## UNIT 8: SECURE SOFTWARE

**Estimated Time in Hours: 10-13**

| Big Idea(s) | Enduring Understandings | Projects & Major Assignments |
|---|---|---|
| 5 System Security<br>7 Risk | 5.3, 5.4, 7.1, 7.2 | - Software Vulnerability Discussion<br>- Buffer Overflow Coding Problems<br>- Hacker Game<br>- Pico CTF Web Exploits |

**Guiding Questions:**

- What are security flaws/vulnerabilities in software?
- Why does software have security vulnerabilities?
- What are the consequences of less secure software?

| Learning Objectives & Respective Essential Knowledge Statements | Materials | Instructional Activities and Classroom Assessments |
|---|---|---|
| 5.3.1 LO: Students will describe common security-related software vulnerabilities.<br>EK: 5.3.1a,b,c,d<br><br>7.1.4 LO: Students will be able to conduct standard security testing and assessments.<br>EK: 7.1.4e<br><br>7.2.3 LO: Students will be able to explain how the logical malleability of software and hardware can allow an adversary to change a system to meet the adversary's goals | • "Hackers & Cyber Attacks: Crash Course Computer Science #32." *YouTube*, uploaded by CrashCourse, 18 Oct 2017, https://www.youtube.com/watch?v=_GzE99AmAQU&feature=youtu.be<br>• Podcast (stop at 4 min 40 sec):<br>Schneier, Bruce and Henage, Dan. "Crypto-Gram February 15, 2020." *The Crypto-Gram* | **What is secure software & why do we care? (1-2-day lesson)**<br>In this lesson students learn about software security issues.<br>• The lesson begins with students completing a quick write in their notebook answering the following question: "What are the consequences of less secure software?" The lesson is then introduced: Today we will be learning about software security issues and identify why we care about the issues. The students watch the Hackers & Cyber Attacks Video and take Cornell notes on the video. After the video, students add key vocabulary from the video into their notebook. The class has a brief discussion about what was learned from the video (be sure to address risks: vulnerabilities in databases, ethical hacking and software updates). Students then listen to the podcast linked left (listen from the beginning of the podcast to 4 min 40 sec). The class has a brief discussion about what they learned in the podcast (be sure to |

| | | |
|---|---|---|
| rather than the system's original objective.<br>EK 7.2.3a,b,c | *Security Podcast*, Libsyn, 15 Feb 2020, https://hwcdn.libsyn.com/p/4/f/f/4ff722d465bd469f/crypto-gram-2020-02.mp3?c_id=66980507&cs_id=66980507&destination_id=19374&expiration=1595268380&hwt=86c49be2206efbe3d3f1fb4e659470f4<br>• Textbook: Stallings, William and Brown, Lawrie. *Computer Security: Principles and Practice, Third Edition*. Pearson, 2015.<br>• Notebook<br>• Cornell Notes (for an explanation of the Cornell note-taking system, visit http://lsc.cornell.edu/study-skills/cornell-note-taking-system/ )<br>• Hacker: Cybersecurity Logic Game, by ThinkFun | discuss how software changes could come from an adversary and how changes in software code from an adversary can be difficult to detect). Students play the Hacker Board game in pairs. Students then read Section 11.1 (Software Security Issues) p.358 - 362 in the textbook and take notes. The lesson is wrapped up with students answering Review Question 11.1. |

| | | |
|---|---|---|
| | (available for purchase on Amazon) | |
| 5.3.1 LO: Students will describe common security-related software vulnerabilities.<br>EK: 5.3.1b<br><br>5.3.2 LO: Students will identify the processes of developing secure software.<br>EK: 5.3.2b | • Textbook:<br>Stallings, William and Brown, Lawrie. *Computer Security: Principles and Practice, Third Edition*. Pearson, 2015.<br>• Notebook<br>• Cornell Notes<br>(for an explanation of the Cornell note-taking system, visit http://lsc.cornell.edu/study-skills/cornell-note-taking-system/ ) | **Understanding Buffer Overflow: (4-5-day lesson)**<br>In this lesson students learn about buffer overflow.<br>• The lesson begins with students completing a quick write in their notebook answering the following question: "Why does software have security vulnerabilities?" Students popcorn their responses. Students read chapter 10 in the textbook and take Cornell notes on the chapter. As the students are reading they update the vocabulary in their notebook. Students complete chapter 10 review questions 10.1, 10.2, 10.5, & 10.9 and problems 10.2, 10.3, 10.4, 10.5, 10.10, and 10.11. |
| 5.3.1 LO: Students will describe common security-related software vulnerabilities.<br>EK: 5.3.1a,b,c,d,e,f<br><br>5.3.2 LO: Students will identify the processes of developing secure software.<br>EK: 5.3.2a,b,c,d | • Textbook:<br>Stallings, William and Brown, Lawrie. *Computer Security: Principles and Practice, Third Edition*. Pearson, 2015.<br>• Notebook<br>• Cornell Notes<br>(for an explanation of the Cornell note-taking system, visit http://lsc.cornell.edu/stu | **Understanding Software Security: (5-6-day lesson)**<br>In this lesson students learn about software security - handling input, writing safe program code, interacting with the operating system and other programs, and handling program output.<br>• The lesson begins with students completing a quick write in their notebook answering the following question: "What are security flaws / vulnerabilities in software?" Students share their response with a partner. On person from each pair shares out. Students read chapter 11 in the textbook and take Cornell notes on the chapter. As students are reading, they update the vocabulary in their notebook. Students answer the following chapter 11 |

| | | |
|---|---|---|
| | dy-skills/cornell-note-taking-system/ )<br>• *Pico CTF*, https://picoctf.com/ | review questions: 11.1, 11.3, 11.4, 11.5, 11.6, 11.13 & 11.16.<br><br>**Assessment:** *Students complete the Web Exploit room in PicoCTF.* |