Secure Coding Practices: A General Guideline

Jay Rovacsek c3146220@uon.edu.au

October 11, 2018

Contents

0.1 Preface

0.1.1 Running with Scissors

Admittedly, the title for this section is very much thanks to one of the first items[1] I read sections of while creating this document. The analogy for development in terms of security could not be more apt for a large portion of the development community.

Why cover this topic? As a security enthusiast and developer, I often found myself looking at a system left untouched until absolutely required, the design choices, logic and knowledge of the language it was written in left with the author. Commonly a requirement for a hotfix was/is needed in a number of this circumstances as a number of critical business services and resources may rely on the system in question.

A large portion of this paper will focus around the more common exploited vectors of web applications, however the vectors commonly exploited in web application settings are commonly exploitable in a desktop application setting, this becomes more and more important to remember as large numbers of commonly used software move to enable cross platform compatibility by utilizing technologies such as Electron[6].

Security as a serious concern is only just now becoming much more "mainstream" to companies than it had previously been, movements pushing HTTPS such as Lets Encrypt[7] or high profile individuals such as Troy Hunt[5] have aided the process of mitigating some of the most easily exploited vectors such as MiTM attacks on unencrypted communications, a plethora of cybersecurity issues however still remain present in modern organisations, with the potential damage to both organisation and individual such as recent breaches in: Sony[2], Equifax[4] and a number of other recent high profile breaches of modern history.

0.1.2 Tripping Over

Security in programming ~~can be a~~ *is* hard beast to tame. Some languages arguably do much better in avoiding accidental issues from being caused by users new to the language or unskilled in understanding potential issues with the code they have written. We can certainly critique early languages for the level of access to the machine they allow a user, without careful consideration in design and a well founded knowledge in the language used issues notorious of early languages. However, in this day and age of highly abstracted languages and frameworks have we traded old demons for new, or do we really have more safety in our computing goals?

As suggested by Wheeler:[3]

> Many programmers don't intend to write insecure code - but do anyway.

0.2 The Most Dangerous Component of Computing

The apparent and dangerous component of computing is never the computer. Users are terrible, will break anything and everything possible and every human is a user. Seasoned veterans of technology or never used a piece of electronics, every human is amazingly fallible in comparison to the machine they attempt to drive.

To avoid heading down the tangent of human computer interaction intricacies, the coverage of human issues in modern and mature languages must be covered.

0.3 Modern Language Issues

0.4 Mature Language Issues

Bibliography

[1] R. C. Seacord, *Secure Coding in C and C++*. Pearson Education, 2005. [Online]. Available: https://www.pearson.com/us/higher-education/program/Seacord-Secure-Coding-in-C-and-C-2nd-Edition/PGM142190.html.

[2] J. Steinberg. (2014). Sony breach, [Online]. Available: https://www.forbes.com/sites/josephsteinberg/2014/12/11/massive-security-breach-at-sony-heres-what-you-need-to-know/.

[3] D. A. Wheeler, *Secure Programming HOWTO*. David A. Wheeler, 2015. [Online]. Available: https://dwheeler.com/secure-programs/Secure-Programs-HOWTO.pdf.

[4] FTC. (2017). Equifax breach, [Online]. Available: https://www.ftc.gov/equifax-data-breach.

[5] T. Hunt. (2018). Https is easy, [Online]. Available: https://www.troyhunt.com/https-is-easy/.

[6] Various. (2018). Electron, [Online]. Available: https://electronjs.org/.

[7] ——, (2018). Lets encrypt, [Online]. Available: https://letsencrypt.org/.