Cameron Winningham

Southern New Hampshire University

8/22/2021

Portfolio Reflection

Throughout this course I have learned about secure coding in C++ and secure coding in general. Since C++ is currently my favorite programming language, I found this course to be fun and enlightening. I will never look at any C++ code the same, and I will certainly go through all my previous C++ projects with a fine-toothed comb looking for security vulnerabilities. Examples of various C++ vulnerabilities such as buffer overflows and code injection can be found in this GitHub repository’s coding zip files.

One of the major things I have learned is to adopt a secure coding standard. Although I will not be majoring in cybersecurity, it is important for each developer of a project to have a complete understanding of how to write secure code. One way to make sure every developer is on the same page is to adopt a secure coding standard. The secure coding standard provides “rules” the developers must abide by to write secure code. An example secure coding standard and overview can be seen in this repository’s presentation video.

Another major aspect of secure coding that I have learned is to not leave security to the end. The practice of secure coding is writing software that is free of bugs, errors, and vulnerabilities. To accomplish this task, security needs to be a high priority and not left as an afterthought. As a best practice, this means developers should concentrate on writing code that is secure from the very beginning instead of leaving the security part till the end. If software is not designed with security in mind, a kid could easily download a canned tool from the internet and steal your data without any hacking skills or knowledge required. Don’t let yourself be an easy target for the 12-year-old script kiddie down the street.

Ensuring businesses have the highest level of security in place to protect personal data is essential for modern businesses of every size. In fact, the Data Protection Act 2018 is the UK's implementation of the General Data Protection Regulation (GDPR) meaning it is law to protect personal data in the UK. If you plan on your project to be used in the UK, you must abide by the GDPR. Doing a full risk assessment for every project is critical to ensure maximum security at the lowest cost.

Technology and times have changed dramatically. The way we used to think about security in the past is having the servers and data stored locally behind physical security devices such as locked doors, security guards, and alarm systems. In the past, when accessing data you must physically be at the location the data is stored or log in directly to those servers. These days, data is stored on the cloud and can be accessed instantly from anywhere on demand. This means that security paradigms of the past cannot be used to protect data and a new security paradigm must be implemented: zero trust. Zero trust is a security paradigm teaches us to “never trust, always verify.” Zero trust by default assumes breach and must verify every single request. As a user, it will not impact anything. As far as the user can tell, nothing has changed, which is a good thing for a seamless and user-friendly experience.

Secure coding practices can vary depending on what software, programming languages, and environments you choose to use. In general, prevention is always better than mitigation. One step in preventing vulnerabilities is to never trust the user: always treat user input as untrusted data that you must sanitize. Trusting the user too much can result in several vulnerabilities such as injection and cross-site scripting. Another step would be to keep things simple and secure. Write as little code as needed to get the job done and choose to use programming languages, tools, and environments that are more secure where possible. Lastly, it is important to implement defense-in-depth. Rather than relying on one layer of security, have multiple so that if one layer fails there are other layers to prevent vulnerabilities.

In conclusion, I started this course with the bare minimum knowledge of writing secure software, and zero knowledge of writing secure C++ software. The knowledge and skills I have gained from this course are essential especially in today’s fast-paced technology filled society. As technology changes, so does the way we look at and implement security. I will use the knowledge gained from this course as my foundation and continue to seek education over time as technology changes to find new secure coding methods that must be utilized to develop secure applications.