The first 3 units introduced concepts which are fundamental to the development of secure software, and the collaborative discussion served as an activity which allowed all the learned concepts to be applied to a real-world situation. My chosen example was a password reset system, and discussion of this topic was made possible through the application of the topics learned during the units. Authoritative resources such as the OWASP Top Ten and MITRE CWE were used to highlight the core vulnerability, while UML was used to show how a password reset system could be attacked in detail, and ISO/IEC terms were integrated in peer responses to facilitate clearer discussion.

Peer responses to the initial post provided additional perspectives to consider, and highlighted the challenges involved in implementing the initial recommendations. As Justus (2021) noted, using a microservice-based approach to improve the security of an RNG, would mean that a new attack vector would appear in the system, due to the networking required for microservice creation. Some potential solutions for securing networks were discussed, including encryption, monitoring, and sandboxing. Smirnov (2021) expanded on this topic and discussed the success of implementing stricter authentication of clients via API gateways and tokens. I overall agree with the idea of securing the network-related aspects of the microservice, although taking a simple approach with preventative measures may be ideal. The founder of Black Hat, Jeff Moss, encourages a mindset of "radical simplicity", which translates to focusing on securing critical business operations, as opposed to attempting to secure the entirety of the business (Security Magazine, 2014). Additional benefits of this philosophy include easier vulnerability patching and an automatic decrease in the size of the attack surface (Ward, 2016; Bocetta, 2020). In light of this, it can be argued that the most secure option would be to use a secure RNG library and truly random seed for that library, in conjunction with a token-based API gateway, or rate limiter to prevent users from accessing the service maliciously.

References

Bocetta, S. (2020) Simplicity is the Key to Enterprise Cybersecurity. Available from: https://blogs.vmware.com/security/2020/03/simplicity-is-the-key-to-enterprise-cybersecurity.html [Accessed 6 September 2021].

Justus, M. (2021) Forum discussion with Shan Swanlow, 18 August.

Security Magazine. (2014) The Hunt for Cybersecurity Solutions at Black Hat 2014. Available from: https://www.securitymagazine.com/articles/85863-the-hunt-for-cybersecurity-solutions-at-black-hat-2014 [Accessed 6 September 2021].

Smirnov, A. (2021) Forum discussion with Michael Justus, 20 August.

Ward, D. (2016) CYBERSECURITY, SIMPLICITY, AND COMPLEXITY: The Graphic Guide to Making Systems More Secure Without Making Them Worse. Web: New America.