Team Discussion: What is a Secure Programming Language?

You should read Chapter 2,6,7,8 of the course text (Pillai, 2017) and Cifuentes & Bierman (2019) and then answer the questions below, adding them as evidence to your e-portfolio.

1. What factors determine whether a programming language is secure or not?
2. Could Python be classed as a secure language? Justify your answer.
3. Python would be a better language to create operating systems than C. Discuss.

Team component

You should discuss your answers within your team, and you can share your team responses with the tutor for formative feedback or discuss it in next week's seminar.

QUESTION 1

A programming language can be considered secure if it can "provide first-language support to address the causes for the most common, significant vulnerabilities found in real-world software" (Cifuentes and Bierman, 2019).

Some of the vulnerabilities that a secure programming language should prevent are the buffer overflows, SQL injections, cross-site scripting (XSS), OS command injections, and information leaks. Also, an interesting finding in one of the studies that explored the sources of cryptographic failures was that 83% of the vulnerabilities were caused by incorrect use of the cryptographic libraries and not due to problems in the libraries themselves. So an important factor in making a secure programming language would be to have resistance to developers' critical mistakes (Lazar et al., 2014).

The analysis conducted by Cifuentes and Bierman revealed that none of the mainstream languages address all the three main exploited vulnerabilities.

QUESTION 2

Python, like any other mainstream languages, presents a series of vulnerabilities. Pillai, A. B. (2017) explored a series of security issues affecting Python such as overflow errors, serialization issues and reading and evaluating input. Moreover, Pillai provided a set of examples on how the language is vulnerable in web applications and how it can be subject to certain attacks such as XXS, DoS and SSTI. These attacks can be mitigated with the solutions adopted by Pillai in his examples.

Python, like all other programming languages, is not considered natively secure as concluded by Cifuentes & Bierman (2019) and indirectly by Pillai (2017). By carefully applying certain secure coding strategies, secure software systems can be developed by using Python.

QUESTION 3

Both languages can be used to write an operating system but Python can not write a full operation system because it is an interpreted language and has limitations to write

instructions for the hardware (History-computer.com, *2022). Also, managing memory is still an issue that can't be addressed adequately for an operating system using Python* (Cifuentes and Bierman, 2019).

Python is a very high-level language (VHLL). This means that Python uses a higher level of abstraction, conceptually further from the underlying machine, than do classic compiled languages such as C, C++, and Fortran, which are traditionally called "high-level languages" (Martelli, et al., 2017).

REFERENCES

Cifuentes, C. and Bierman, G. (2019). What is a Secure Programming Language? DOI:10.4230/LIPIcs.SNAPL.2019.3.

History-computer.com *(2022). C vs Python: Compared*. [online] History Computer. Available from:https://history-computer.com/c-vs-python/ [Accessed 2 Jul. 2022].

Lazar, D., Chen, H., Wang, X. & Zeldovich, N. (2014) 'Why does cryptographic software fail'. *Proceedings of 5th Asia-Pacific Workshop on Systems*, 2014 Beijing, China. ACM, 7.

Pillai, A.B. (2017) *Software Architecture with Python*. Birmingham, UK: Packt Publishing. Available

from:https://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=nlebk&AN=1513359&site=eds-live[Accessed: 1 July 2022].

Martelli, A., Ravenscroft, A. & Holden, S. (2017) *Python in a Nutshell : A Desktop Quick Reference*. Sebastopol: O'Reilly Media, Incorporated.