**6-1 Journal**

Dylan Jeffery

SNHU

CS-410

Brian West

2/14/2022

Define: What is a security vulnerability?

A security vulnerability is a defect or product of flawed design that allows for unintended behavior within our software. Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source are all security vulnerabilities.

Identify: What kinds of vulnerabilities would be identifiable in C++ code?

Some of the vulnerabilities we might identify can include string vulnerabilities (violating format-string rules can allow an attacker to execute arbitrary code through the format string), invalid string formats (accepting an external string as an argument is bad), unsigned integer wrapping (this can result in integer underflow and overflows) and incorrect type conversions (which can lead to undefined behaviors).

Purpose: Why would you be looking for vulnerabilities during legacy to C++ conversion rather than during testing?

It is possible to miss things during testing and when performing legacy to C++ conversion we may discover previously unknown vulnerabilities. As we tend to have much more user data available at the conversion phase, we also may be more aware of an issue we face in production.

Solutions: How do you determine the appropriate fix to a security vulnerability?

Following STRIDE, we can decompose our system into relevant components, analyze them for susceptibility, and mitigate these threats. We then repeat the process for each component. For each component we must address each of the following threats (that apply)

Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege.

We must also understand that it is impossible to predict if the mitigation we integrate will be effective at the system level, but each component must be secured to prevent serious problems.