Programming language concepts

What is ReDOS and what part do 'Evil Regex' play?
The Regular Expression Denial of Service (ReDoS) attack make use of the fact that most regular expression implementations can reach extreme situations that cause them to work very slowly (exponentially related to input size). An attacker can then cause a program using a regular expression (Regex) to enter these extreme situations and then hang for a very long time. (Weidman, n.d.)
Regex expressions that are known to be vulnerable to this type of attack are known as evil regex. It could include grouping with repetition, as well as repetition and alternation with overlapping inside the grouping. (Weidman, n.d.)

What are the common problems associated with the use of regex? How can these be mitigated?
A common problem with regular expressions is that they are compiled at run-time and the regular expression compiler does not give much feedback on potential errors. And also, the patterns used are prone to errors, in terms of how they are implemented by developers. (Larson, 2018).
Tools like EGRET and ACRE can be used to check regex patterns for common mistakes (Larson, 2018).

How and why could regex be used as part of a security solution?
Regex can be used validate inputs in an application as this ensures that the correct type of data is inserted into programs (Larson& Kirk, 2016). Regex Patterns could be used to ensure that passwords are always complicated to a certain level.

References:
Larson, E. (2018) Automatic Checking of Regular Expressions. 18th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM).

Larson, E., Kirk, A. (2016) Generating Evil Test Strings for Regular Expressions, IEE International Conference on Software Testing, Verification and Validation (ICST).

Weidman, A. (n.d) Regular Expression Denial of Service. Available from: https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS [Accessed 23 Apr 2022].