# Unit 3 – SSD Team discussion

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

A programming language is as secure as the way a user operates it, therefore declaring a programming language as secure or not is tricky. However, we can classify programming languages as safe if the chances of coming across errors are lower. These errors include:

- Buffer errors. Buffer errors are common in programming languages operating using memory buffer. Buffer errors can lead to a buffer overflow. Buffer overflow happens when the amount of data surpasses the capacity of the memory buffer. Additionally, because of an unseemly validation of input data, a hacker can read or write data outside the intended buffer (ImmuniWeb, 2020).
- Information leak. Data leaks can occur if the programming language is harder to use. More complex programming languages can make up multiple lines of code, making the debugging process harder and less efficient. Insufficient debugging can lead to unresolved critical errors, allowing external threats to penetrate the system.
- Input validation. Input validation is performed to validate the data entering an Information System. By validating the inputs, the system can reject data from potentially untrusted sources, trying to compromise it. (OWASP, n.d).
- Cross-Site Scripting. Cross-Site Scripting or XSS are attacks where malicious scripts are injected into trusted websites. The attacker operating the script can implement it into the trusted website's code and send it to an unsuspecting user. As the user's browser thinks that the script belongs to the trusted website, it runs it and leaks data browsing data to the hacker. These can include cookies, passwords saved in the browser, session tokens, etc. (OWASP, n.d).

These errors can affect almost all programming languages; however, the safest languages are the ones that can implement security fixes and debug easier and faster.

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

Python, among other programming languages, suffers from vulnerabilities and external malicious attacks. As stated above, declaring a programming language secure is not easy. Python's most common vulnerabilities are non-sufficient input validation and Cross-Site Scripting such as SQL Injections and Cross-Site Request Forgery (WhiteSource, n.d). Nonetheless, with the enormous scientific community Python has developed, bugs and security issues can be detected and shared among the community, resulting in a much faster resolution. Furthermore, with Python being a high-level logical programming language, the debugging process is easier and more probable in detecting code anomalies.

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

Today, the most common programming language for creating operating systems is C. Microsoft Windows, Linux, macOS, Smartphone kernels, Databases, Embedded Systems, and 3D movies are all written and created with C (Munoz, n.d). While there are multiple other languages where programmers would be more productive, C is still the most common language because it provides libraries for almost every architecture (Munoz, n.d).

Python would not be a better solution to create operating systems, as it is classified as a very high-level programming language, used mostly as a productivity tool. Therefore, with Python programmers are not able to access the computer's hardware and perform low-level data manipulation (stackoverflow, 2012).

Moreover, historically, the first operating systems were written using the Assembly programming language, and even today, most high-level languages except C require a vast amount of Assembly development to provide an appropriate runtime environment (OSDev.org, n.d).

References:

ImmuniWeb (n.d) Buffer Errors [CWE-119] Available from: https://www.immuniweb.com/vulnerability/buffer-errors.html [Accessed 25 May 2021].

Imperva (n.d) Buffer Overflow Attack Available from: https://www.imperva.com/learn/application-security/buffer-overflow/ [Accessed 25 May 2021].

Kirsten, S., (n.d) Cross Site Scripting (XSS) Available from: https://owasp.org/www-community/attacks/xss/ [Accessed 25 May 2021].

Munoz, D., (n.d) After All These Years, the World is Still Powered by C Programming Available from: https://www.toptal.com/c/after-all-these-years-the-world-is-still-powered-by-c-programming [Accessed 25 May 2021].

OSDev.org (n.d) Languages Available from: https://wiki.osdev.org/Languages Accessed 25 May 2021].

OWASP Cheat Sheet Series (n.d) Input Validation Cheat Sheet Available from: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html [Accessed 25 May 2021].

stackoverflow (2012) Is it possible to create an operating system using Python Available from: https://stackoverflow.com/questions/10904721/is-it-possible-to-create-an-operating-system-using-python [Accessed 25 May 2021].

WHITESOURCE (n.d) What are the Most Secure Programming Languages? Available from: https://www.whitesourcesoftware.com/most-secure-programming-languages/ [Accessed 25 May 2021].