**Programming language concepts**

1. **What is ReDOS and what part do ‘Evil Regex’ play?**
   * The Regex Denial of Service (ReDoS) is a type of sophisticated DoS, which involves cyber-attackers exploiting the Regex’s worst case and lead to an application Dos (Roichman & Weidman, 2012; Davis et al., 2018), which can be defeated via ‘selective memoization’ (Davis et al., 2021). Regex is called ‘evil’ if it can get stuck based on particularly formatted inputs, such as grouping constructs with repetitions and inside the repeated group there is a repetition and an alternation with overlapping, such as ‘(a+)+’, ‘([a-zA-Z]+)\*’, etc. (Roichman & Weidman, 2012; Larson & Kirk, 2016).
2. **What are the common problems associated with the use of regex? How can these be mitigated?**
   * Besides the ReDoS mentioned above, it is worth noting that there are not many robust tools for ReDoS-safety validation (Roichman & Weidman, 2012; Li et al., 2021). Nevertheless, ‘accepting known good’ and ‘rejecting known bad’ are two main sanitisation strategies to validate the inputs by Regex (Roichman & Weidman, 2012). An example of an attempt to automate the verification of Regex is the ‘Automatic Checking of Regular Expressions’ (ACRE) method, developed by Larson (2018) and based on the knowledge of common mistakes that are made when using Regex, such as incorrect use of character sets and line anchors.
3. **How and why could regex be used as part of a security solution?**
   * Some security software solutions, such as Google CodeSearch, leverage meta-Regex and advanced operators to detect certain strings in open-source codes, such as ‘evil Regex’ (Roichman & Weidman, 2012). This is an example of ‘Cloud Security as a Service’ (Hussain & Hussain, 2021) to enhance the security of applications further.

**References**

Davis, J. C., Coghlan, C. A., Servant, F., & Lee, D. (2018) The impact of regular expression denial of service (ReDoS) in practice: an empirical study at the ecosystem scale. In *Proceedings of the 2018 26th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering* (pp. 246-256).

Davis, J. C., Servant, F., & Lee, D. (2021) Using selective memoization to defeat regular expression denial of service (ReDoS). In *2021 IEEE symposium on security and privacy (SP)* (pp. 1-17). IEEE.

Hussain, M. E., & Hussain, R. (2021) Cloud Security as a Service Using Data Loss Prevention: Challenges and Solution. In *International Conference on Internet of Things and Connected Technologies* (pp. 98-106). Springer, Cham.

Larson, E. (2018) Automatic checking of regular expressions. In *2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM)* (pp. 225-234). IEEE.

Larson, E., & Kirk, A. (2016) Generating evil test strings for regular expressions. In *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)* (pp. 309-319). IEEE.

Li, Y., Chen, Z., Cao, J., Xu, Z., Peng, Q., Chen, H., ... & Cheung, S. C. (2021) {ReDoSHunter}: A Combined Static and Dynamic Approach for Regular Expression {DoS} Detection. In *30th USENIX Security Symposium (USENIX Security 21)* (pp. 3847-3864).

Roichman, A., & Weidman, A. (2012) Regular Expression Denial of Service.