Team Discussion: What is a Secure Programming Language?

1. What factors determine whether a programming language is secure or not?

Secure programming languages have following features:

- A language with less security holes
- A language which ensures data confidentiality and integrity

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

Python 2 has some security vulnerabilities:

- Input evaluation: To solve the problem, replace input() with raw_input() which does not pass the contents to eval.
- Overflow errors: The xrange() function can produce overflow error. To solve the problem, replace it with range() function which produces less errors.
- Object serialization: The pickle and cpickle modules are used for object serialization. They do not do type checks on objects. To solve the problem, replace those modules with json or yaml.

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

Developing an operating system was done in C many years ago and the reason behind it was that C is a low-level language which resembles to the system language. After the emergence of Python, some people started to use this language for OS development because Python has lots of libraries which make the development process much easier.

Team Discussion: Exploring the Cyclomatic Complexity’s Relevance Today

The Cyclomatic Complexity is commonly considered in modules on testing the validity of code design today. However, in your opinion, should it be? Does it remain relevant today? Specific to the focus of this module, is it relevant in our quest to develop secure software? Justify all opinions which support your argument and share your responses with your team.

Using Cyclomatic Complexity has some advantages and disadvantages.

Advantages:

- finding complex functions with too many lines of code and breaking them into smaller functions
- determining the number of required test cases
- lowering the complexity means lowering security risks in the system

Disadvantages:

- measuring the number of pathways through a body of code
- considering only the code that directly implements a method or function without considering what happens behind a method call.
- leading to more security vulnerabilities

From security perspective, some security experts believe that complexity of code lead to security problems. The attackers use the complexity as hiding places for code vulnerabilities. So, it is good practice to use metric tools, but others believe that some codes are complex by nature and using metric tools make the situation harder and pave the way for more vulnerabilities.

Team Activity: Debate: Microservices and Microkernels

Tanenbaum-Torvalds debate- Agreeing/disagreeing with “Torvalds has been proven wrong and it only took nearly thirty years. Microservices and microkernels are the future. “

The Tanenbaum-Torvalds Debate is a debate about two operating systems- Minix and Linux. The developer of Minix -Tanenbaum- commented on Linux. He believed that the operating system was obsolete because it had monolith style architecture and was not portable (DiBona et al., 1999). Tanenbaum also supported Minix as a microkernel- based system. According to him, microkernels were the winners.

Recent migrations to microservices attest Tanenbaum justification that the era of microservices is coming. As companies grow and applications’ demands increase, the architecture becomes more worthwhile (Chandler, 2022). Migrating to microservices is even more important when new features are added to monolith-based architecture. The independence of components in microservice systems make them easy to scale.

Although Torvalds made good arguments on supporting monolith-based architecture, the above-mentioned evidence proves that it is the era of microservices.

References

- DiBona C & Ockman S. (1999). Appendix A: The Tanenbaum-Torvalds Debate. Available from: https://www.oreilly.com/openbook/opensources/book/appa.html [Accessed 30 August 2022]
- Chandler H. (2022). Microservices vs monolithic architecture. Available from: https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith [Accessed 31 August 2022]