UNIT 7: Principles of Software Design

Estimated Time in Hours: 5

Big Idea(s)	Enduring Understandings	Projects & Major Assignments
2 Establishing Trust	2.2, 2.3	- Practice recognizing and differentiating the principles using resources such as the GenCyber principles card game. - Practice minimization by installing a simple firewall and configuring its rules (a raspberry pi firewall example is given).

Guiding Questions:

- What are principles and how do they differ from rules?
- What is the difference between simplicity and minimization?
- What is the difference between domain separation and process isolation?
- What is the difference between resource encapsulation and information/data hiding?
- Why are fail-safe defaults important, especially in software design?
- Do these 11 principles guarantee security?

Learning Objectives & Respective Essential Knowledge Statements	Materials	Instructional Activities and Classroom Assessments
2.2.1 LO: Students will describe the principle of simplicity, which is about ensuring that systems are easy to understand, maintain and test so as to be more secure.	• Computer, lecture slides, projector, graphic organizers, access to Internet • Resource for GenCyber 10 Principles (note: does not cover all 11 principles presented here, but a great resource for the others):	• Introduce the 11 Principles using Saltzer and Schroeder's 1975 paper "The Protection of Information in Computer Systems" (note: originally 8, now expanded to 11). Note: the 11 Principles are introduced in this unit; however, they are expanded upon in future relevant units. • Show students common web home pages (i.e., Google, Yahoo). Ask them which one holds a more simplistic design and how it is easier to understand, maintain, and test.

	Hale, Ghandi, Morrison, and Rausch. "Introduction to Cybersecurity First Principles." *GitHub*, uploaded by mlhale, https://mlhale.github.io/nebraska-gencyber-modules/intro_to_first_principles/README/ • GenCyber Principles Card Game: "Game Instructions." Cyber Realm, *gencybercards.com*, https://gencybercards.com/instructions	
2.2.1a EK: Simple designs are easier to understand, maintain and test for security problems. 2.2.1b EK: Simplicity is also known as "Economy of Mechanism."	• Economy of Mechanism: "Engineering Maintainable Android Apps – Economy of Mechanism." *YouTube*, uploaded by intrigano, 31 Oct 2017, https://youtu.be/TNBSSIrKXnE	• Have students watch the YouTube video to define Economy of Mechanism and why it is needed. • Challenge students to answer why is could be a bad idea to include functions and features that aren't needed.

2.2.1cEK: A simple design incorporates a careful analysis of what is needed	• Guide on steps to simplify cybersecurity: Chiock, Mario. "8 Steps to Simplify Cybersecurity." *SecurityRountable.org*, 4 Apr 2019, https://www.securityroundtable.org/8-steps-to-simplify-cybersecurity/?doing_wp_cron=1589219255.4833331108093261718750	• Walk through the steps provided by the web source to demonstrate how to achieve simplicity.
2.2.2LO: Students will use the principle of abstraction to represent complicated concepts more simply and to allow solutions to be transferred to other contexts. 2.2.2a EK: Abstraction is reducing the complexity of an object down to its essentials in a way that is understandable.		• Task students with matching examples of abstraction with what they represent (e.g., a car instrument panel & the car's parts, an illustration of a person & a picture of them, a tablet & the code running on it).
2.2.2LO: Students will use the principle of abstraction to represent complicated concepts more simply and to allow	• Abstraction from binary numbers: Domas, Chris. "The 1s and 0s behind cyber warfare." *YouTube*,	• Show the YouTube video where Chris Domas uses abstraction to categorize binary data into a visual representation that can be interpreted quickly. Ask students to identify specific ways he uses abstraction.

solutions to be transferred to other contexts. 2.2.2b EK: Good and elegant design involves using abstraction.	uploaded by TED, 30 June 2014, https://youtu.be/cWpRxyqDgpM	• Frame abstraction as a way to represent complicated concepts more easily. Ask them what colors are often used for danger, or what shapes are often tied to ideas. Stress this importance for developing UI and GUI. • Assign students abstract nouns and have them come up with a graphic representation of them.
2.2.3 LO: Students will apply the principle of minimization by decreasing the number of ways in which attackers can exploit a program or device.	• Raspberry Pi activity [install a simple firewall (UFW) on the pi and configure rules to block traffic from specific IP addresses, ports, etc.]: "Securing your Raspberry Pi." *RaspberryPi.org*, https://www.raspberrypi.org/documentation/configuration/security.md	• Show examples of minimization which students can relate to: turning off Wi-Fi when not in use and disabling Bluetooth when not in use. This can be as simple as not leaving a car running when it is parked and not in use. • Students can practice minimization by installing a simple firewall (resource reference UFW for the Raspberry Pi) and making firewall rules. These can be tested with multiple pis.
2.2.3a EK: The attack surface of a software environment is the sum of the different points where an unauthorized user can try to enter data or to extract data from an environment.	• Attack surface: "The Threat Landcsape Attack Surface." *YouTube*, uploaded by Muhammad Farooq, 28 May 2019, https://youtu.be/gvkWKKbTkx8	• Show the YouTube video introducing attack surfaces. Use a viewing guide with the video and have students write down attack surface examples in the video and how minimization can remedy them.
2.2.3b EK: Minimizing the attack surface decreases the	• Minimize attack vectors on Android:	• Task students with exploring how to minimize the attack surface on their own phones.

opportunity to find an exploitable vulnerability in the system. 2.2.3d EK: Common mechanisms and access should be minimized. 2.2.3c EK: The human interface should be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly.	Raphael, JP. "10 Android Settings that'll strengthen your security." Computerworld, *computerworld.com*, 20 Nov 2018, https://www.computerworld.com/article/3268079/android-security-settings.html • Minimize attack vectors on iOS: Whittaker, Zack. "Cybersecurity 101: Five settings to secure your iPhone or iPad." Tech Crunch, *TechCrunch.com*, 19 Feb 2019, https://techcrunch.com/2019/02/19/cybersecurity-101-guide-ios-12-privacy/	
2.3.1 LO: Students will give examples of the principle of domain separation, which allows for the enforcement of rules governing the entry and use of	• Introduce concept of domains: "Introduction to Domains." *YouTube*, uploaded by Eli the	• After introducing the domains, ask students to list examples of domains in real-life. Examples include residential vs commercial areas; sidewalks vs roadways; and restaurants with their own seating areas.

domains by entities outside the domain. 2.3.1a EK: A domain refers to a collection of data or instructions that warrant protection.	Computer Guy, 16 Feb 2011, https://youtu.be/ut_oLhMhJsYa (Show 2:48 – 10:55)	• Students can work together to brainstorm ways in which these domains can be protected.
2.3.1b EK: Communications between domains are allowed only as authorized.		• Explain the school's network as sets of domains. Students and teachers can both access the network, but likely with different restrictions. • Ask students if they are allowed to access the teacher/staff network and whether that would violate domain separation.
2.3.2 LO: Students will know that the principle of process isolation prevents tampering or interference from/by other processes. 2.3.2a EK: A process is a program running on a computer.		• Demo process isolation by opening Task Manager (Ctrl+Shift+Esc in Windows). Explain how each item in the processes tab is separated. • Ask students why it is important to isolate processes. What happens if a program crashes or freezes? Demo ending a process and discuss how it only affects the target process.
2.3.2a EK: A process is a program running on a computer. 2.3.2b EK: Each process has a region of the memory (address space), which only it can access. 2.3.2c EK: Processes have to use defined communications	• Process isolation demo: "Cyber security Process Isolation." *YouTube*, uploaded by Phyllis adkins, 26 July 2018, https://youtu.be/HrY9KaCfKDs	• Explain why processes must be separated. Demo the example from the video if resources are available. • Have students research the terms namespace, resource control, and process isolation technologies. These are used by the operating system to achieve process isolation.

mediated by the operating system to communicate with other processes.		• Other examples include Google Chrome separating processes by tab, allowing one tab to crash without harming the others.
2.3.3b EK: Encapsulation allows access or manipulation of the class data in only the ways the designer intended	phpdevster. "Encapsulation. This is the easier of the two concepts to understand…" [Comment on the online forum post *ELI 5: Abstraction vs Encapsulation?*], Reddit, *Reddit.com*, https://www.reddit.com/r/javascript/comments/3shetz/eli5_abstraction_vs_encapsulation/	• Explain the resource part of resource encapsulation first. Students should be familiar with computer components by this point. Ask them to list the resources of a computer. This principle is often difficult for students to understand. Use a simplified example as explained on the website provided: "Consider another real-world example: your house. Your house is an encapsulation of how people go in and out of it. You must enter and leave only through doorways. People can't just randomly enter from any side or direction. The equivalent of a non-encapsulated house would be a house that has no walls, and a roof supported only by a few pillars." Example: medicine is encapsulated inside a gelatin capsule.
2.3.3a EK: Examples of resources are the memory, disk drive, network bandwidth, battery power, and a monitor. It can also be system objects such as shared memory or a linked list data structure. 2.3.3 LO: Students will explain the importance of encapsulating resources, i.e., creating well-defined interfaces around	• Encapsulation explanation: "What is Encapsulation." *YouTube*, uploaded by OOP Channel, 27 Mar 2017, https://youtu.be/bSpPwVFEbO8	• Use a viewing guide to review encapsulation examples and definition from the linked YouTube video.

resources to set rules for how the resources should interact.		
2.3.4c EK: Granting only those privileges necessary for a user to accomplish assigned duties improves accountability and limits accidental misuse. 2.3.4a EK: A privilege is a right for the user to act on managed computer resources.	• Least privilege example: "Least Privilege vs. Shared Accounts." *YouTube*, uploaded by Centrify, 23 Apr 2015, https://youtu.be/GXQsD5-noPM • Access Control List (ACL) manipulation in Linux: Newell, Glen. "An introduction to Linux Access Control Lists (ACLs)." Red Hat, *RedHat.com*, 6 Feb 2020, https://www.redhat.com/sysadmin/linux-access-control-lists	• Use a viewing guide with the linked YouTube video. Students should be able to describe how least privilege was used in the video's example. • Ask students how least privilege improves accountability and limits accidental misuse in this scenario and others.
2.3.4b EK: Least privilege is the concept and practice of restricting access rights for users, accounts, and computing processes to only those resources absolutely required to perform routine, legitimate activities.		• Challenge students by tasking them to research the various types of access control (mandatory, role-based, etc.). They can research the types of access control individually and collaboratively create a poster with descriptive terms of each access control type. • Have students practice least privilege by manipulating Access Control Lists (ACL) in Windows and Linux (linked resource).

2.3.4 LO: Students will explore the principle of least privilege, which is about differentiating among types of access control (mandatory, role-based, discretionary, and rule-based access controls) and analyzing which to use for selective restriction of access to a place or other resource.		
2.3.5 LO: Students will break down how the principle of layering is a strategy for slowing down an attack because the attacker has to conquer each layer before moving on to the next. 2.3.5a EK: A layer is a separate level that must be conquered by an attacker to breach a system.	• Layering (defense in depth) introduction: "Network Security | Defense in Depth." *YouTube*, uploaded by Network Direction, 9 July 2019, https://youtu.be/IiWFMIgKaqQ	• Show the students a simple graphic of layering, such as a castle or fort. Have the students list the individual layers used. Does each layer make it more difficult for an adversary to achieve their goals? What happens if one layer fails? • Alternatively, use a viewing guide with the video to assess the students similarly.
2.3.5b EK: Multiple independent layers require integration and independent management to get the full benefits of layered protection.		• Introduce the types of layering in an effective defense: physical, technical, and administrative. These will be covered multiple times in future sections, but it is important for students to distinguish these categories early.
2.3.6 LO: Students will know that the principle of data hiding is about allowing only necessary aspects of a data structure or a	• Information hiding introduction:	• Make students explain the difference between encapsulation and information hiding.

record to be observed or accessed. 2.3.6a EK: Data hiding can help prevent users/programmers/processes from updating/changing data in invalid ways or by mistake.	"Information Hiding." *YouTube*, uploaded by Udacity, 23 Feb 2015, https://youtu.be/zaqiMBoGFO4	• Ask students to research ways information hiding is achieved in computers.
2.3.7a EK: The principle of modularity says that individual components are capable of executing a unique part of the desired functionality and is achieved through system design. Because of this modular design, security upgrades can happen in one component without having to overhaul the entire system. 2.3.7b EK: A system's components may be separated and recombined.	• Modularity in product design: "WHY MODULAR PROJECT DEISGN?" *YouTube*, uploaded by Modular Management, 18 July 2017, https://youtu.be/p6liu6Ro1bE	• Ask students how computers employ modularity. If a hard drive fails, can another replace it? If a second monitor is desired, can it be added? • Have students research systems (technical and non-technical) which employ modular design. Is modularity a benefit in these cases?
2.3.8 LO: Students will define the principle of fail-safe defaults, which restricts how privileges are initialized when a subject or object is created. 2.3.8a EK: When something does not work or the system fails, the	• The beginning of this video shows a couple good examples of fail-safe not involving computers (lawn push mowers, automatic shopping mart doors):	• Discuss how the ultimate goal of a fail-safe is to protect the assets. With fail-safe, security will beat availability. • Present the students with scenarios and have them judge whether it should be or is fail-safe. Should functionality be sacrificed for security?

system must return to a secure state	"Nuclear Reactor Fail-safe." *YouTube*, uploaded by Randy Dobson, 2 July 2017, https://youtu.be/y_kePiYWl4w	
2.3.8b EK: A secure state is a condition when no subject can access any object in an unauthorized manner 2.3.8c EK: Turning off permission causes a security problem. Please read the following comment from the creator about this EK.		• Explain the broader goal is that a system should come with secure initial settings and should reset to a secure state if rebooted/restarted/reset. This could be applied to the software you install on your computer or the device you connect to your home network. Note: the concept of fail-safe in computers is different than fail-safe in physical security. In physical security, fail-secure is the closer definition to this principle.
8.1.1h EK: Cybersecurity events have led to the development of various cybersecurity career paths and various needs in order to prepare people for these new types of jobs.		• Explore a relevant career, such as cyber instructor.