Specification

Project Overview

Creation of a smart tool to automate or semi-automate the data extraction and normalization process at Telstra.

Project background

In Telstra, large and varied data sets are used to detect security threats and anomalies. These data sets pertain to all types of logs ranging from access logs to network & security logs.

The data, generated from multiple technologies comes in different shapes as the result of varying

- Naming conventions
- Configurations
- Logging standards
- File formats

> For example, Telstra's next generation firewall data has a number of vendors in this space (Palo Alto, Cisco, Fortinet) each one of them have different log formats with similar data.

Normalization is the pre-requisite to all analysis.

Logs that contain unknown format or are a currently unmapped format can not be analysed by current 3rd party integrations (Splunk, ElasticSearch etc).

The data in these logs needs to be normalised into a valid format after which it can be fed into existing 3rd party integrations allowing for their security analysis.

Current Solution

The current solution requires human effort to understand the logs. This is followed by manually mapping fields from previously unknown formats to a common schema.

This can be characterised as:

- Time consuming
- Prone to human errors.

Project Motivation

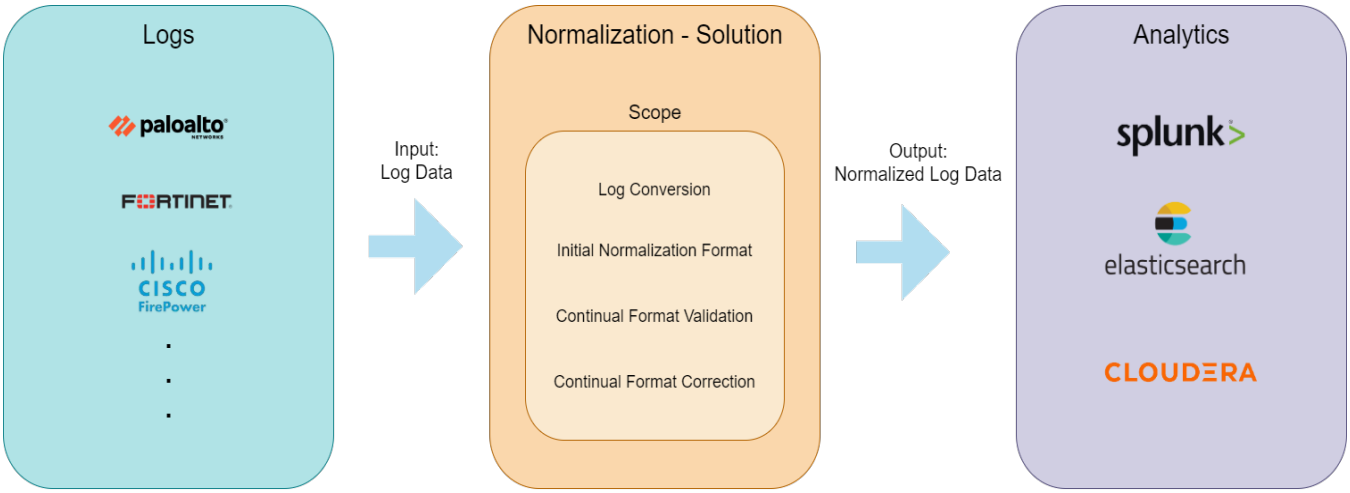
The motivation for this project is to reduce the time expenditure on log normalisation, improve error margins and simplify the cluttered nature of the current process.

This project will produce value to the business and can be characterised as

1. Reduction of human hours spent on manual normalisation.
2. Ingestion of more ever-changing data for security analysis.
3. Increase ability to secure Telstra's Systems, as this normalised data can now be analysed by their 3rd party integrations (Splunk, ElasticSearch)
4. Mitigating the costs of the security breaches from the enormous amount of confidential data Telstra handles.

Project goal

To engineer a consistent approach to data normalization.



Not in Scope:

- Log Data Ingestion
- Analytics

Example of analytic integrations used at Telstra

Elastic ECS

Splunk CIM

Examples of Logs used at Telstra

Palo Alto (most recent format)

Fortinet

Cisco Firepower

Project Details

Core issue

Telstra currently takes in terabytes of log files from firewalls, load balancers, servers, and other network devices on a daily basis. Once collected, these log files are parsed and are then ingested by analytics platforms to be analysed in real time. As log specs vary heavily between device manufacturers /providers, data fields and contents must be normalised to allow for meaningful analysis and comparison to take place. Currently, this normalisation process requires significant manual effort – we are tasked with investigating possible routes towards creating an automated or semi-automated normalisation.

Current Solution:

Primary users:

Engineers / interns / analysts that would have in the past had to investigate and determine mappings themselves.

Secondary users:

Data scientists / analysts / existing automated ML platforms that ingest the processed data.

Current process:

1. Take a ~1000 line file
2. Engineer or Intern will manually match fields based on:
    a. Logging specification or interface agreement provided by logging vendor (not always available)
    b. Guesswork
3. Engineer / Intern creates parser to convert data into a normalized format for consumption.
4. Outputted normalized logging file is a well formed data table, works well for a lot of end user.

Pain points:

- Problem only occurs infrequently, only when adding new software. So building a solution might not be worth it.
- Format changing overtime of log files, new features
- Logging specification not always available
- Different methods of data extraction
- Labor intensive
- Introduce error from manual human work

Output consumers:

Technology	Use Volume	Priority
Splunk	1000's	High
Elastic Search	100's	High
Cloudera	Unknown	Medium

- No common information model
- 3 different teams, inconsistency

Input consumers:

Vendors	Format	Format Notes	Example
Palo Alto	value, value, value, value, value...	Values separated by commas, field depends on position	FUTURE_USE, Receive Time, Serial Number, Type...
Fortinet	Field=value Field=value Field =value...	Field specified with value by equals sign and separated by space	date=2017-11-15 time=11:44:16 logid="0000000013" type="traffic"...
Cisco Firepower	Field = value, Field = value....	Field specified with value by equals with spaces and separated by comma	Group = groupname, Username = user, IP = IP_address ...
Many more vendors			

Note: Same vendors can have different logging formats based on configuration

Input scale and format:

Formats	Volume	Devices
Many, JSON, XML, CSV, Txt	3-5Tb a day	10,000's, Routers, Servers, Endpoints

Solution Expectations:

- Literature review / survey of current open source or commercially available tools that could solve problem.
    - Outlining tools, value, pros, cons, cost, and configuration settings.
    - Common practices in industry
    - Analysis of effectiveness. For example, discussion and comparison around the mapping accuracy of:
        - Rule-based matching
        - Machine learning or statistical inference models
        - Existing mapping processes (may require that Telstra provides data)
- A design for a tool with an interactive interface that can output recommended mappings / format.

The client has expressed no strong preference regarding how the tool is delivered - they are open to web solutions, desktop applications etc.

Scope:

- Our solution should provide recommendations towards automating or partially automating the normalization of log data from a variety of log formats and vendors.
- Our solution should be capable of producing output to be analysed by Splunk or Elasticsearch

Out of Scope:

- No system ingest requirement, just normalization.

Areas of Extension:

- Our proposal could include methods for automatically identifying changes to log and data formats provided by existing systems/network devices and adapting normalization processes to reflect these changes.

Output Expectations

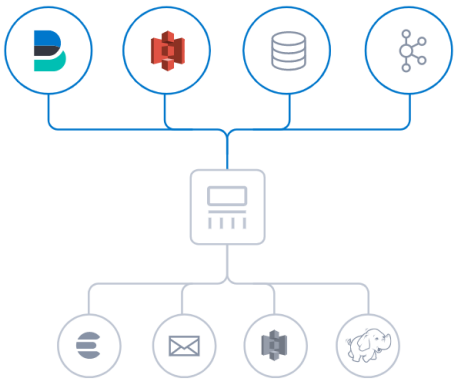
Elasticsearch

Elasticsearch is a **distributed**, **free** and open **search** and **analytics** engine for **all types of data**: including

- textual,
- numerical
- geospatial
- structured
- unstructured

Similar Products - Logstash

Logstash dynamically ingests, transforms, normalizes, and provides access to this new transformed data.
It is sub-component to Elasticsearch which can be retrofitted to difference search/analysis engines, by configuration.



Plugins - Premade Field Extraction:

Allows use of external plugins for applications, allowing ease of integration with existing technology and continual relevance.

- Fortinet Plugin
- Palo Alto Plugin
- Cisco Firepower Plugin

Manual Field Extraction

https://www.elastic.co/guide/en/logstash/current/field-extraction.html

There is a variety of methods to hand create field extraction templates, based on the complexity of data. In an intuitive format.

Example:

Example

```
Apr 26 12:20:02 localhost systemd[1]: Starting system activity accounting tool...
```

Filter Template

```
filter {
  dissect {
    mapping => { "message" => "%{ts} %{+ts} %{+ts} %{src} %{prog}[%{pid}]: %{msg}" }
  }
}
```

Output

```
{
  "msg"        => "Starting system activity accounting tool...",
  "@timestamp" => 2017-04-26T19:33:39.257Z,
  "src"        => "localhost",
  "@version"   => "1",
  "host"       => "localhost.localdomain",
  "pid"        => "1",
  "message"    => "Apr 26 12:20:02 localhost systemd[1]: Starting system activity accounting tool...",
  "type"       => "stdin",
  "prog"       => "systemd",
  "ts"         => "Apr 26 12:20:02"
}
```

Notes

Notes - Joel

Core issue

Telstra currently takes in a broad range of log files from firewalls, load balancers, servers, and other network devices. These logs are collected in analytics platforms, have their contents parsed and info extracted, and are then analysed in real time. As each provider has their own log specs, data fields and contents must be normalised to allow for meaningful analysis and comparison to take place. Currently, this normalisation process requires significant manual effort – we are tasked with investigating possible routes towards creating an automated or semi-automated normalisation.

Current normalisation method:

1. Engineer takes sample file
2. Engineer examines log spec if available / manually inspects log contents, then standardises fields/fieldnames based on lookup from documentation and guesses.
3. Engineer uses this information to build a parser to convert data into a consumable form

Users

Engineers / interns / analysts that would have in the past had to investigate and determine mappings themselves.

Downstream Users

Data scientists / analysts / existing automated ML platforms that make use of the processed data

Target platforms

Solution should be able to normalise data and then output mappings for Splunk and Elastic. Cloudera should also be targeted if possible but sits at a lower priority than Splunk/Elastic.

Data formats taken in:

- JSON, XML, CSV, + many others

Data throughput

- 3-5 TB of data ingested per day
- Tens of thousands of devices sending regular log data

Expected outcomes:

- A proposal or recommendation that addresses problem. Perhaps a literature review / survey of current open source or commercially available tools that could solve problem.
- Outline of the tools, their value, pros and cons, cost, and configuration settings.
- Some analysis of how effective the solution is. For example, discussion around the mapping accuracy of rule-based matching vs the mapping accuracy we can expect to see with machine learning or statistical inference models.
- A design for a tool with an interactive interface that can output recommended mappings / formats
- NOTE: client indicated that they had no strong preferences on if the solution was web based, desktop based etc.

Areas for extension:

If time allows, our proposal could include methods for determining when the format of logs provided by a system/network device changes and adapting to these changes.

Client stated they will provide further information to us regarding:

- Examples of log file contents and pre/post normalisation forms
- Current internal process for handling this normalisation, including documentation / process guides if possible.

Useful references provided by client:

- **Elastic ECS (https://www.elastic.co/guide/en/ecs/current/ecs-field-reference.html)**
- **Splunk CIM (https://docs.splunk.com/Documentation/CIM/4.18.0/User/Overview)**
- **Palo Alto most recent format is defined https://docs.paloaltonetworks.com/pan-os/10-0/pan-os-admin/monitoring/use-syslog-for-monitoring/syslog-field-descriptions.html**
- **Fortinet: https://docs.fortinet.com/document/fortigate/6.4.5/fortios-log-message-reference/524940/introduction**
- **Cisco Firepower: https://www.cisco.com/c/en/us/td/docs/security/firepower/Syslogs/b_fptd_syslog_guide/security-event-syslog-messages.html**

Notes - Luke

Problem Space:

- Extract common security fields for analytic value
- Not a new problem
- Same vendors can have different logging based on configuration
- Servers running operating system we haven't seen before, and there are log files on these files, we want to use these logs to analyze the system for security purposes, to see what that system is doing
- Describes userId in different format, so it is worthless

Current Solution:

Main user :

Range from intern to experienced engineer.

Secondary user:

data scientist, will see the output (be good to know what they want to ingest).

Current process:

1. Take a ~1000 line file
2. Engineer or Intern will manually match fields
    a. Using logging specification, interface agreement, from logging vendor (not always available)
    b. Guessing
3. Use programming language and libraries to parse data into a format, just to consume
4. Outputted normalized logging file is a well formed data table, works well for a lot of end user

Pain points:

- Problem only occurs infrequently, only when adding new software. So building a solution might not be worth it.
- Format changing overtime of log files, new features
- Logging specification not always available
- Different methods of data extraction
- Labor intensive
- Introduce error from manual human work

Solution Expectations:

Proposals:

Input log to system -> interactive display -> output recommended format

Preferred application type : No specific preferred preference

Scope:

- Normalizing the data process
- Output for analytics tool
    - Produce output to be consumed from Splunk or Elasticsearch
- Handle format changing overtime of log files
- No system ingest requirement, just normalization
- Proposal or recommendation to address the problem, a review or survey of existing tools, and what value those tools add, pros and cons of them
- Analysis of the tool effectivity (might need to use Telstra data, what percentage of mapping are we expected will machine learning improve this)
- Want to know what is common practice in industry, what algorythmn or analytic tool to make this task easier. BUT actually wants us to build our automated or semi-automated tool.

Output consumers:

Technology	Use Volume	Priority
Splunk	1000's	High

Elastic Search	100's	High
Cloudera	Unknown	Medium

- No common information model
- 3 different teams, inconsistency

Input consumers:

- 5Tb a day being ingested of logs
- Routers, Servers, Endpoints, all generate logs in different file format
- Logs from most common sources