Project Summary

Author: Bill Grow [bill.j.grow@gmail.com]
Date: Nov. 6, 2022

Background

The requirement for this assignment was to test Endpoint Detection and Response (EDR) agent software updates, by creating a framework for generating endpoint activity on a host machine and producing logs for the executed activity. This application would live on a set of hosts, with varying operating systems, running the latest builds of the EDR agent. The tool should execute a series of predefined or custom processes, store attributes of those executed processes, then generate logs for the activity in a machine friendly format. The scope of this project was to generate logs for: custom processes, file CUD and network data transmission.

Solution

For this project, I chose to implement the features using a Rails application with SQLite, for a fairly light footprint on the host machine, coded in the organizations current stack. Even though I'm not using Controllers, Views or any web server, Rails still provides an excellent framework for this tool, in a structure a team can easily follow.

The Application centers around one main Model, the `EndpointProcess`. This model is responsible for storing the core data around the process triggered. Creating a new `EndpointProcess` (`EndpointProcess.create(command: 'ls ~')`), triggers a call to a service class named `RedCanary::ExecuteProcess`; which spawns a new process and reports back its PID, name, start time, etc.. This data is used to complete the `EndpointProcess` record.

Two additional Models were created to store data regarding specific process types: `FileActivity` and `NetworkActivity`. These tables contain attributes such as `:file_path` or `:destination_port` and belong to a required `EndpointProcess`, which maintains the actual command (`touch`, `rm`, `curl`, etc.) executed. Additional activity types can be added following this pattern.

Finally, all activity can be exported with the `RedCanary::ExportLogs` service class. This class lets a user export logs in JSON (default) or CSV formats. Additional formats could be easily added to this class. A time-frame can also be specified, to return a subset of logs in a desired range. Additional filters could be added to scope logs as needed.

A rake file has been configured to make running all these commands easier. A full set of tests can be run (Process, files, network, logs) with the command `rake red_canary:run_tests | tee log.json`, which will write the output results to a file called `log.json`.

This project also includes a full suite of tests for all models and services. This test suite can be viewed in the `/spec` folder and can be run with the command `rspec`.

Currently this tool is configured and tested to run on Linux and Mac. To add Windows functionality locales could be added for each platform and used to replace hard-coded commands with the platform specific equivalent.