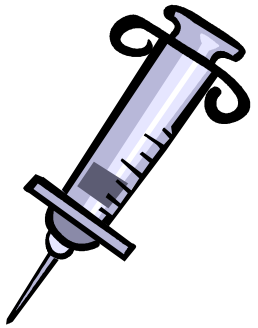
GD Scanner

- Most difficult design issue with a GD scanner is to determine how long to run each interpretation

GD scanner contains:



Digital Immune System



- A comprehensive approach to virus protection developed by IBM and refined by Symantec
- Motivation for development has been the rising threat of Internet-based virus propagation
- Two major trends in Internet technology have had an increasing impact on the rate of virus propagation in recent years:
  - integrated mail systems
  - mobile-program systems (i.e., *moving* cross-platform malicious code)
- Objective of the system is to provide rapid response time so that viruses can be stamped out almost as soon as they are introduced

Digital Immune System

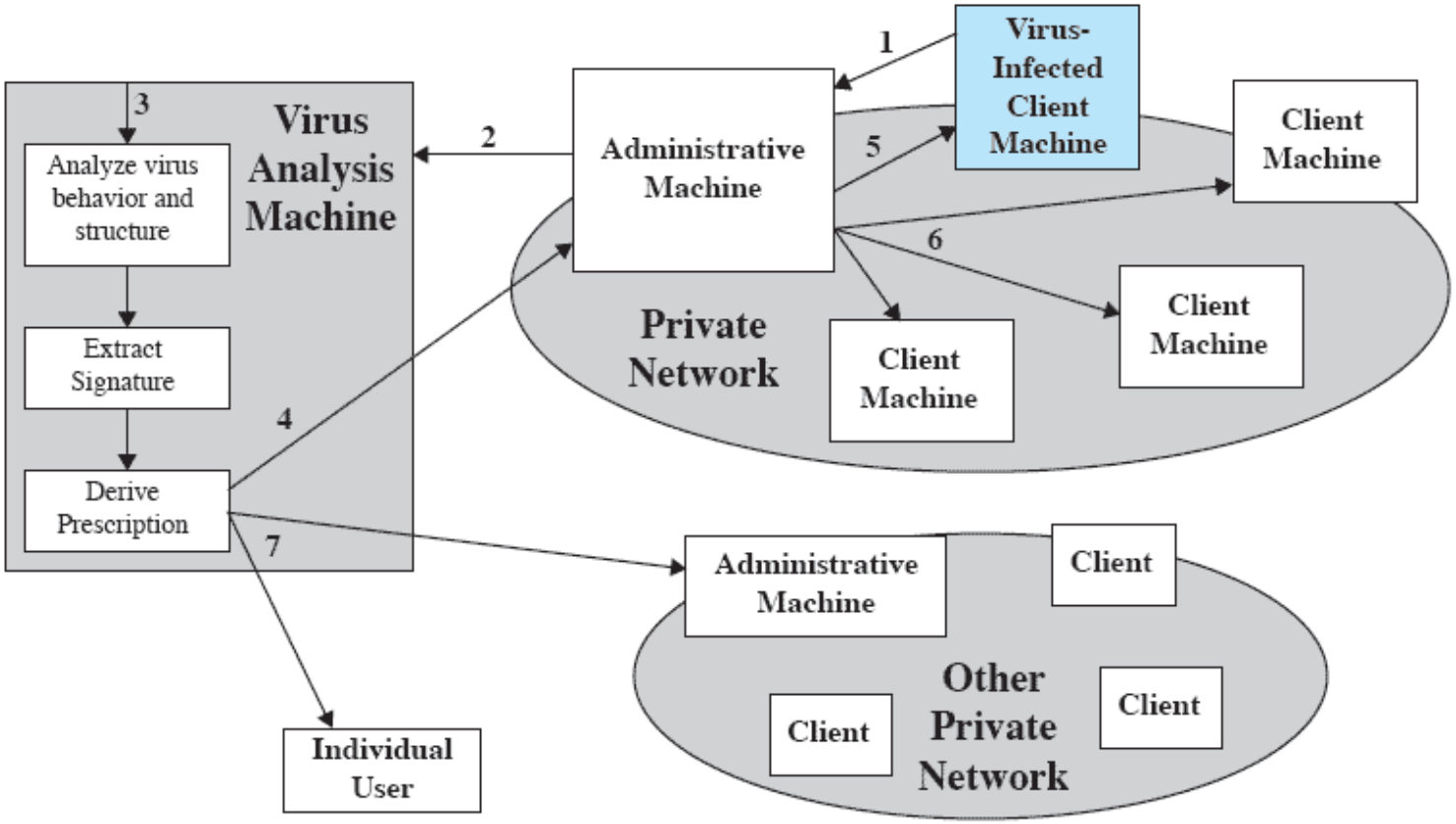
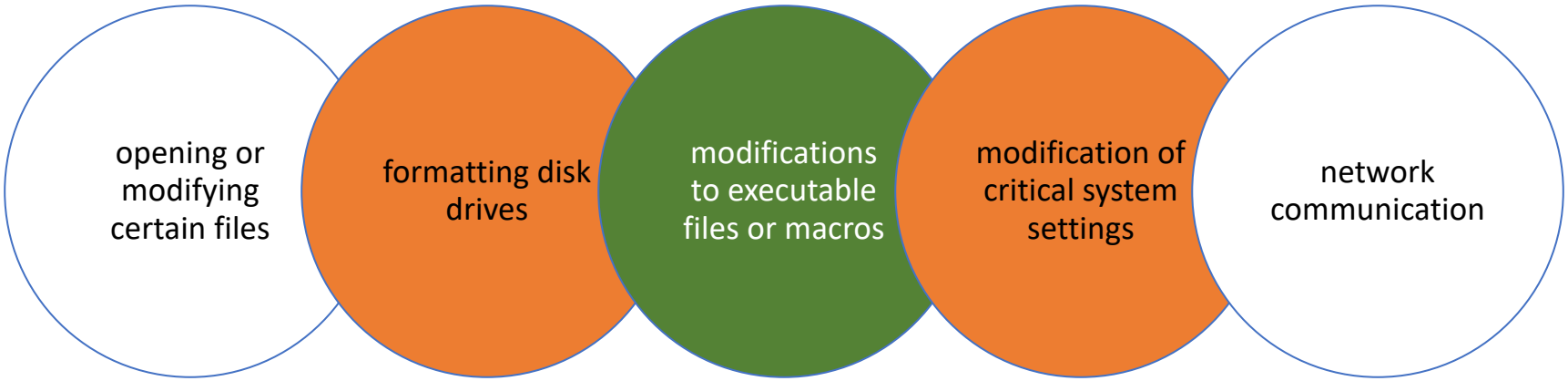


Figure 15.9 Digital Immune System

Behavior Blocking Software

- Integrates with the operating system and monitors program behavior in real time for malicious actions
- Monitored behaviors may include:



Behavior-Blocking Software Operation

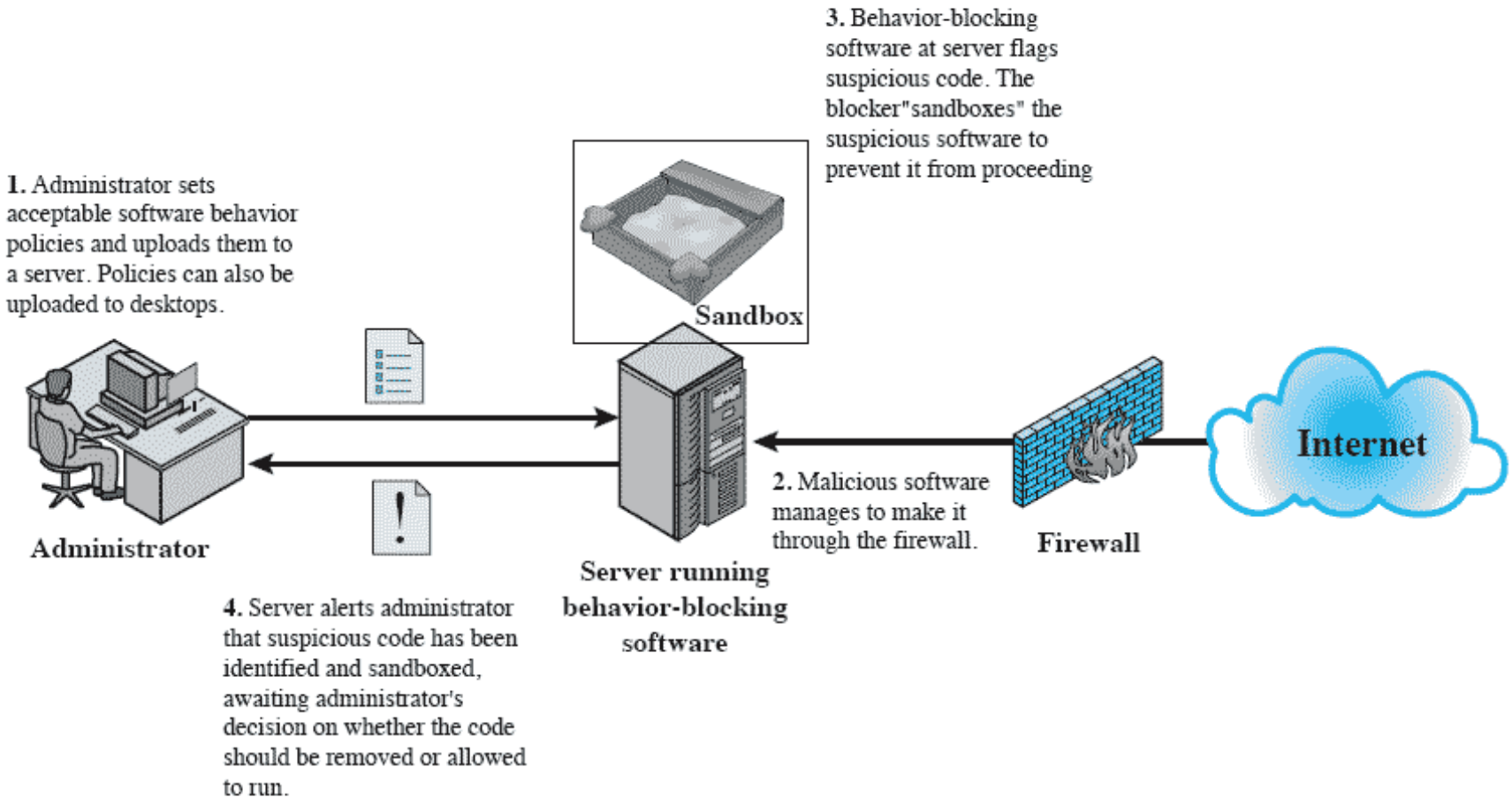
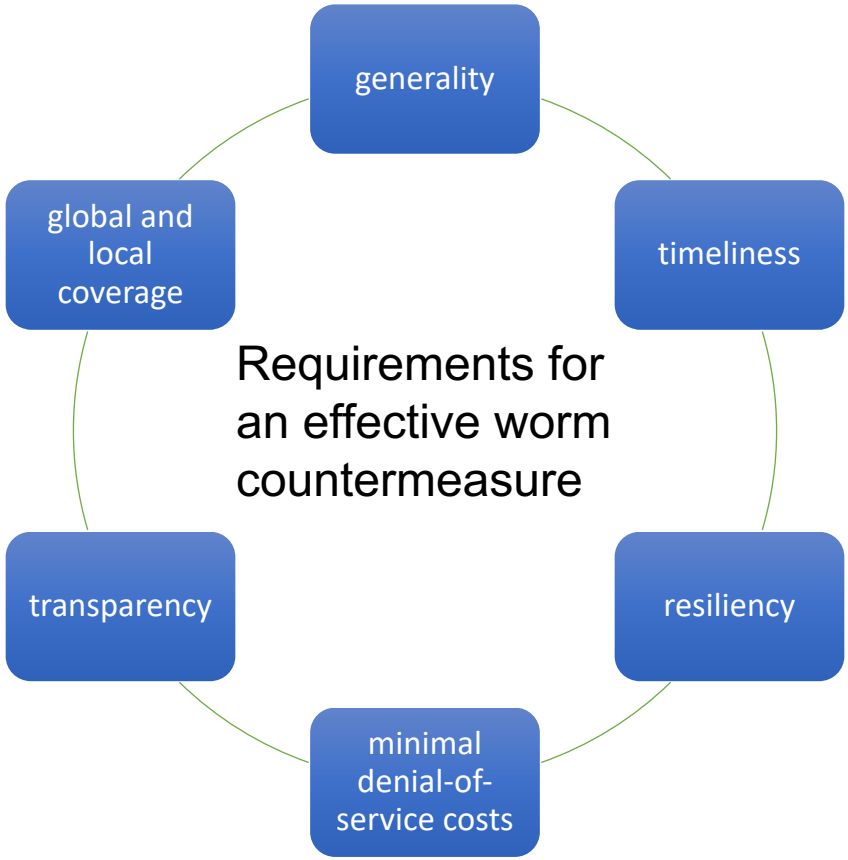


Figure 15.10 Behavior-Blocking Software Operation

Worm Countermeasures

- Once a worm is resident on a machine, antivirus software can be used to detect it
- Worm propagation generates considerable network activity so activity and usage monitoring can be the basis of a defense



Classes of Worm Defense

A. Signature-based worm scan filtering

- generates a worm signature which is then used to prevent worm scans from entering/leaving a network/host

B. Filter-based worm containment

- similar to class A but focuses on worm content rather than a scan signature

C. Payload-classification-based worm containment

- network-based techniques examine packets to see if they contain a worm

D. Threshold random walk (TRW) scan detection

- exploits randomness in picking destinations to connect to as a way of detecting if a scanner is in operation

E. Rate limiting

- limits the rate of scanlike traffic from an infected host

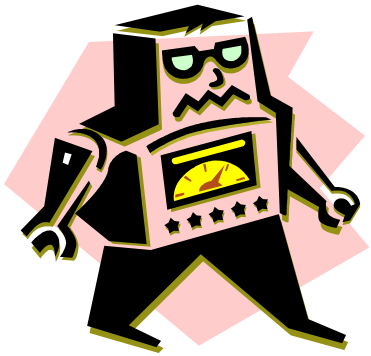
F. Rate halting

- immediately blocks outgoing traffic when a threshold is exceeded either in outgoing connection rate or diversity of connection attempts

Botnet and Rootkit Countermeasures

Botnet

- IDS and digital immune systems are useful against bots
    - once bots are activated and an attack is underway these countermeasures can be used to detect the attack
- The primary objective is to try to detect and disable the botnet during its construction phase



Rootkit

- Can be difficult to detect and neutralize
- Many of the administrative tools can be compromised
- Countering rootkits requires a varitey of network and computer level security tools
- Network-based and host-based intrusion detection systems can look for the code signatures of known rootkit attacks in incoming traffic
- Host based anitvirus software can also be used to recognize the known signatures
- Do some sort of integrity check

Summary

- Authentication
  - password based
  - token based
  - biometric
- Access control
  - discretionary
  - role-based
- Intrusion detection
  - host-based
  - audit records
- Malware defense
  - antivirus approaches
  - worm countermeasures
  - bot countermeasures
  - rootkit countermeasures
- Buffer overflow attacks
  - compile-time defenses
  - real-time defenses
- Windows 7 security
  - access control scheme
  - access token
  - security descriptors