Unit 4 - Programming language concepts

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

Regular Expressions are used to check inputs for specific criteria or patterns. Some Regex patterns can cause loops or crashes due to the number of possible combinations it will attempt. The regex engine will attempt all possible combinations to find a pattern and when one fails, it will return to a previous position to attempt other paths. This is known as backtracking. Regular expression Denial of Service (ReDOS) is a form of denial-of-service attack which exploits this vulnerability to crash or slow a program (Weidman, N.D.)

The term Evil Regex describes a Regex pattern which is susceptible to the above attacks when faced with difficult inputs. A Bad actor can exploit Evil Regex by creating an input designed to cause the Regex to loop or crash.

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

There are many issues associated with the use of Regex, aside from the security and performance issues listed above. Some studies have shown that many developers have issues with the non-intuitive syntax, understanding the performance and security risks associated with Regex, and performing sufficient testing. These issues can be mitigated with the use of tools to assist in the writing, comprehension and testing of Regex and further education (Michael et al., 2019). Furthermore, tools such as Automatic Checking of Regular Expressions (ACRE) can be used to check for common mistakes (Larson, 2018)

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

Regex can be used as a tool for security in many ways. The primary use would be for input validation, ensuring the integrity of data entered to a database. Furthermore, it can be used to enforce password complexity, to apply a criteria to passwords in order to ensure a minimum password strength.

References

Weidman, A. (n.d.) Regular expression Denial of Service – ReDoS. Available from: https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS [Accessed 01/11/22]

Michael, L.. Donohue, J., Davis, J., Lee, D. & Servant, F. (2019) Regexes are Hard: Decision-Making, Difficulties, and Risks in Programming Regular Expressions *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)* pp. 415-426

Larson, E. (2018) "[Research Paper] Automatic Checking of Regular Expressions," *2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2018, pp. 225-234