J. Irvine Secure Software Development

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?**

There are several factors that determine whether a programming language is secure or not. According to the book "Software Architecture with Python" by Anand Balachandran Pillai, some of these factors include:

Programming languages that are faster to write, easy to read and understand are less likely to have security vulnerabilities (Cauligi, 2017).

Ruef et al (2015) found that keeping the programming language short and simple led to less vulnerabilities being reported.

A programming language that comes with a set of well-tested and compact standard library modules is more likely to be secure.

A programming language that has strong memory management features is more likely to be secure.

A programming language that has strong type safety features is more likely to be secure.

There are several best practices that can be followed to write secure code in any programming language. Use the latest version of the language, keeping the code up-to-date with security patches, using secure coding practices, and using third-party libraries that have been audited for security.

Ruef, A., Hicks, M., Parker, J., Levin, D., Mazurek, M.L. and Mardziel, P., 2016, October. Build it, break it, fix it: Contesting secure development. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 690-703).

Cauligi, S., Soeller, G., Brown, F., Johannesmeyer, B., Huang, Y., Jhala, R. and Stefan, D., 2017, September. Fact: A flexible, constant-time programming language. In 2017 IEEE Cybersecurity Development (SecDev) (pp. 69-76). IEEE.

2. **Could Python be classed as a secure language? Justify your answer.**

Python is considered to be a relatively secure programming language and is one of the fastest growing languages (Van Rossum, 2007).

Python has a Strong Community that actively contributes to the language's security. Security vulnerabilities are quickly identified there are regular updates and patches.

There are many built-in security features that help developers write secure code. For example, Python's standard library includes modules for cryptographic operations.

Python manages memory allocation and deallocation automatically, relieving developers of low-level memory manipulation responsibilities. This prevents common vulnerabilities such as buffer overflows making Python programs less susceptible to memory-related security issues.

Python has numerous sets of standard libraries and frameworks that are widely used and maintained with security mechanisms built-in. This reduced the risk of vulnerabilities in commonly used functionality.

There is a large amount of third-party security tools and frameworks for Python that enable developers to find and limit security risks for example Bandit.

Django and Flask are Python frameworks that have built-in security features that help developers create secure web applications. These input validation, protection against common web vulnerabilities (e.g., Cross-Site Scripting, Cross-Site Request Forgery), and secure session management.

Python does have has these security advantages, however software security will be dependent on the use of secure coding practices, and vulnerability management through testing and ensuring that ther are regular updated and patches being applied.

Van Rossum, G., 2007, June. Python Programming Language. In USENIX annual technical conference (Vol. 41, No. 1, pp. 1-36).

https://bandit.readthedocs.io/en/latest/

https://www.djangoproject.com/

https://flask.palletsprojects.com/en/2.3.x/

3. **Python would be a better language to create operating systems than C. Discuss.**

Python is a high-level programming language that is known as simple to learn and use (Zehra et al, 2020). It can be used for scripting, web development, and data analysis. C is a low-level programming language that is often used for system programming and embedded systems (Kernighan and Ritchie, 2002).

While Python has many advantages over C in terms of ease of use and readability, it is not necessarily better than C for creating operating systems. Operating systems are designed using low-level access to hardware and system resources, which C is better suited for than Python. C also has better performance than Python as it is a compiled language, whereas Python is an interpreted language. Albalooshi and Mahmood (2017) found that there was superior software reusability with Python and C when multiple inheritance is in use.

There has been operating systems written in Python, for instance Raspbian by the Raspberry Pi Foundation.

Although Python has many advantages over C such as being easy to use and learn and has a wide community. The drawback with Python is that it cannot create operating systems as they need low-level access to hardware and system resources, which C, at this moment in time, can do.

Albalooshi, F. and Mahmood, A., 2017. A comparative study on the effect of multiple inheritance mechanism in java, c++, and python on complexity and reusability of code. International Journal of Advanced Computer Science and Applications, 8(6).

Kernighan, B.W. and Ritchie, D.M., 2002. The C programming language.

Srinath, K.R., 2017. Python–the fastest growing programming language. International Research Journal of Engineering and Technology, 4(12), pp.354-357.

Zehra, F., Javed, M., Khan, D. and Pasha, M., 2020. Comparative analysis of c++ and python in terms of memory and time.

http://www.raspbian.org/