Secure Software Development: Design Document

Background

The National Cyber Security Centre (NCSC) is the Netherlands' consolidated data hub and cyber security knowledge centre. NCSC's objective is to strengthen Dutch society's digital resilience, resulting in a better, broader, and stable digital world. The NCSC provides expert insight into cyber-security innovations, threats, and risks (Government of the Netherlands, N.D.).

Domain-Specific Requirements

Operating systems

- Use of Linux desktops by specialist police unit, since its inception in 2003.
- Upgraded to 2200 Ubuntu Linux workstations.

Technologies

- Use of cloud solutions limits management and development departments as data increases.
- Uses only free and open-source solutions based on open standards and developed publicly.
- Open-source software and open standards identified as a strategic choice and future-proofing.
- Mandatory availability of source code on the internet to be audited.

Others

- GDPR compliance to ensure data privacy and security.
- Monolithic approach due to scalability concerns.

Source: Hillenius (2013).

Proposed System

One of NCSC's main tasks is to continuously monitor all suspect sources on the internet and to alert public organisations of new threats (Government of the Netherlands, N.D.).

The system that will be developed for the Dutch NCSC will allow authorised employees to search, amend and create entries for the suspect sources database. These can then be accessed and utilised by other staff members to assist in the work they do.

The system will need to adhere to the domain-specific requirements mentioned above. It is assumed that the system will be used by three groups of users, each with their own set of access permissions and controls: Specialists, Administrators and External Authorities.

Figure 1 highlights sample use cases for each role. Figure 2 describes the process flow of a specialist viewing and adding a new source to the database.

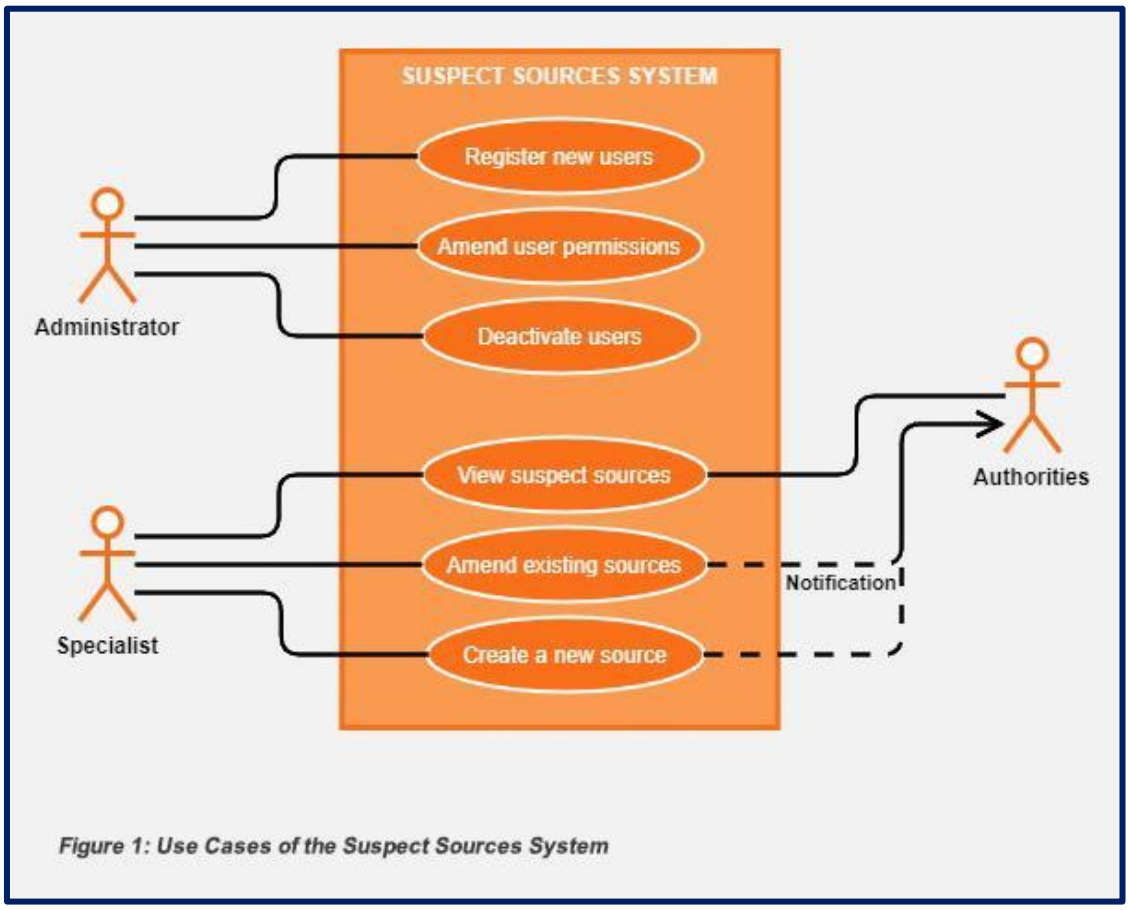

Figure 1: Use Cases of the Suspect Sources System

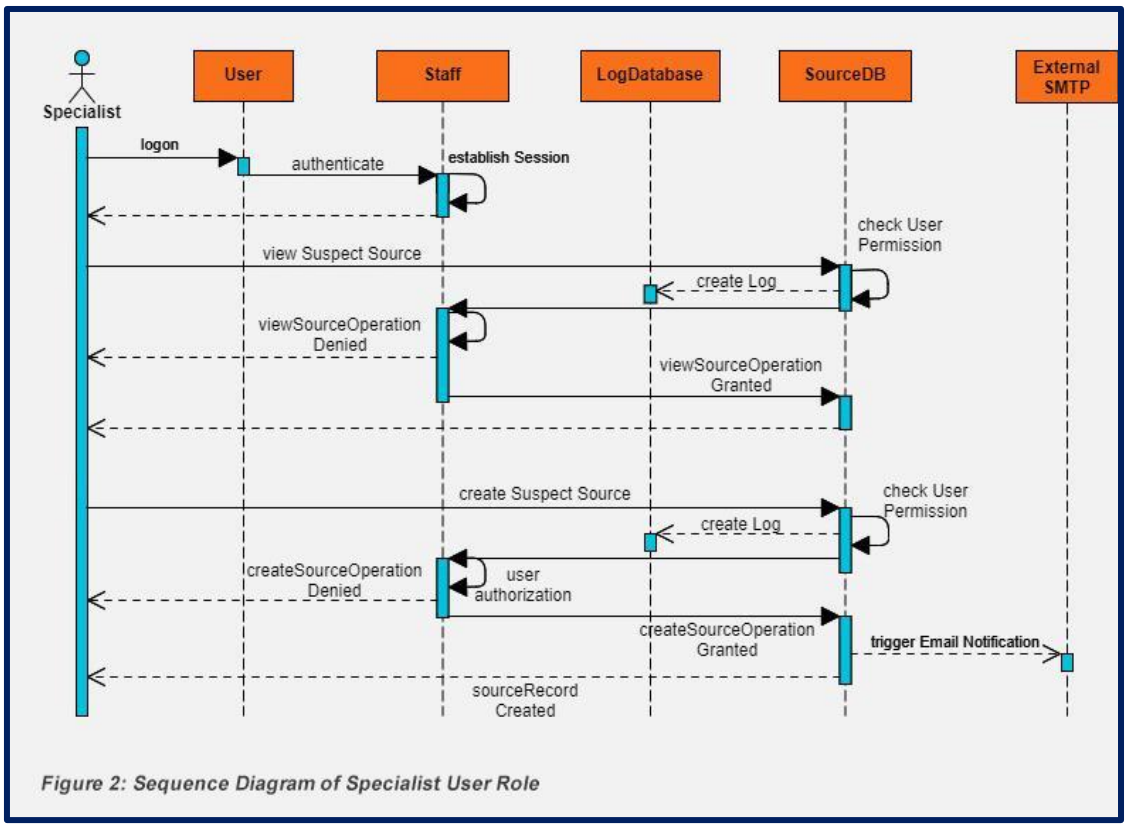

Figure 2: Sequence Diagram of Specialist User Role

The system prototype consists of a CLI that is linked to the suspect sources database. An email notification will be triggered via an external SMTP if a new source is added. Figure 3 describes the architecture of the Prototype.

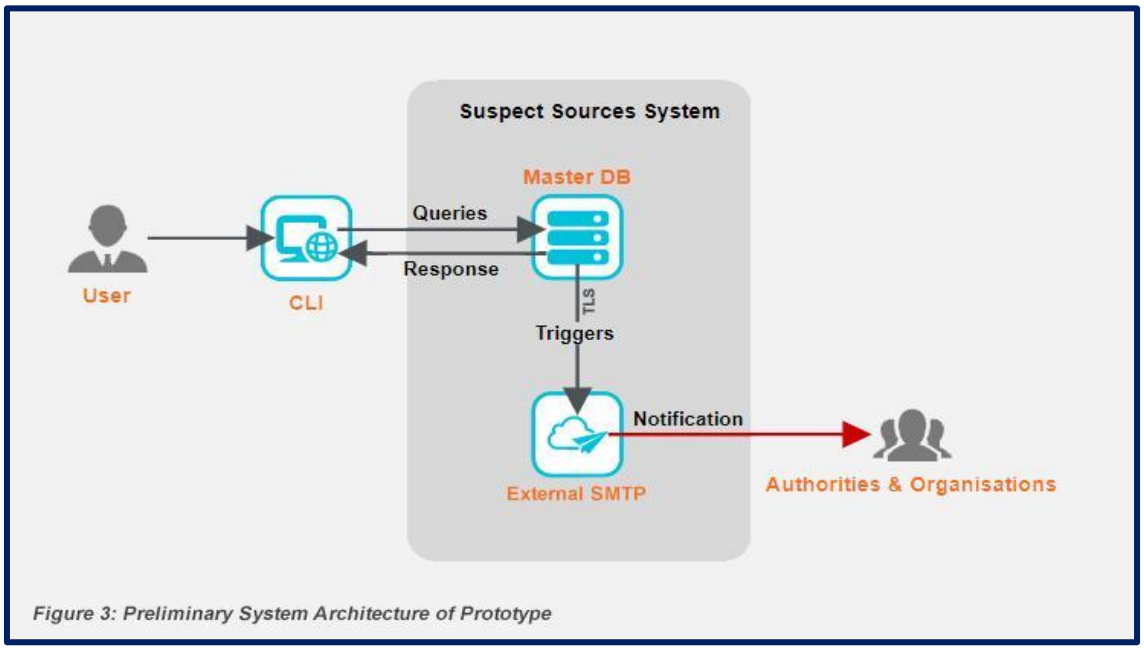

Figure 3: Preliminary System Architecture of Prototype

Challenges

Heralded by principles of the Police Data Act (Wpg) and General Data Protection Regulation (GDPR), privacy by design raise challenges in the application, including commitment management (data retention, sharing, removal) and integrating privacy-enhancing measures in the application design (Spiekermann, 2012; Politie, 2019).

During the SDLC implementation phase, it is vital to avoid any issues related to code readability and reusability to ensure the application's long-term stability (Hijazi et al., 2014).

As an insecure web application undermines data security and privacy, it is key to have security risk-mitigating controls incorporated with reference to the latest OWASP Top 10.

The decision to apply monolithic applications has its drawbacks as fine-grained scaling is impossible without a deployment on every occasion and thus, negatively impact the ability to swiftly add new features (Kalske et al., 2018).

Solutions

To abide by the strict regulations mandated by the Wpg and GDPR, data protection and information security are essential (Politie, 2019). For the proposed system, secure authorisation and logging are introduced to emphasise accountability and documentation.

Only employees who are authorised are given access to data. Personal data, such as user information, is only kept long enough for its collection purpose.

To assure long-term stability, coding antipatterns such as Spaghetti Code (unorganised and incoherent long code), Big Ball of Mud (lack of apparent architecture), and Copy and Paste Programming (duplicating functionality instead of using methods or inheritance) should be avoided (Baum et al, 2018). Instead, modular-written code with sufficient documentation is implemented to ensure scalability and maintainability.

The system will use proactive controls as advised by OWASP. For example, the Principle of Least Privilege will be strictly enforced for database access, and all user inputs will be sanitised and validated (OWASP, 2018).

Tools and Libraries

Created in Python using the web-based IDE Codio:

- PostgreSQL Database to conform to open-source requirements.
- Pylint to check for syntax errors and enforce PEP-8 styling.
- Fernet's symmetric encryption for encrypting and decrypting database credentials and content.
- Argon2 for password hashing.

Reference List

Baum, D., Dietrich, J., Anslow, C. and Müller, R. (2018) Visualizing Design Erosion: How Big Balls of Mud are Made. *2018 IEEE Working Conference on Software Visualization (VISSOFT)*: 122-126. Available from: https://ieeexplore.ieee.org/abstract/document/8530139 [Accessed 23 March 2021].

Government of the Netherlands (N.D.) Fighting Cybercrime in the Netherlands. Government of the Netherlands. Available from: https://www.government.nl/topics/cybercrime/fighting-cybercrime-in-the-netherlands [Accessed 15 March 2021].

Hillenius, G. (2013)' Open Source Only' at Dutch Police Internet Forensics. European Union. Available from: https://joinup.ec.europa.eu/collection/open-source-observatory-osor/news/open-source-only-dutch-p [Accessed 15 March 2021].

Hijazi, H., Alqrainy, S., Muaidi, H. & Khdour, T. (2014) Identifying Causality Relation between Software Projects Risk Factors. *International Journal of Software Engineering and Its Applications* 8(2): 51-58.

Kalske, M., Mäkitalo, N. & Mikkonen, T. (2018). Challenges When Moving from Monolith to Microservice Architecture. Available from https://www.researchgate.net/publication/323312732_Challenges_When_Moving_from_Monolith_to_Microservice_Architecture [Accessed 21 March 2021].

OWASP (2017) OWASP Top 10 – 2017: The Ten Most Critical Web Application Security Risks. Available from: https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf [Accessed 20 March 2021].

OWASP (2018) Proactive Controls. Available from: https://owasp.org/www-project-proactive-controls/ [Accessed 23 March 2021].

Politie (2019) Privacy Statement. Available from https://www.politie.nl/algemeen/privacy.html?sid=228463d3-72e3-4434-8947-933a8e3d3756 [Accessed 20 March 2021].

Spiekermann, S. (2012). The Challenges of Privacy by Design. *Communications of The ACM – CACM* 55 (7): 38-40. DOI: 10.1145/2209249.2209263.