What is ReDOS and what part do 'Evil Regex' play?

ReDoS stands for Regular expression Denial of Service.

It is possible for many regex patterns to be evaluated very slowly if the input is carefully crafted. In an attempt to ge the program to hang, an attacker inputs a string (e.g. **aaaaaaaaaaaaaaaaaaaaaaaa!)** which is known to make some expressions evaluate very slowly. The above string increases evaluation time exponentially with the addition of just a few extra characters. (OWASP, n.d)

Evil regex are regex expressions that are known to be susceptible to this kind of attack. The can contain grouping with repetition, and repetition inside the grouping too. Some examples include: **(a+)+** and **([a-zA-Z]+)***

What are the common problems associated with the use of regex? How can these be mitigated?

One problem with regular expressions is that they are compiled at run-time This means that the regular expression compiler does not give any feedback on potential errors. Errors are easy to make when writing regex patterns.

One mitigation for this would be to use a regex pattern checker like ACRE, which checks the regex patterns included in a program for common errors. Among the checks are checks for incorrect use of character sets (enclosed by []), wildcards (represented by .), and line anchors (^ and $) (Larson, 2018).

How and why could regex be used as part of a security solution?

- **Input Validation.** RegEx can be used to ensure that input doesn't contain certain characteristics and protect from SQL injection, for example
- **Password Security.** RegexPatterns could be written to ensure that passwords meet a certain level of complexity (e.g. 15 characters long and must include a lowercase, uppercase, number and special character).
- **Firewall Rules.** You can use RegEx to create rules to block requests to certain file types. *.json *.js etc.

References

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

OWASP Regular expression Denial of Service: https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS [accessed 1/4/2022]