September 2023

Configuration-driven Logging Deprecation Without Software Updates

Ashok Chandwaney

Branden Archer

Dan Hastings

Betty Liu

Configuration-driven Logging Deprecation Without Software Updates

ABSTRACT

Software that executes on client devices may generate log files that can be utilized for troubleshooting and debugging purposes. When generation of certain log data is no longer useful, logging is disabled. However, disabling logging requires the pushing of a binary update to client devices. In situations where some fraction of client devices does not receive updates, such unwanted logging continues, which is wasteful of device resources. This disclosure describes techniques to automatically disable logging on client devices without a binary update. The binary on a client device is configured to automatically stop logging after a certain date unless a configuration update is received that instructs it to do otherwise. If the device does not receive such an update, logging automatically stops on the date. If the device receives a configuration update to continue logging, logging is continued. Zombie logging is thus eliminated.

KEYWORDS

- Data logging
- Zombie logging
- Telemetry
- Software update
- Sunset provision
- Time-to-live (TTL)
- Configuration update
- Expiration date

BACKGROUND

Typically, logging on client devices is disabled and old log files are removed when they are no longer useful. At present, disabling logging or removal of unwanted log files requires the pushing of a binary update to the client device. In many cases, some fraction of the devices do not receive the updates. Consequently, unwanted logging on such devices is not disabled.

Zombie logging continues unabated on such devices and consumes device resources such as battery and network while there is no actual need for the logs being generated.

DESCRIPTION

This disclosure describes techniques to automatically disable old (or no longer useful) logging on client devices that do not receive binary updates. The binary (executable) that is installed on a client device is configured to automatically stop sending a given type of log data on a certain date, unless it receives a configuration update that instructs it to do otherwise. If the device does not receive such an update, logging automatically stops on the date. If the configuration is updated, and if the device receives the updated configuration (but not a binary) update, it can continue logging based on the new configuration. Consumers of log data can be notified in advance of an upcoming expiration date for continued receipt of log data.

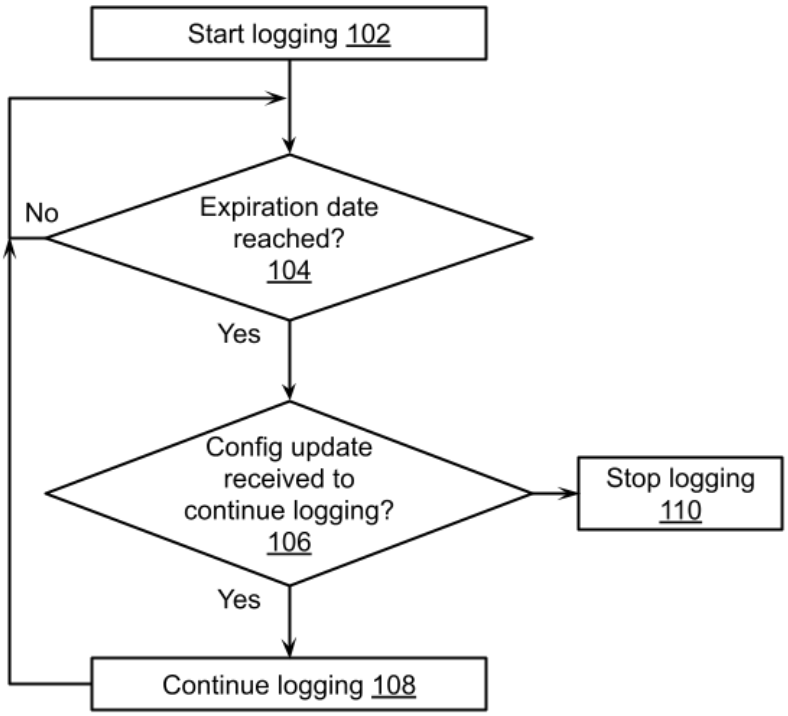

Fig. 1: Deprecating logging without binary updates

Fig. 1 illustrates deprecating logging without a software update (update to the binary executable). A binary received or installed at a client device includes code to perform logging, e.g., to store log data and transmit it for debugging purposes. Based on the configuration, logging is started (102), and continues until an expiration date for logging is reached (104). Upon reaching the expiration date, if no configuration update is received (106) to continue logging, logging is stopped (110). If a configuration update is received with an instruction to continue logging (e.g., till a later date), logging is continued (108).

In this manner, by automatically disabling logging unless a low-friction action is taken to re-enable it, irrelevant logging that consumes battery and network resources is reduced or eliminated without having to update the binary. Reduction of unneeded logging also reduces back-end computational and storage costs, and also reduces compliance risks.

The described techniques automatically disable logging without any updates to the binary and are therefore suitable in situations where not all client devices may receive binary updates as they are released. The techniques are applicable to any device or application that logs data, in particular, devices or apps that log data in bandwidth-constrained environments, e.g., mobile devices or apps. The techniques reduce the volume of unused or irrelevant telemetry that is generated and uploaded. Zombie logging is eliminated.

CONCLUSION

This disclosure describes techniques to automatically disable logging on client devices without a binary update. The binary on a client device is configured to automatically stop logging after a certain date unless a configuration update is received that instructs it to do otherwise. If the device does not receive such an update, logging automatically stops on the date.

If the device receives a configuration update to continue logging, logging is continued. Zombie logging is thus eliminated.

REFERENCES

1. "Configuration update management in AWS control tower," available online at https://docs.aws.amazon.com/controltower/latest/userguide/configuration-updates.html accessed Aug. 20, 2023.
2. "Deleting Amazon S3 log files - Amazon simple storage service," available online at https://docs.aws.amazon.com/AmazonS3/latest/userguide/deleting-log-files-lifecycle.html accessed Aug. 20, 2023.
3. "Logging and monitoring in AWS identity and access management," available online at https://docs.aws.amazon.com/IAM/latest/UserGuide/security-logging-and-monitoring.html accessed Aug. 20, 2023.
4. "How to delete old log and model upload files in planning analytics," available online at https://www.ibm.com/support/pages/how-delete-old-log-and-model-upload-files-planning-analytics accessed Aug. 20, 2023.