

## **Team Discussion – Michael Geiger – What is a secure Programming Language?**

The study by Cifuentes & Bierman (2019) states that 64% of vulnerabilities in software can be traced back to four categories, buffer errors, injections, information leak and permissions, privileges and access control. It is also emphasized that an unambiguous definition of a secure programming language is difficult, since even those that are labeled as secure are not (Cifuentes & Bierman, 2019). However, factors and influences that contribute to a secure program can be identified and be able. First of all, the choice of the programming language counts, since each language has its own advantages as well as security gaps. However, it should also be emphasized that the type of programming, the design, is the key factor that decides whether a program is secure or has vulnerabilities. In addition, quality control within the framework of testing and penetration testing offers the possibility of identifying threats and weak points in order to be able to take suitable countermeasures.

Making a concrete decision on whether Python can be classified as a safe programming language on this basis is an insufficient approach. Python does offer some security aspects, such as the garbage collector and the enforcing information flow

Policies by Jeeves, however, has errors in relation to other threats such as buffers, which are addressed by e.g. C (Cifuentes & Bierman, 2019). Other potential Python vulnerabilities are noted by Pillai (2017) in server side template injection, denial of service (DoS) and cross-site scripting (XSS). However, these vulnerabilities can be mitigated with the right design. With this consideration it can be seen that every programming language has its own advantages and disadvantages and when creating a program it is primarily important to know the program-typical weak points in order to reduce them in a targeted manner.

## **References:**

Cifuentes, C. & Bierman, G. (2019) What is a Secure Programming Language? Leibniz International Proceedings in Informatik. 3: 1-15. Available from: <https://drops.dagstuhl.de/opus/volltexte/2019/10546/pdf/LIPIcs-SNAPL-2019-3.pdf> [Accessed 24 March 2022].

Pillai, A. (2017) Software Architecture with Python. Available from: <https://eds.s.ebscohost.com/eds/ebookviewer/ebook/bmxlYmtfXze1MTMzNTIfX0FO0?sid=4d9d2041-d6cc-4e55-8e28-344420d21f12@redis&vid=0&hid=http://eds.s.ebscohost.com/&format=EB> [Accessed 23 March 2022].