CPS221 Lecture: Introduction to Security

last revised 11/17/14

Objectives

1. To introduce the notions of confidentiality, integrity, and availability of data
2. To discuss issues specific to operating systems, networks, and database systems

Materials:

1. Projectable of "Security Triad" from Stallings *Computer Security*
2. Projectable of attack costs table from Stallings
3. Projectable of security principles from Hailperin pp. 399-400
4. Ability to show system root certificates on laptop
5. Ken Thompson article "Reflections on Trusting Trust"
6. Projectable of Kerberos Authentication (Casad figure 23.9)
7. Projectable of Figure 7.9 from Stallings
8. Projectable of Stallings Table 7.1
9. Projectable of two IPSec modes (Forouzan fig. 30.5)
10. Ability to login to mysql on laptop and on joshua
11. Projectable of mysql privileges from p. 914-915 of manual
12. View example from Intro to DBMS lecture, including projectable of code for creating view and grant statement

I. Introduction

A. The topic of computer system security is, of course, a huge one. Our goal is just to look at a few key concepts:

1. Goals of security and kinds of threats and attacks
2. Authentication
3. Some issues specific to operating systems (Host security)
4. Some issues specific to networks (Network security)
5. Some issues specific to database systems (Information security)

B. We cover the topic here in this course because some of the fundamental issues are cross-cutting ones that affect operating systems, networks, and database systems. (In fact, this is one place where there is significant overlap between full courses in these three areas.)

II. **Security Goals**

A. The NIST (National Institute of Standards and Technology) Security Handbook definition of computer security:

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

B. In general, security is concerned with preserving the confidentiality, integrity, and availability of information in the face of attacks that are often malicious or at least intended to subvert safeguards.

PROJECT: Security Triad

1. Confidentiality has to do with two issues:

   a. Data confidentiality: Ensuring that information can only be <u>seen</u> by those who are authorized to do so.

      Example: a bank will need to control who may have access to information about a customer's accounts

   b. Privacy: Ensuring that individuals have appropriate control or influence over what information about them can be collected and stored and how that information is disclosed to others.

      Example: Many organizations are required by law to have stated privacy policies

2. Integrity has to do with ensuring that information can only be changed by those who are authorized to do so.

   Example: a bank will need to control who may change the balance in a customer's accounts

3. Availability has to do with ensuring that information can be accessed when a legitimate user needs to access it.

   Example: a bank will need to ensure that information about a customer's account is available to be seen and or modified whenever that customer performs a transaction.

4. In designing a system, there are often tradeoffs between availability and the other goals - e.g. one could build a system that provided very high privacy if it had no network connection, but that might well stand in the way of availability to users who were not in the same place as the system.

C. Sometime writers add one or more additional goals:

1. Authentication has to do with ensuring that both parties in the communication are, in fact, who they claim to be.

2. Auditability has to do with being sure that proper functioning of the security mechanisms can be verified.

3. Accountability has to do with ensuring that if an invalid operation is performed, it will be possible to trace it to the perpetrator (which has a deterrent effect).

D. Another way to look at what we're trying to guarantee is called the Operational Model of Security: Protection = Prevention + (Detection + Response).

1. We want to prevent things that violate CIA from happening.

2. But if they do occur we want to detect that this has occurred and make an appropriate response to minimize negative impact.

III. **The Nature of Threats**

Security is concerned with dealing with a variety of different kinds of threat to a computer system.

A. Unauthorized disclosure - a threat to information confidentiality

1. Exposure of confidential information to those who shouldn't have access to it.

   a. Can be the result of a deliberate action by an insider

      Example: An employee of a restaurant stealing credit card numbers of customers

   b. Can be the result of carelessness

      1. Careless use of file protection mechanisms - e.g. on most Unix-like systems the default protection setting for a file is "readable by anyone"

      2. Theft or careless disposal of media (or computers containing media) containing confidential information

         Some examples (from (Stallings, 2008) p. 67)

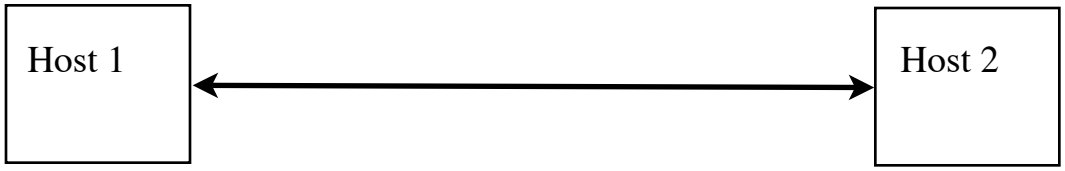
         a. "In December of 2004, Bank of America employees backed up and sent to its backup center data tapes containing the names, addresses, bank account numbers, and Social Security numbers of 1.2 million government employees enrolled in a chargecard account. None of the data were encrypted. The tapes never arrived and indeed have never been found ..."

         b. "In April of 2005, San Jose Medical group announced that someone had physically stolen one of its computers and potentially gained access to 185,000 unencrypted patient records."

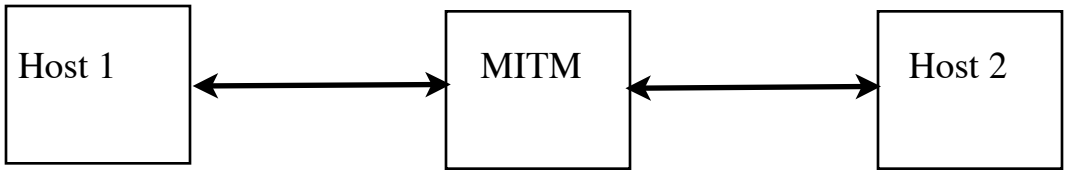
2. Interception of network traffic (“sniffing” - recall the Wireshark lab)

3. "Man-in-the-middle": a system interposes itself between two hosts - e.g.

   What the hosts think is happening



   What is actually happening



   (i.e. the MITM impersonates Host 1 to Host 2 and vice versa, thus being able to see an potentially manipulate all information passing between them)

   a. This can happen even if encryption is being if the MITM can interpose itself before the hosts exchange public keys.

   b. This can be prevented if the hosts can independently verify each other - e.g. by the use of certificates.

4. Inference:

   a. Network traffic analysis - even if an attacker cannot learn the content of a user’s network traffic (due to encryption), the attacker may be able to learn something from the pattern of sites the user visits or other people the user communicates with.

   b. Inferences from statistical databases

      Example: Suppose a system includes a facility for looking up the

average salary for a group of people. Is it possible to find out the salary of a single individual? Yes - if one an obtain an average from two different groups, identical except for including or not including this one person.

5. Intrusion by an attacker

    a. Exploiting flaws in mechanisms intended to control access to information

    b. Activities such as breaking and entering

6. Covert channels - mechanisms for transferring information from higher classification compartment to a lower classification compartment without going through the normal (presumably secured) channels.

    Example: A high classification program may be altered to transmit information by using the existence or non-existence of certain files to transmit information to another program a bit at a time. (The contents of the file may be secured - but information is transmitted by the mere fact of its existence or non-existence.)

B. Deception: a threat to confidentiality and/or integrity

1. Masquerading:

    a. An attacker can learn the password of a user and then can use this to gain access to or modify information the user has legitimate access to:

    b. Use of a Trojan horse

    A Trojan horse is a program that performs a useful function but also performs a malicious one - e.g. a game program that spreads a virus.

2. Falsification: Unauthorized modification of a file or a database table.

    a. Misuse of authorized access by an insider

    b. Careless use of protection mechanisms

3. Man-in-the-middle attacks can also alter information.

4. Replaying a captured message to cause the action specified to be done again.

    Example: An attacker might obtain a copy of a message authorizing an electronic funds transfer to an account belonging to the attacker. A replay attack would involve the attacker inserting a second copy of the message into the network, with the result that the EFT operation is done twice.

5. Repudiation, where the originator of a message denies having sent it, or the recipient of a message denies having received it.

C. Threats to availability:

1. Incapacitation

    a. Physical attack on hardware or communication links

    b. Malicious software

2. Obstruction - e.g. denial of service

D. Misappropriation of system resources

E. Attacks of various sorts can actually be quite costly

PROJECT Table of Attack costs from Stallings p. 31

IV. Motivation for Attacks

Why would someone want to attack a computer system?

ASK

A. Proving one's prowess.

B. Malice.

C. Desire to gain financial advantage by:

1. Stealing money (perhaps via a stolen credit card)

2. Holding resources hostage for ransom (e.g. encrypting all the files on a hard drive and then demanding a payment to decrypt)

D. Espionage (industrial or military)

E. Cyberwarfare.

V. Some Security Terminology

A. Terminology pertaining to attacks:

1. A vulnerability is a weakness in system security that potentially allows someone to attack the system. A vulnerability requires three elements to be present:

   a. A system flaw.

   b. An attacker must have access to the flaw. (One of the benefits of a layered approach to security is that a flaw at a lower layer will not be a vulnerability if a higher layer prevents attacker access to it

   c. An attacker must have the capability to exploit the flaw.

2. An exploit is some software or strategem (e.g. a sequence of commands or input that is entered at some point) that takes advantage of a vulnerability to actually attack the system.

   The method that is utilized by an exploit is called the attack vector (e.g. a buffer overflow vulnerability).

3. A risk is the potential for loss or damage that might result from the exploiting of some vulnerability. (Note that a vulnerability is not necessarily a risk - e.g. if exploiting it would not result in loss or damage.)

4. A previously-unknown vulnerability is known as a zero-day threat from the time it is first discovered until the time a patch is created (and applied) to deal with it.

B. A trusted system is one that is relied upon to enforce some security policy. Should this break down, the security of other systems relying on it may be compromised.

   The National Security Agency's "Orange Book" defines various "evaluation classes" specifying the requirements for different levels of assurance in the trustworthiness of a system.

VI.Building Secure Systems

A. Security mechanisms can be deployed to

   1. Prevent attacks

   2. Detect attacks, with a view to

      a. Recovery

      b. Holding the responsible party accountable

B. Many security mechanisms rely on four basic ideas:

1. Encryption

2. Authentication of users

3. Control of physical access to systems

4. Logging of security-related events, to facilitate recovery (if a suspicious event is discovered) and/or accountability

5. We will discuss encryption in the next lecture. In this lecture we will discuss authentication and say a little about the physical access, but will not discuss logging further.

6. We will also talk about some of the applications of these ideas to security in operating systems, networks, and database systems.

C. Although technical measures to ensure security are vital, another issue that needs to be considered is the phenomenon of social engineering. This broad term encompasses numerous different psychological tricks that exploit human weaknesses to accomplish an attacker's goals. Examples?

ASK

1. Phishing

2. Vishing (= voice phishing)

3. Tailgating

4. Virus hoaxes (the program advertised as removing a certain virus actually installs malware itself!)

   (A common example of this on MacIntosh's is a program called MacDefender)

5. Baiting

D. Hailperin discusses a number of basic principles in his book

PROJECT and walk through

1. Some of these principles are known by other names. For example, the custom book uses the following names for some of the principles mentioned by Hailperin:

   a. Economy of mechanism - Keep it simple.

   b. Fail-safe (and fail-noisy) defaults - Implicit deny

   c. Open design is contrasted with the undesirable approach known as security through obscurity.

   d. Least privilege - a user should have only the set of privileges that are necessary to perform his or her duties.

      (Contrast this with the "everything" privileges associated with being root on a Unix system.)

   e. Separation of duties - more than one person should be required to perform certain critical tasks - reducing the possibility that rogue individual might be able to do something improper on his or her own.

      A familiar example (from outside the computer world) is the requirement that most companies have checks for large amounts require two signatures.

   f. Defense in depth - layered security.

g. Complete mediation - every access to information should be checked, instead of relying on a check performed some time in the past.

2. The custom book also mentions two principles not in Hailperin:

   a. Job rotation

   b. Diversity of defense

3. Three other principles not mentioned in either book:

   a. End-to-end security.states that responsibility for security (e.g. encryption of data) should reside with the end-points of the communication (e.g. in the application layer of two hosts communicating over a network) rather than in intermediate nodes.

   b. Security by design states that a system should be designed to be secure from the ground up, rather than trying to layer security onto a system after the fact.

   c. Use of vetted (thorougly tested through use) components for security-sensitive functions, rather than developing them from scratch.

VII. **Authentication**

A. There are many cases in which it is important to be sure that the person at the other end of a communication is who he/she claims to be (example: the many situations in which a password is utilized.)

B. In general, there are four ways a person might be able to prove he/she is who he/she claims to be.

   1. Something the user <u>knows</u>

a. A password

b. Personal information such as mother's maiden name, place of birth, etc.

c. An algorithm

2. Something the user has

   e.g. an ATM card - used in conjunction with a PIN to ensure that it has not been lost or stolen

3. Something the user is

   e.g. static biometrics such as fingerprints, iris scans

4. Something the user does

   e.g. dynamic biometrics such as voiceprints or handwriting characteristics (not just the appearance, but speed and pressure)

C. Passwords are the most commonly used form of authentication.

1. Because of this, they are a particular focus of attack by crackers. (Cracking is the term commonly used for a person who attempts to learn a password illicitly). What are some ways a cracker might attempt to learn a password?

   ASK

   a. If passwords are short, sheer trial-and-error may work

      i. This possibility is what lies behind requirements such as minimum password length. Note that the effort to discover a password by trial and error goes up exponentially with the length.

      ii. This is also the rationale for requiring various kinds of characters. (If the password includes only lowercase letters, the effort goes

up as 26^{length}, but if the password is required to include both cases as well as digits, it goes up much faster as 62^{length}.

iii. As a further protection, many systems inject some delay after each unsuccessful password attempt, or terminate attempts after several unsuccessful tries - both of which are aimed at automated attempts at guessing by trial and error.

b. Exploiting user mistakes (e.g. writing a password down)

c. Popular password attack (trying frequently used passwords - e.g. see http://www.cbsnews.com/news/the-25-most-common-passwords-of-2013/ but there are also much longer lists.

d. Researching a particular user in an attempt to guess likely passwords. (For example, shortly before the 2008 election, the private email account of Vice-Presidential candidate Sarah Palin was hacked. This appears to be the way this was done.)

e. Running a Trojan horse login program. (Such a program is left running on a computer system (generally a publicly accessible one) and displays the normal username and prompts, but then captures what an unsuspecting user types and emails it to the cracker.)

2. A particular vulnerability arises if a cracker can gain access to the file containing the encrypted passwords, because then the cracker can run an offline dictionary attack, encrypting multitudinous possibilities to see if any matches an entry in the file.

a. This is possible because - by design - there is nothing secret about the hashing algorithm used for encrypting passwords. (There does not need to be, since it is one-way.)

b. Historically, on Unix the file that was originally used to store the passwords (/etc/passwd) is readable to any user logged on to the system.

1. This was necessary because this file also stored information about users that needed to be publicly available (e.g. historically their office and phone numbers - though modern versions store this elsewhere)

2. This was felt to be safe because passwords were stored in encrypted form.

3. However, this became a serious vulnerability, because hackers figured out that they could try to encrypt password guesses (such as "password" or the user's name) and compare the encrypted result to what was stored in the password file, without having to actually try to log in using the guessed password (a more time-consuming process, and one that could result in suspicious entries in system logs)

c. Modern Unix-like systems - and most other systems - store encrypted passwords in a file that is not publicly readable for just this reason.

   In the case of Linux, this is /etc/shadow. (The password field in ./etc/passwd is blank - the password isn't stored there anymore)

   DEMO: try to access /etc/shadow as an ordinary user, then using sudo

d. Since a cracker may find a way to gain access to the password file on any system (not just Unix) two other techniques are used to make offline dictionary attacks more difficult.

   1. The use of password "salt".

      a. On many systems, instead of encrypting just the user's password, the system encrypts the password combined with a random "salt" value. This value is also stored in the password table, and when a user attempts to log in, the password the user supplies is combined with this value for encryption.

b. What this does is to require a cracker to try a password combined with every possible salt value to see if it matches any entry in the password table - e.g. if a 12 bit salt is used, a cracker must try all 4096 possible combinations with something like "secret" or "password" to see if any matches an entry in the password table.

2. Using a deliberately slow encryption algorithm. This poses little burden when a legitimate user logs in, but makes the job of testing possible passwords for matches much more time-consuming.

D. Of course, the need to authenticate is often two-way. For example, when doing electronic banking, it is important to know that the site to which you give your password is really your bank, and not some impostor site whose creator will then use the password you supply to get funds from your account.

(This is otherwise a danger because a router along the way may be hijacked by an intruder to redirect traffic destined for the bank to the intruder instead.)

1. To facilitate this, a genuine site will use a certificate specifying its identity and public key.

2. The certificate is digitally signed by a certificate authority (CA) whose identify and public key are known to the client. The client can therefore verify the integrity of the certificate and hence the identity of the server and validity of the public key.

3. Of course, this depends on the client trusting the certificate authority. (The CA serves as a trusted third party, whom the client trusts to have verified the server's identity and public key before issuing a signed certificate.) This may be as a result of the CA being one of the root CA's known to the user (or more likely the user's browser); or the CA may itself be verified by another CA known to the user.

SHOW System root Certificates on my Mac (Utilities / Keychain Access - go through and note different kinds)

E. One commonly used authentication system is the Kerberos system, which was first developed as part of Project Athena at MIT.

PROJECT: Kerberos Authentication

VIII. Access Control

A. While authentication is concerned with a person (or system) establishing its identity (who it is), access control has to do with what a person can do once identity is established.

B. Another term with similar meaning is "authorization". (For example, the documentation for db2 uses terms like "authorization ID" to describe the level of access needed to perform various functions.)

C. There are two general approaches to access controls:

1. The discretionary access control approach (often abbreviated DAC) allows the owner of a piece of information to control who has access to it.

That, for example, is the way that file access is managed on most workstations - the owner of a file is allowed to set various bits to control access to the file, or to manage an access control list (ACL) that governs this.

2. The mandatory access control approach (often abbreviated MAC) makes access control a matter of organizational policy.

a. This becomes especially important when the system stores sensitive information that only certain users are allowed to see - e.g. the military has a three-level system of classified, secret, and top secret;

many companies use classifications like "for internal access only" or "proprietary".

b. Under MAC, who may access various kinds of information is controlled by organizational policy, rather than by the creator/owner of the information.

    i) Users are classified by the access permitted - e.g. a military user may not be allowed to see any classified information, or may be allowed to see only classified information at the classified level, or may be allowed to access secret level (which also allows classified, of course.)

    ii) Frequently information is also classified into compartments -

        a) So, for example, a document dealing with use of biological weapons by terrorists might be assigned to the "biological weapon" compartment and to the "terrorim" compartment. (A given piece of information can be assigned to multiple compartments.)

        b) The access level assigned to a user would then also specify compartments - e.g. if a user is be allowed to see top secret (and hence secret) information in the "nuclear" and "terrorism" compartments could see this document - but a user allowed to access top secret documents in the "nuclear" and "weapons" compartments would not.

c. One form of MAC policy that is often used is called the Bell-LaPadula model. This model is specifically concerned with preserving information confidentiality rather than integrity.

    i. Subjects and objects both have confidentiality levels assigned.

ii. A subject is allowed to see objects at its own level and below (e.g. secret can see secret, confidential, and public). This is called "read-down".

iii. However, a subject can only write objects at its own level or higher - hence a "secret" subject can write to secret and top secret objects - but not to public or confidential ones.

Why this seemingly strange requirement?

ASK

i. To prevent intentional or accidental leaks of information to a lower classification level.

ii. Recall that this model is concerned with confidentiality but not integrity - e.g. a subject at the "secret" level could alter or delete a top-secret object (assuming he/she knew the name)

IX. Issues Specific to Operating Systems

A. As you know, operating system security is typically based on user identity established via a password. Only a user that can properly authenticate as a known user is allowed to access the system in the first place, and what that user can do is determined by the identity.

B. Attempts to bypass operating systems, or to hijack the system, are often made at one of four points.

1. The password system - we have discussed this already

2. Attempting to get a legitimate user to run a malicious program. (Trojan horses).

a. In earlier days, viruses were often spread by an infected program that was shared with other users - e.g. a game.

b. Today, Trojan horses may be embedded in programs downloaded from the web, or as macros embedded in documents sent in email attachments.

c. When a user runs a Trojan horse, the malicious code in the Trojan horse runs with the same access to the file system as the user running it.

i. Of course, this means that a Trojan horse run as the "root" user on a Unix-like system is a particular danger.

ii. For this reason, the preferred mode of operation on most such systems is for administrators to run as ordinary users, using sudo only when necessary. (This is what requests to Authenticate on systems like Ubuntu mean - the utility needs to use sudo and requires a password to do so.)

3. Backdoors left in legitimate software.

   A particularly insidious example of how this might be done is the approach discussed in Ken Thompson's article "Reflections on Trusting Trust"

   DISCUSS

4. Exploiting insecure code - we will discuss various kinds of code insecurities later when we discuss secure programming.

C. Of course, the "holy grail" of crackers is to obtain root level access to a system.

1. Once such access is obtained, a hacker may install a software package known as a rootkit, which modifies system-level behavior in such a way

as to disguise the intruder's presence on the system

Examples:

a. The code for the system's "list processes" API may be modified so that the hacker's processes are not reported

b. The system's directory listing API code may be modified so that the hacker's files are not reported

2. An example of how a rootkit might be installed

   PROJECT Stallings Figure 7.9

3. Detection and removal of rootkits is complicated by the fact that the very tools that would be used (examination of running processes and directory listings) have been rendered unreliable by the rootkit.

   One approach that is used is to access information directly - without going through system APIs and then compare this to the result obtained by going through the API's.

D. Physical security is also important, lest an attacker be able to bypass operating system protections

1. Example: an attacker who can boot his own operating system from a medium like a CD can generally access all the files on the computer's disk without regard to operating system-managed file protection.

   For this reason, Linux systems often include a boot loader program will only allow the computer to boot from specified devices. The boot loader is itself password protected.

2. Example: an attacker who can take a disk out of the case can access any file on it without regard to operating system-managed file protection.

   This is why if a stolen laptop contains confidential information that is not encrypted, that information can easily be accessed. (Fortunately,

though, those who steal laptops are often more interested in the value of the hardware itself, not the information on it!)

X. Issues specific to networks

A. We consider in this section only issues peculiar to two parties communicating via a network. We assume basic operating system security on both ends.

B. Malware

1. Malware is short for "malicious software". There are many different kinds of malware.

   PROJECT Stallings Table 7.1 - Go over types

2. There are many ways in which malware can be spread from one system to another.

   a. Deliberate penetration using password cracking.

   b. Deliberate penetration using buffer overflows.

   c. Trojan horse programs available on the web.

   d. Macro viruses in documents or spreadsheets sent by email. (This builds on the fact that some software - e.g. Microsoft Office - allows macros in documents that can access the user's files or perform other "risky" operations.)

   e. Macro viruses in the body of an email. (This builds on a mail program that allows executable code in an email)

3. One sort of malware that calls for particular note is malware that compromises a system in such a way that an attacker can later use it for other purposes, such as sending spam email or denial of service attacks.

Such a compromised system is called a “bot” (short for robot) and a network of such compromised systems is called a “botnet”.

C. Denial of Service Attacks

1. A denial of service attack attacks the availability of a system, rendering it unable to perform its intended function. Such an attack may be motivated by

   a. A desire to harm the “victim”, either due to malice or a desire to show off the attackers prowess

   b. A desire to gain a financial advantage of some sort - perhaps by extorting “protection money” from the target.

2. DOS attacks typically make use of the fact that it is possible to include a phony source address is an IP packet. IP contains no provision for verifying that a packet actually came from the system it claims to have come from.

3. One sort of DOS attack, known as flooding, involves sending so much traffic to a server that either its network connection or its ability to service requests is overloaded. The former may result in many legitimate access attempts being lost, as legitimate packets are dropped. The latter may result in response being unacceptably slow, to the point of even timing out.

   a. A flooding attack may be originated from a system that has a higher network bandwidth than the site being attacked. Of course, the source address in the packets is usually spoofed to prevent the source of the attack from being discovered.

   b. A flooding attack may be originated by sending packets to other servers on the network, but with the source address spoofed as being the target system. This causes these servers to respond to the target system, flooding it with traffic.

c. A distributed denial of service attack involves using a botnet to flood a target system with traffic. Though each individual bot may have low network bandwidth, the cumulative impact of a botnet may bring down even a high bandwidth server.

4. Another sort of DOS attack involves establishing numerous TCP connections to a server, thus overloading its internal connection table.

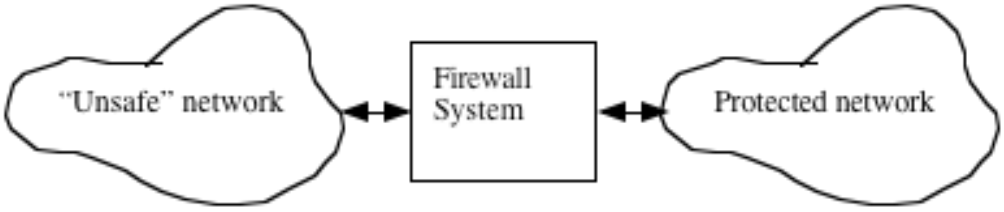
D. Preventative measures

1. Encryption of confidential information.

   It is always best to assume that anything that is transmitted over the Internet unencrypted can be read! While this may not be a concern with ordinary email or web browsing traffic, it is a significant concern with sensitive information or passwords.

2. Firewalls

   a. A firewall may be a separate, hardened computer system placed between a local area network and an unsafe network such as the Internet.



   b. Or, a firewall may be a component of the network stack on an individual computer.

   c. In either case, a firewall serves as a filter, either passing through or blocking packets according to rules that are specified as part of its configuration.

      SHOW: ssh to joshua as root, and do `iptables -L`

d. In a world in which there were no security loopholes in applications, a firewall would be unnecessary. But given the reality of applications having vulnerabilities, a firewall can serve to prevent undesired access.

   Example: suppose a database houses sensitive data, and it is desired to allow the database server to be accessed on a LAN, but disallow accesses from outside, lest an attacker find a vulnerability that allows to the database. This can be done by configuring the firewall to block connections to the database from the external network

3. IPSec

   a. IPSec (IP Security) is a collection of extensions to the IP protocol, which are optional in IPv4 but will be built into IPv6. In effect, IPSec adds a new layer to the network stack, either between the Transport layer and the Network layer, or just below the Network Layer.

   b. IPSec operates in two different modes. These two modes can be illustrated as follows:

      PROJECT: Two IPSec modes

      i. In transport mode, it provides protection (via encryption) for the payload of packets, but not for the IP headers - so an observer can know where a packet is coming from and where it is going - but can learn nothing about its actual content. This can be used for host-to-host security - presuming, of course, that the IP protocol stack on both hosts supports it.

      ii. In tunnel mode, IPSec encrypts an entire IP packet and places it in a new packet which is sent to a router that supports IPSec.

(a) The destination router decrypts the encapsulated packet and sends it on - typically onto a LAN.

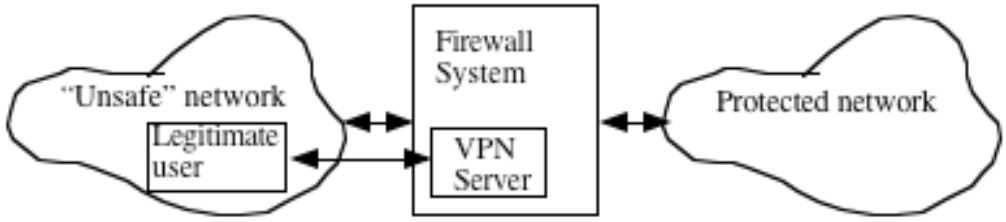
(b) The new packet records as its sender the system that created the new packet and the router that will unpack it as its destination; the original origin and destination of the encapsulated packet are encrypted and therefore not visible while the packet is traveling between the two routers.

4. Virtual Private Networks

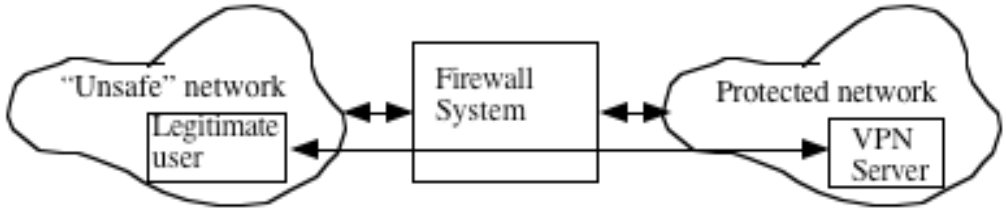
a. A firewall can create tensions between protection and legitimate use.

Example: Suppose, to protect a database, the firewall is configured to disallow external access. This would seem to also preclude accesses by legitimate users from home or while traveling.

b. One way to address these tensions is through the use of Virtual Private Network (VPN), which makes use of IPSec A VPN setup looks like this:



or



1. A legitimate external user first establishes a connection to the VPN, using his/her authentication credentials.

2. Then the legitimate user can use the VPN server as an intermediary to access resources on the protected network as an "insider".

    a. The legitimate user sends packet directed at a server on the inside to the VPN server. The actual IP-level packet (which includes its destination IP and port inside the protected network) are both encrypted and authenticated..

    b. The VPN server decrypts the authenticated packet from the legitimate user, and then places it on the internal network, on the other side of the firewall.

    c. The same approach is used for outbound packets going from systems "inside" the firewall to the user outside, to protect the security of information traveling over the Internet and to assure the user that the packets really come from the "inside" system.

        Example: I prepared most of this lecture at home. Gordon's firewall does not allow direct access to NAS1. However, I was able to access it using Gordon's VPN - which required me to authenticate with my Gordon password before allowing me to access the file server.

5. Sandboxes

    a. Running a program in a sandbox means running it in an environment where its capabilities are restricted to "safe" operations.

    For example, Java applets run in a web browser are run in a sandbox where the following operations - among others - are prohibited

    (1) Any access to the file systems on the computer on which the applet is running

(2) Any network connection to any system other than the one from which it was downloaded [which might lead to getting around a firewall]

b. Of course, there will be times when the legitimate functionality of an applet requires access outside the sandbox.

Example: Gordon's VPN is implemented by a Java applet that must, of course, be able to access the file system on the computer it is running on to support file transfers between a host outside Gordon's firewall and a file server inside it.

c. Java also supports the notion of a "signed applet" - an applet which is cryptographically signed by a trusted entity. Such an applet can be allowed access privileges outside the sandbox.

6. A network may also incorporate an Intrusion Detection System - which monitors activity, looking for suspicious activity - which may prevent penetration if an attack can be detected before it has succeeded, or at least can facilitate recovery following a successful attack (before the intruder can completely cover his tracks.)

VII. **Issues specific to database systems**

A. Database system security is built on top of operating system security. The file(s) in which the database is stored have protections set so that only the dbms can access them. (This is typically done by having the dbms run under a special user id used only for it, and having the database owned by this user)

B. The dbms, in turn, generally has its own user authentication mechanisms, so that logging on to the dbms is a separate and distinct step from logging on to the computer on which it is running.

1. In fact, most dbms's allow a user to log on to the database remotely without having the actually log onto the computer running it.

   DEMO: Log onto mysql on dbms from my computer

   ```
   mysql -h dbms.cs.gordon.edu -u cps221 -p
   ```

2. Some dbms's may allow a user who is logged in to the computer to access the database under the same username.

   ssh to dbms
   ```
   db2
   connect to cps221
   ```

3. The DBMS, in turn, controls access to individual tables or columns within tables on the basis of the identity established by a user. The precise details of the privilege structure varies a bit from dbms to dbms (though the SQL syntax is the same).

4. For example, mysql allows a database administrator to grant privileges to users on all databases, specific databases, specific tables within a database, or even specific columns within a table.

5. There are a wide variety of different privileges that can be granted - most of which are meaningful only at one of the levels just noted (e.g. the privilege to create a table is applicable only at the database level.)

   PROJECT: List of privileges from mysql manual pp. 914-915

   a. Note the ALL option

   b. Note that the privilege names are standard in SQL, though not all dbms's implement them (e.g. REFERENCES in mysql).

6. The SQL GRANT statement can be used to grant privileges to a specific user, or to all users (GRANT ... TO PUBLIC)

C. One important feature of SQL is the concept of a view. A view is a "virtual table" that - among other uses - can be used to limit what a particular user may see.

DEMO: use db2

```
connect to db2demos user bjork;
set schema registrar;
select * from course_taken;
update course_taken set grade = 'C'
    where id =  '5555555' and
          department = 'BCM' and
          course_number = '101';
select * from course_taken;

connect to db2demos user aardvark;
set schema registrar
select * from course_taken;
select * from student_info;
update course_taken set grade = 'A'
    where id = '1111111'
```

PROJECT code for creating student_info view.

1. A view is treated like a table by the DBMS.

2. In particular, it is possible to specify distinct user access rules for a view from the table it is based on.

   SHOW Grant used for view student_info

VIII.Ethical Issues

Because computer systems are custodians of private information about individuals, numerous ethical issues related to privacy and other matters arise.

DISCUSS AS CLASS