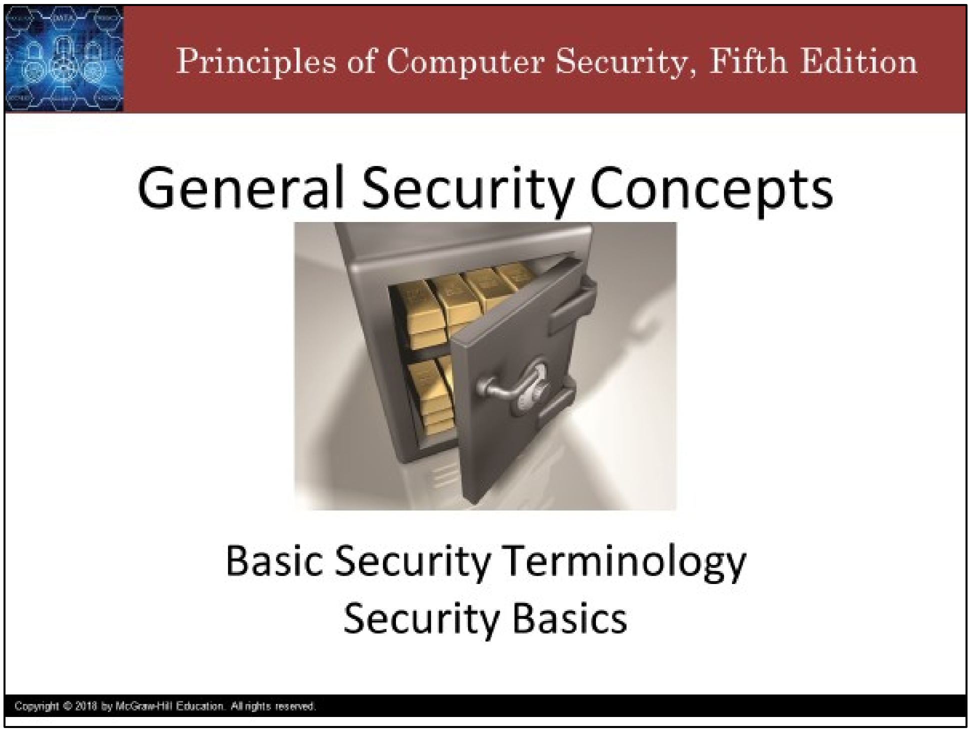
General Security Concepts: Security Basics

Slide 1



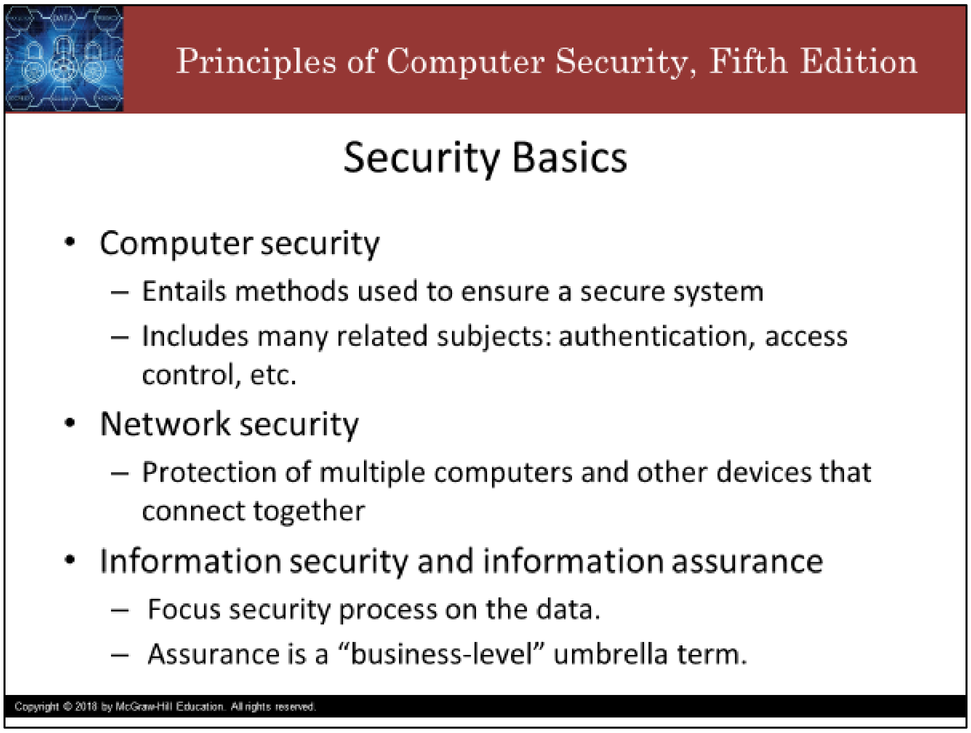
Principles of Computer Security, Fifth Edition

General Security Concepts

Basic Security Terminology
Security Basics

Howdy! In this video, we will introduce some basic security terminology, the InfoSec triad, the operational model of computer security, and the NIST Cybersecurity Framework.

Slide 2



Principles of Computer Security, Fifth Edition

Security Basics

- Computer security
  - Entails methods used to ensure a secure system
  - Includes many related subjects: authentication, access control, etc.
- Network security
  - Protection of multiple computers and other devices that connect together
- Information security and information assurance
  - Focus security process on the data.
  - Assurance is a “business-level” umbrella term.

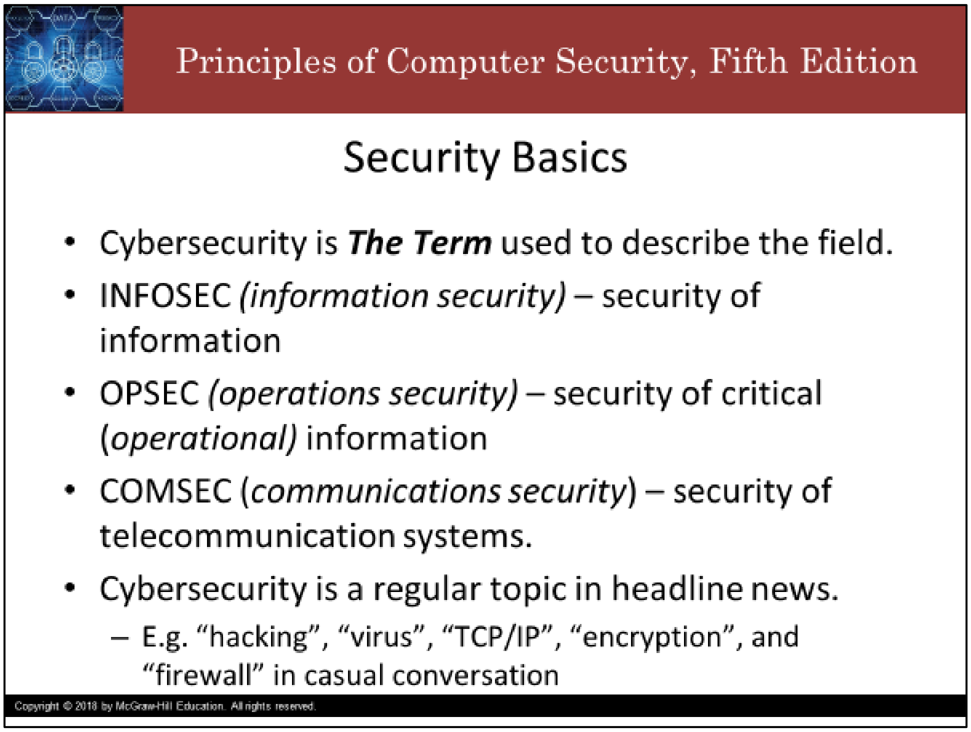
Computer security has many meanings and related terms. Things like authentication and access control must be addressed in broad terms of computer security.

Seldom in today’s world are computers not connected to other computers in networks; hence we now have the term *network security.*

Information security and *information assurance* place the focus of the security process not on the hardware and software being used but on the data that is processed by them.

Information assurance includes protection of user data (such as confidentiality, integrity, and availability). It encompasses both digital and physical protection. These protections apply to data at rest and in transit. However, the term is best used to describe the business-level practice of information risk management, something like a 10,000-foot view of InfoSec, requiring a very broad and holistic view that includes not just technical details but also corporate governance, regulatory compliance, incident response, and all the rest.

Slide 3



Principles of Computer Security, Fifth Edition

Security Basics

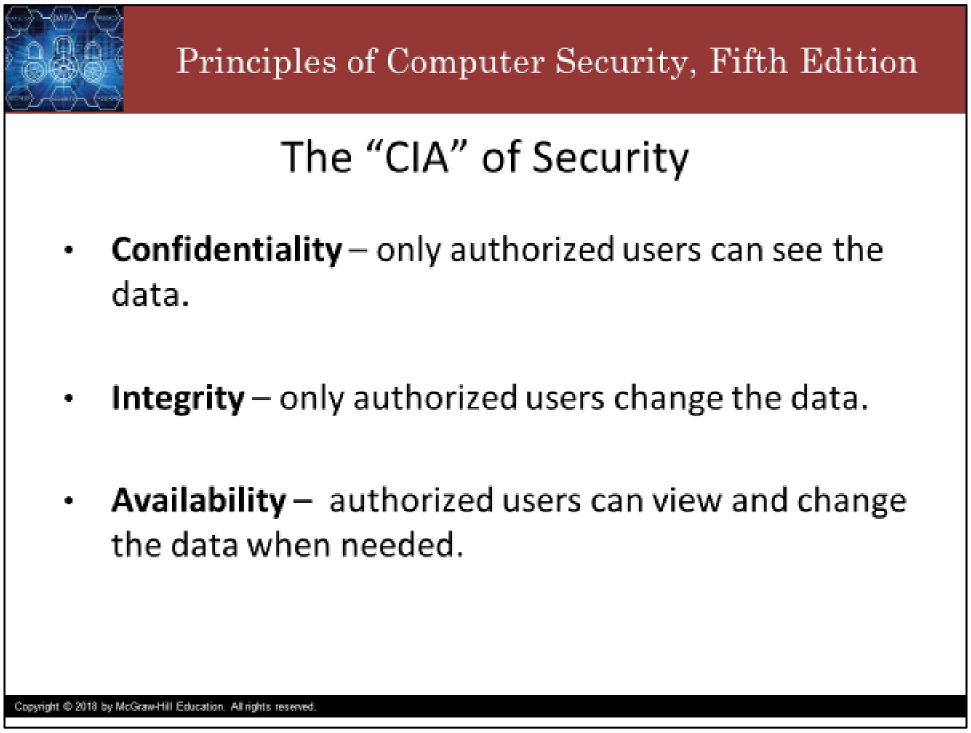
- Cybersecurity is ***The Term*** used to describe the field.
- INFOSEC *(information security)* – security of information
- OPSEC *(operations security)* – security of critical (*operational)* information
- COMSEC (*communications security*) – security of telecommunication systems.
- Cybersecurity is a regular topic in headline news.
  - E.g. "hacking", "virus", "TCP/IP", "encryption", and "firewall" in casual conversation

Cybersecurity is The Term that describes the entire field. Anything that has to do with security of connected systems, including the people that use them, is included. Cybersecurity is a vast a diverse field with room for everyone. Literally, it's May 2021 when I am recording this, and there are an estimated 464,420 jobs openings in cybersecurity in the United States (https://www.cyberseek.org/). Texas has about 10% of them. So, quick plug for cybersecurity as a career: we not only want you, we need you. We sometimes borrow or inherit terms from the government or military, like INFOSEC, OPSEC, and COMSEC.

Cybersecurity has become regular headline news these days, with reports of break-ins, data breaches, fraud, and a host of other calamities. The general public has become increasingly aware of its dependence on computers and networks and consequently has also become interested in the security of these same computers and networks. As a result of this increased attention by the public, several new terms have become commonplace in conversations and print.

With our increased daily dependence on computers and networks to conduct everything from making purchases at our local grocery store, banking, trading stocks, and receiving medical treatment to driving our children to school, ensuring that computers and networks are secure has become of paramount importance. Computers and the information they manipulate has become a part of virtually every aspect of our lives.

Slide 4



Principles of Computer Security, Fifth Edition

The "CIA" of Security

- **Confidentiality** – only authorized users can see the data.
- **Integrity** – only authorized users change the data.
- **Availability** – authorized users can view and change the data when needed.

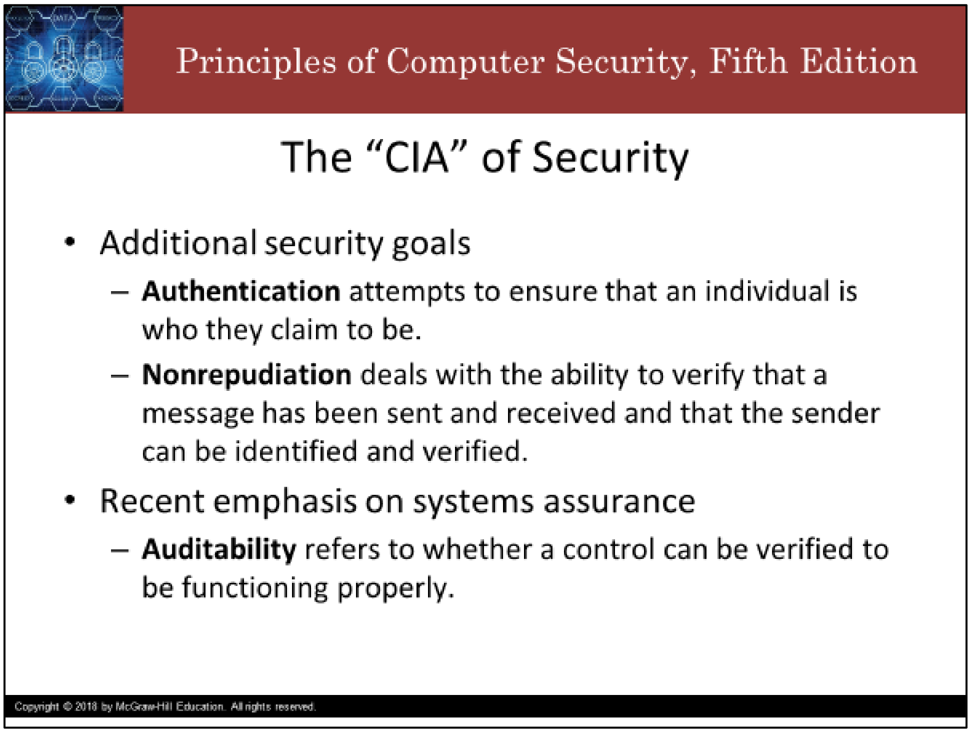
Often called the "InfoSec triad," the security goals of confidentiality, integrity, and availability are the foundation of information security.

Confidentiality ensures that only those individuals who have the authority to view a piece of information may do so.

Integrity deals with the generation and modification of data. Only authorized individuals should ever be able to create or change (or delete) information.

Availability ensures that the data, or the system itself, is available for use when the authorized user wants it.

Slide 5



Principles of Computer Security, Fifth Edition

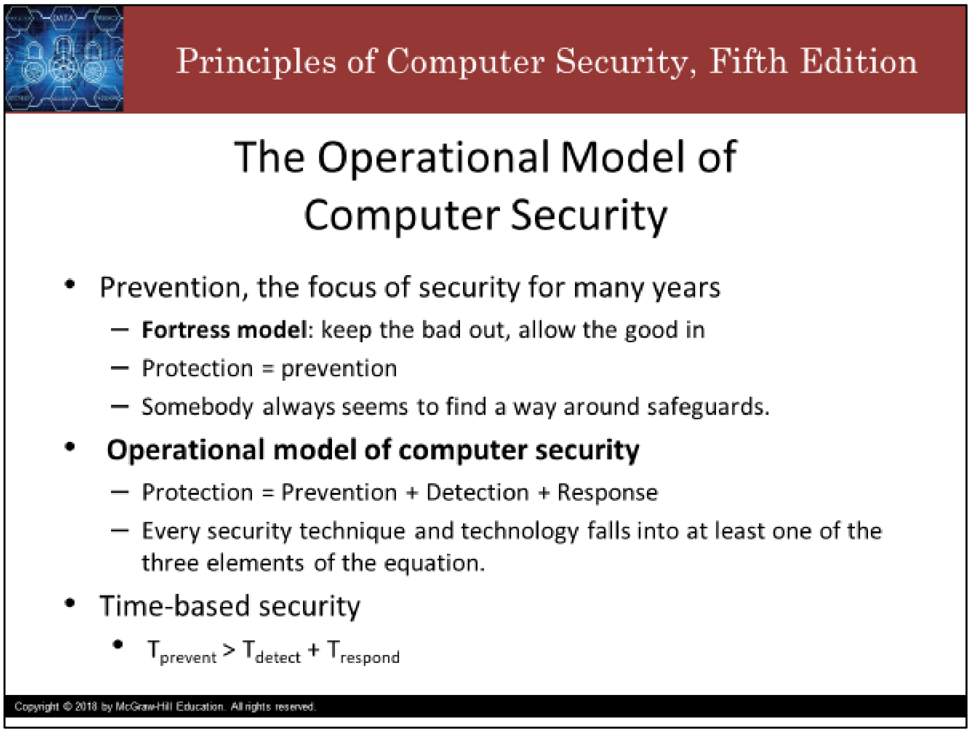
The “CIA” of Security

- Additional security goals
  - **Authentication** attempts to ensure that an individual is who they claim to be.
  - **Nonrepudiation** deals with the ability to verify that a message has been sent and received and that the sender can be identified and verified.
- Recent emphasis on systems assurance
  - **Auditability** refers to whether a control can be verified to be functioning properly.

Some lists of fundamental security goals include more than the 3 goals of the InfoSec triad. Common additions include authentication, nonrepudiation, and auditability. Some lists will include anonymity, too.

While there is no universal agreement on authentication, auditability, and nonrepudiation as additions to the original CIA of security, there is little debate over whether **confidentiality**, **integrity**, and **availability** are basic security principles. Understand these principles because one or more of them are the reason most security hardware, software, policies, and procedures exist.

Slide 6



Principles of Computer Security, Fifth Edition

The Operational Model of Computer Security

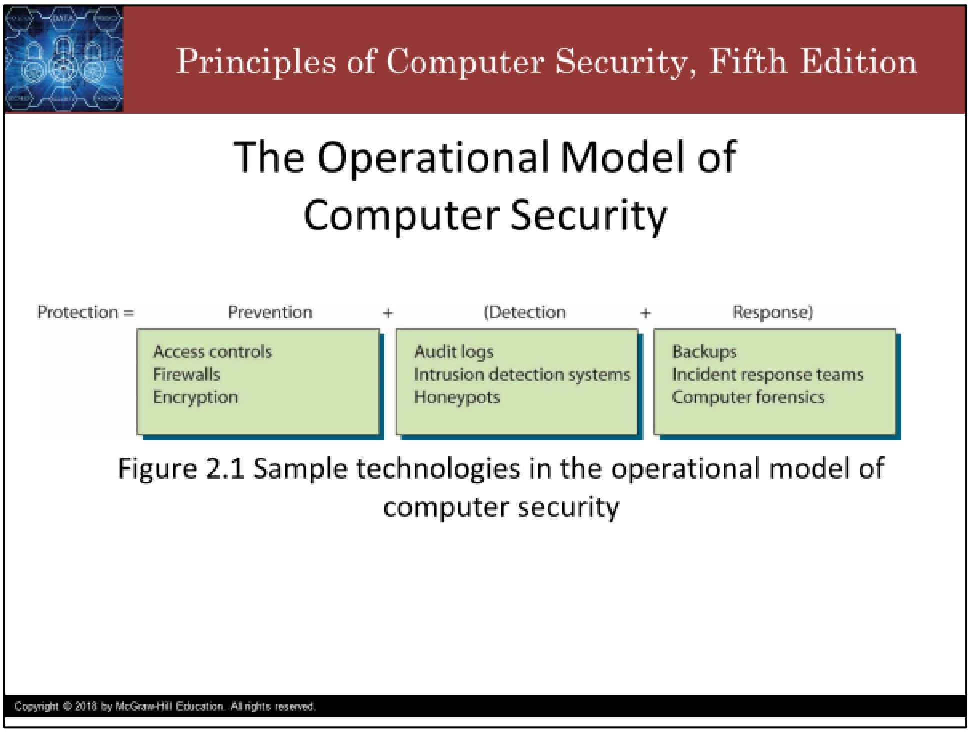
- Prevention, the focus of security for many years
  - **Fortress model**: keep the bad out, allow the good in
  - Protection = prevention
  - Somebody always seems to find a way around safeguards.
- **Operational model of computer security**
  - Protection = Prevention + Detection + Response
  - Every security technique and technology falls into at least one of the three elements of the equation.
- Time-based security
  - $T_{prevent} > T_{detect} + T_{respond}$

The fortress model was the original model for computer security. The idea was to build enough defenses (a fortress strong enough to withstand all attacks), and your system would be secure. If it's not secure yet, add more defenses. However, it turns out that building an impenetrable system security fortress (especially when trying to protect an existing system) is a pipe dream. Somebody always finds a way to break in (or out).

So, we need multiple prevention techniques and also technology to alert us when prevention has failed and to provide ways to address the problem. This results in a modification to the original security equation with the addition of two new elements—detection and response.

Time-based security is a constraint on the security equation to focus on risk management, arguing that this results in more effective and efficient use of resources. Time-based security considers the amount of time that a security mechanism can withstand an attack. In order for a protection mechanism (which goes in the prevention bucket of the operational model equation) to be worthwhile, the amount of time offered by the mechanism should be greater than the time it takes to detect the attack plus the response time of the organization. That is, the organization should be able to take action to stop the attack before it can succeed.

Slide 7



Principles of Computer Security, Fifth Edition

The Operational Model of Computer Security

Protection = Prevention + (Detection + Response)

Prevention	Detection	Response
Access controls Firewalls Encryption	Audit logs Intrusion detection systems Honeypots	Backups Incident response teams Computer forensics

Figure 2.1 Sample technologies in the operational model of computer security

Here are some examples of mechanisms used for security and the buckets into which they are placed in the operational model of computer security.

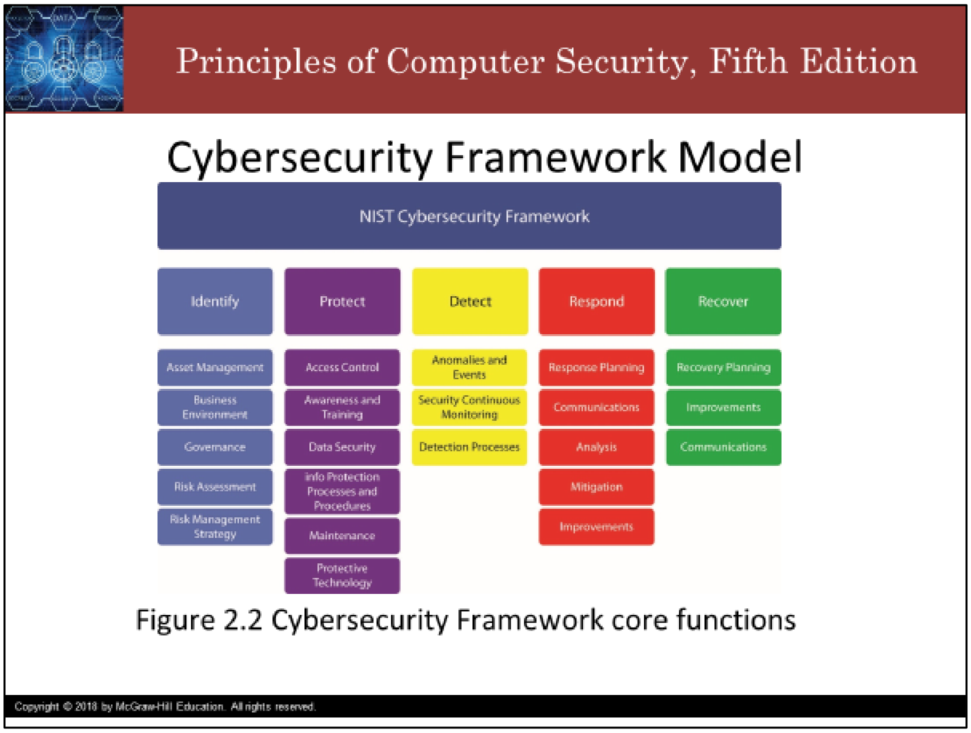
Access control, firewalls, and crypto are prevention mechanisms. They can slow down or even outright stop attacks.

Audit logs, IDS, and honeypots (which are decoy systems) are detection mechanisms. They reveal the presence of an attacker.

Backups and incident response (including forensics) are response mechanisms. They come into play after an attack (but sometimes during) and reveal what happened, and they help to restore the system.

A modern secure system needs mechanisms in every bucket. This is what is called defense-in-depth and is something we will talk about again soon.

Slide 8



Principles of Computer Security, Fifth Edition

Cybersecurity Framework Model

NIST Cybersecurity Framework

Identify	Protect	Detect	Respond	Recover
Asset Management	Access Control	Anomalies and Events	Response Planning	Recovery Planning
Business Environment	Awareness and Training	Security Continuous Monitoring	Communications	Improvements
Governance	Data Security	Detection Processes	Analysis	Communications
Risk Assessment	Info Protection Processes and Procedures		Mitigation	
Risk Management Strategy	Maintenance		Improvements	
	Protective Technology			

Figure 2.2 Cybersecurity Framework core functions

In 2013, President Obama signed an executive order directing the U.S. National Institute of Science and Technology (NIST) to work with industry and develop a cybersecurity framework. This was in response to several significant cybersecurity events where the victim companies appeared to be unprepared. The resulting framework, titled *Framework for Improving Critical Infrastructure Cybersecurity*, was created as a voluntary system.

The NIST Cybersecurity Framework is a risk-based approach to implementation of cybersecurity activities in an enterprise. The framework provides a common taxonomy of standards, guidelines, and practices that can be employed to strengthen security efforts.

Its purpose is to complement and enhance the risk management efforts of companies through:

1. Determining their current cybersecurity posture
2. Documenting their desired target state with respect to cybersecurity
3. Determining and prioritizing improvement and corrective actions
4. Measuring and monitoring progress toward goals
5. Creating a communication mechanism for coordination among stakeholders

It is composed of five core functions: Identify, protect, detect, respond and recover.

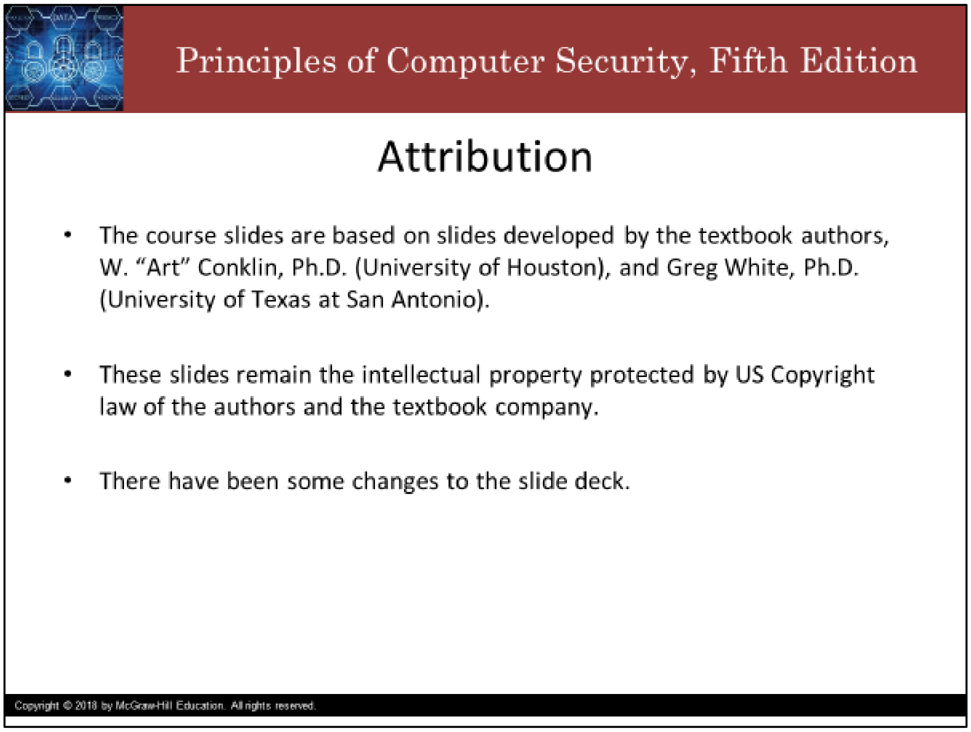
Identify and Protect describe actions taken before an incident.

Detect is the core function associated with intrusion detection or the beginning of an incident response.

Respond and Recover detail actions that take place during the post-incident response.

In addition to the five functions, the framework has levels of implementations referred to as tiers. These tiers represent the organization’s ability from the lowest level of “Partial” (Tier 1) to the highest level of “Adaptive” (Tier 4).

Slide 9



Principles of Computer Security, Fifth Edition

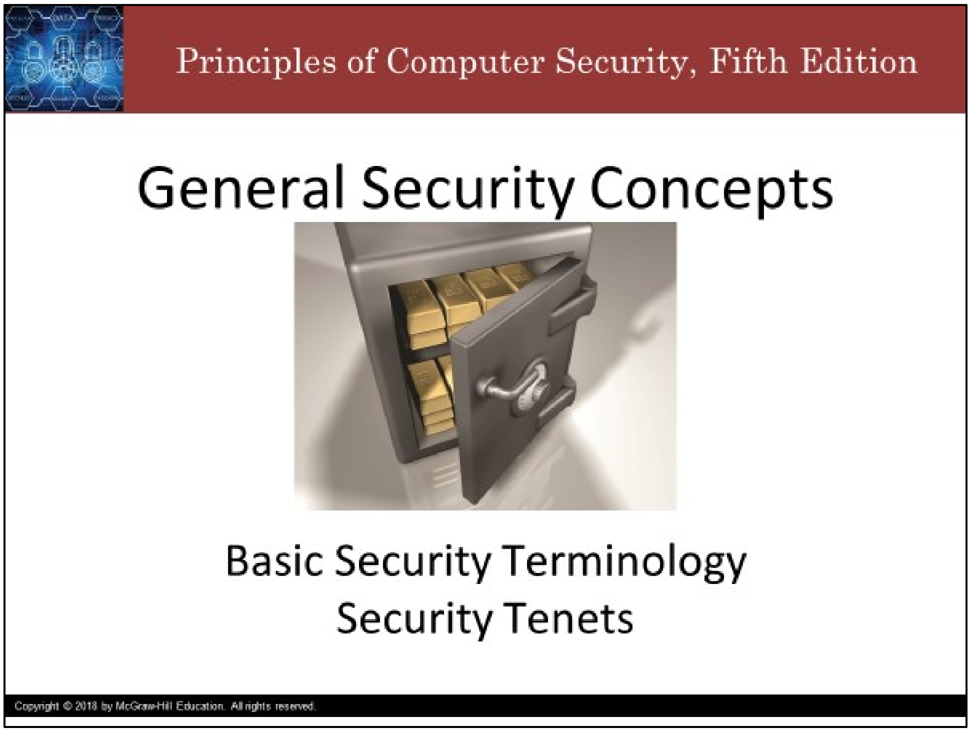
Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Thank you and take care.

General Security Concepts: Security Tenents

Slide 1



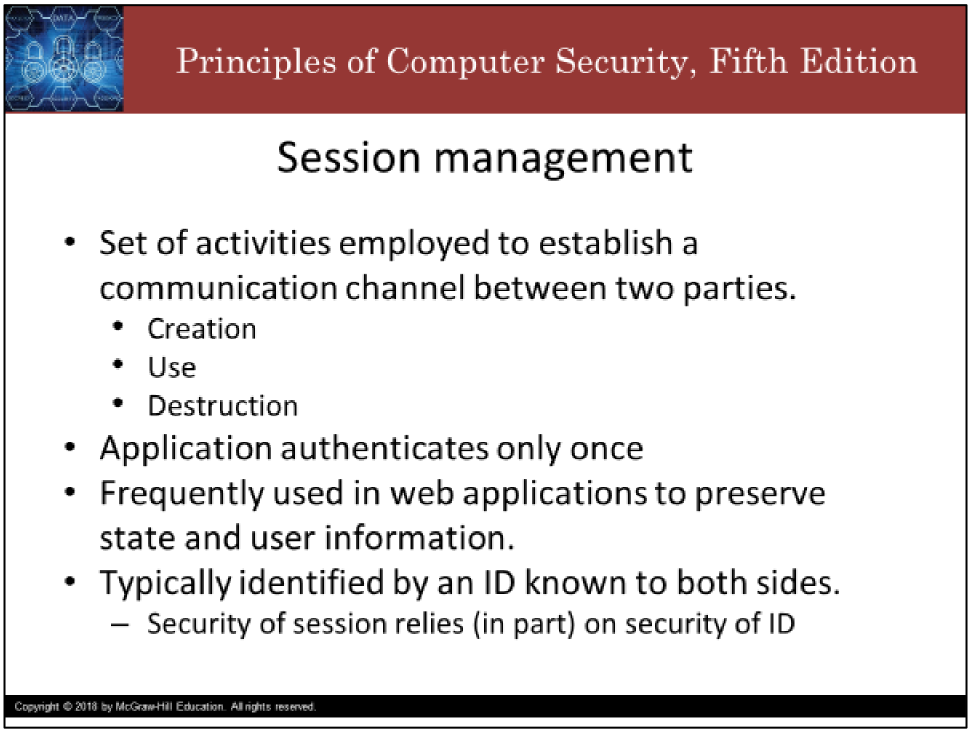
Principles of Computer Security, Fifth Edition

General Security Concepts

Basic Security Terminology
Security Tenets

Howdy! In this video, we will discuss the operational tenets of session management, exception management, and configuration management.

Slide 2



Principles of Computer Security, Fifth Edition

Session management

- Set of activities employed to establish a communication channel between two parties.
  - Creation
  - Use
  - Destruction
- Application authenticates only once
- Frequently used in web applications to preserve state and user information.
- Typically identified by an ID known to both sides.
  - Security of session relies (in part) on security of ID

Session management is a set of activities employed to establish a communication channel between two parties.

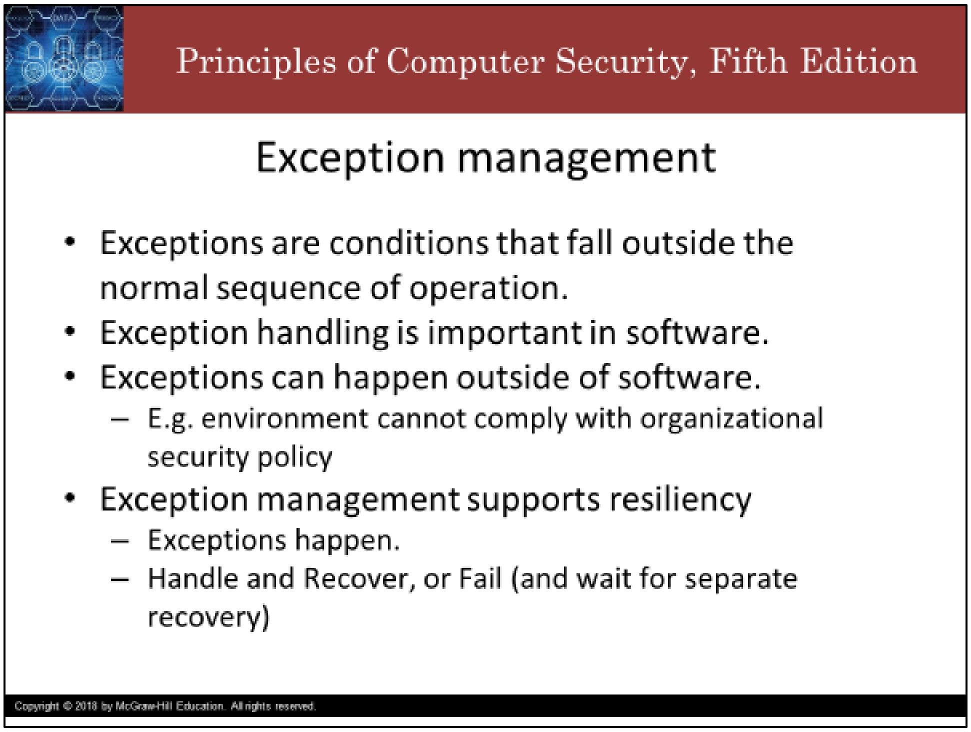
Session management includes all the activities necessary to manage the session, from establishment, during use, and at completion of the conversation.

Session management allows an application to authenticate once and have subsequent activities ascribed to the authenticated user.

Sessions are frequently used in web applications to preserve state and user information between normally stateless clicks.

Sessions are typically identified by an identifier known to both sides. The id can be used as a token for future identification if confidentiality is required. Then the channel should be secured by an appropriate level of cryptographic protection because the session represents the continuity of a security condition established during authentication. The level of protection that should be afforded to the session id should be commensurate with the level of security initially established.

Slide 3



Principles of Computer Security, Fifth Edition

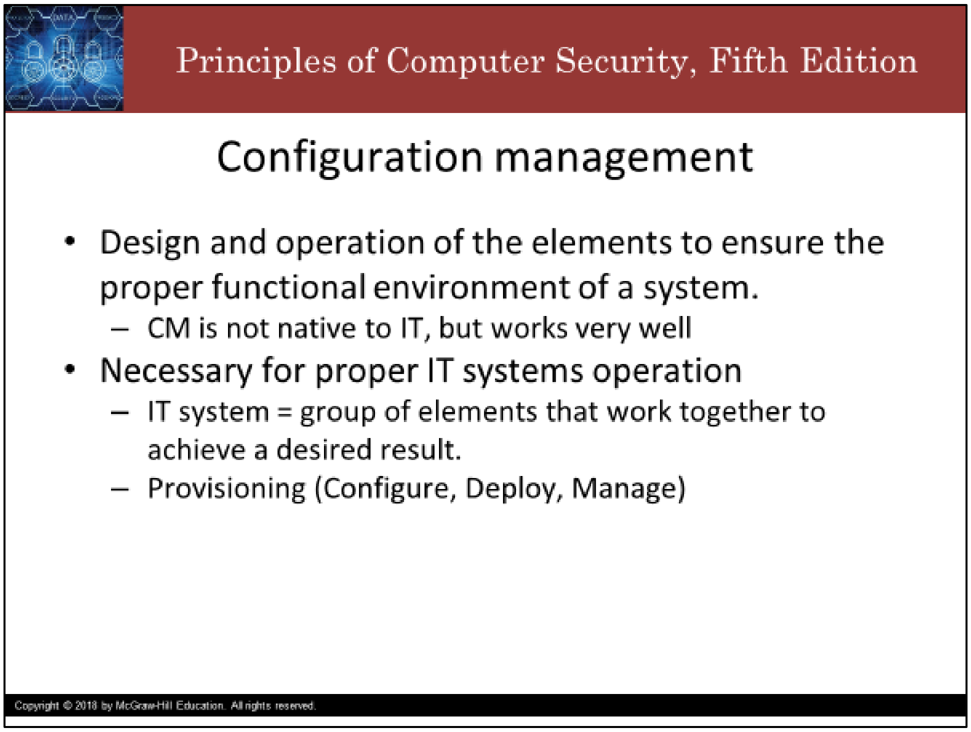
Exception management

- Exceptions are conditions that fall outside the normal sequence of operation.
- Exception handling is important in software.
- Exceptions can happen outside of software.
  - E.g. environment cannot comply with organizational security policy
- Exception management supports resiliency
  - Exceptions happen.
  - Handle and Recover, or Fail (and wait for separate recovery)

Exceptions are conditions that fall outside the normal sequence of operation, whether by error or malicious action. Exceptions are changes to normal processing and need to be managed. The special processing required by exceptions can result in errors either locally or in other processes in a system. How to handle exceptions is an important consideration during software design and development. Exception management is more than software exception handling. When a system operation exception occurs, the system must effectively handle it. Sometimes this can even mean operating outside normal policy limits temporarily. Exceptions can be non-technical in nature. For example, some environments cannot comply with organizational security policy.

Exception management includes documenting the exceptions, determining mitigations to deploy to limit risk, approving the exception if sufficient mitigation can be achieved, and monitoring the exceptions. Exceptions will happen. The only question is how they're handled. The system must handle the condition and recover, or it must fail and wait to be recovered by some other process or action.

Slide 4



Principles of Computer Security, Fifth Edition

Configuration management

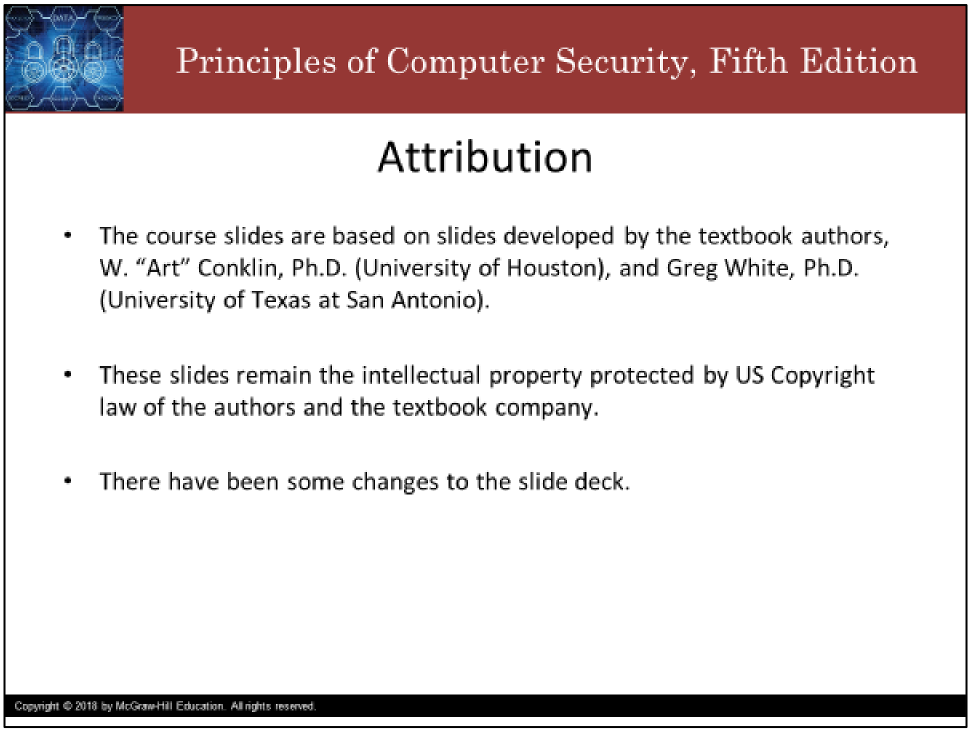
- Design and operation of the elements to ensure the proper functional environment of a system.
  - CM is not native to IT, but works very well
- Necessary for proper IT systems operation
  - IT system = group of elements that work together to achieve a desired result.
  - Provisioning (Configure, Deploy, Manage)

Configuration management is the design and operation of the elements to ensure the proper functional environment of a system throughout its life.

Configuration management is necessary for the proper operation of IT systems. It deals not only with the configuration of the environment and the devices within it but also with their deployment and management.

For many systems, the lack of CM, or an ineffective implementation of it, can be very expensive and sometimes can have such catastrophic consequences as failure of equipment or loss of life. Considering the dependence of critical infrastructure on connected computing systems, it should be clear that CM is important for cybersecurity. Recently in the news -- it's May 2021 right now in case you are watching this in the future, hello future folk!) -- anyways, a hacker group recently shut down an oil pipeline with a ransomware attack. The company ended up paying the 5-million-dollar ransom. I would bet that an ineffectual implementation of CM was in part to blame for the attack and the inability of the company to recover without paying the ransom.

Slide 5



Principles of Computer Security, Fifth Edition

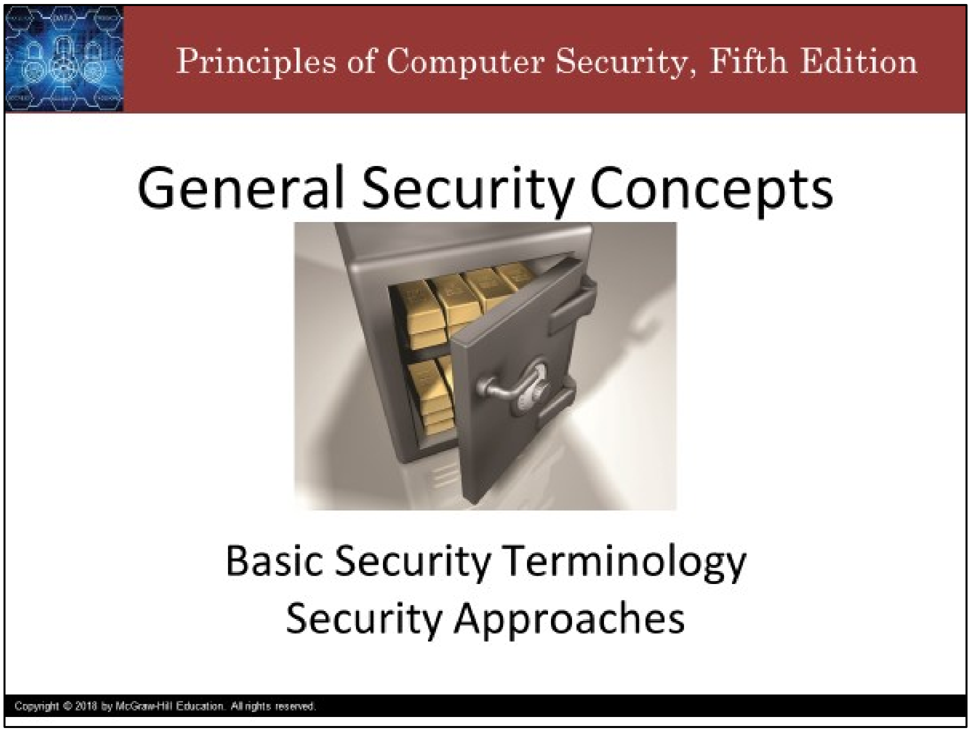
Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Thank you and take care.

General Security Concepts: Security Approaches

Slide 1



Principles of Computer Security, Fifth Edition

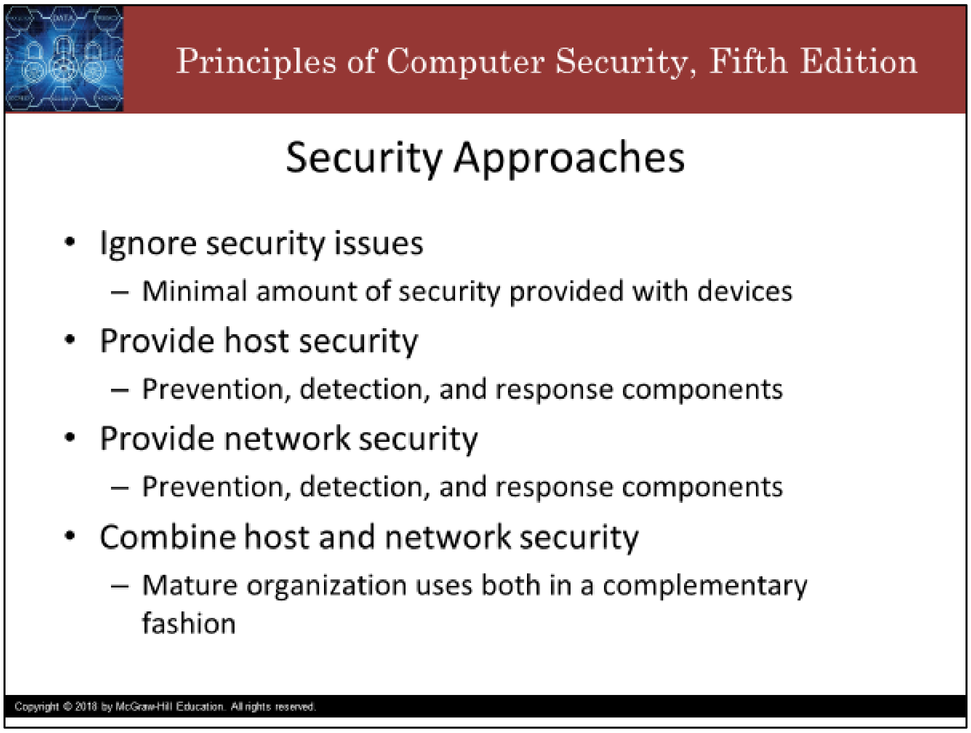
General Security Concepts

Basic Security Terminology

Security Approaches

Howdy! In this video, we will discuss 4 approaches to security: none, host security, network security, and using both host and network security.

Slide 2



Principles of Computer Security, Fifth Edition

Security Approaches

- Ignore security issues
  - Minimal amount of security provided with devices
- Provide host security
  - Prevention, detection, and response components
- Provide network security
  - Prevention, detection, and response components
- Combine host and network security
  - Mature organization uses both in a complementary fashion

There are multiple tools and strategies that an organization can use to address system security. Generally speaking, we can put these into two boxes: host security and network security. These two boxes are somewhat orthogonal, although there is some overlap between them. However, the important point to make is that they can be applied independently, or in concert, or not at all. Thus we have four security approaches: do nothing, apply host-only security, apply network-only security, or apply both.

If an organization decides to ignore security, it has chosen to utilize the minimal amount of security that is provided with its workstations, servers, and other devices. No additional security measures will be implemented. Each "out of the box" system has certain security settings that can be configured, and they really should be. To actually protect an entire network, however, requires work in addition to the few protection mechanisms that come with systems by default. Unfortunately, too often, system defaults are not secure.

A good example of this is routers for home networking. The default credentials for most commercially available routers are easy to find on the Internet. Most Linksys routers have a default IP address of 192.168.1.1, and the admin username and password are some combination of none and "admin." If you don't change these, then anyone within range of your WiFi network can connect directly to the router, log in as admin, and effectively take control of your network. Suffice it to say, ignoring security issues and relying on defaults configuration is the worst approach.

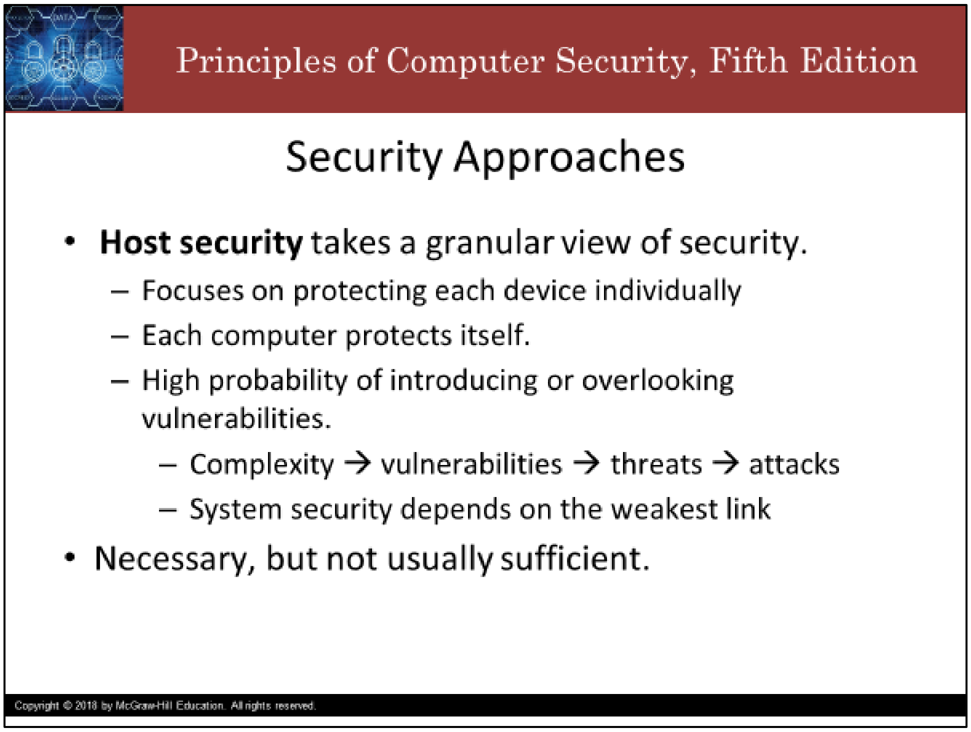
Host security focuses on protecting each device individually.

Network security focuses on protecting access to the devices on the network.

If your system is a gated community, host security is the alarm system and the locks in your house, which operate independently of the alarm system and locks, or lack thereof, in your neighbor's house. Network security is the gate, fence, and the guard that control access to the neighborhood as a whole.

The best approach is to use both host and network security. This is an application of the security principle of defense in depth.

Slide 3



Principles of Computer Security, Fifth Edition

Security Approaches

- **Host security** takes a granular view of security.
  - Focuses on protecting each device individually
  - Each computer protects itself.
  - High probability of introducing or overlooking vulnerabilities.
    - Complexity → vulnerabilities → threats → attacks
    - System security depends on the weakest link
- Necessary, but not usually sufficient.

Host security focuses on protecting each computer and device individually instead of addressing protection of the network as a whole.

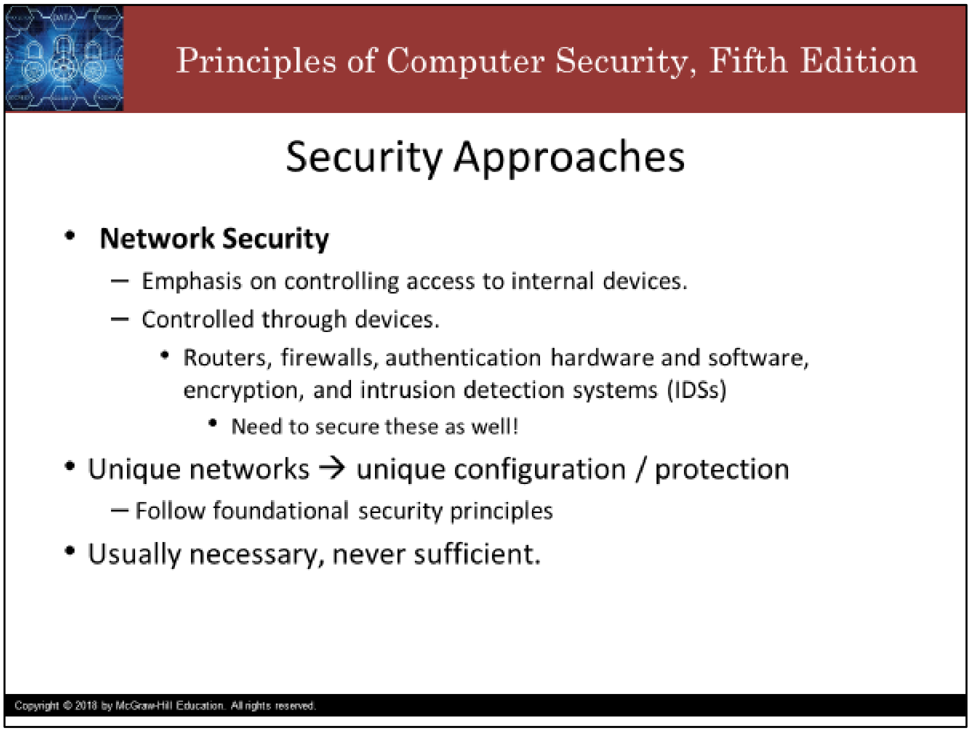
When host-only security is used, each computer is expected to protect itself.

Without network security, there is a high probability of introducing or overlooking vulnerabilities.

Most environments contain devices with different operating systems and different versions of those operating systems and different types and versions of installed applications. Each operating system has security configurations that differ from those of other systems, and different versions of the same operating system may even have configuration variations between them. As the complexity of the system grows, so, too, does the probability of vulnerabilities, which lead to attacks. The security of the whole system often comes down to the security of the weakest link. Finding and securing that weakest link is difficult in large and complex systems.

Host security is important and should always be addressed. Security, however, should not stop there since host security can (and should) be combined with network security. If individual devices have vulnerabilities, then network security can provide another layer of protection that will slow down or stop attackers who have gotten that far into the environment.

Slide 4



Principles of Computer Security, Fifth Edition

Security Approaches

- **Network Security**
  - Emphasis on controlling access to internal devices.
  - Controlled through devices.
    - Routers, firewalls, authentication hardware and software, encryption, and intrusion detection systems (IDSs)
      - Need to secure these as well!
- Unique networks → unique configuration / protection
  - Follow foundational security principles
- Usually necessary, never sufficient.

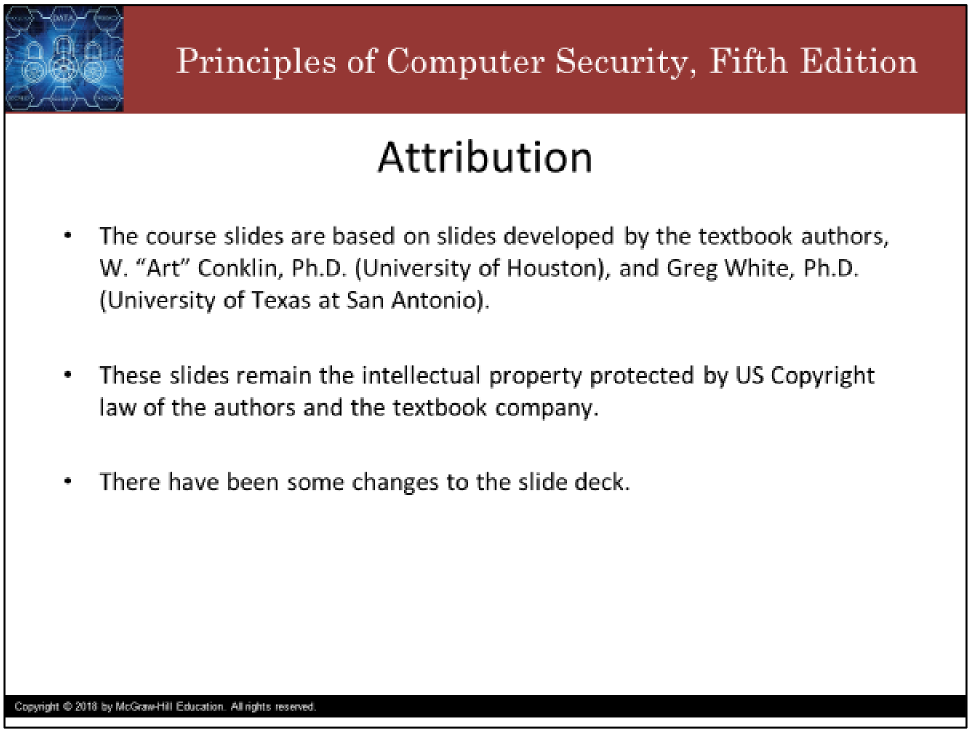
In some smaller environments, host-only security may actually be a viable option, but as systems become connected into networks, security should include the actual network itself.

In **Network Security,** emphasis is placed on controlling access to devices within the network from entities outside of the network.

This control is effected through devices such as routers, firewalls, and intrusion detection systems. These devices are themselves hosts, and therefore need to also be subject to host security mechanisms.

Network environments tend to be unique entities because no two networks have exactly the same topology and devices. Since networks have so many variations, there are many different ways they can be configured and protected. Each approach may be implemented in several ways, but both host and network security need to be addressed for effective overall system security. There are foundational security principles that are useful for designing, building, and operating secure systems.

Slide 5



Principles of Computer Security, Fifth Edition

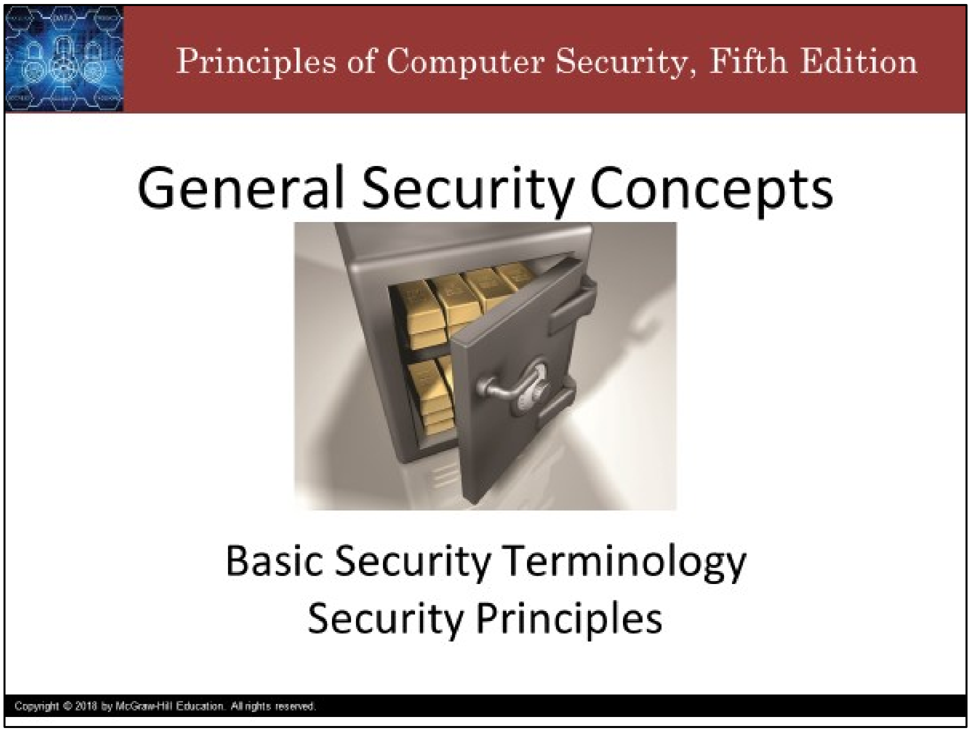
Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Thank you and take care.

General Security Concepts: Security Principles

Slide 1



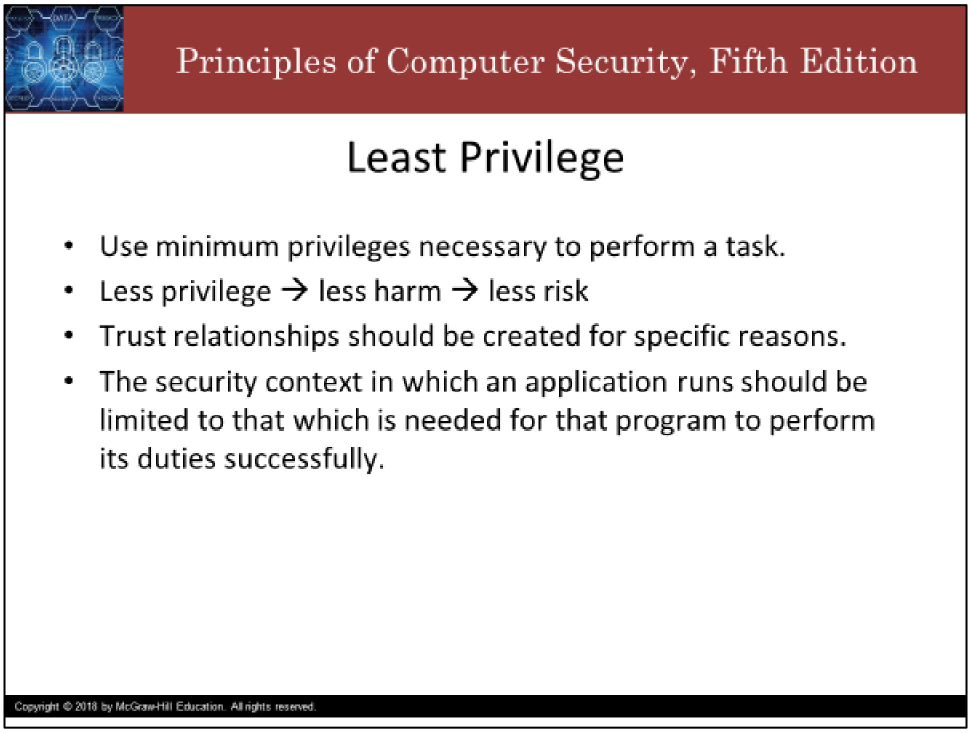
Principles of Computer Security, Fifth Edition

General Security Concepts

Basic Security Terminology
Security Principles

Howdy! In this video, we discuss several foundational principles for designing, building, and operating secure systems.

Slide 2



Principles of Computer Security, Fifth Edition

Least Privilege

- Use minimum privileges necessary to perform a task.
- Less privilege → less harm → less risk
- Trust relationships should be created for specific reasons.
- The security context in which an application runs should be limited to that which is needed for that program to perform its duties successfully.

One of the most fundamental principles in security is **least privilege**. This concept is applicable to many physical environments as well as network and host security.

A subject, which can be a user, application, or process, should have only the rights and privileges required to perform its task and no additional permissions.

By limiting an object's privilege, we limit the amount of harm that can be caused.

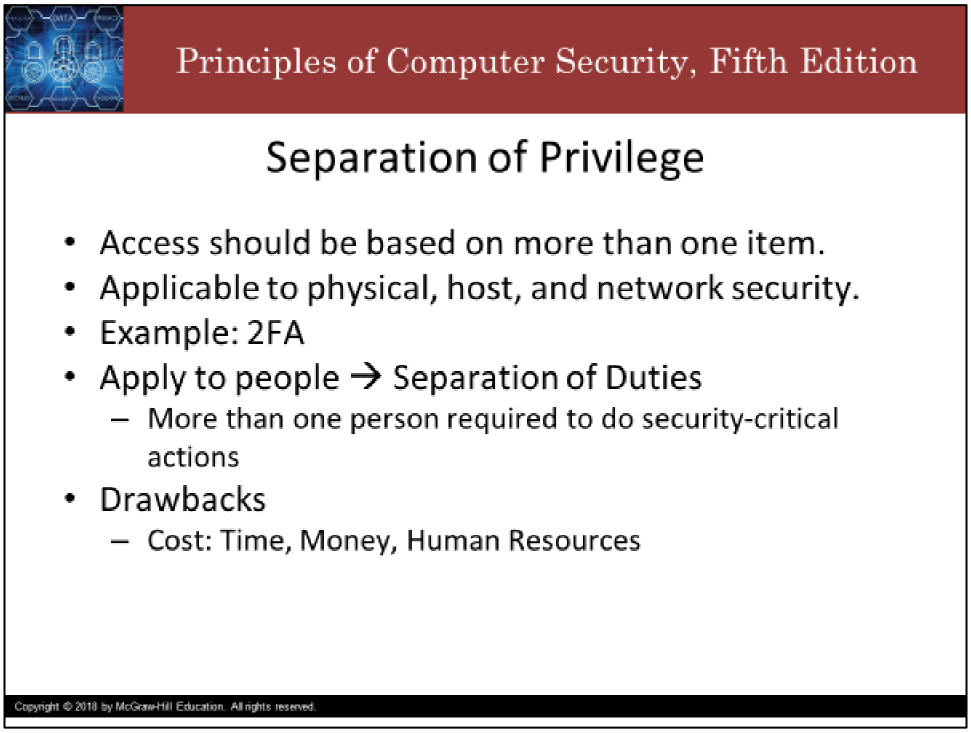
For example, users may have access to the files on their workstations and a select set of files on a file server, but no access to critical data that is held within the database.

This rule helps an organization protect its resources -- devices and information -- and helps ensure that whoever is interacting with these resources has a valid reason to do so.

The concept of least privilege applies to more network security issues than just providing users with specific rights and permissions. When trust relationships are created, they should not be implemented in such a way that everyone trusts each other simply because it is easier. One domain should trust another for very specific reasons, and the implementers should have a full understanding of what the trust relationship allows between two domains. If one domain trusts another, do all of the users automatically become trusted, and can they thus easily access any and all resources on the other domain? Is this a good idea? Is there a more secure way of providing the same functionality? If a trusted relationship is implemented such that users in one group can access a plotter or printer that is available in only one domain, it might make sense to simply purchase another plotter so that other, more valuable, or sensitive resources are not accessible by the entire group.

Another issue that falls under the least privileged concept is the security context in which an application runs. All applications, scripts, and batch files run in the security context of a specific user on an operating system. They execute with specific permissions as if they were a user. The application may be Microsoft Word and run in the space of a regular user, or it may be a diagnostic program that needs access to more sensitive system files and so must run under an administrative user account, or it may be a program that performs backups and so should operate within the security context of a backup operator. The crux of this issue is that a program should execute only in the security context that is needed for that program to perform its duties successfully. In many environments, people do not really understand how to make programs run under different security contexts, or it may just seem easier to have all programs run under the administrator account. If attackers can compromise a program or service running under the administrator account, they have effectively elevated their privilege and have much more control over the system and much more potential to cause damage.

Slide 3



Principles of Computer Security, Fifth Edition

Separation of Privilege

- Access should be based on more than one item.
- Applicable to physical, host, and network security.
- Example: 2FA
- Apply to people → Separation of Duties
  - More than one person required to do security-critical actions
- Drawbacks
  - Cost: Time, Money, Human Resources

Protection mechanisms can grant access based on a variety of factors. One of the key ideas is to use more than one piece of information to make those decisions.

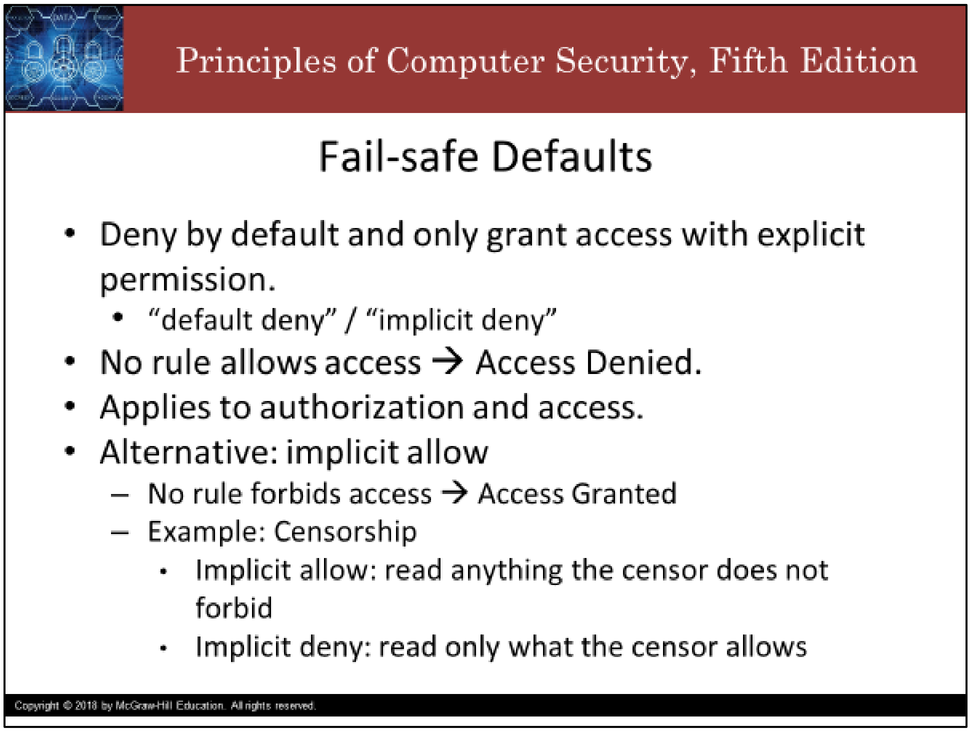
An example, hopefully, a familiar one, of an application of this principle in practice is multi-factor authentication. That is, requiring at least 2 forms of identification, typically a password and a physical device or attribute (like a fingerprint) when authenticating users. The authentication privilege is thus separated into possession of at least two of the identifying pieces of information. In order to log in, a user must have both. An attacker with only one will not be able to log in.

When applied to personnel, the principle of separation of privilege becomes separation of duties. That is, tasks (maybe all, maybe only some) should be divided into different duties and those duties assigned to different people. This way, no one person can abuse the system for gain. For example, one person is responsible for making purchases, and another person is responsible for making payments.

Separation of privilege provides a certain level of checks and balances, but it does have some drawbacks. This fact – that things that are good for security are bad for other properties of the system -- is so common in security that there is a security principle coming up later that deals with it directly.

The chief drawback is the cost required to accomplish a privilege- or duty-separated task. That cost is paid in time, money, and human resources.

Slide 4



Principles of Computer Security, Fifth Edition

Fail-safe Defaults

- Deny by default and only grant access with explicit permission.
  - "default deny" / "implicit deny"
- No rule allows access → Access Denied.
- Applies to authorization and access.
- Alternative: implicit allow
  - No rule forbids access → Access Granted
  - Example: Censorship
    - Implicit allow: read anything the censor does not forbid
    - Implicit deny: read only what the censor allows

Fail-safe defaults is the principle that failures should leave the system in a safe state.

This is also referred to as default deny or implicit deny.

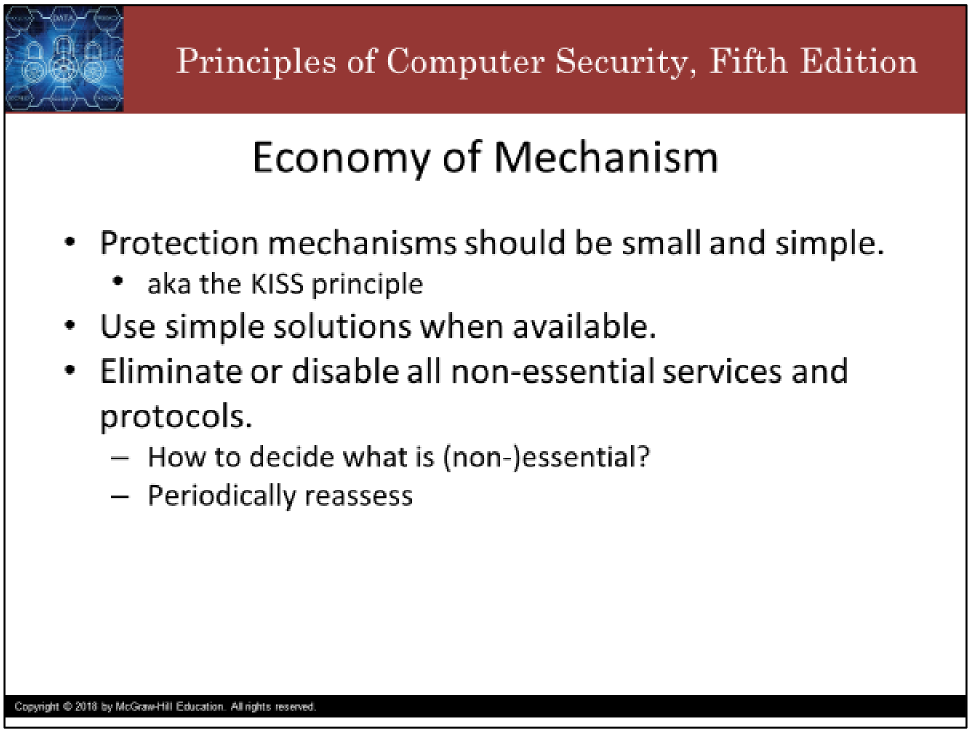
Frequently in the network world, administrators make many decisions concerning network access. Often a series of rules will be used to determine whether or not to allow access (which is the purpose of a network firewall). If a particular situation is not covered by any of the other rules, fail-safe defaults require that access should not be granted. In other words, if no rule would allow access, then access should not be granted. Fail-safe defaults apply to situations involving both authorization and access.

My favorite example of this principle, or, rather, of this principle being violated, is in the movie Diehard. If you haven't seen it, you should. It's great. In the movie, the "terrorists," led by Hans Gruber, are trying to steal a lot of money from a vault. The vault has many layers of security (an application of defense in depth), which the hacker on the team defeats one after the other until the final layer, which is an unhackable time-based electromagnetic lock. So, what does Hans do? He orchestrates a situation in which the good guys decide their best move is to switch off the power to the building. When they do this, the electromagnetic lock loses power, and the vault swings open. Ta-Da! 640 million dollars of bearer bonds now belong to Hans. Do you see how the design of the security system violates the principle of fail-safe defaults? That's right! The vault should fail into the _locked_ state, not into the unlocked state. When the power goes out, the vault should become completely unopenable. That would have made for a short movie, but good security *should be* boring like that.

The alternative to implicit deny is to allow access unless a specific rule forbids it. An example of these two approaches is in programs that monitor and block access to certain websites. One approach is to provide a list of specific sites that a user is not allowed to access. Access to any site not on the list would be implicitly allowed. The opposite approach (the implicit deny approach) would block all access to sites that are not specifically identified as authorized. As you can imagine, depending on the specific application, one or the other approach will be more appropriate. Which approach you choose depends on the security objectives and policies of your organization.

Going back to the Diehard example. Apparently, the vault designers chose the implicit allow approach, which was wrong. But, there are good reasons to choose implicit allow on things like electromagnetic locks. One example is fire doors. In case of a fire and the power goes out, people should not be trapped in the building. All the exits should fail open (at least, egress should not be prevented… in secure installations, ingress should still be guarded against with exceptions for emergency response teams, like firefighters). So, cybersecurity professionals have to think about things like this.

Slide 5



Principles of Computer Security, Fifth Edition

Economy of Mechanism

- Protection mechanisms should be small and simple.
  - aka the KISS principle
- Use simple solutions when available.
- Eliminate or disable all non-essential services and protocols.
  - How to decide what is (non-)essential?
  - Periodically reassess

Security and complexity are at odds with each other. The more complex something is, the harder it is to build and manage (i.e., secure, maintain, and operate). Edsger Dijkstra once noted that "if debugging is the process of removing bugs, then programming must be the process of putting them in." The more complex something is and the more lines of code it takes to implement it (or for provisioning or configuration), the more opportunities there are for errors to be made. Thus, if complexity is bad, then simplicity is good. So, the principle of economy of mechanism says to keep it simple.

Putting this principle into practice means using simple solutions when available. (This will come up again later in another favorite example: never roll your own crypto).

It also means reducing the number of things the system is expected to do, which fits well with the principles of least privilege and separation of privilege.

Part of a process known as system hardening, which you may come across at some point, is to eliminate or disable all non-essential services and protocols.

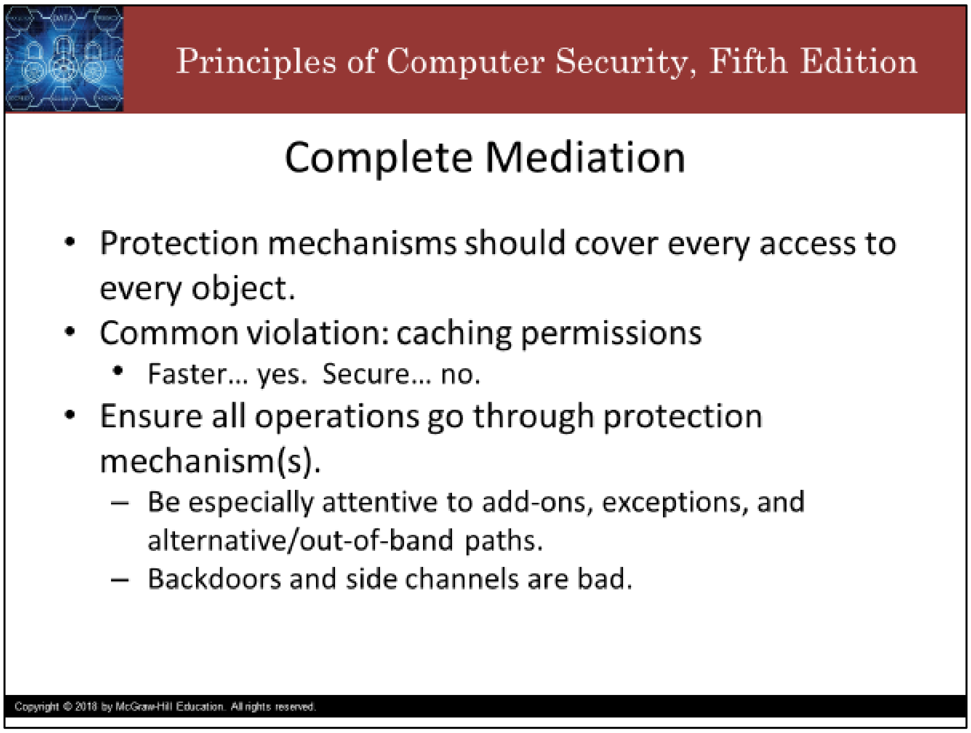
This begs the question, however, of how to determine which services and protocols are essential and which are not.

The answer is: you should know. Now, that doesn't seem helpful. But, actually, it is. The point is that if you cannot decide whether a service is essential, then you do not know enough about your system, and you need to go back to the drawing board to make it simpler and easier to understand.

One tactic you could take to figure out what is essential is to ruthlessly apply the principles of least privilege and fail-safe defaults and assume all services are non-essential, disable everything, and then proceed from the null set of services to enable only those that are absolutely necessary for the system to operate.

No matter what you do, expect to engage in a never-ending cycle of reassessment to find the right balance between functionality, usability, and security.

Slide 6



Principles of Computer Security, Fifth Edition

Complete Mediation

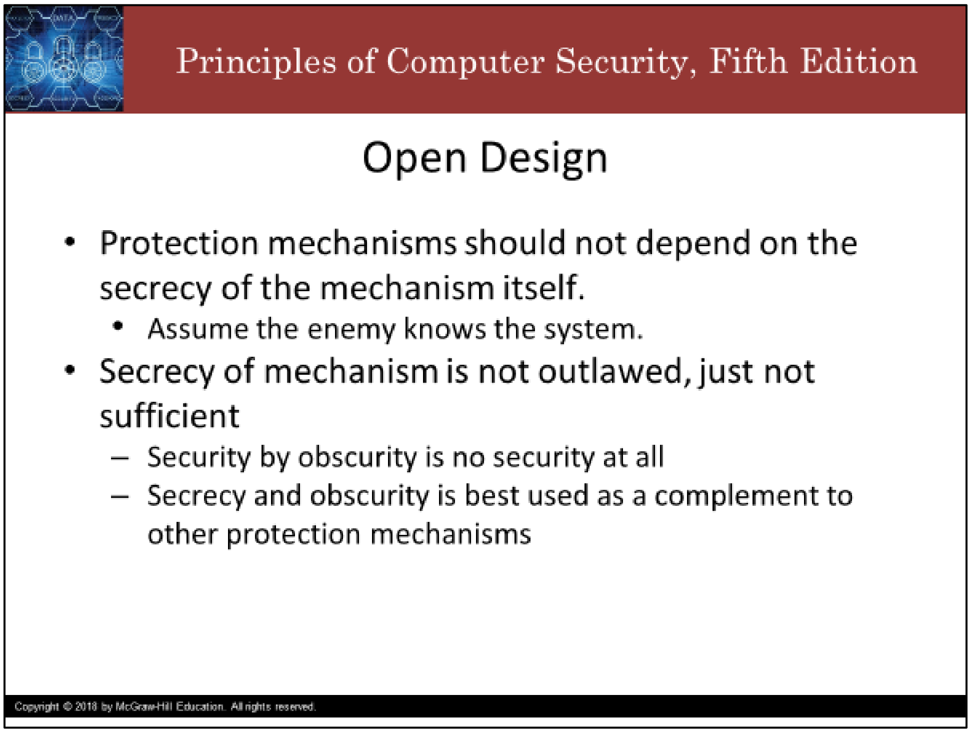
- Protection mechanisms should cover every access to every object.
- Common violation: caching permissions
  - Faster… yes. Secure… no.
- Ensure all operations go through protection mechanism(s).
  - Be especially attentive to add-ons, exceptions, and alternative/out-of-band paths.
  - Backdoors and side channels are bad.

One of the fundamental tenets of a protection system is to check all access requests for permission. Each and every time a subject requests access to an object, the permission must be checked; otherwise, an attacker might gain unauthorized access to an object.

The principle of **Complete mediation** states that every request for access should be checked for authorization.

Complete mediation also refers to ensuring that all operations go through the protection mechanism. When security controls are added after the fact, it is important to make certain that all process flows are covered by the controls, including exceptions and out-of-band requests. If an automated process is checked in one manner, but a manual paper backup process has a separate path, it is important to ensure all checks are still in place. When a system undergoes disaster recovery or business continuity processes or backup and restore processes, these too require complete mediation.

Slide 7



Principles of Computer Security, Fifth Edition

Open Design

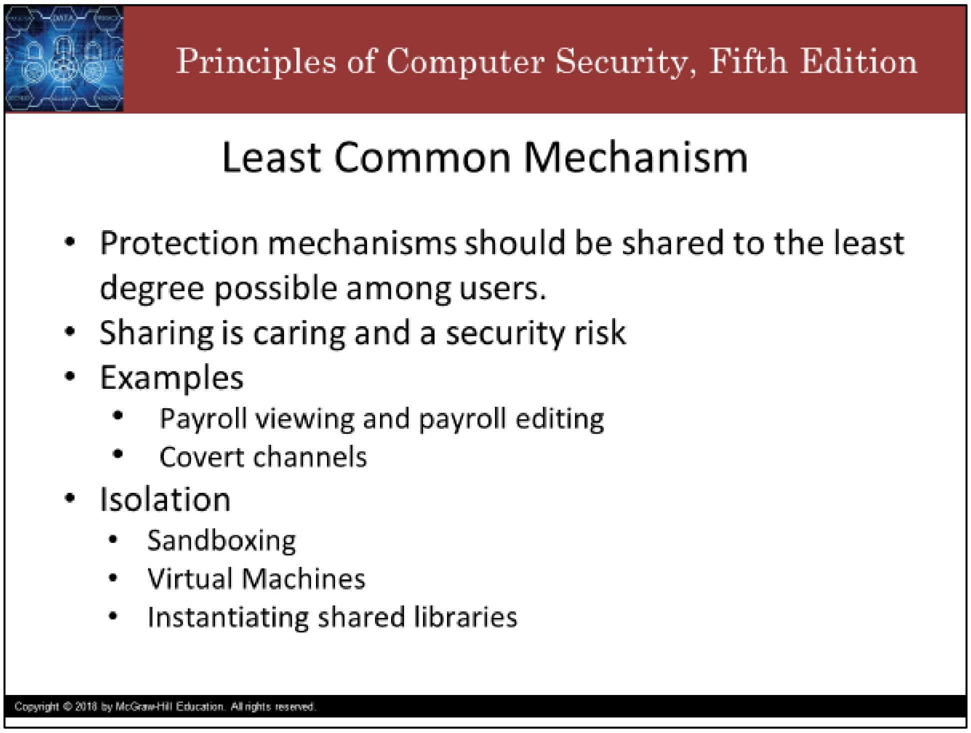
- Protection mechanisms should not depend on the secrecy of the mechanism itself.
  - Assume the enemy knows the system.
- Secrecy of mechanism is not outlawed, just not sufficient
  – Security by obscurity is no security at all
  – Secrecy and obscurity is best used as a complement to other protection mechanisms

The principle of **Open design** states that the protection of an object should not rely upon secrecy of the protection mechanism itself.

This principle is foundational to modern cryptography, where it is well-established that the security of the encryption must rest solely in the key and not at all in the algorithm. We will discuss cryptography in more depth in another module, but to underline this point, it is worth noting that the advanced encryption standard (AES), which is a national standard for encryption in the US, is fully specified in public documents and has nonetheless withstood all publicly-known attacks. In fact, the competition to select the algorithm which would become AES required that all candidate algorithms were open and subjected to long periods of intense analysis by cryptographic experts around the world and did result in improvements to the algorithms. The subsequent winner, an algorithm called Rijndael, is so good as a direct result of the principle of open design.

It is important to note, though, that this principle does not exclude the practice of keeping the design or implementation of protection mechanisms secret. It simply states that secrecy alone is not sufficient for protection. Attempting to protect something by making it secret or very hard to understand is popularly derided as being an approach called security by obscurity. It is considered to be a poor approach since it simply attempts to hide an object, and it doesn't implement a security control to protect it. The problem is that information wants to be free and, eventually, someone will figure out how the system works. An organization can use security by obscurity measures to try to hide critical assets, but other security measures should also be employed to provide a higher level of protection.

Slide 8



Principles of Computer Security, Fifth Edition

Least Common Mechanism

- Protection mechanisms should be shared to the least degree possible among users.
- Sharing is caring and a security risk
- Examples
  - Payroll viewing and payroll editing
  - Covert channels
- Isolation
  - Sandboxing
  - Virtual Machines
  - Instantiating shared libraries

The principle of **Least common mechanism** states that mechanisms used to access resources should be dedicated and not shared.

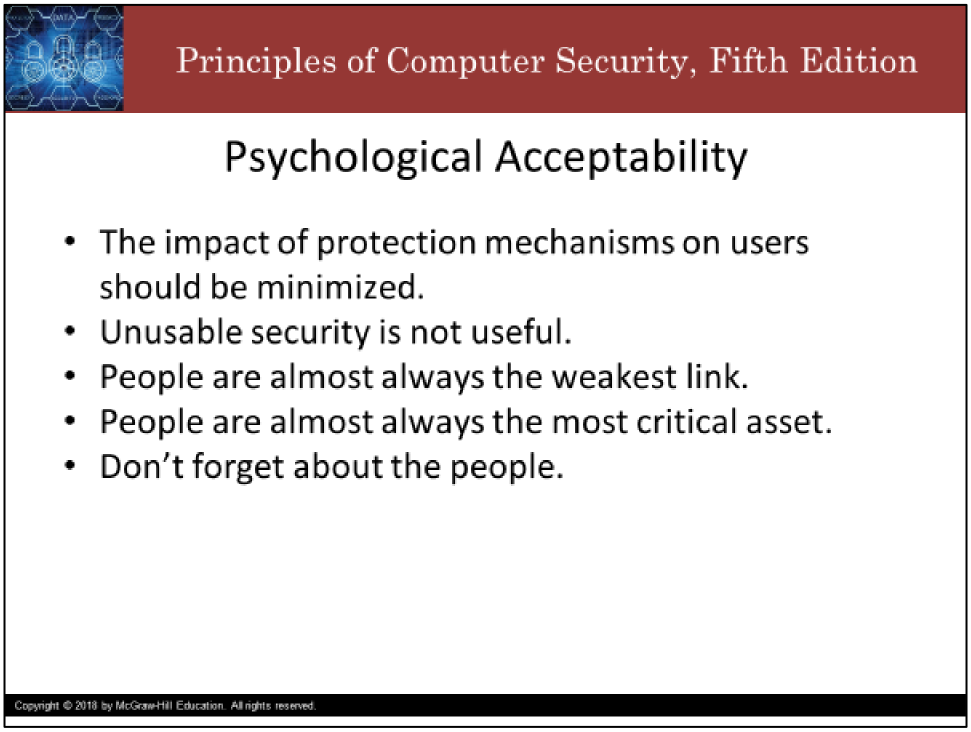
The primary motivation for this principle is that the sharing of mechanisms allows a potential cross-over between channels resulting in a protection failure mode.

One example of this type of failure relates to the principles of least privilege and separation of privilege. A system which allows employees to view their payroll information should be separate from the system which allows payroll information to be changed. It's the same data, but the mechanism of access needs to be kept separate to prevent, for example, employees giving themselves raises every year or reducing the wages of others.

Another example, which is a bit more pure in the sense that the LCM principle captures it best, is the problem of covert channels. In secure systems, there are often processes (software and human) that should not communicate with each other. One example of this would be process that works with classified information and processes which work with unclassified information. We don't want classified information flowing into unclassified systems, as this would undermine the security goal of confidentiality. If those processes share any resource in a way that makes even a single bit of information available to both, that resource can be used to create a covert channel (a channel which was not intended to be used for information transfer), and information can leak from one process to the other.

Common examples of the **least common mechanism** and its isolation principle are common in ordinary systems. Sandboxing is a means of separating the operation of an application from the rest of the operating system. Virtual machines perform the same task between operating systems on a single piece of hardware. Instantiating shared libraries, in which separate instantiation of local classes enables separate but equal coding is yet another. The key is to provide a means of isolation between processes so information cannot flow between separate users unless specifically designed to do so.

Slide 9



Principles of Computer Security, Fifth Edition

Psychological Acceptability

- The impact of protection mechanisms on users should be minimized.
- Unusable security is not useful.
- People are almost always the weakest link.
- People are almost always the most critical asset.
- Don’t forget about the people.

Also called the principle of least astonishment, the principle of **Psychological acceptability** applies to the users' acceptance – and, importantly, usage -- of security measures.

Users play a key role in the operation of a system, and if security measures are perceived to be an impediment to the work a user is responsible for, then a natural consequence may be that the user bypasses the control. Although a user may understand that this could result in a security problem, the perception that it gets in the way of their work will present pressure to bypass it.

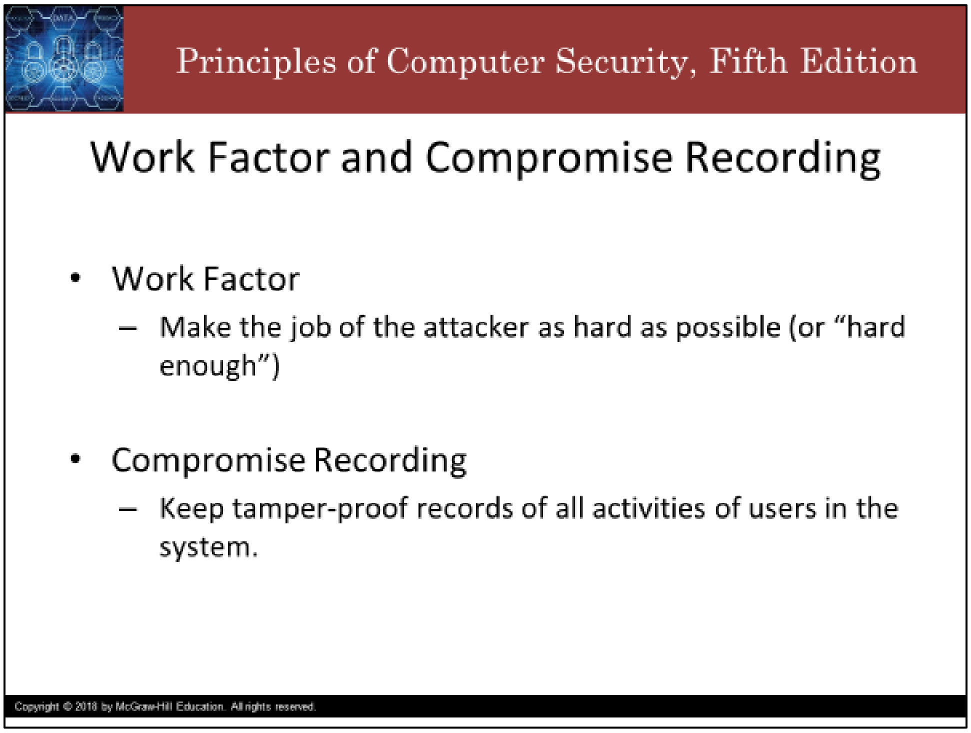
An apocryphal example of this is requiring long and complex passwords, which often results in the practice of writing passwords on paper and posting them on boards and monitors and under keyboards.

This highlights another extremely important facet of security. The fact that this is but 1 of the 8 principles you have so far seen should not be taken as an indication that it is but 1/8th of the total picture. Ask almost any security practitioner, and they will tell you that the human element is by far the most challenging and critical part of security.

Psychological acceptability is sometimes overlooked by security professionals who focus too much on technical issues and how they perceive the threat. They are focused on the threat, which is their professional responsibility, so the focus on security is natural, and it aligns with their professional responsibilities. However, this alignment between security and professional work responsibilities does not always translate to other positions in an organization. Security professionals, particularly those designing the security systems, should not only be aware of this concept but pay particular attention to how security controls will be viewed by workers in the context of their work responsibility and not with respect to security for its own sake.

I hesitate to call myself an expert in security, but I grant that I am more experienced than most people who do not work in cybersecurity. I am perfectly comfortable with jumping through hoops and paying the costs of things like 2-factor authentication, very long passwords, and complete mediation. But, I know many people who think 2-factor authentication is a pain or who have criminally short passwords or get annoyed by having to re-authenticate whenever they access a new system. My level of psychological acceptability is different than yours. I have to be aware of this. Because if I design a system with only myself in mind, it probably won't be psychologically acceptable to most people, and therefore it either won't be used, or various workarounds will be employed that degrade and subvert the protection mechanisms.

Slide 10



Principles of Computer Security, Fifth Edition

Work Factor and Compromise Recording

- Work Factor
  - Make the job of the attacker as hard as possible (or "hard enough")
- Compromise Recording
  - Keep tamper-proof records of all activities of users in the system.

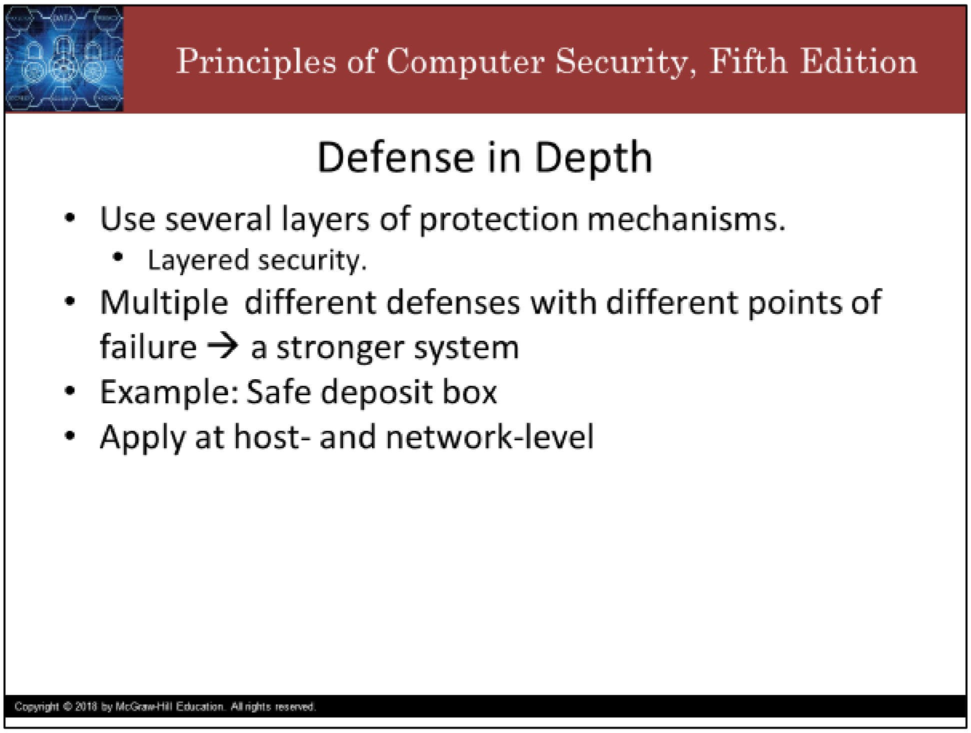
There are two more principles that I am sad to see get omitted from lists of foundational security principles based on those published in 1975 by Saltzer and Schroeder, which is where the preceding principles came from. The textbook misses them, too, despite citing the original source. They should not be missed.

One is work factor. It simply says to design the system such that the work required by the attacker to defeat the protection mechanisms is as hard as possible. Actually, with the more mature approach to security through risk management, I ought to say that work factor should be high enough that the risk of an attacker succeeding is brought down to an acceptable level.

The other is compromise recording. It simply says that the system should include a tamper-proof mechanism for recording the activities of the users of the system. The idea behind keeping these records is, as the name implies, to enable posthoc analysis of the actions of an attacker in order to determine what assets were attacked and in what ways and to inform the incident response process. With a risk-management approach, it may not be necessary (or useful) to record all activities. There is some level of recording, or some set of activities to record, that will be sufficient to capture the most salient information for incident response.

These principles are actually hiding in other places. Work factor is a crucial part of defense in depth. Compromise recording is a crucial part of auditability, one of the triple-A security goals that go along with CIA.

Slide 11



Principles of Computer Security, Fifth Edition

Defense in Depth

- Use several layers of protection mechanisms.
  - Layered security.
- Multiple different defenses with different points of failure → a stronger system
- Example: Safe deposit box
- Apply at host- and network-level

Speaking of defense-in-depth, here it is!

Defense in depth is a principle characterized by the use of multiple, different defense mechanisms with the goal of improving the defensive security posture.

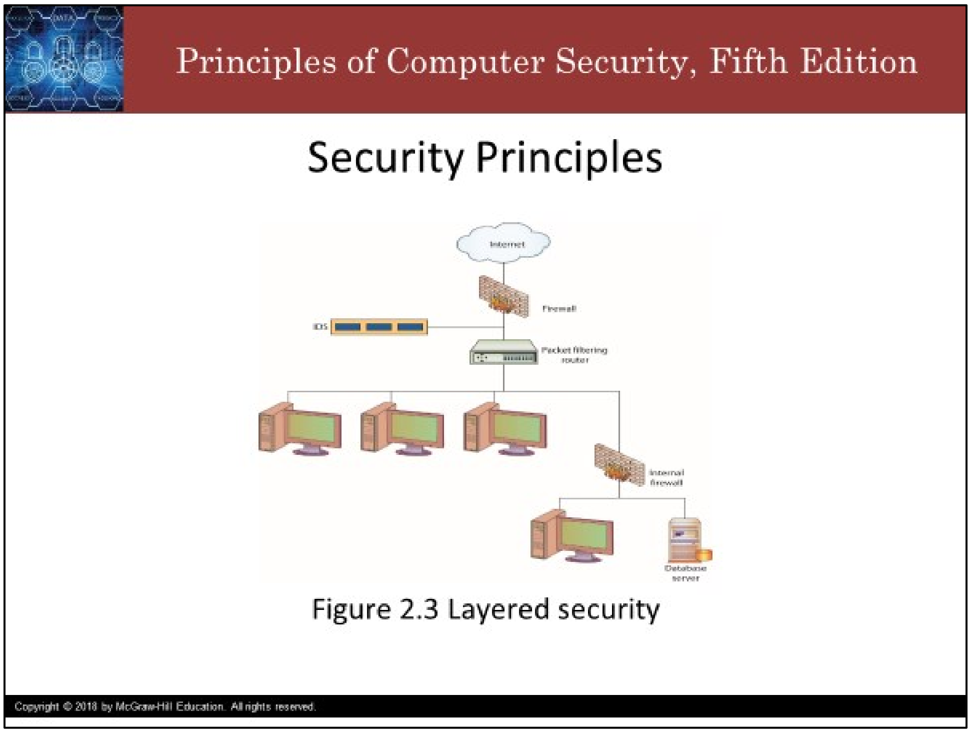
Another term for defense in depth is **layered security**.

Single points of failure represent just that, an opportunity to fail. By using multiple defenses that are different, with differing points of failure, a system becomes stronger. While one defense mechanism may not be 100 percent effective, the application of a second defense mechanism to the items that succeed in bypassing the first mechanism provides a stronger response. There are a couple of different mechanisms that can be employed in a **defense-in-depth** strategy: **layered security** and diversity of defense. Together these provide a defense-in-depth strategy that is stronger than any single layer of defense.

Examples of defense in depth abound. I've already given you one example from the movie Diehard. Another example is a safe deposit box at a bank. The box requires 2 keys, one which you have and one which the bank has. The bank's key can only be used by a bank employee. The box is in a vault. Access to the vault and the bank employee's access to their key requires you to authenticate yourself as one who has authority to access the box (the possession of the key is necessary but not sufficient). The vault is only accessible at certain times (i.e., when the bank is open). Access to the bank itself may be protected by guards and other access control mechanisms. So, if you wanted to access without authorization the contents of a safe deposit box (i.e., "steal"), your best bet is probably social engineering (attacking the people). All other methods, like tunneling in, are likely more work than reward.

Networks should utilize the same type of layered security architecture.

Slide 12



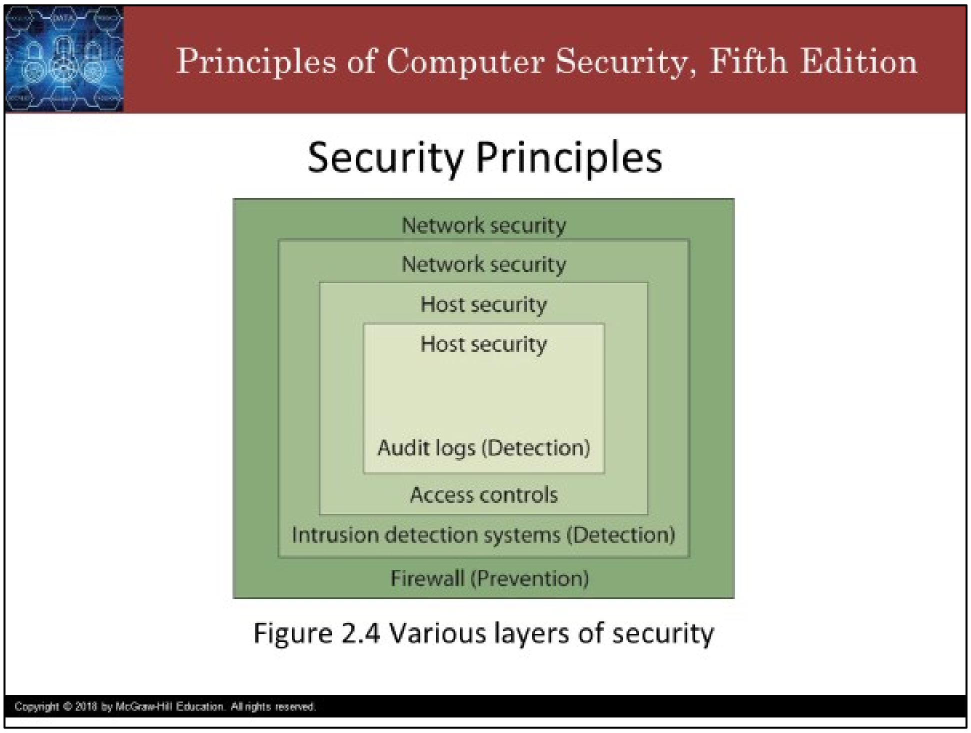
Principles of Computer Security, Fifth Edition

Security Principles

Figure 2.3 Layered security

There is no 100 percent secure system, and there is nothing that is foolproof, so a single specific protection mechanism should never be solely relied upon. It is important that every environment have multiple layers of security. These layers may employ a variety of methods, such as routers, firewalls, network segments, IDSs, encryption, authentication software, physical security, and traffic control. The layers need to work together in a coordinated manner so that one does not impede another's functionality and introduce a security hole.

Slide 13



Principles of Computer Security, Fifth Edition

Security Principles

Network security

Network security

Host security

Host security

Audit logs (Detection)

Access controls

Intrusion detection systems (Detection)

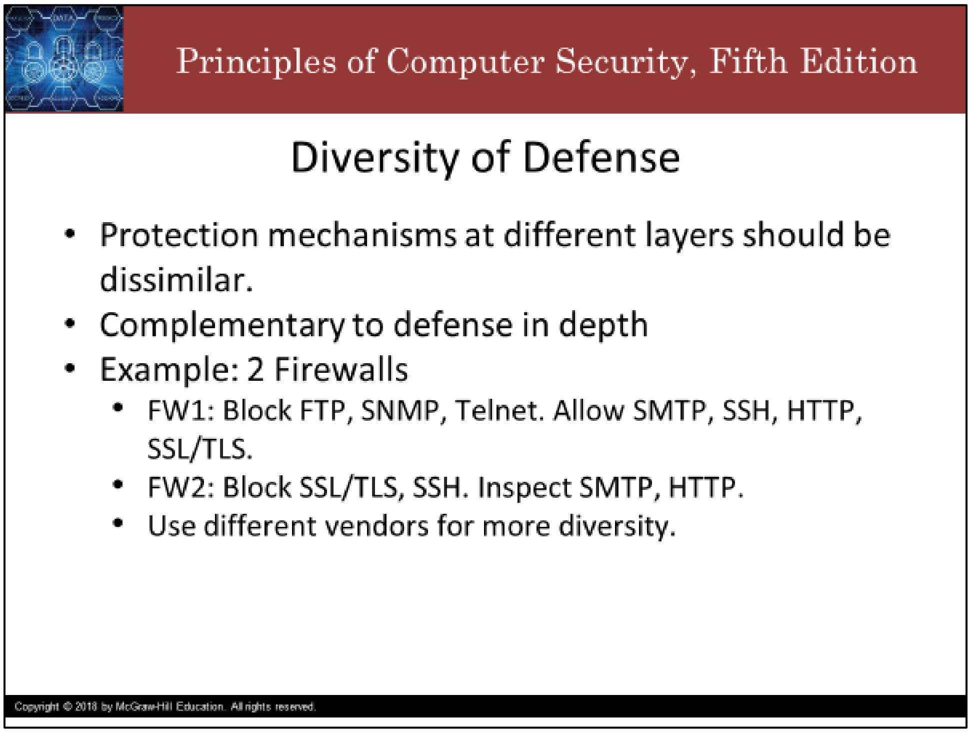
Firewall (Prevention)

Figure 2.4 Various layers of security

An example of how different security methods can work against each other is exemplified when firewalls encounter encrypted network traffic. An organization may utilize encryption so that an outside customer communicating with a specific web server is assured that sensitive data being exchanged is protected. If this encrypted data is encapsulated within Secure Sockets Layer (SSL) or Transport Layer Security (TLS) packets and then sent through a firewall, the firewall may not be able to read the payload information in the individual packets.

The layers usually are depicted starting at the top, with more general types of protection, and progressing downward through each layer, with increasing granularity at each layer as you get closer to the actual resource. This is because the top-layer protection mechanism is responsible for looking at an enormous amount of traffic, and it would be overwhelming and cause too much of a performance degradation if each aspect of the packet were inspected. Instead, each layer usually digs deeper into the packet and looks for specific items. Layers that are closer to the resource have to deal with only a fraction of the traffic that the top-layer security mechanism does, and thus looking deeper and at more granular aspects of the traffic will not cause as much of a performance hit.

Slide 14



Principles of Computer Security, Fifth Edition

Diversity of Defense

- Protection mechanisms at different layers should be dissimilar.
- Complementary to defense in depth
- Example: 2 Firewalls
  - FW1: Block FTP, SNMP, Telnet. Allow SMTP, SSH, HTTP, SSL/TLS.
  - FW2: Block SSL/TLS, SSH. Inspect SMTP, HTTP.
  - Use different vendors for more diversity.

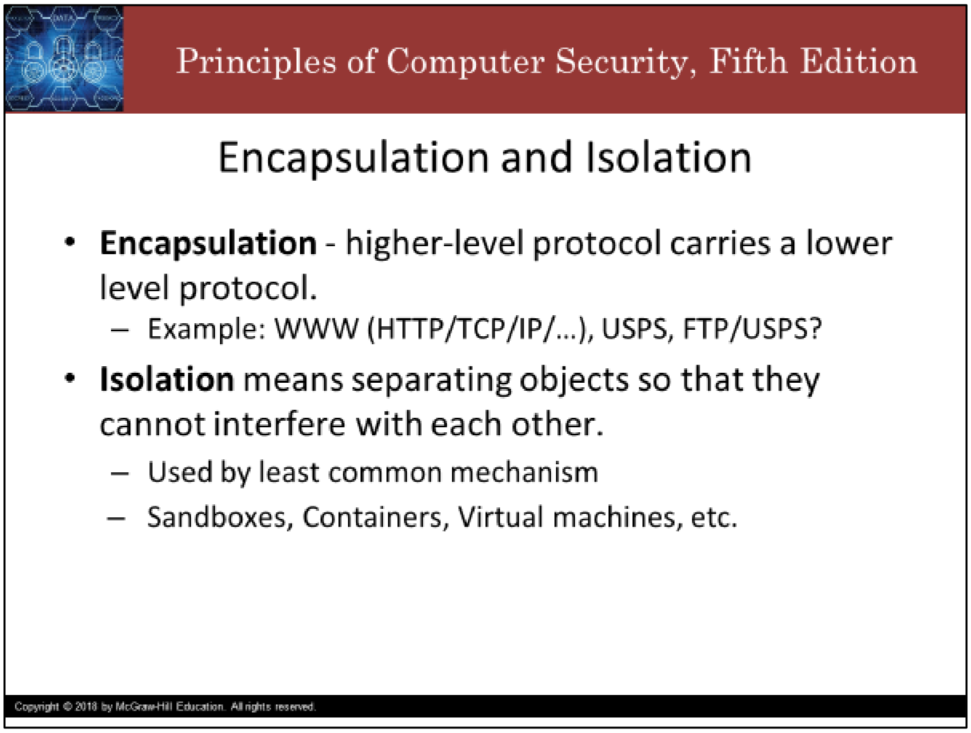
Diversity of defense is a concept that complements defense in depth.

It involves making different layers of security dissimilar.

If attackers know how to get through a system that comprises one layer, they may not know how to get through a different type of layer that employs a different system for security.

If an environment has two firewalls that form a demilitarized zone (DMZ), for example, one firewall may be placed at the perimeter of the Internet and the DMZ. This firewall analyzes the traffic that is entering through the specific access point and enforces certain types of restrictions. The other firewall may then be placed between the DMZ and the internal network. When applying the diversity-of-defense concept, you should set up these two firewalls to filter for different types of traffic and provide different types of restrictions. The first firewall, for example, may make sure that no FTP, SNMP, or Telnet traffic enters the network but allow SMTP, SSH, HTTP, and SSL traffic through. The second firewall may not allow SSL or SSH through and may interrogate SMTP and HTTP traffic to make sure that certain types of attacks are not part of that traffic.

Slide 15



Principles of Computer Security, Fifth Edition

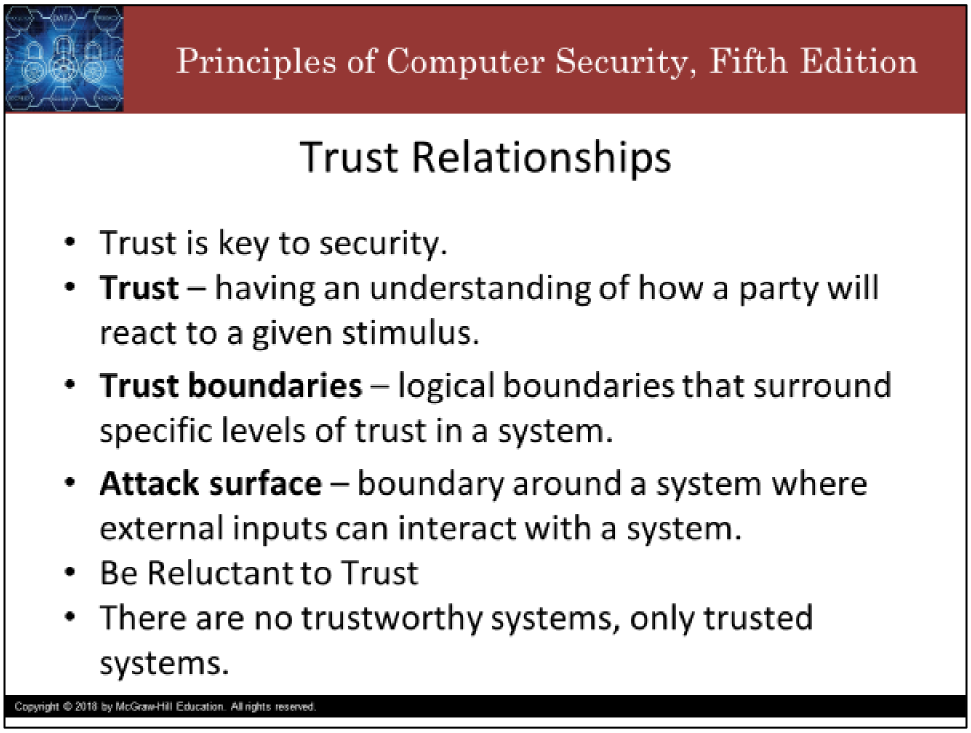
Encapsulation and Isolation

- **Encapsulation** - higher-level protocol carries a lower level protocol.
  - Example: WWW (HTTP/TCP/IP/…), USPS, FTP/USPS?
- **Isolation** means separating objects so that they cannot interfere with each other.
  - Used by least common mechanism
  - Sandboxes, Containers, Virtual machines, etc.

Encapsulation is when a higher-level protocol carries a lower-level protocol. The lower protocol is contained (encapsulated) in the data portion of the higher protocol. Two very familiar examples of this are TCP over IP and the US Postal Service. The web is delivered in many ways, one of the most common being hypertext. The hypertext transfer protocol (HTTP) is an application-layer protocol. It rides across the Internet in the data portion of transmission control protocol (TCP) packets at the transport layer. TCP packets ride in the data portion of Internet protocol (IP) packets at the Internet layer. IP packets ride in the data portion of a link-layer protocol, which in turn is sent out along physical connections like Fiber, WiFi, Ethernet, Bluetooth. The postal service encapsulates correspondence in envelopes and packages. USPS handles routing the envelope or package to its destination. That correspondence can be anything, like coursework, or legal documents, or even HTTP requests, or file transfers. You may laugh… but considering the price of storage and the uninspiring speeds of the US broadband network, once the data gets big enough, it is, in fact, faster to put data onto a hard drive and drop it in the mail than it is to send it over the Internet.

Isolation is a principle used by the least common mechanism. Isolation means separating objects (like data and processes) so that they cannot interfere with each other. In the immortal words of The Offspring: "You gotta keep 'em separated." The most common examples of isolation in practice are sandboxes, containers, and virtual machines.

Slide 16



Principles of Computer Security, Fifth Edition

Trust Relationships

- Trust is key to security.
- **Trust** – having an understanding of how a party will react to a given stimulus.
- **Trust boundaries** – logical boundaries that surround specific levels of trust in a system.
- **Attack surface** – boundary around a system where external inputs can interact with a system.
- Be Reluctant to Trust
- There are no trustworthy systems, only trusted systems.

Trust is a key principle in security. I would even go so far as to say that it is the first principle. So, in this case, it is last, but not least, and in fact, most.

Trust means knowing (having justified belief) what someone else will do given a particular stimulus. For example, I trust that when I say Howdy to my students, they will say Howdy back to me.

Security decisions are based on trust relationships.

When you establish a trust relationship between two systems, you are granting objects on one system access to view or modify objects on the other system, and possibly vice versa. You are trusting, to some extent (the least possible extent), that those objects will act correctly and securely.

Changes in trust occur at trust boundaries, which are logical boundaries around specific levels of trust in a system.

When information from outside the system enters the system as input, it is crossing a trust boundary. A decision must be made as to whether to trust the information. Generally speaking, data from users is not trusted as-is. It must undergo input validation. The user must be authenticated. The operation must be authorized. Only then will the data be trusted.

The boundary around a system where data crosses from outside to inside has the special name: attack surface.

A key practice for securing systems is minimizing the attack surface, which means limiting the trust of information coming from outside the system.

Many, if not all, security failures are rooted in a violation of a trust relationship. Because of the nature of trust (in order for connected systems to interoperate, it must be a 2-way street) and the level of risk it entails, the sage advice is to develop and maintain a culture of reluctance to trust.

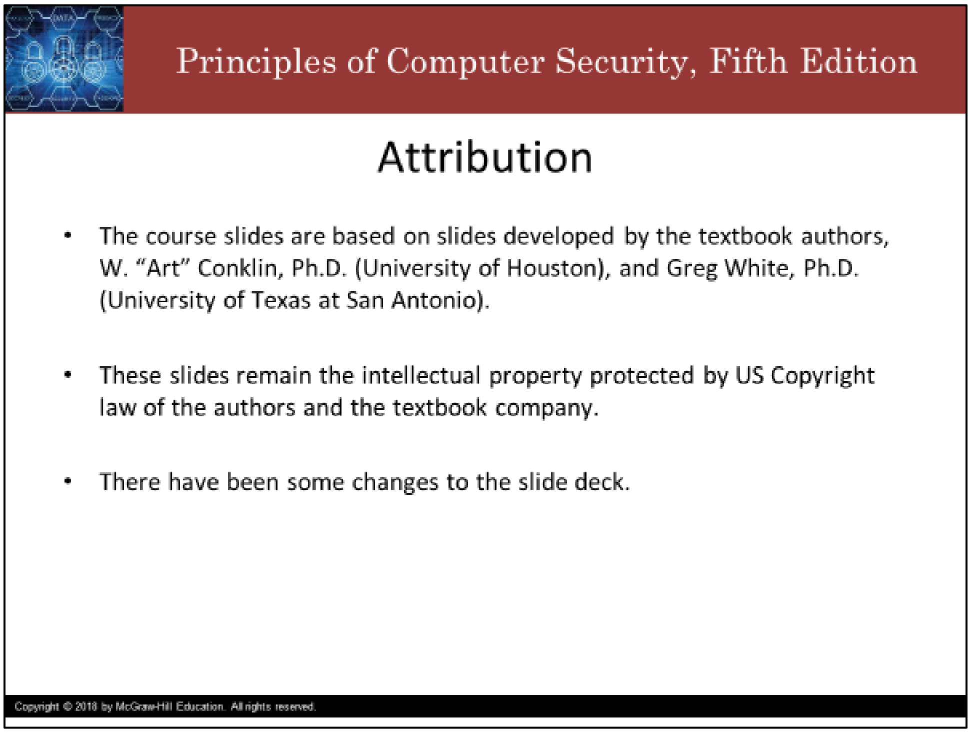
Trust should be limited to only that which is needed for correct operation.

Excessive trust increases risk without benefit.

Another bit of advice is that there are no trustworthy systems, only trusted systems.

These last two are good life advice, too.

Slide 17



Principles of Computer Security, Fifth Edition

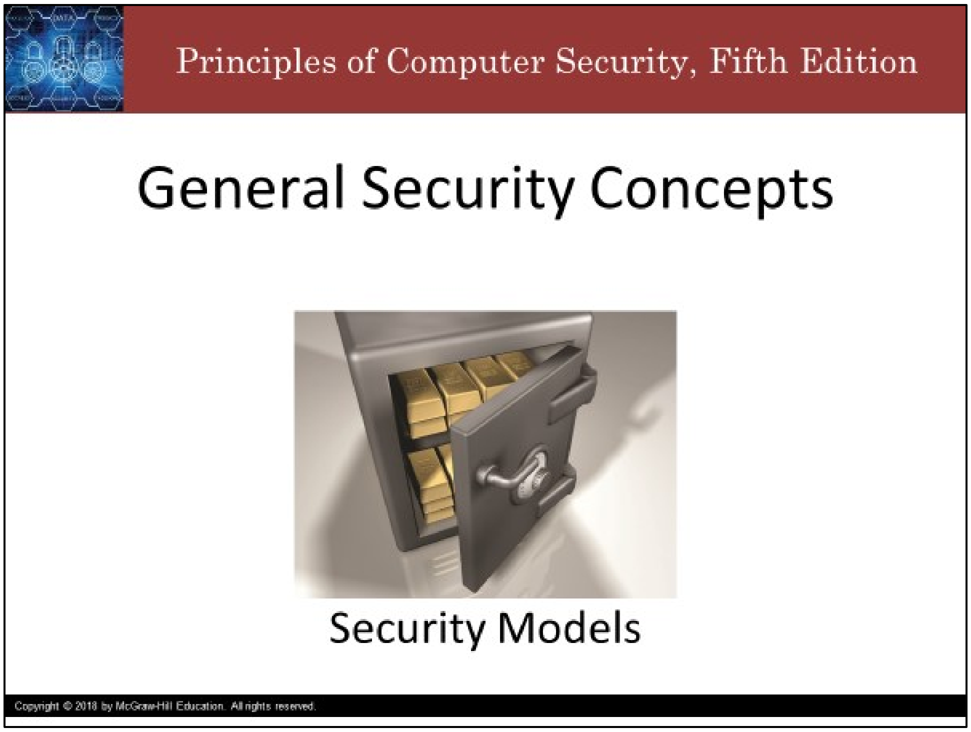
Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Thank you and take care.

General Security Concepts: Security Models

Slide 1



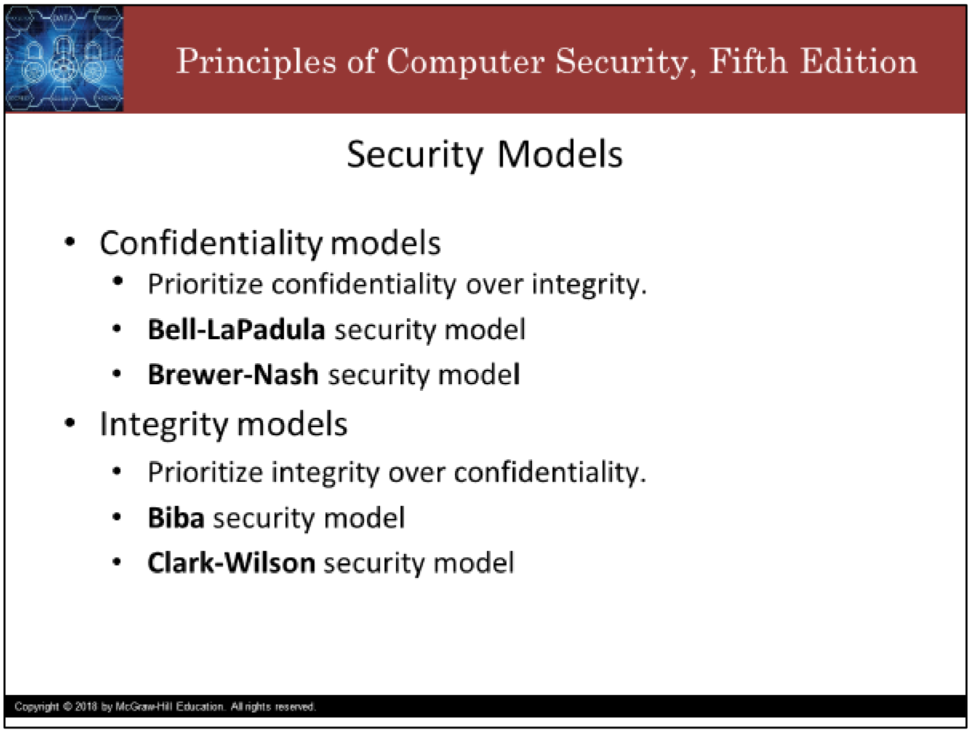
Principles of Computer Security, Fifth Edition

General Security Concepts

Security Models

Howdy! In this video, we discuss confidentiality and integrity security models. In particular, we will see the Bell-LaPadula, Brewer-Nash, Biba, and Clark-Wilson security models.

Slide 2



Principles of Computer Security, Fifth Edition

Security Models

- Confidentiality models
  - Prioritize confidentiality over integrity.
  - **Bell-LaPadula** security model
  - **Brewer-Nash** security model
- Integrity models
  - Prioritize integrity over confidentiality.
  - **Biba** security model
  - **Clark-Wilson** security model

An important issue when designing the software that will operate and control secure computer systems and networks is the security model that the system or network will be based upon. The security model dictates the implementation of the security policy that has been chosen and the mechanisms that enforce the properties deemed most important by the system designers.

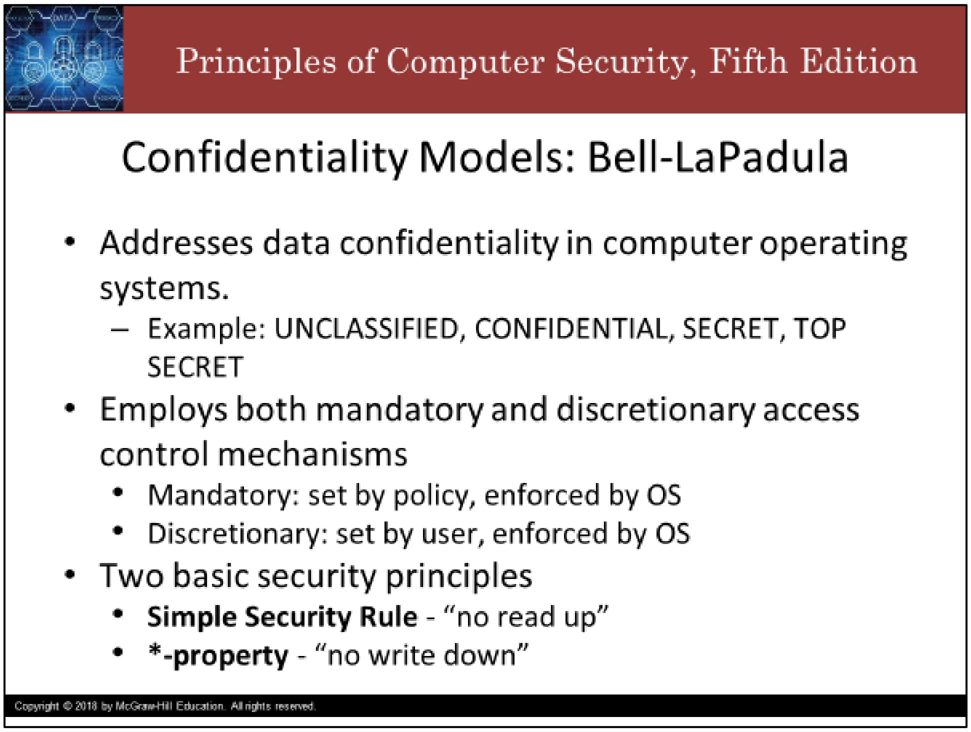
For example, if confidentiality is considered paramount, the model should make certain no data is disclosed to unauthorized individuals. A model enforcing confidentiality may allow unauthorized individuals to modify or delete data, as this would not violate the tenets of the model because the true values for the data would still remain confidential. Of course, this model may not be appropriate for all environments.

In some instances, the unauthorized modification of data may be considered a more serious issue than its unauthorized disclosure. In such cases, the model would be responsible for enforcing the integrity of the data instead of its confidentiality.

Choosing the model on which to base the design and implementation of the protection mechanisms is critical if you want to ensure that the resulting system accurately enforces the security policy desired. This, however, is only the starting point, and it does not imply that you have to make a mutually exclusive choice between confidentiality and integrity, as both are important. Not to mention, there are other security goals like availability that the policies and mechanisms must address.

We will see two very well-known examples of each of confidentiality models and integrity models. For confidentiality, we will see the Bell-LaPadula and Brewer-Nash models. For integrity, we will see the Biba and Clark-Wilson models.

Slide 3



Principles of Computer Security, Fifth Edition

Confidentiality Models: Bell-LaPadula

- Addresses data confidentiality in computer operating systems.
  - Example: UNCLASSIFIED, CONFIDENTIAL, SECRET, TOP SECRET
- Employs both mandatory and discretionary access control mechanisms
  - Mandatory: set by policy, enforced by OS
  - Discretionary: set by user, enforced by OS
- Two basic security principles
  - **Simple Security Rule** - “no read up”
  - ***-property** - “no write down”

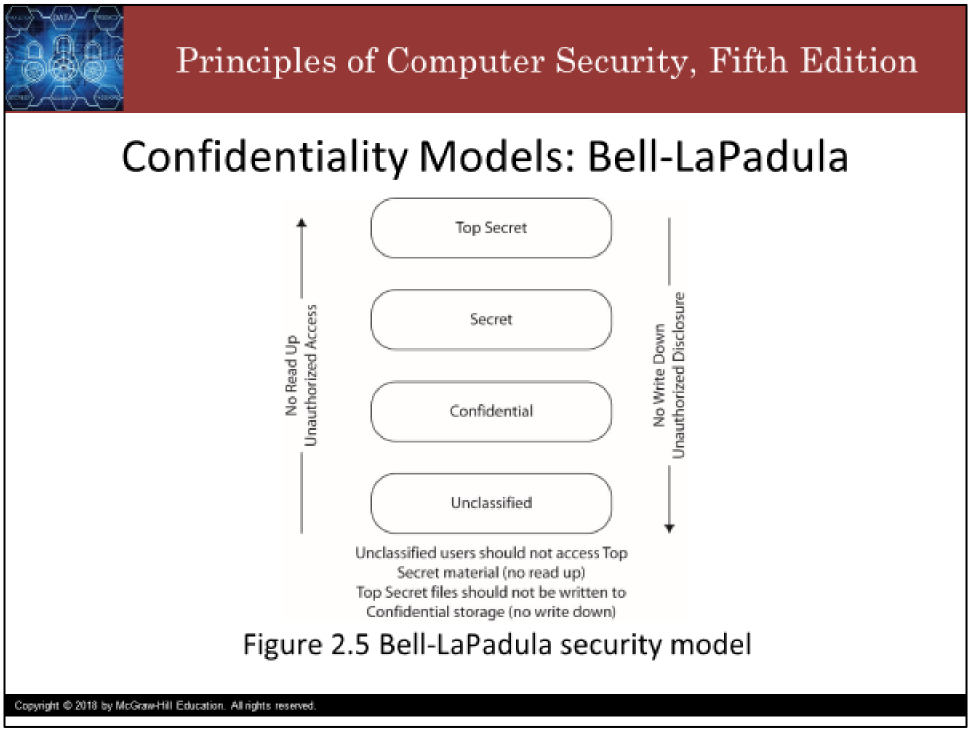
The U.S. military encouraged the development of the **Bell-LaPadula security model** to address data confidentiality in computer operating systems. This model is especially useful in designing multilevel security systems that implement the military's hierarchical security scheme, which includes levels of classification such as *Unclassified, Confidential, Secret,* and *Top Secret.* Similar classification schemes can be used in industry, where classifications might include *Publicly Releasable, Proprietary,* and *Company Confidential.*

Bell-LaPadula employs both mandatory and discretionary access control mechanisms to implement its two basic security principles.

The first of these principles is called the **Simple Security Rule**, which states that no subject can read information from an object with a security classification higher than that possessed by the subject itself. This means that the system must prevent a user with only a Secret clearance, for example, from reading a document labeled Top Secret. This rule is often referred to as the "no read up" rule.

The second security principle enforced by the Bell-LaPadula security model is known as the ***-property**. This principle states that a subject can write to an object only if the target's security classification is greater than or equal to the object's security classification. This means that a user with a Secret clearance can write to a file with a Secret or Top-Secret classification but cannot write to a file with only a Confidential classification. This, at first, may appear to be a bit confusing since this principle allows users to write to files that they are not allowed to view, thus enabling them to actually destroy files that they don't have the classification to see. This is true, but keep in mind that the Bell-LaPadula model is designed to enforce confidentiality, not integrity. Writing to a file that you don't have the clearance to view is not considered a confidentiality issue; it is an integrity issue.

Slide 4



Principles of Computer Security, Fifth Edition

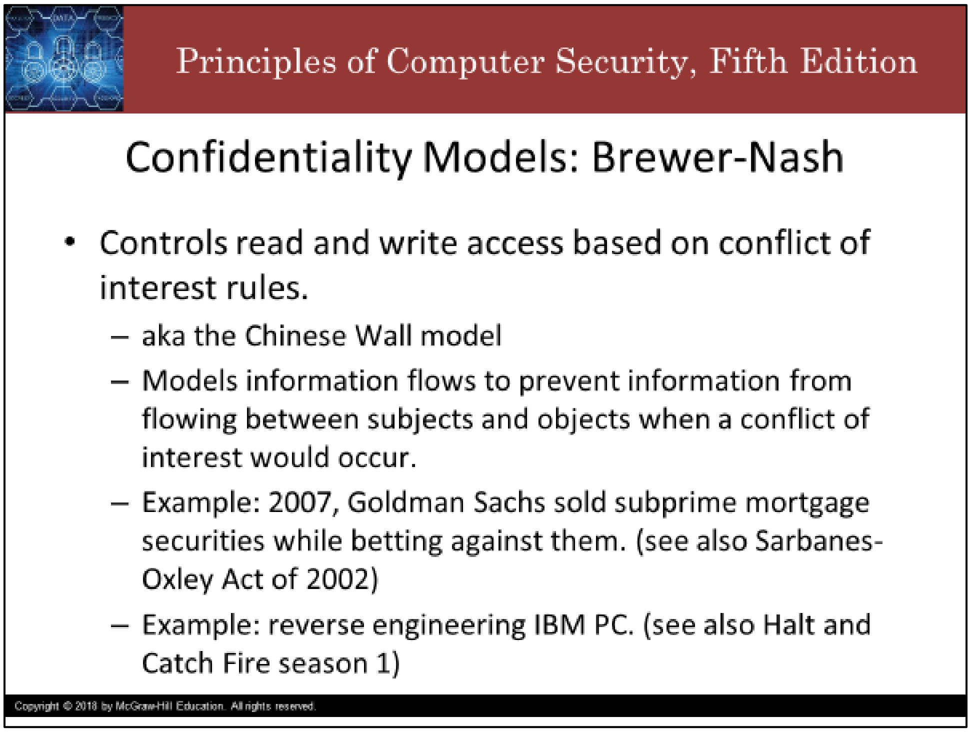
Confidentiality Models: Bell-LaPadula

Unclassified users should not access Top Secret material (no read up)
Top Secret files should not be written to Confidential storage (no write down)

Figure 2.5 Bell-LaPadula security model

The star property allows a subject to write to objects at equal or higher security classification but prohibits writing to objects at lower levels of security classification. Why not? Shouldn't a subject with SECRET clearance who can read UNCLASSIFIED objects be allowed to write to them? From a confidentiality security perspective, the answer is NO. The reason for this prohibition is to mitigate the risk of information disclosure from high security to low security. The system is designed to prevent data to be disclosed to those without the authorization to view it. If it were possible for a subject with TOP SECRET classification to write to an object with CONFIDENTIAL classification, then some other subject which only has CONFIDENTIAL classification could view it, which would be a violation of the confidentiality security policy. This is why Bell-LaPadula enforces the no write down rule.

Slide 5



Principles of Computer Security, Fifth Edition

Confidentiality Models: Brewer-Nash

- Controls read and write access based on conflict of interest rules.
  - aka the Chinese Wall model
  - Models information flows to prevent information from flowing between subjects and objects when a conflict of interest would occur.
  - Example: 2007, Goldman Sachs sold subprime mortgage securities while betting against them. (see also Sarbanes-Oxley Act of 2002)
  - Example: reverse engineering IBM PC. (see also Halt and Catch Fire season 1)

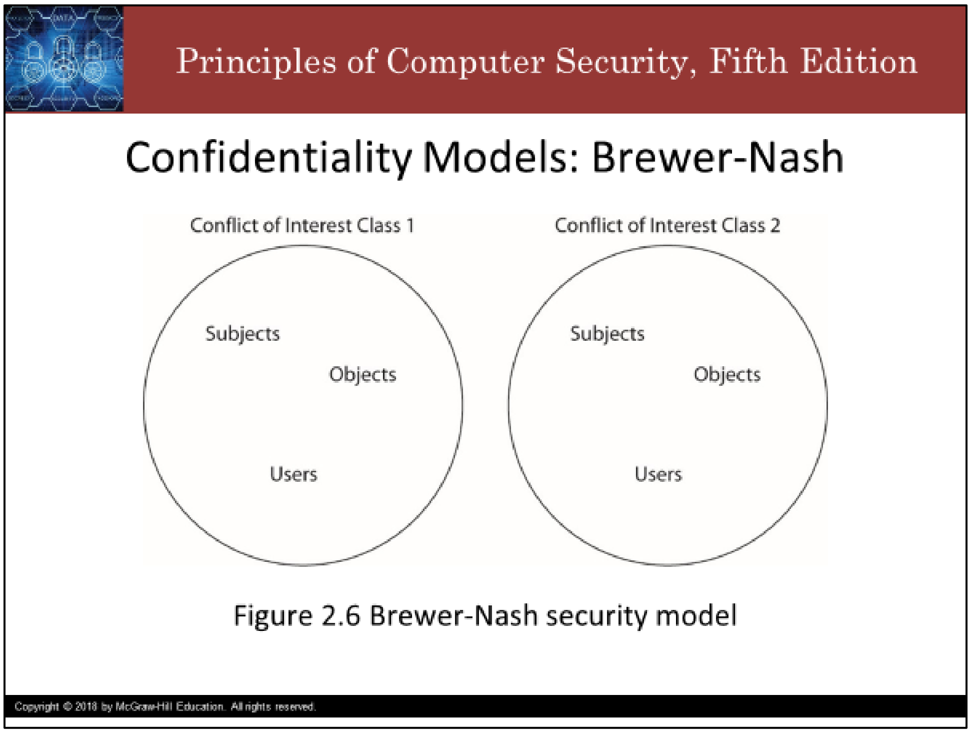
One of the tenets associated with access is need to know. Separate groups within an organization may have differing needs with respect to access to information. The **Brewer-Nash security model** takes into account user conflict-of-interest aspects and uses them to control read and write access.

Brewer-Nash enforces a wall between different groups of an organization who have different interests. For this reason, it is also known as the Chinese Wall model, named after the Great Wall of China.

There are many examples of this model in practice. It is actually used in several fields besides security, including finance, journalism, and law. A famous violation of the model (a context in which the wall was supposed to be enforced but was not) was committed by Goldman-Sachs in 2007 when they marketed and sold subprime mortgage-backed securities with one hand and bet against them with the other.

Another example, this one from computing, and a bit more of a happy story (unless you're IBM) is that of reverse-engineering the BIOS firmware of an IBM PC to create a hardware clone that can run IBM PC applications, which helped to fuel the PC revolution and the diversity of computing options available today. This kind of reverse engineering process is still used today. It works by having two groups separated by a Chinese wall. One group works with the hardware to reverse engineer the driver code required to work with the hardware (they do not look at any proprietary code or documentation, they just work with the hardware and figure out from that interaction alone how the code must work). They then write up detailed documentation about what they learned about the algorithms and protocols that must be used. That information is then passed to the other team, which writes original code to meet the specifications in the reverse-engineered documentation. This process, called clean-room design, helps to legally protect the new code from claims of copyright infringement.

Slide 6



Principles of Computer Security, Fifth Edition

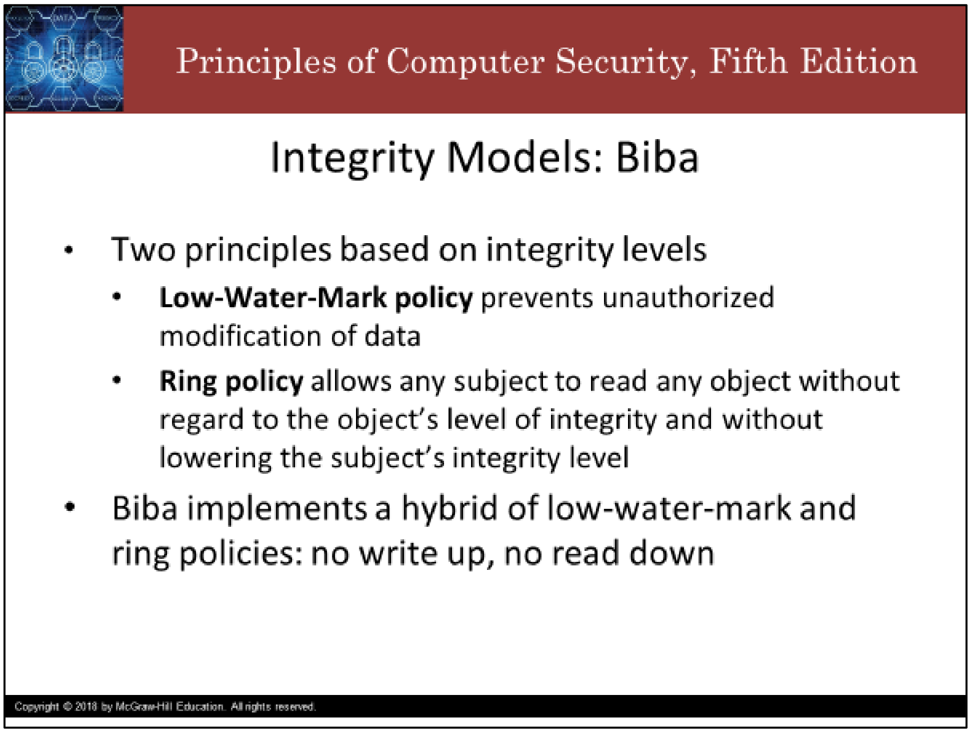
Confidentiality Models: Brewer-Nash

Figure 2.6 Brewer-Nash security model

To apply the Brewer-Nash security model, separate groups are defined, and access controls are designed to enforce the separation of the groups. Then, information flows are modeled to detect and prevent information flowing between subjects and objects where a conflict of interest exists, that is, between subjects and objects in different conflict of interest classes.

Another example of this model in practice is a company whose clients are competitors. An example of such a company could be a bank, or a research firm, or an equipment supplier. The data kept by this company about its clients must be separated by a Chinese wall so that one client cannot gain information about its competitors through its business with the company. In this type of scenario, the conflict of interest classes contain the data about the clients. Anyone who accesses the data for client A is prevented from accessing the data about client B, whose data is in a different conflict of interest class.

Slide 7



Principles of Computer Security, Fifth Edition

Integrity Models: Biba

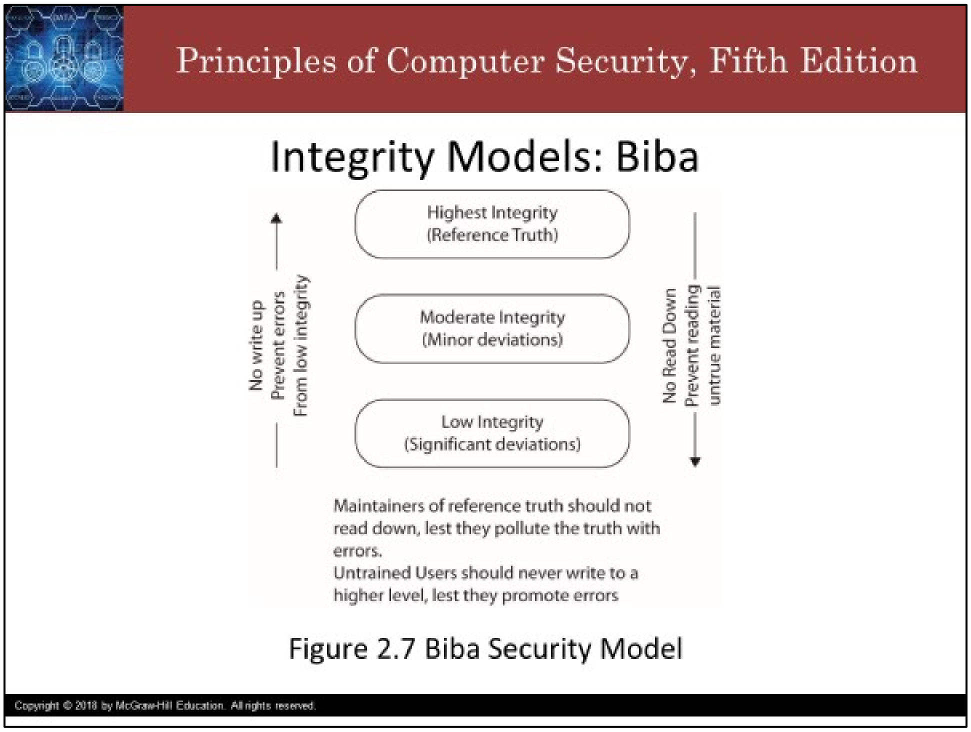
- Two principles based on integrity levels
  - **Low-Water-Mark policy** prevents unauthorized modification of data
  - **Ring policy** allows any subject to read any object without regard to the object’s level of integrity and without lowering the subject’s integrity level
- Biba implements a hybrid of low-water-mark and ring policies: no write up, no read down

A principle of integrity levels is that data with a higher integrity level is believed to be more accurate or reliable than data with a lower integrity level. Integrity levels indicate the level of "trust" that can be placed in information at the different levels. Integrity levels differ from confidentiality levels in another way—they limit the modification of information as opposed to the flow of information.

An initial attempt at implementing an integrity-based model was captured in what is referred to as the **Low-water-mark policy**. This policy, in many ways, is the opposite of the *-property of Bell-LaPadula in that it prevents subjects from writing to objects of a higher integrity level. The policy also contains a second rule that states that the integrity level of a subject will be lowered if it reads an object of a lower integrity level. The reason for this is that if the subject then uses data from that object, the highest the integrity level can be for a new object created from it is the same level of integrity of the original object. In other words, the level of trust you can place in data formed from data at a specific integrity level cannot be higher than the level of trust you have in the subject creating the new data object, and the level of trust you have in the subject can only be as high as the level of trust you had in the original data. The final rule contained in the Low-Water-Mark policy states that a subject can execute a program only if the program's integrity level is equal to or less than the integrity level of the subject. This ensures that data modified by a program only has the level of trust (or integrity level) that can be placed in the individual who executed the program.

While the Low-Water-Mark policy certainly prevents unauthorized modification of data, it has the unfortunate side effect of eventually lowering the integrity levels of all subjects to the lowest level on the system (unless the subject always views files with the same level of integrity). This is because of the second rule, which lowers the integrity level of the subject after accessing an object of a lower integrity level. There is no way specified in the policy to ever raise the subject's integrity level back to its original value. A second policy, known as the **Ring policy**, addresses this issue by allowing any subject to read any object without regard to the object's level of integrity and without lowering the subject's integrity level. This, unfortunately, can lead to a situation where data created by a subject after reading data of a lower integrity level could end up having a higher level of trust placed upon it than it should.

Slide 8



Principles of Computer Security, Fifth Edition

Integrity Models: Biba

Highest Integrity (Reference Truth)

Moderate Integrity (Minor deviations)

Low Integrity (Significant deviations)

No write up
Prevent errors
From low integrity

No Read Down
Prevent reading
untrue material

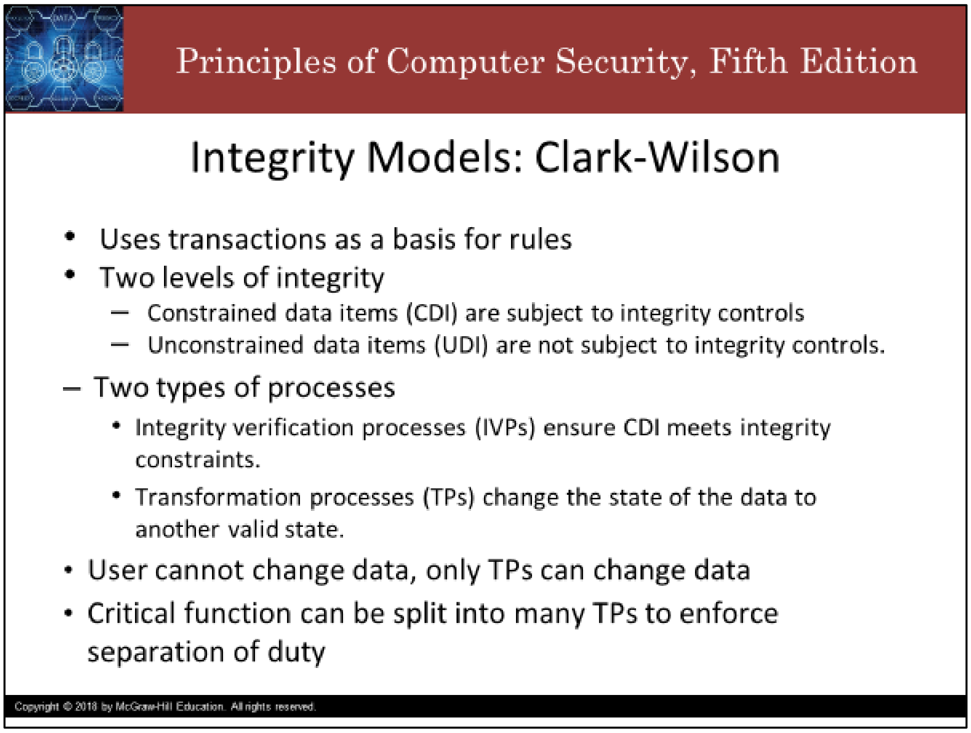
Maintainers of reference truth should not read down, lest they pollute the truth with errors.
Untrained Users should never write to a higher level, lest they promote errors

Figure 2.7 Biba Security Model

The Biba security model implements a hybrid of the Ring and Low-Water-Mark policies. Biba's model is, in a way, the opposite of the Bell-LaPadula model in that what it enforces are "no write-up" and "no-read-down" policies. It also implements a third rule that prevents subjects from executing programs of a higher level. The Biba security model thus addresses the problems mentioned with both the Ring and Low-Water-Mark policies.

One of my favorite tongue-in-cheek examples of the application of the Biba security model is that I can talk to my students, and they can listen to me because I am a higher-integrity process than they are, but I cannot listen to them since their low-integrity data would pollute my high-integrity Truth. This is why I do not respond to their emails. I am simply not allowed to read them!

Slide 9



Principles of Computer Security, Fifth Edition

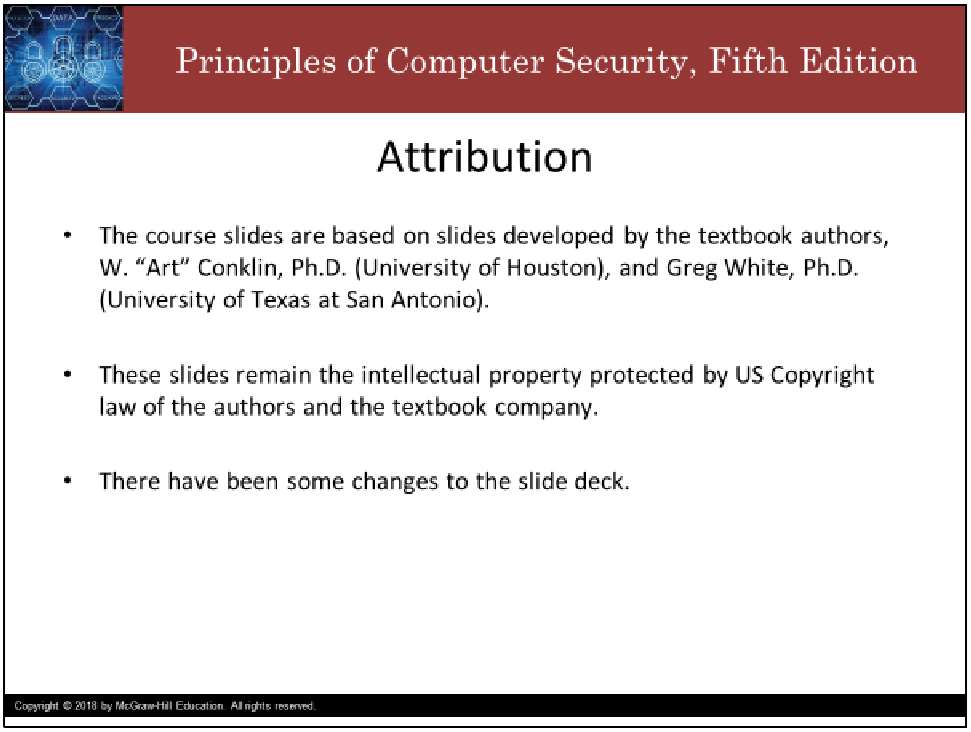
Integrity Models: Clark-Wilson

- Uses transactions as a basis for rules
- Two levels of integrity
  - Constrained data items (CDI) are subject to integrity controls
  - Unconstrained data items (UDI) are not subject to integrity controls.
- Two types of processes
  - Integrity verification processes (IVPs) ensure CDI meets integrity constraints.
  - Transformation processes (TPs) change the state of the data to another valid state.
- User cannot change data, only TPs can change data
- Critical function can be split into many TPs to enforce separation of duty

The Clark-Wilson security model takes an entirely different approach than the Biba and Bell-LaPadula models, using transactions as the basis for its rules. It defines two levels of integrity only: constrained data items (CDIs) and unconstrained data items (UDIs). CDI data is subject to integrity controls while UDI data is not. The model then defines two types of processes: integrity verification processes (IVPs), which ensure that CDI data meets integrity constraints (to ensure the system is in a valid state), and transformation processes (TPs), which change the state of data from one valid state to another. Data in this model cannot be modified directly by a user; it must be changed by trusted TPs, access to which can be restricted (thus restricting the ability of a user to perform certain activities).

As an example, consider account data held by a bank, things like name, address, account balance, the design of their checkbook. Keeping things like the account balance confidential is important, but it is more important that the account balance's integrity be maintained. In the Clark-Wilson model, the account balance would be a CDI because its integrity is a critical function for the bank. A client's preference for their checkbook is not a critical function and would be considered a UDI. Since the integrity of the account balance is of primary importance, changes to a person's account balance must be accomplished through the use of a TP. Ensuring that the balance is correct would be the duty of an IVP. Only certain employees of the bank should have the ability to modify an individual's account, which can be controlled by limiting the number of individuals who have the authority to execute TPs that result in account modification. Certain very critical functions may actually be split into multiple TPs to enforce the principle of separation of duties. This limits the authority any one individual has so that multiple individuals will be required to execute certain critical functions.

Slide 10



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. “Art” Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Thank you and take care.