System Security

- The Security Problem
- Authentication
- Program Threats
- System Threats
- Securing Systems
- Intrusion Detection
- Encryption
- Windows NT

The Security Problem

- Security must consider external environment of the system, and protect it from:
  - unauthorized access.
  - malicious modification or destruction
  - accidental introduction of inconsistency.
  - These are management, rather than system, problems.
- Easier to protect against accidental than malicious misuse.
- We say that the system is secure if its resources are used and accessed as intended under all circumstances.

The Security Problem…

- Unfortunately total security can not be achieved.
- It is easier to protect against accidental misuse than malicious misuse.
- Intruder or cracker: attempt to breach the security
- Threat: potential of security violation such as discovery of vulnerability
- Attack: an attempt to break security
- Form of malicious access
  - Breach of confidentiality: Unauthorized reading of data (theft of information)
  - Breach of integrity: Unauthorized modification of data
  - Breach of availability: Unauthorized destruction of data
  - Theft of service: Unauthorized use of resources.
  - Denial of service: Preventing legitimate use of the system (denial of service)
- Absolute protection against malicious use is not possible, however cost of perpetrator can be sufficiently high to deter unauthorized attempts.

The Security Problem…

- Four levels of security measures must be taken.
  - Physical
    - ✓ Against armed or surreptitious entry by intruders.
  - Human
    - ✓ Careful screening of users to reduce the chance of unauthorized access.
  - Network
    - ✓ No one should intercept the data on the network.
  - Operating system
    - ✓ The system must protect itself from accidental or purposeful security beaches.
- A weakness at a high level of security allows circumvention of low-level measures.
- In the remainder of this chapter we discuss security at OS level

Security measures at OS level

- User authentication
  - Verifying the user's authentication
- Program threats
  - Misuse of programs unexpected misuse of programs.
- System threats
  - Worms and viruses
- Intrusion detection
  - Detect attempted intrusions or successful intrusions and initiate appropriate responses to the intrusions.
- Cryptography
  - Ensuring protection of data over network

Authentication

- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities.
- Easy to understand and use
- However,
  - they can be easily guessed, illegally transferred
  - Visual or electronic monitoring, network sniffing
- Passwords must be kept secret.
  - Frequent change of passwords.
  - Use of “non-guessable” passwords.
  - Log all invalid access attempts.
- Passwords may also either be encrypted or allowed to be used only once.
  - For given a value of x, f(x) is calculated and stored. It is difficult to guess x by seeing f(x).
- One time passwords
  - Password is different for each session.
- Example: My mother’s name is K…
  - Password “MmnisK”

Authentication…

- Biometrics
  - Palm and hand readers: temperature map, finger length, finger width and line patterns.
  - Fingerprint readers.
- Two factor authentication scheme
  - Password plus fingerprint scan

Program Threats

- Trojan Horse
  - Code segment that misuses its environment.
  - Exploits mechanisms for allowing programs written by users to be executed by other users.
    - ✓ Consider the use of "." character in a search path. The "." tells the shell to include the current directory. If the user sets current directory to a friends directory, the program runs in users domain and effects friends directory.
- Trap Door
  - The designer of the code might leave a hole in the software that only she is capable of using.
  - Specific user identifier or password that circumvents normal security procedures.
  - Could be included in a compiler.
- Logic Bomb
  - Security threat only under certain circumstances.
- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers.)

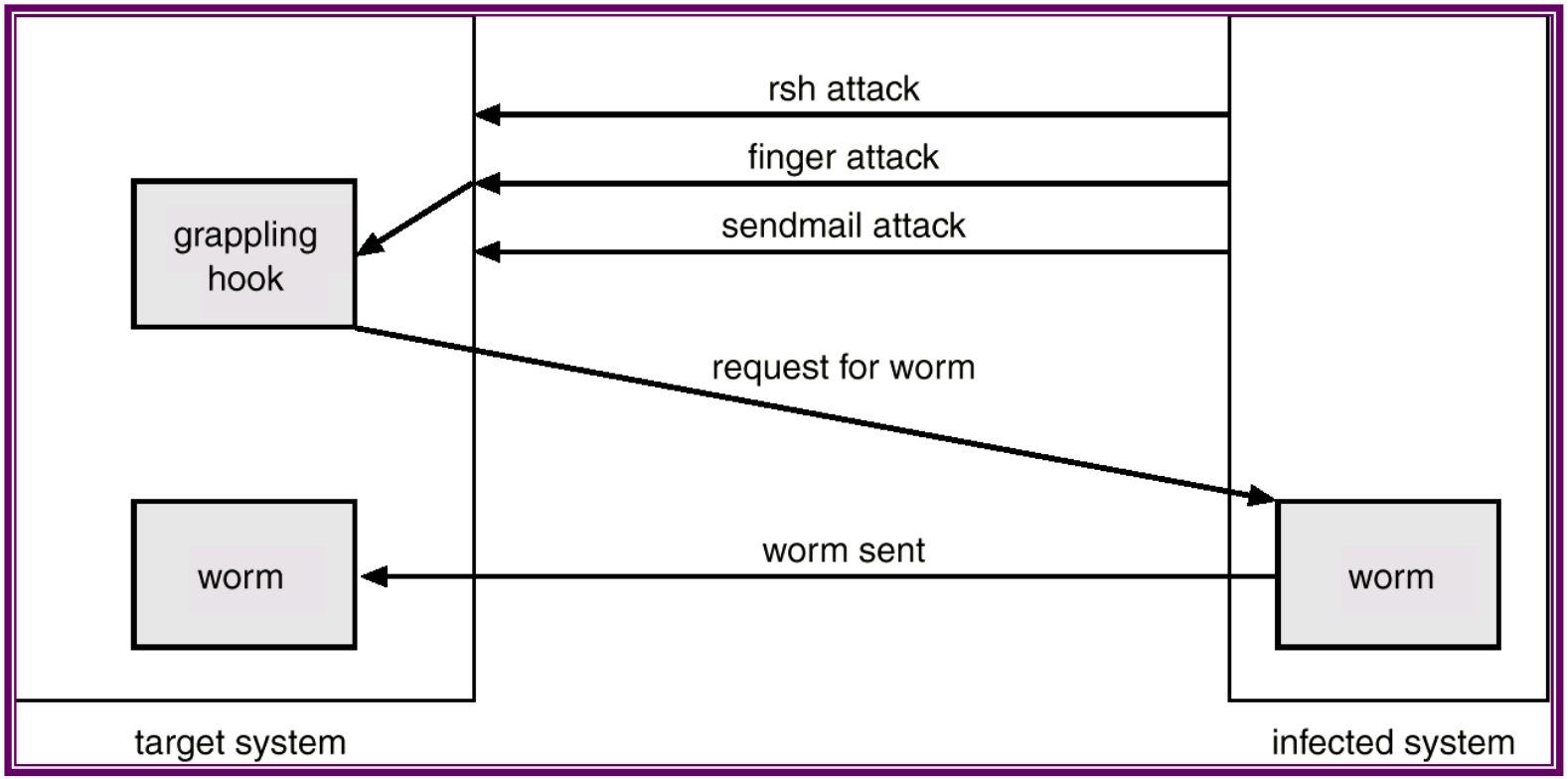
Program Threats…

- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers.)
- The attacker determines the vulnerability and writes a program to do the following.
  - Overflow an input-field, command-line argument, or input buffer until it writes into the stack.
  - Overwrite the current return address on the stack with the address of the exploit code in the next step.
  - Write a simple set of code for the next space in the stack that includes commands that the attacker wishes to execute, for example, spwan a shell.

System Threats

- Worms – use spawn mechanism; standalone program
  - The worm spawns copies of itself, using up systems resources and perhaps locking out system use by all other processes.
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs.
  - Grappling hook program uploaded main worm program.
- Viruses – fragment of code embedded in a legitimate program.
  - Mainly effect microcomputer systems.
  - Downloading viral programs from public bulletin boards or exchanging floppy disks containing an infection.
  - MSWORD (micros)
    - ✓ RTF format is safe
  - *Safe computing*.
- Denial of Service
  - Overload the targeted computer preventing it from doing any useful work.
  - Downloading of a page.
  - Partially started TCP/IP sessions could eat up all resources.
  - Difficult to prevent denial of service attacks.

The Morris Internet Worm



Threat Monitoring

- Check for suspicious patterns of activity – i.e., several incorrect password attempts may signal password guessing.
- Audit log – records the time, user, and type of all accesses to an object; useful for recovery from a violation and developing better security measures.
- Scan the system periodically for security holes; done when the computer is relatively unused.

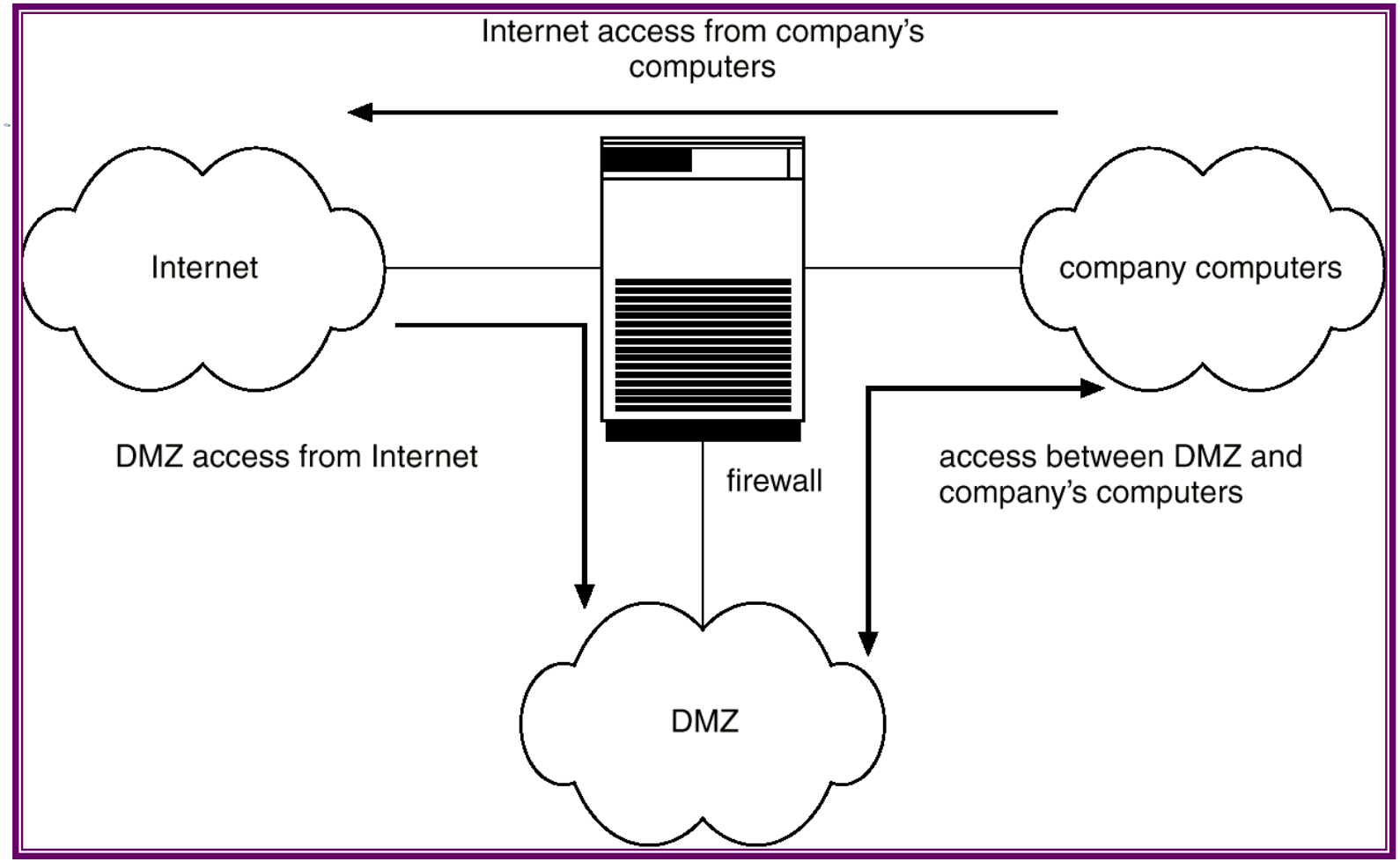
Threat Monitoring (Cont.)

- Check for:
  - Short or easy-to-guess passwords
  - Unauthorized set-uid programs
  - Unauthorized programs in system directories
  - Unexpected long-running processes
  - Improper directory protections
  - Improper protections on system data files
  - Dangerous entries in the program search path (Trojan horse)
  - Changes to system programs: monitor checksum values

FireWall

- A firewall is placed between trusted and untrusted hosts.
  - A firewall is a computer or router that sits between trusted and untrusted systems. It monitors and logs all connections.
- The firewall limits network access between these two security domains.
- **Spoofing**: An unauthorized host pretends to be an authorized host by meeting some authorization criterion.

Network Security Through Domain Separation Via Firewall



DMZ: Demilitarized zone

Intrusion Detection

- Detect attempts to intrude into computer systems.
- Wide variety of techniques
  - The time of detection
  - The type of inputs examined to detect intrusion activity
  - The range of response capabilities.
    - ✓ Alerting the administrator, killing the intrusion process, false resource is exposed to the attacker (but the resource appears to be real to the attacker) to gain more information about the attacker.
- The solutions are known as intrusion detection systems.
- Detection methods:
  - Auditing and logging.
    - ✓ Install logging tool and analyze the external accesses.
  - Tripwire (UNIX software that checks if certain files and directories have been altered – I.e. password files)
    - ✓ Integrity checking tool for UNIX.
    - ✓ It operates on the premise that a large class of intrusions results in anomalous modification of system directories and files.
    - ✓ It first enumerates the directories and files to be monitored for changes and deletions or additions. Later it checks for modifications by comparing signatures.
- System call monitoring
  - Detects when a process is deviating from expected system call behavior.

Data Structure Derived From System-Call Sequence

Open, read,mmap, mmap, open, getrlimit,mmap,close

system call	distance = 1	distance = 2	distance = 3
open	read getrlimit	mmap	mmap close
read	mmap	mmap	open
mmap	mmap open close	open getrlimit	getrlimit mmap
getrlimit	mmap	close	
close			

Open, read,mmap, open, open, getrlimit,mmap,close

Cryptography

- Eliminate the need to trust the network.
- Cryptography enables a recipient of a message to verify that the message was created by some computer possessing a certain key.
- Keys are designed to be computationally infeasible to derive from the messages.

Encryption

- Encrypt clear text into cipher text.
- Properties of good encryption technique:
  - Relatively simple for authorized users to incrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of the algorithm but on a parameter of the algorithm called the encryption key.
  - Extremely difficult for an intruder to determine the encryption key.
- *Data Encryption Standard* substitutes characters and rearranges their order on the basis of an encryption key provided to authorized users via a secure mechanism. Scheme only as secure as the mechanism.
- RSA : public/private key algorithm is popular