CASE STUDY

CREATING AN AUTOMATED TESTING TOOL FOR SIEM RULES WITHIN THE DVLA

Thomas Pritchard for BSc (Hons) Computing (Software Engineering)

Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Purpose

SIEM (Security Information & Event Management) tools ingest logs from a variety of sources and runs them against rules to trigger alerts for activity of security concern. For manually created rules, these can be implemented into the live system & placed into silent mode for testing, but there is no mechanism to force logs matching these rules to be produced, if not naturally occuring on the network.

A proof-of-concept tool was requested exploring the possibility of a full tool to gereate specific logs to test alerts, with these **objectives**:

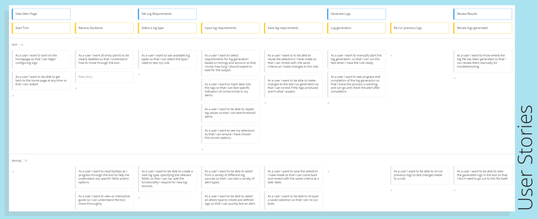
- To identify and evaluate potential tools and techniques to assist with **sample log generation**, and how these tools could be integrated with a **frontend**.
- To evaluate the **feasibility** and potential benefits of a standalone testing tool for SIEM rules by demonstrating a **proof-of-concept** for an automated log generation tool.
- To determine whether an automated testing tool could generate fake logs similar enough to genuine logs that they can be **ingested by a SIEM tool**, with **field changes** input by **users** with varying levels of expertise at the DVLA.

Research

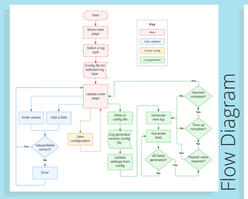
Multiple Log Generation backend tools have been compared, but only one has been found to meet requirements: Peter Scopes' **Log Generator**

This is an open-source tool offering good functionality, but requires additional functionality to be added - along with a user-friendly frontend.

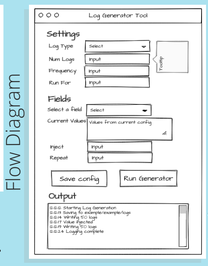
Design



User Stories



Flow Diagram



Wireframe

Language: Python
Used as basis for Log Generator tool, some development already required to improve functionality of log generator.

GUI: Tkinter
Well documented, python integration, platform agnostic.

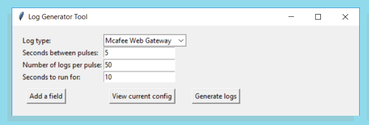
Methods: User Stories, Wireframe, Flow diagram
User-first approach, stories enabled 'chunking' tasks.
Wireframe enabled communication with client.
Flow diagram developed understanding of components fitting together, especially with utilising a third-party Log Generator.

Implementation

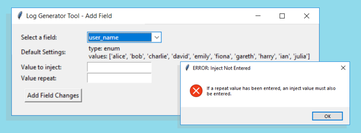
New functionality was required to be developed for the open-source **Log Generator back end**;

- Public and Private IP log field generation
- Epoch time log field generation
- Max log generation run time
- Inject a value into one or more log fields

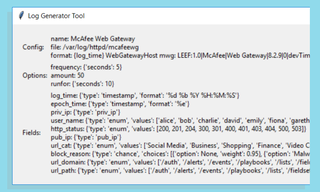
A GUI was created to allow users to set up and generate their own sample logs, to be used for testing their SIEM rules.



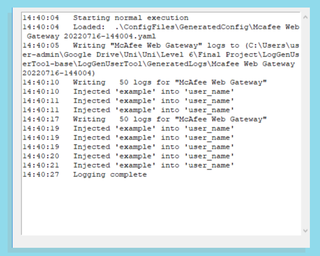
Users can select a log type, and enter settings to be used for the generation, including run times, how frequently to produce logs, and how many logs to produce.



Users can select a log field based on the log type chosen, and inject values one or more times, into one or more fields. An example error can be seen for incorrect entry.



The full configuration file can be viewed, this is the file used by the generator as a template to produce logs. If a user enters new values, they are added/updated and a new configuration file created.



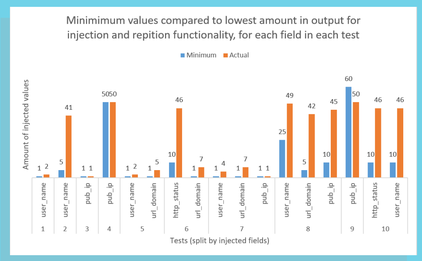
This configuration file is passed as an argument to the updated log generator component, which ingests the settings and begins generating logs based on the set rules and values. Output is displayed to the user to show progress, and they can opt to re-run with the same setting or amend further and re-run.

The two components of the updated log generator in the backend, and the newly created GUI intend to form a proof-of-concept tool meeting the three objectives and showing the **potential for a full-featured tool**.

Development has mostly stopped at the MVP stage for the case study purposes (with a couple of additional features, such as allowing multiple injection fields in one log generation).

Should full development go ahead, there are multiple user stories in the backlog already, but the results of testing have identified some further.

Results



Testing took place with a variety of injected field values and repetition variations. Analysing results (orange bars) against the expected minimums (blue bars) shows that the proof-of-concept was **successfully able to generate more than the minimum** required injection values in all cases but one, which was expected to fail. Also indicated is a potential issue with repetition values causing higher than expected numbers.

100% Sample logs were generated with a variety of test settings, and output logs tested against a regex string from the SIEM tool used for log matching. Results showed **100%** of the logs generated **matched**, evidencing it is feasible to produce fake logs for SIEM tool ingestion. Two issues were also identified with settings for generation frequency and number of logs to produce potentially affecting output.

User testing took place with multiple types of user, general feedback was **positive**, some areas for **future development** were identified mostly around documentation/learning and signposting in the tool.

Conclusions & Future

From this testing, it can be confirmed **all objectives have been met**.

- A Log Generator tool meeting requirements has been identified, tested and integrated with a frontend. In order to achieve this, some additional functionality was added into the generator at development.
- A standalone proof-of-concept tool has been created, which can be built on and eventually introduced to a testing environment, evidencing feasibility. Users identified benefits over current process.
- Testing has shown that 100% of the logs generated were similar enough to genuine logs to be ingested by a SIEM tool successfully, injection test have proved the ability to add fields, and users of varying levels have provided positive testing feedback.

Recommendations for **future improvements** have been made which should be utilised to form requirements and user stories for development sprints going forwards, some of these include:

- Performance enhancements for large generations, i.e. multithreading
- Fixes for 'log frequency' setting issue
- Improvement in value repetition functionality
- Improved user warnings and error messages
- Improved user documentation & training tooltips