Joseph Silva Jr.

SNHU

CS 405: 6-1 Journal: Don’t Leave Security to the End

12/04/2021

* **Explain what the following statement means as a best practice in secure coding: “Don’t leave security to the end.”**

The “Don’t leave security to the end” practice means do not put developing security measures at the end of the development of the project. Ignoring this practice can cause an issue with the project itself and/or the protection of the project due to unreliable code. Depending on the project and the safety set of code, the safety areas and the entire project might not be compatible with each other and thus might have to make a team start the project over in order to accommodate the security sections of code needed in the project.

* **Describe the steps you can take to prevent threats.**

The steps I would take in order to prevent threats would be the following:

1. Using a multi-layer security system to make sure my program is not only relying on 1 set of defenses.
2. Request from my team leader and/or third-party users for random ethical hacking events in order to be given up-to-date issues when developing the project instead of having to return to the project after an attack has already been successful.
3. Peer reviews before section of code is pushed to main project.
4. Multiple automated testing such as JUnit testing.

* **Provide one example that you can include in your Project Two presentation of how you plan to ensure that security is addressed intrinsically and not left until an issue is discovered—for instance, the use of unit testing.**

One example I would include in my project two to ensure security is addressed intrinsically would be the unit testing such as JUnit testing. I have used JUnit testing in a prior class and the testing was conducted quickly and accurately by the system. Also, this class has shown me to use other testing systems to locate errors / warning / messages which were not discovered in the IDE being used to create the project. These other testing systems / programs are ones such as Cppcheck. Both the unit testing and Cppchecking program allowed additional reviews of the code which can find mistakes that were missed due to human error and/or other computer programs.

**CITATIONS:**

1. Bellairs, R. (n.d.). *What is secure coding?* Perforce Software. Retrieved December 4, 2021, from https://www.perforce.com/blog/sca/what-secure-coding.