Model as a Service

Modern Streaming Data Science with Apache Metron

Casey Stella
@casey_stella

2017

Hi, I'm Casey Stella!

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:
  - We ingest network data from many different sources.

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:
  - We ingest network data from many different sources.
  - That data is then normalized and enriched

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:
  - We ingest network data from many different sources.
  - That data is then normalized and enriched
  - Potential threats will be identified in the enriched data as it streams by and the likelihood of the threat will be ranked

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:
  - We ingest network data from many different sources.
  - That data is then normalized and enriched
  - Potential threats will be identified in the enriched data as it streams by and the likelihood of the threat will be ranked
  - The enriched data will be indexed for further analysis in a realtime index and in a batch index

Apache Metron

- Apache Metron is a streaming cybersecurity analytics solution. Roughly speaking:
  - We ingest network data from many different sources.
  - That data is then normalized and enriched
  - Potential threats will be identified in the enriched data as it streams by and the likelihood of the threat will be ranked
  - The enriched data will be indexed for further analysis in a realtime index and in a batch index
- Better enrichment $\Longrightarrow$ more context $\Longrightarrow$ better insights

Enrichment $\Longrightarrow$ Data Science

Threat identification is a hard thing, it turns out. The key to the solution is enabling better threat identification.

Enrichment $\Longrightarrow$ Data Science

Threat identification is a hard thing, it turns out. The key to the solution is enabling better threat identification.

- Fixed rules are too coarse and multiply aggressively.

Enrichment $\Longrightarrow$ Data Science

Threat identification is a hard thing, it turns out. The key to the solution is enabling better threat identification.

- Fixed rules are too coarse and multiply aggressively.
- Statistical baselining can adapt better than rules in some circumstances, but has scale challenges as well

Enrichment $\Longrightarrow$ Data Science

Threat identification is a hard thing, it turns out. The key to the solution is enabling better threat identification.

- Fixed rules are too coarse and multiply aggressively.
- Statistical baselining can adapt better than rules in some circumstances, but has scale challenges as well
- Machine learning can adapt better, but can be hard to scale at high velocity

Enrichment $\Longrightarrow$ Data Science

Threat identification is a hard thing, it turns out. The key to the solution is enabling better threat identification.

- Fixed rules are too coarse and multiply aggressively.
- Statistical baselining can adapt better than rules in some circumstances, but has scale challenges as well
- Machine learning can adapt better, but can be hard to scale at high velocity

The ideal solution is to enable the mixing of rules, statistical baselining and machine learning.

Characteristics of a Solution

- There's a lot of data and it comes at you fast

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive
  - If we assume models are state-free, they look a lot like traditional microservices and can be scaled by replication

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive
  - If we assume models are state-free, they look a lot like traditional microservices and can be scaled by replication
- Integration of new models and retirement of old models must be seamless

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive
  - If we assume models are state-free, they look a lot like traditional microservices and can be scaled by replication
- Integration of new models and retirement of old models must be seamless
  - Registry and Autodiscovery through zookeeper

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive
  - If we assume models are state-free, they look a lot like traditional microservices and can be scaled by replication
- Integration of new models and retirement of old models must be seamless
  - Registry and Autodiscovery through zookeeper
- Many instances of many models will be running at once

Characteristics of a Solution

- There's a lot of data and it comes at you fast
  - Build atop scalable infrastructure: Storm + Hadoop
- Data Scientists rarely standardize on one technology or library for all problems
  - Bring your own Language and Library
  - Model interaction can be done via exposing models as REST endpoints
- Data Science can be computationally expensive
  - If we assume models are state-free, they look a lot like traditional microservices and can be scaled by replication
- Integration of new models and retirement of old models must be seamless
  - Registry and Autodiscovery through zookeeper
- Many instances of many models will be running at once
  - Share resources fairly by using Yarn for deployment and model lifecycle management

Requirements of a Model Creator

We want to ask as little of data scientists and model creators as possible:

Requirements of a Model Creator

We want to ask as little of data scientists and model creators as possible:

- Handle training your model wherever and however you like

Requirements of a Model Creator

We want to ask as little of data scientists and model creators as possible:

- Handle training your model wherever and however you like
- Make your model stateless

Requirements of a Model Creator

We want to ask as little of data scientists and model creators as possible:

- Handle training your model wherever and however you like
- Make your model stateless
- Provide a REST interface serving up your model

Requirements of a Model Creator

We want to ask as little of data scientists and model creators as possible:

- Handle training your model wherever and however you like
- Make your model stateless
- Provide a REST interface serving up your model

From there, MaaS will handle deployment, management and model auto-discovery

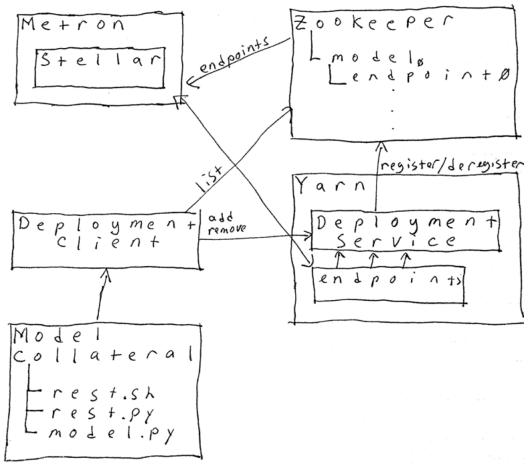
Model as a Service: Stellar

Now that we have a deployment and autodiscovery solution, we need to interact with the models. We think of **Stellar** as like Excel functions that we can run on streaming data:

- Compose a rich set of built-in functions with user defined functions
- Provide simple primitives around the functions: boolean operations, conditionals, numerical computation.

Model interaction is enabled as a set of Stellar functions that discover and interact with model endpoints.

Model as a Service: Architecture



Interlude: Domain Generating Algorithms

Domain Generating Algorithms are used by botnets to communicate with compromised computers in an evasive way. Traffic to a fixed command and control host would be noticed and firewall rules could be modified to cut communication channels cleanly. Instead, the botnet command and control hosts must move around to evade detection. Typically this involves periodically generating a synthetic domain in a repeatable way and having the compromised computers attempt to connect to a few candidate synthetic domains daily with some moderate hope of guessing the right one. This traffic is small enough to be lost in the shuffle of a large organization, making the evasion effective.

Demo

- One solution for detecting synthetic domains is to construct a classifier that must be run against every domain going across the network.
- At the BSides DFW Conference in 2013, ClickSecurity presented a model written in Python for detecting synthetic domains.
- We can expose this model as a REST endpoint, deploy via MaaS and interact with the model via Stellar

```
dga_model_endpoint := MAAS_GET_ENDPOINT('dga')
dga_result_map := MAAS_MODEL_APPLY( dga_model_endpoint
                                  , { 'host' : domain_without_subdomains }
                                  )
dga_result := MAP_GET('is_malicious', dga_result_map)
is_dga := dga_result != null && dga_result == 'dga'
```

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

- A UI for model deployment

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

- A UI for model deployment
- Model Monitoring w/ dashboards

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

- A UI for model deployment
- Model Monitoring w/ dashboards
- Alternative endpoint formats (e.g. gRPC)

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

- A UI for model deployment
- Model Monitoring w/ dashboards
- Alternative endpoint formats (e.g. gRPC)
- Automatic conversion of common model output formats to an endpoint

Next Steps

Going forward, we'd like to add a few things to make this an even more complete solution

- A UI for model deployment
- Model Monitoring w/ dashboards
- Alternative endpoint formats (e.g. gRPC)
- Automatic conversion of common model output formats to an endpoint
- A searchable model registry

Questions

Thanks for your attention! Don't forget to come to the cybersecurity Bird of a Feather session Thursday.

- Find me at http://caseystella.com
- Twitter handle: @casey_stella
- Email address: cstella@hortonworks.com