Minimal threat analysis on logs

1. Identification of threats

1.1. Threats to confidentiality

- Unauthorized access to log messages: In the event of unauthorized access, attackers may be able to extract sensitive information such as personal data, passwords, or API keys from log messages.
- Logging of sensitive information: Logging of sensitive information may occur due to an error and/or oversight.
- Exposure through insecure transmission paths: Logs that are transported via networks could be intercepted and read.

1.2. Threats to integrity

- Log message manipulation: An attacker may modify the contents of logs to conceal malicious activity or inject false information.

1.3. Threats to Availability

- Denial of service attacks on the logging system: Attackers could generate a large number of logs that overload the system.
- Data loss due to inadequate backup: In the absence of proper backup, logs may be lost or deliberately deleted.

2. Implemented protection measures

The integrity of the data is ensured through the implementation of hashing with salt and SHA-256. This approach ensures the integrity of the data by storing it in the form of a non-reversible hash value, thereby safeguarding its authenticity and preventing unauthorized access. The integrity of the data can be verified by comparing the calculated hash value with the one sent, thus confirming the data's integrity and ensuring its reliability. The incorporation of salt serves to enhance the complexity of the system, thereby making rainbow table attacks more challenging to execute (if separated from the original message). The second Measure will address confidentiality and integrity using the AES encryption method. AES provides strong symmetric encryption and effectively protects data from unauthorized access. Additionally, incorrect decryption can serve as an indicator of a lack of integrity.

3. Analysis of timestamp services

Timestamp services are mechanisms that ensure the provision of log messages is accompanied by a verifiable timestamp. This practice is intended to protect against the potential manipulation of the timestamp and to guarantee the authenticity of the sequence of events, thereby ensuring integrity and non-repudiation. One possible implementation of this process would be the utilization of Trusted Timestamp Authorities (TSAs) via a service or tools such as OpenTSA or OpenSSL.