J. Irvine Secure Software Development

Read Larson (2018) and Weidman (n.d.) then answer the questions below, adding them as evidence to your **e-portfolio**.

You may want to complete this activity in conjunction with or after completing Seminar 2 preparation.

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

ReDoS is a Regular expression Denial of Service that exploits Regular Expression implementations could be in a position where they work very slowly. In a ReDoS attack, the attacker will create a denial of service by using strings that takes a long time to process. This is done according to The OWASP Foundation through exploitation of evil regex patterns.

The OWASP® Foundation defines an evil regex attack as a pattern that has repetition in the grouping and inside the repeated group. Some examples given are:

- (a+)+
- ([a-zA-Z]+)*
- (a|aa)+
- (a|a?)+
- (.*a){x} for x \> 10

These can be vulnerable if the input is aaaaaaaaaaaaaaaaaaaaaaaa!

To avoid an Evil Regex attack you can use multithreading and stop the attacker from killing anticipated processes. If a regex evaluation is taking a long time it should be stopped. You should also refrain from sharing regex patterns online and do not write code that fits the patterns for an Evil Regex attack.

https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS#

https://sec.okta.com/articles/2020/04/attacking-evil-regex-understanding-regular-expression-denial-service

J. Irvine Secure Software Development

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

Regular expressions (regex) are powerful tools for pattern matching and text processing, however a common problem is Regular expression Denial of Service (ReDoS) attacks. ReDoS attacks exploit Regular Expression implementations that reach extreme situations causing them to work slowly.

Regex can be difficult to read and understand, especially for complex patterns. This can lead to errors and bugs in code.

To mitigate an Evil Regex attack you can use multithreading and stop the attacker from killing anticipated processes. If a regex evaluation is taking a long time it should be stopped. You should also refrain from sharing regex patterns online and do not write code that fits the patterns for an Evil Regex attack.

https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS#

https://sec.okta.com/articles/2020/04/attacking-evil-regex-understanding-regular-expression-denial-service

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

Regex can be used as a security measure across multiple layers of a corporation's infrastructure. It can be used used to filter and sanitise user input as a defense mechanism against attacks. Regex can be used by developers to impact the behaviour of firewalls and malware detection programmes. The regex rules can be applied to help system administrators identify potentially dangerous content in files and ensure these are quarantined.

https://learn.microsoft.com/en-us/archive/msdn-magazine/2010/may/security-briefs-regular-expression-denial-of-service-attacks-and-defenses

https://owasp.org/www-community/attacks/Regular_expression_Denial_of_Service_-_ReDoS#

https://sec.okta.com/articles/2020/04/attacking-evil-regex-understanding-regular-expression-denial-service

You can share your responses with tutor for formative feedback or discuss it in this week's seminar.