**Collaborative Discussion 2: The Pros and cons of logging – The impact of log4j**

Logging is essential to record application-related events, including errors, warnings, and state changes, thus enabling to identify the root cause of performance-related issues (Chuvakin & Peterson, 2010; Ots, 2021). Moreover, logging is critical for having auditable application-level information for compliance-related and regulatory purposes, especially in regulated industries, such as finance and healthcare (Putz et al., 2019). It can also be useful to detect any time-sensitive trends and understand how to focus development efforts and ensure further reliability of the application (Chuvakin & Peterson, 2010; Rivera-Ortiz & Pasquale, 2020; Ots, 2021). Nevertheless, sensitive information should not be placed inside the logs, thus being compliant with applicable regulations, such as GDPR (Ots, 2021). Logs should be kept to a minimum to ensure only the required information is stored and to avoid unnecessary costs and time-consuming operations to analyse superfluous logs (Chuvakin & Peterson, 2010; Ots, 2021). Based on specific logs, alerts can be triggered to enable proactive monitoring and intervention to ensure the stability of the application (Chuvakin & Peterson, 2010; Rivera-Ortiz & Pasquale, 2020; Ots, 2021).

It is challenging to handle considerable volumes of heterogeneous logs derived from different sources (Ekelhart et al., 2018). Instead of rules-based methods that cannot understand the context of logged events, Ekelhart et al. (2018) proposed a semantic approach that leverages the background of the logs to derive causal relationships to enhance security monitoring. However, due to vulnerabilities in open-source libraries used for logging (Ots, 2021), such as ‘Log4Shell’ in Apache’s java library ‘Log4j’, considering its advanced capabilities to communicate with various internal services too, cyber-attackers could control devices connected to the Internet remotely (Berger, 2021). As a result, Apache released four patches to tackle this and related vulnerabilities and cloud vendors worldwide, including Amazon Web Services (AWS), along with their customers, were impacted and had to upgrade the library to ensure a fix was in place (Berger, 2021).

Logging-related libraries particularly, especially because they deal with and record numerous application- and database-level information over time, should be vetted appropriately and regularly, not only via company-specific processes but open-source community-based tools to ensure that security vulnerabilities are detected more promptly and mitigate their impact on businesses and, ultimately, society.

**References**

Berger (2021) What is Log4Shell? Dynatrace. Accessed on July 7th, 2022, from <https://www.dynatrace.com/news/blog/what-is-log4shell/>.

Chuvakin, A., & Peterson, G. (2010) How to do application logging right. *IEEE Security & Privacy*, 8(4): 82-85.

Ekelhart, A., Kiesling, E., & Kurniawan, K. (2018) Taming the logs-Vocabularies for semantic security analysis. *Procedia Computer Science*, 137: 109-119.

Microsoft (2021). Guidance for preventing, detecting, and hunting for exploitation of the Log4j 2 vulnerability. Security. Accessed on July 7th, 2022, from <https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/>.

Ots, K. (2021) Logging and Monitoring. In *Azure Security Handbook* (pp. 39-57). Apress, Berkeley, CA.

Putz, B., Menges, F., & Pernul, G. (2019) A secure and auditable logging infrastructure based on a permissioned blockchain. *Computers & Security*, 87: 101602.

Rivera-Ortiz, F., & Pasquale, L. (2020). Automated modelling of security incidents to represent logging requirements in software systems. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (pp. 1-8).