CPSC:480 Software Engineering Exercise 4

10/12/2022

Background

The Keyfactor C Agent is a distributed software component that communicates with the Keyfactor platform to manage digital certificates and keys on a device (e.g. car, smartlock, pacemaker). The agent makes an HTTP request to Keyfactor to acquire an agent session, which contains a list of jobs for the agent. Each job will start with another HTTP request to get configuration details for that job.

A job can be any of the following:

- Inventory – Look at pre-defined locations on the device, identify any certificates and keys at that location, and send the certificates to Keyfactor
- Manage – Take a certificate from Keyfactor and add or remove it at a pre-defined location on the device.
- Enrollment – Generate a new key and Certificate Signing Request (CSR), pass it to Keyfactor to have a certificate issued, and install the certificate+key.
- Fetch logs – Collect local agent logs and pass to Keyfactor.

You can find the source code at https://github.com/kilgallin/Keyfactor-CAgent, organized as follows:

- "lib" folder holds 3rd-party code for base64 and JSON formatting
- "openssl_wrapper" and "wolfssl_wrapper" folders hold interfaces to cryptography libraries used for generating keys and certificates.
- agent.c contains the main function and core session registration and job scheduling logic.
- inventory.c, management.c, enrollment.c, and fetchlogs.c hold implementations of the respective job types.
- All other .c files hold helper code related to one particular aspect or concern of the agent program. You will also find .h, .json, .txt/.md files and a makefile.
- Some of the code includes pre-processor switches to build the code for different hardware systems. You can gloss over these for this assignment.

Exercise

Do the following in groups of 2-3 students:

1. For each .c file in the root folder (minus utils.c), identify its main concern and write a sample corresponding requirement ("As a user, I want ___").
2. For each of the following cross-cutting concerns, identify blocks of code in three different files that relate to the concern and briefly describe how/why; state the filename and line numbers for each block.
   - Logging (excluding blocks in logging.c/.h)
   - Data validation
   - Persistence
   - Licensing
   - Memory management
   - Select one other cross-cutting concern of your choice
3. Identify one of the above cross-cutting concerns that the agent might be able to handle better in at least one case and briefly explain how.
4. Identify one challenge that might be likely to come up with localization of the agent, & explain how the code could be modified to make this easier.

Write up a document (approximately 1-2 pages) with the responses for items 1-4. Include the names of everyone involved. One member of the 2-3-person team should submit to "exercise 4" on Brightspace as PDF by **Thursday, Oct 13, 11:59 PM** (tomorrow).

Grading:

1. 26% (1% per concern analysis, 1% per requirement)
2. 54% (3% per block)
3. 10%
4. 10%