Team Discussion: What is a Secure Programming Language?

Unit 3

Programming Languages: History, Concepts & Design

1. Draft 1 by Neelam
What factors determine whether a programming language is secure or not? (Cifuentes & Bierman, 2019)

- All programming languages are insecure, including even the most recent ones that come with claims that they are designed to be “secure”.
- real criticism is the lack of a common understanding of what “secure” might mean in the context of programming language design.
- Authors propose a simple data-driven definition for a secure programming language: that it provides first-class language support to address the causes for the most common, significant vulnerabilities found in real-world software.
- **three of the top four most common vulnerabilities are actually issues that can be considered to be in the realm of programming language design**: buffer errors, injection errors, information leak errors.
- Developers do not write incorrect code because they want to. Vulnerable code is written inadvertently because our mainstream programming languages do not provide the right abstractions to support developers in writing vulnerable-prone code.
- Buffer errors are introduced because of manual management (allocation, reallocation, and deallocation) of pointers.
- Injection errors are introduced because of the representation of code as strings, use of manual string concatenation, and sanitisation of strings that reach a sensitive location (e.g., an SQL execute statement).
- And information leak errors are introduced because of manual tracking of sensitive data, trying to ensure it does not leak to less sensitive objects.
- Abstractions in programming languages introduce different levels of cognitive load. The easier it is for an abstraction to be understood, the more accepted that abstraction becomes. For example, managed memory is an abstraction that frees the developer from having to manually keep track of memory (both allocation and deallocation). As such, this abstraction is widely used in a variety of mainstream and non-mainstream languages. At the same time, performance of the developed code is also relevant to developers. If an abstraction introduces a high performance overhead, it makes it hard for that abstraction to be used in practice in some domains. For example, managed memory is not often used in embedded systems or systems programming due to the difficulty of predicting its performance overhead.
- In 1958, the LISP language introduced the concept of managed memory by adding garbage collection in the runtime of the language. Garbage collection provides a solution that frees up developers from the cognitive load of having to allocate, reallocate and deallocate memory objects correctly. Developers do not need to specify any allocation and deallocation instructions because it is done under the hood by the runtime of the language.

Amendments by Thien

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

   According to Cristina Cifuentes and Gavin Bierman (SNAPL 2019), three vulnerabilities are widely exploited year over year: buffer errors, injection errors, and information leak errors. A programming language is secure if it has addressed:

   ○ **Memory safety**: to protect the programs/ applications from security vulnerabilities when dealing with memory access such as buffer overflows and memory leaks. Memory safety can be assured using tracing garbage collection to insert runtime checks on every memory access. Another approach uses static program analysis and automated theorem proving to ensure that the program is free of memory errors.

   ○ **Confidentiality**: It is hard to prevent data leakage at the programming language level. The developers are supposed to handle exceptions thrown by the program to ensure that no sensitive data is visible to the end-users. However, a secure programming language must not expose too many details about the errors but the error codes, coming along with the documentation on managing those errors. Data encryption and decryption should be encouraged or enabled by default when dealing with data.

   ○ **Data Integrity**: All the data flow must be protected throughout the application life cycle. Providing techniques such as digital signatures, timestamping, or running the data through a hash calculator would help ensure the data has not been tampered.

   ○ There are also mistakes caused by developers when using unsafe/outdated modules/dependencies with unintended vulnerabilities, leading to information leakage.

Draft 1 by Neelam

2. Could Python be classed as a secure language? Justify your answer. (Pillai, 2017, Chapter 6)

YES	NO
Python is a very readable language with simple syntax. It comes with a set of well-tested and compact standard library modules. All of this seems to indicate that Python should be a very secure language.	four types of security issues with Python (console issues), namely, those with reading input, evaluating expressions (eval function), overflow errors, and serialization issues (pickle module).
	Security issues from web applications written in Python such as Django, Flask, Pyramid, and others. Some possible attacks: Server Side Template Injection (SSTI), Denial of Service, Cross-Site Scripting(XSS)
	YES, if precautions are taken. However, mitigation of attacks is possible – must take some measures while writing the codes, and take specific measures while handling passwords etc. Furthermore, the Python Open Web Application Security Project (OWASP) project is a free, third-party project aimed at creating a hardened version of Python more resilient to security threats than the standard Cpython. It is part of the larger OWASP initiative. The Python OWASP project makes available its Python bug-reports, tools, and other artifacts via the website and associated GitHub projects.

Amendments by Thien

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

   No.

   Pillai(2017) has demonstrated some vulnerabilities in both Python 2.x and Python 3.x version:

   ○ Failures in reading and evaluating users' input lead to exposing global variables and other valuable information about the program such as function names, packages used, and so on. Furthermore, malicious hackers can leverage this exploit to crash the Python interpreter and thereby possibly gain control over the system.

   ○ Overflow errors in several built-in functions such as xrange(), range(), and len().

   ○ Server-Side Template Injection, XSS, SQL Injection errors in web frameworks written in Python such as Django, Flask, Pyramid, and others allow the attackers to modify the request with illegal parameters.

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

- Python: a high level programming language (i.e use strong abstraction, does not deal with registers, memory addresses…)
- However, operating systems (which consist of a shell and kernel) are mainly done in C (Python cannot access hardware)
- Python is mostly an interpreted language, which means a dedicated interpreter is needed to run the code. No compiler.

- Note: Shell which interact with user and Kernel which interact with Hardware
- Source: https://stackoverflow.com/questions/10904721/is-it-possible-to-create-an-operating-system-using-python
- https://www.quora.com/Is-it-possible-to-write-an-operating-system-using-python

Amendments by Thien

○ Python is a high-level programming language, and it does not natively provide constructs to talk directly to the hardware and perform low-level data structure manipulation.

○ Python is an interpreted language, which means a dedicated interpreter(CPython) is needed to run the code. The memory management techniques will be provided by the interpreter and the underlying operating system.

Sources

- Cifuentes C. & Bierman, G. (2019) “What is a Secure Programming Language?”, 3rd *Summit on Advances in Programming*. Available from: https://drops.dagstuhl.de/opus/volltexte/2019/10546/pdf/LIPIcs-SNAPL-2019-3.pdf
- Pillai, A.B. (2017) *Software Architecture with Python*, Birmingham, Packt.
- https://stackoverflow.com/questions/10904721/is-it-possible-to-create-an-operating-system-using-python
- https://www.quora.com/Is-it-possible-to-write-an-operating-system-using-python

Thien’s sources

- References

- Pillai, A.B. (2017). Software Architecture with Python. Packt Publishing Ltd.

- Cifuentes, C. and Bierman, G., 2021. *What is a Secure Programming Language?*. [online] Drops.dagstuhl.de. Available at: <https://drops.dagstuhl.de/opus/volltexte/2019/10546/> [Accessed 27 November 2021].

- En.wikipedia.org. 2021. *Manual memory management - Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/Manual_memory_management [Accessed 27 November 2021].

- En.wikipedia.org. 2021. *Memory safety - Wikipedia*. [online] Available at: https://en.wikipedia.org/wiki/Memory_safety [Accessed 27 November 2021].

- SearchSecurity. 2021. *Exception handling best practices call for secure code design*. [online] Available at: https://www.techtarget.com/searchsecurity/feature/Exception-handling-best-practices-call-for-secure-code-design [Accessed 27 November 2021].

- Tripwire, I., 2021. *Data Integrity Follow Up: Ways to Protect Your Data*. [online] The State of Security. Available at: https://www.tripwire.com/state-of-security/risk-based-security-for-executives/connecting-security-to-the-business/data-integrity-follow-ways-protect-data/ [Accessed 27 November 2021].

- I. and Haapala, A., 2021. *Is it possible to create an operating system using Python?*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/10904721/is-it-possible-to-create-an-operating-system-using-python [Accessed 27 November 2021].

- Quora. 2021. *Is it possible to write an operating system using python?*. [online] Available at: https://www.quora.com/Is-it-possible-to-write-an-operating-system-using-python [Accessed 27 November 2021].

- How much of an operating system could be written in, P., Brocious, S. and J, M., 2021. *How much of an operating system could be written in, say, Python?*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/190464/how-much-of-an-operating-system-could-be-written-in-say-python [Accessed 27 November 2021].