Christopher Ivey

CS-405-T1205 Secure Coding 22EW1

Southern New Hampshire University

Dr. Wasim Alim

October 20, 2022

As a developer understanding and adopting the right criteria and requirements for a

systematic software secured applications is specifically essential to creating an application that

does not leave security to the end. Complying with industry secure coding best practices for

secure coding standards, it is critical for developers to adopt implementation of defense-in-depth

security layers through the whole software application development lifecycle. Beginning the

development process with Zero Trust Security protocols using secure coding standards, a

developer will be able to test and run the software security, reliability, safety, and encryption

related properties. Running consistent test on the software that is being created is a very

important continuous step that needs to be done, to ensure the security performance capabilities

of the system.

Unit testing Levels range based on the depth of the security vulnerability that is looking

to be prevented by the developer. Detailed code analysis for each specific programming language

is important to become aware of, as a developer. Developers must realize that automation is their

friend. Automation is a very cost-effective way to include security protocols into the software

code, finalizing in a reduction of the probability for human error. Software Developers are

encouraged to make sure code is secure and simple. By doing so, the application is easier to

secure.

Security should be one of the top priorities when implementing any system design and

functionality for a systematic solution. A major protection point is to decrease the attack surface.

This can be achieved by shrinking data entry points on a network, turning off new software

features, and considering a user credential reduction. Because vulnerability scanning and

remediation is a critical component of any effective security program, there will always be

exceptions that will throw the systems of track if it cannot be patched or updated, leaving a

system’s exposed to a known vulnerability.

Additional steps to ensure that security is addressed intrinsically and not left until an

issue is discovered, include developer checklist based on design and code inspections. This is to

ensure that the designs implementations are free from known problems. Having a checklist as

well as providing the developer with the meaning behind the nature of the items on the list will

equip the developer with the knowledge to prevent from personally misusing the checklist, which

can lead to unwanted changes to the design implementation. Checklist also aid in keeping

documentation on the problems and dates and times that the designed code was assessed for

vulnerabilities. Developer checklist will document the reason behind the vulnerability. For

Project two I plan to use the implementation of developer checklist, to ensure the safety of future

programs. By using the experience of previously known vulnerabilities, it will give me a

guideline to follow. More like a directional navigator for developer safe secure code, in

compliance with industries best coding practices.

Before learning about a Zero Trust Security Model, I wasn't aware that it will monitor

and protect the data at all interval states of the encryption policies (at rest, in transit, and in use).

I didn't know that business data transfers continuously between SaaS applications, IaaS

applications, Data centers, Remote users, and IOT devices. After analyzing more research, I

would say that it is essential to create procedural Zero Trust Security methods to assess and

provide protection to the vulnerability risk points of multiple doorways for a cyber hacker to

enter and cause havoc. The zero trust best practice methods provide increased accountability and

puts greater responsibility on the users to protect the content on the company's computer

network. Zero Trust security methods is a trending topic in software development lifecycles to

combat the cybersecurity threats.

Zero Trust security models make sense. Never trust and always verify can be seen in

today's world. From two factor authentication for apps like Facebook and Instagram, or the

access code that Gmail uses when you are logging into your email account from an unknown

device. Literally, you must verify on a known device, or the user won’t be able to enter the site.

Convincing a developer who is opposing the Zero Trust security measure would be easy! I would

simply tell them, "Hey if you don't secure your system or program with this protective security

measure, then you are leaving your application open to vulnerabilities for hackers to intercept

valuable information.

**APA Citation**

CISA, R. D. (P. A. R. T. N. E. R. |. (2021, October 26). The importance of a company information security policy. Linford & Company LLP. Retrieved October 4, 2022, from <https://linfordco.com/blog/information-security-policies>