# ReDOS and Regex

ReDOS is a form of denial of service attack that exploits weaknesses in regular expressions, with those being expressions for strings of information that often come in a standard format such as dates or numbers. If a regular expression is poorly designed, this can cause programmes to take a considerably long amount of time to execute certain inputs, which in turn leads to a denial of service. Evil regex (regular expressions) are regular expressions which lead to these denials of service and, according to Weidman (2024), can be exploited by attackers on systems which use regular expressions.

A common problem with regex patterns is that some symbols can be used in different expressions. Larson (2018) provides an example of this, stating that “The ^ symbol could
refer to the “beginning of the string”, a negated character set, or a ^ symbol depending on where it is relative to other elements in the regular expression”. This level of complexity can lead to mistakes being easy to make during development. This feeds into another common problem with regex in that it doesn’t give feedback due to being compiled at run time. Because of this, developers could be unaware of a potential flaw in their code prior to deployment, which could be exploited by attackers. One way to mitigate this would be through rigid testing and validation. Developers could test inputs that use regex independently of their full code using unit testing, which would highlight a flaw in this without having to wait for deployment. This allows developers to fix the issue without risk of exploitation.

Regex does, however, allow us some security benefits when used effectively. According to OWASP (2024), regex can be used as an input validation technique. For example, if a password should only be numerical, we can set a parameter that passes numerical values using the “\d” regex statement. While this is a simple example, regex statements can be built to be more complex to limit the variation in allowed input, which in turn limits the type of attacks bad actors can use.

## References

Larson, E., (2024) Automatic Checking of Regular Expressions. Available at: http://fac-staff.seattleu.edu/elarson/web/Research/acre.pdf [Accessed 19 October 2024].

OWASP (2024) Input Validation Cheat Sheet. Available at: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html [Accessed 19 October 2024].

Weidman, A., (2024) Regular expression Denial of Service - ReDoS. Available at: https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS [Accessed 19 October 2024].