BUFFALO STATE
The State University of New York

Cybersecurity First Principles

Prof. Charles Arbutina
&
Prof. Sarbani Banerjee

Computer Information Systems Department

Google CS[4]HS

What are First Principles?

The principles are basic, foundational propositions regarding what qualities of a system contribute to cybersecurity. These principles guide tradeoffs during system design that contribute to security.

The discussion of 10 cybersecurity first principles is adapted from **National Security Agency (NSA)**

1. Domain Separation
2. Process Isolation
3. Resource Encapsulation
4. Least Privilege
5. Layering
6. Abstaction
7. Information Hiding
8. Modularity
9. Simplicity
10. Minimization

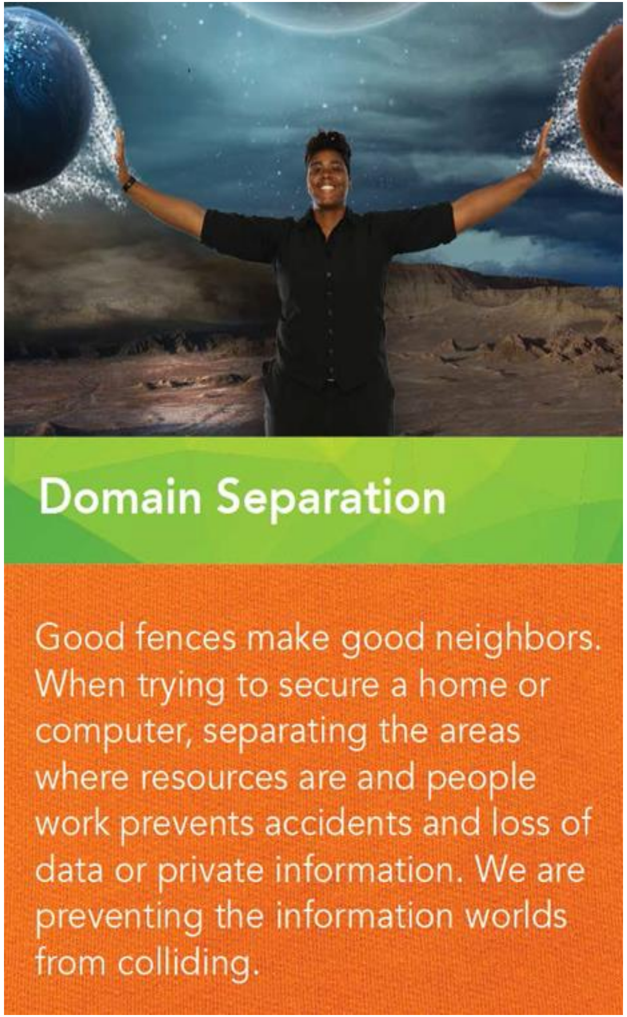
1. Domain Separation

What is a Domain?

- In a computer, domain refers to a collection of data or instructions that warrant protection.
- Outside of a computer, a domain can be an area of responsibility or control.
- Separating domains allows for enforcement of rules governing the use of domains by entities outside the domain.
- During system testing, test data should be separated from “real” data, such as personal info to avoid disclosure.

Examples

- Computer processors run in two domains: Supervisor & User In supervisor domain it can directly access RAM while in user domain, it cannot access memory other programs or the OS.
- A virtual machine is a domain that is separate from other virtual machines (or containers)



Domain Separation

Good fences make good neighbors. When trying to secure a home or computer, separating the areas where resources are and people work prevents accidents and loss of data or private information. We are preventing the information worlds from colliding.

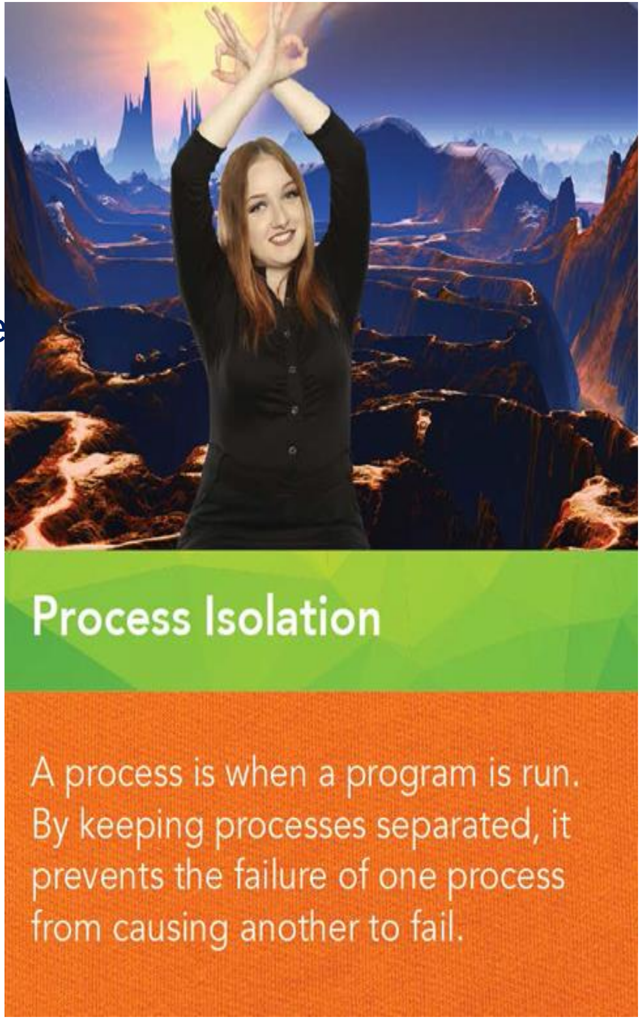
2. Process Isolation

What is a Process?

- A process is a program running on a computer. Each process has a region of memory (address), which only it can access.
- Isolating the process address space from other address space prevents tampering or interference from/by other processes.

Examples

- A word processor, a database, and a browser running on a computer are all running in different addresses spaces. Process isolation ensures that each one cannot influence the others address space.
- A non-technical example of process isolation is when a prosecutor and defense attorney run their cases in court. It would be a problem if either had access to each other’s work. Keeping their work separate protects it from misuse by the other party.



Process Isolation

A process is when a program is run. By keeping processes separated, it prevents the failure of one process from causing another to fail.

3. Resource Encapsulation

What is a Resource?

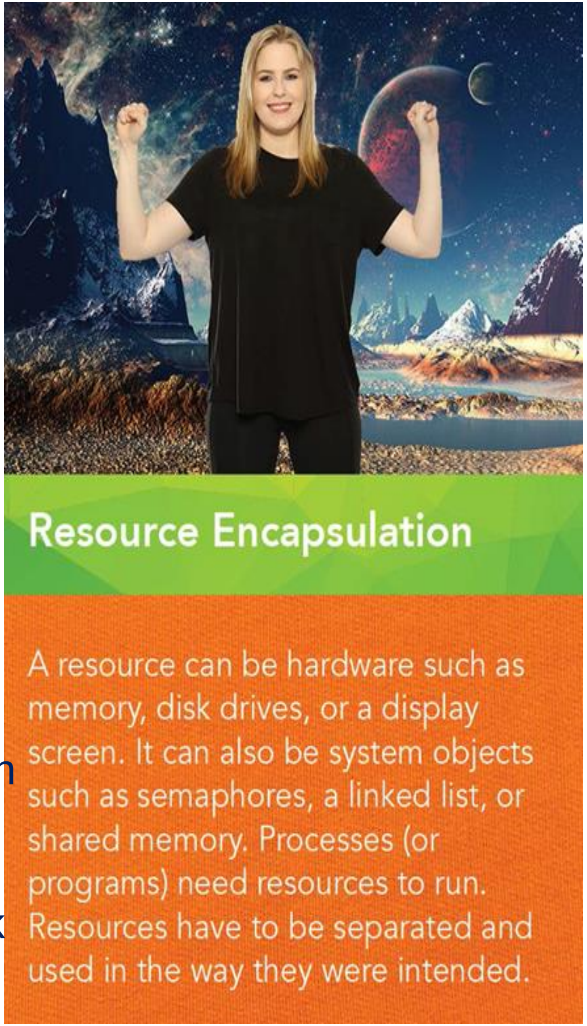
- A computer has many resources. It can be memory, disk drive, network bandwidth, battery power, or monitor. It can be system objects such as shared memory or a linked list data structure.

What is encapsulation?

- Encapsulation finds its origin in object-oriented programming. In OOP, a class definition encapsulates all data and functions to operate on the data. The goal is to allow access or manipulation of the class data in only the ways the designer intended.

Examples

- The application logic of a website allows access and manipulation of database records in defined ways. Here the database is a resource encapsulated by the website application logic.
- A flag pole allows certain operations (raise flag,lower flag,unhook flag). No one needs to know how flag pole works internally.



Resource Encapsulation

A resource can be hardware such as memory, disk drives, or a display screen. It can also be system objects such as semaphores, a linked list, or shared memory. Processes (or programs) need resources to run. Resources have to be separated and used in the way they were intended.

4. Least Privilege

What is a privilege?

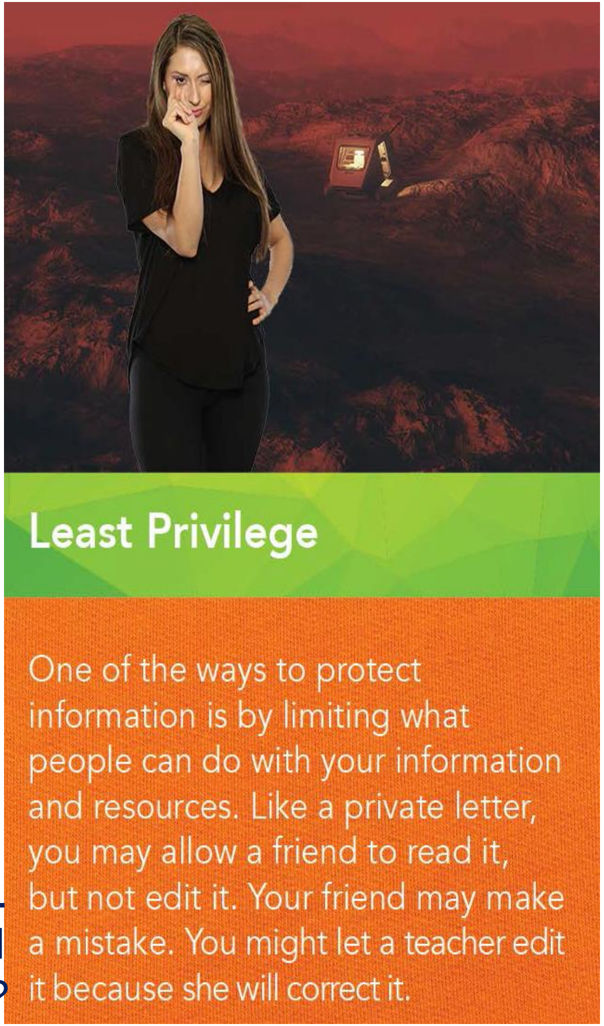
- It is a right for the user to act on managed computer resources.

Why should privileges be minimized?

- Minimizing the number of privileges granted to a user for accomplishing assigned duties improves accountability and limits accidental misuse.

Examples

- When a person gets a new computer, they log onto it using an administrative account which has privileges to install software & hardware, add or delete any users, program or file. Now, if the person opens a malicious phishing attachment, the malware will run with administrative privileges. If privileges were lowered to a regular user the malware wouldn’t have administrative privilege.
- If a user doesn’t need a permission, why give it to them? Should a military radio operator have permission to access nuclear silo?



Least Privilege

One of the ways to protect information is by limiting what people can do with your information and resources. Like a private letter, you may allow a friend to read it, but not edit it. Your friend may make a mistake. You might let a teacher edit it because she will correct it.

5. Layering

What is a Layer?

- In the context of computer security, a layer is a separate level that must be conquered by an attacker to breach a system.
- Layering slows down an attacker. The attacker needs to conquer each layer before moving on to the next.

Examples

- A moat is an outer layer that protects a castle. The next layer that an intruder has to go through is the high walls. All of this has to be done by the intruder while avoiding the watchful guards. Finally, the intruder needs to scale the inner walls before getting to the king's treasure.
- Firewall, intrusion detection systems, internal encryption, access control and personnel controls are examples of layers typically employed to protect enterprise data and programs.

Layering

Cyber security uses multiple layers of defenses for protecting information. If one layer is defeated then the next one should catch it.

6. Abstraction

What is abstraction?

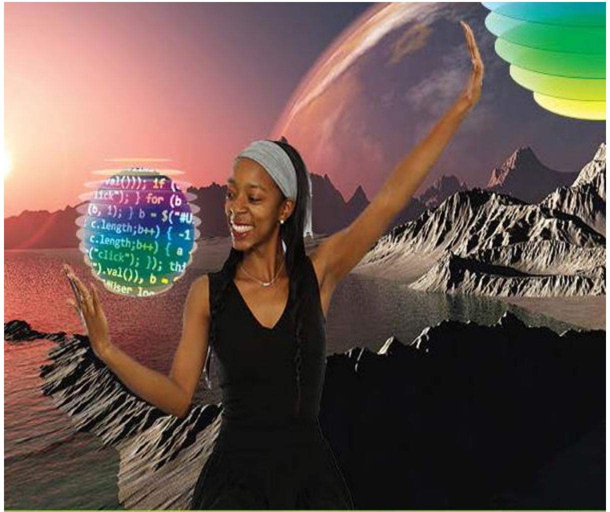
- Abstraction is the concept that something complicated can be represented more simply. All models are abstractions - since it reduces complexity of an object to something understandable.

How does abstraction contribute to cybersecurity?

- Remove/reduce any clutter that can distract the programmer or user from using a resource correctly.
- Only provide necessary details, while reducing the complexity to a set of essential characteristics.
- Excess complexity may hide malicious behaviors.

Examples

- The gauges in a car are an abstraction of the car's performance.
- A map is an abstraction of an area.
- A model airplane is an abstraction of a real airplane and may be used to test aerodynamics.



Abstraction

Abstraction is a fancy word for summarizing or explaining in a way that we can easily understand. A map is an abstraction of the Earth. The speedometer on a car as an abstraction for how fast the car is going.

7. Information Hiding

How does data hiding contribute to cybersecurity?

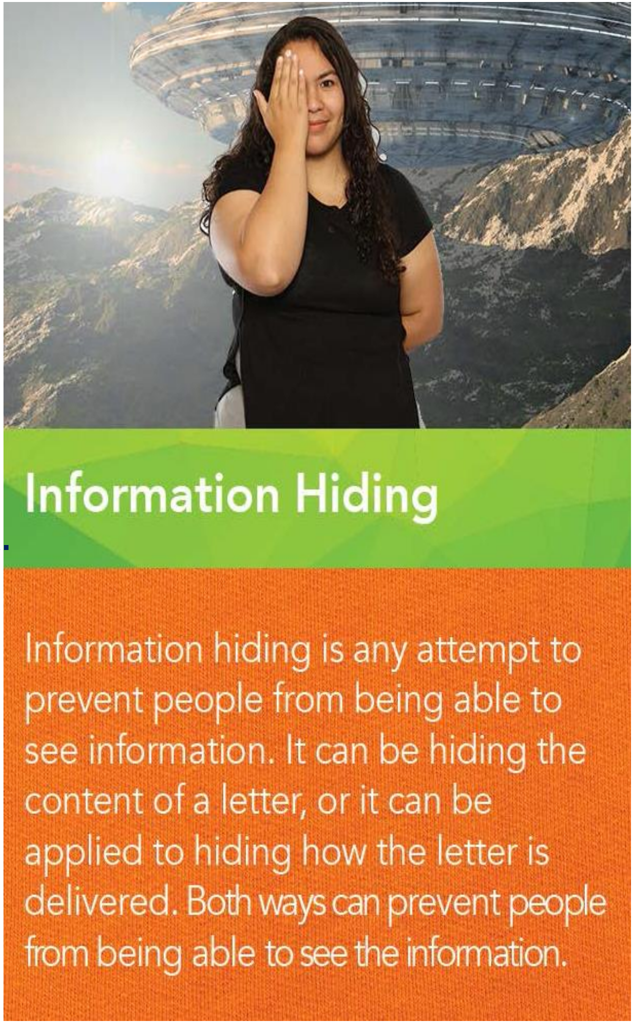
- Only allow necessary aspects of a data structure or a record to be observed or accessed. Log all access attempts.

Examples

- A stack data structure exposes only the data at the top of the stack using simple push and pop instructions. The operating system applies access control to different regions of the stack.
- Websites don’t need to load all of a user’s data to show a list of usernames - they only need the username, the rest of the record fields can be hidden.

Implications

- Programmer or user frustration if allowed access is not sufficient to carry out the task.



Information Hiding

Information hiding is any attempt to prevent people from being able to see information. It can be hiding the content of a letter, or it can be applied to hiding how the letter is delivered. Both ways can prevent people from being able to see the information.

8. Modularity

What is modularity?

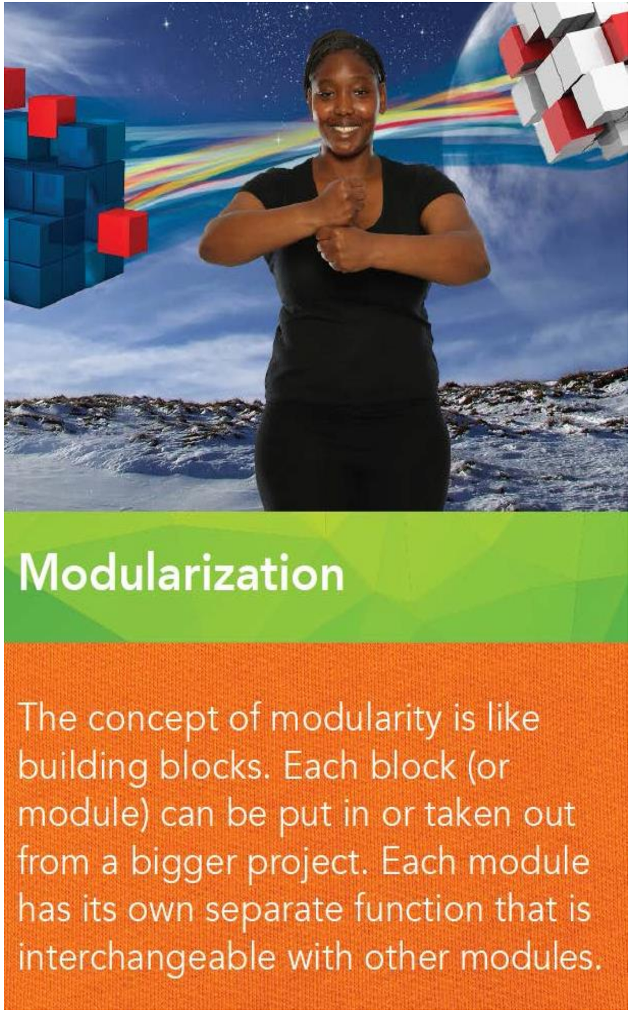
- Modularity is a design technique that separates functionality of a program into independent, interchangeable components.
- Each component or module is self-sufficient and capable of executing a unique part of the desired functionality through well-designed interfaces.

How does modularity contribute to cybersecurity?

- Modules can be mutually-untrusting
- Compartmentalization is possible using modularization. It contains damage to a single module.
- Using modules means that you can swap out a bad part. If batteries weren't modules, any time a battery died you would need to throw out the entire electronic device it was in.

Examples

- Electronic circuits
- Lego blocks
- Network nodes



Modularization

The concept of modularity is like building blocks. Each block (or module) can be put in or taken out from a bigger project. Each module has its own separate function that is interchangeable with other modules.

9. Simplicity

How does simplicity contribute to cybersecurity?

- The lack of complexity allows system designers and programmers to identify unwanted access paths.
- Users can easily translate their general protection goals to appropriate system security configurations.

Examples

- Interface designs that allow correct application of security features.
- Intuitive and straightforward access control rules
- Easy to follow and maintain program statements.

Implications

- Testers will be able to cover all possible combinations and spot problems sooner.
- Simplicity may feed aspirations to add complexity!

Simplicity

The less complicated something is, the less likely it is to have problems. It is also easier to troubleshoot and fix. Keep It Simple!

10. Minimization

What is minimization?

- Having the least functionality necessary in a program or device

How does minimization contribute to cybersecurity?

- Decrease the number of ways in which attackers can exploit a program or device.

Examples

- Turn off unnecessary features.
- Block unnecessary ports using a firewall.
- Reduce the amount of code.

Implications

- Expanding feature sets in future versions can be difficult.
- Reduce test combinations.

Minimization

Minimization refers to having the least functionality of a program or device. The goal of minimization to simplify and decrease the number of ways the software can be exploited. This can include turning off the ports that are not needed, reduce the amount of code running, and turn off unneeded features.

Citations

First Principles – definitions, examples and implications

https://mtu.instructure.com/courses/1267944/files/79717623/download?verifier=iBraay1D8a5AF4FfENCMm6gZZbZhsDG7T2Ihag1R&wrap=1

10 Principles: GenCyber Card Game

https://gencyber.utulsa.edu/wp-content/uploads/2016/10/10-Principles-GenCyber-Card-Game.pdf

Questions/Comments

Thank You!