Large-Scale Indicator Caches

Built using Analysis Pipeline and the Elastic Stack

FloCon 2020

Indicator caches make it quick and easy to find the presence of specific indicators such as IPs or domain names in flow traffic and later associate those cache records with full flow data to avoid expensive searches of the full repository. We tested an indicator cache system capable of processing 40 billion records/day.

Background

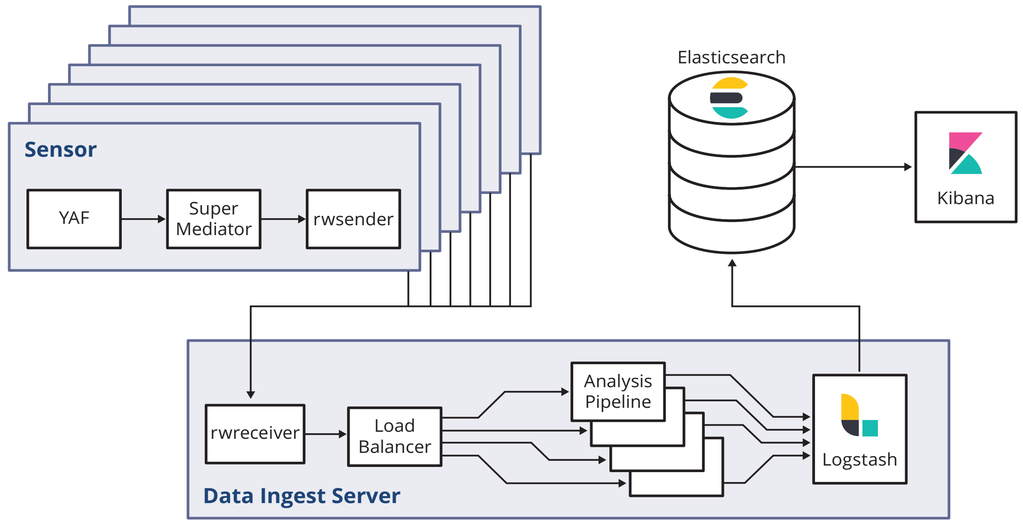
The caches use the following technologies:

- IPFIX: A protocol used for export and storage of network flow information.
- YAF (Yet Another Flowmeter): Processes raw packet data into IPFIX flow records with deep packet inspection information.
- Super Mediator: An IPFIX mediator that is used here for aggregation, data enhancement and file output.
- rwsender/rwreceiver: Tools from the SiLK toolset that are used to move the IPFIX files.
- Analysis Pipeline: A streaming analysis tool capable of aggregating the cache indicators and creating the cache alerts.
- Elastic Stack: Logstash, Elasticsearch, and Kibana are used to collect, store and display the alerts on a large scale.

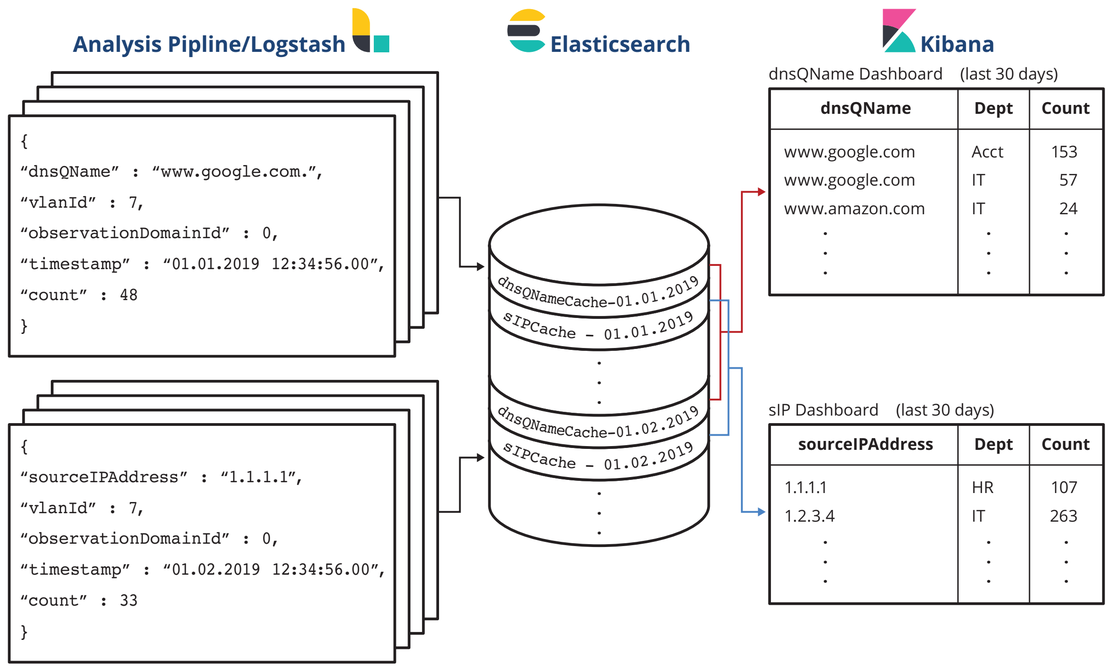
Analysis Pipeline Configuration

```
FILTER dns
    silkAppLabel == 53
END FILTER

STATISTIC dnsQNameCache
    FILTER dns
    FOREACH vlanId dnsQName obsDomainId
    RECORD COUNT
    UPDATE 10 MINUTES
END STATISTIC
```



The sensors use YAF to turn raw packet data into IPFIX flow Records and then Super Mediator to aggregate, enhance, and output the IPFIX files. The files are then load balanced across multiple Analysis Pipeline instances which generate alert logs. The logs are picked up by Logstash, inserted into Elasticsearch, and then grouped and viewed using Kibana.



Logs get emitted from Analysis Pipeline in JSON format, containing the cached item, information to identify the sensor, a timestamp, and a count of how many times it has been seen since the last alert. The items are then stored in Elasticsearch by date and by cache type, and then grouped and displayed by cache item and source using Kibana

Elasticsearch Shard Heuristic

N = expected number of unique indicators for a single cache in 1 day

- $N < 3{,}000{,}000$: 1 shard per index
- $3{,}000{,}000 < N < 10{,}000{,}000$: 2 shards per index
- $N > 10{,}000{,}000$: $\lceil N/5{,}000{,}000 \rceil$ shards per index

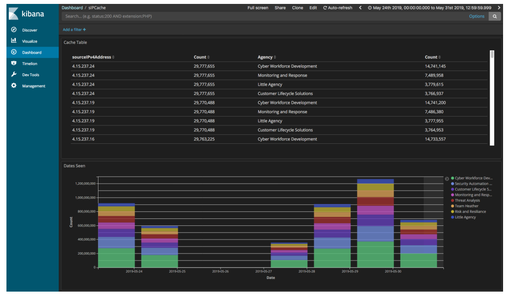
Can be tuned individually for each cache

Cache Query Performance

Calculated on the source IP Cache with 1 week of data

Total Record Count	4,588,939
Records Found	223,168
Query Time	923 ms

Kibana Dashboards



Screenshot of the source IP Cache Dashboard in Kibana

Indicator Cache Records to Full Flow

- Cache records can be associated with full flow data stored in a distributed file system using Apache Spark support built into Elasticsearch
- Data is loaded from Elasticsearch into DataFrames in Spark
- Efficient Spark queries can be constructed using date and department information to restrict searches to a much smaller subset of the data

Carnegie Mellon University
Software Engineering Institute

Dillon Lareau | jdlareau@sei.cmu.edu
Anusha Sinha | asinha@sei.cmu.edu