Shooting the moving target: machine learning in cybersecurity

Ankit Arun
PatternEx
San Jose, CA

Ignacio Arnaldo
PatternEx
San Jose, CA

Abstract

We introduce a platform used to productionize machine learning models for detecting cyberthreats. To keep up with a diverse and ever evolving threat landscape, it is of paramount importance to seamlessly iterate over the two pillars of machine learning: data and models. To satisfy this requirement, the introduced platform is modular, extensible, and automates the continuous improvement of the detection models. The platform counts more than 1000 successful model deployments at over 30 production environments.

1 Introduction

The cybersecurity community is embracing machine learning (ML) to transition from a reactive to a predictive strategy for threat detection. In fact, most cyberthreats exhibit distinctive activity patterns, allowing practitioners to leverage ML to accurately identify attacks. However, while there is a plethora of research on detecting attacks using ML [1], the findings are rarely deployed in real-world solutions.

The limited adoption of ML in cybersecurity is explained by the following challenges: a) the diversity of the threat landscape [2] requires the creation and deployment of a large number of models; b) threats keep evolving to bypass defenses, requiring detection models to be frequently updated.

To alleviate model management effort and to simultaneously tackle the *moving target* problem, we present a scalable, extensible, and automated machine learning platform designed to keep the detection models deployed in production environments up to date. Our platform is designed to satisfy the following requirements:

1. To maintain and to enable the extension of the datasets required to retrain detection models. Each dataset (one per model) contains examples of a particular attack, as well as a representative sample of benign activity. In this paper, we refer to these datasets as "golden datasets".
2. To support modifications to the modeling strategy (namely the addition of new features), and to update the deployment logic accordingly.
3. To seamlessly deploy updated models in production.
4. To do the aforementioned points in minimal time.

2 Overview of our machine learning platform

Figure 1 shows a schematic representation of our platform. In the following, we briefly describe the different modules.

Golden dataset repository The golden datasets are stored in a repository accessed by threat researchers, data scientists, and ML engineers. The repository is stored in Amazon S3.

Configurable data pipelines To simplify and speed up both data ingestion and changes in the feature extraction logic, we have created a configurable and extensible log parsing and feature computation engine.

The parsing engine relies on Protocol buffers (`protobuf`) messages expressed in plain text to convert raw logs into a structured format. The *Log Parsing Engine* in Figure 1 shows a snippet of the `protobuf` message. The logic needed to extract the fields that make up the structured format is declared in `fields` blocks, each composed of the following parameters:
- `name`: the name of the extracted field
- `display_name`: the display name of the extracted field
- `data_type`: the type of extracted field
- `index`: the relative position of the raw data field(s) needed to extract the new field
- `definition`: the definition of the transformation required to extract the new field, declared as a SQL expression.

With this approach, edits to the extraction and transformation logic correspond to configuration changes rather than changes in the platform codebase. To achieve scalability, we rely on Spark jobs to perform the parsing and extraction logic.

In a similar way, features are also expressed as a `protobuf` messages (as shown in the *Feature Compute Engine* module in Figure 1). The extraction of the features is performed by

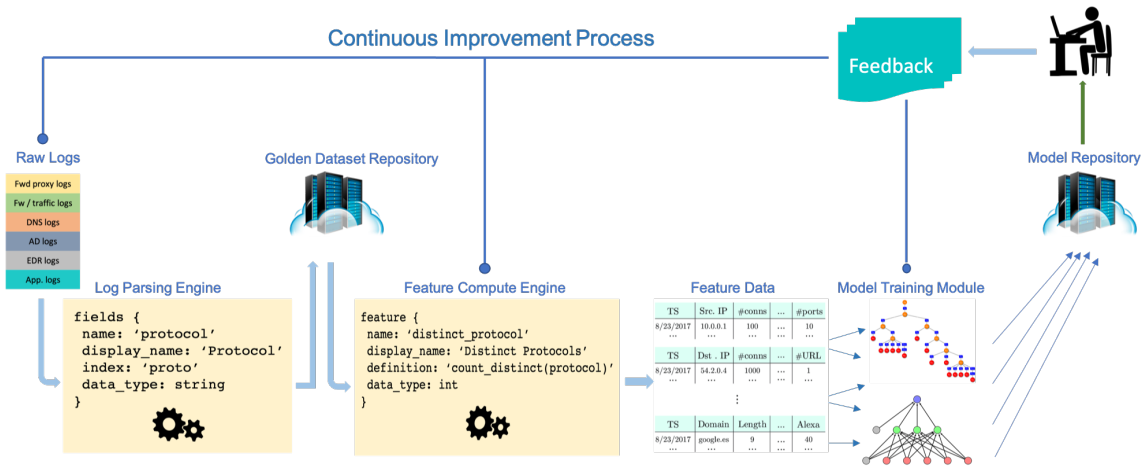

Figure 1: The presented machine learning platform implements a continuous improvement process based on end user feedback to maintain a wide range of cyberattack detection models deployed in production environments up to date.

a Spark job that reads the messages and applies the transformations indicated in the `definition` fields. Note that the definitions are again SQL expressions, and that changes to feature extraction logic (including the addition of new features) entail only the modification of the *feature messages*.

Model training and validation In a nutshell, this module retrieves the newly extracted features and trains machine learning models using the standard machine learning libraries *scikit-learn* and *TensorFlow*[1].

Model repository After training, the models are stored at a central location, making it a one stop shop for all the models.

Model distribution and deployment All the serviced environments share the same parsing and feature computation logic, and periodically pull the models from the repository. This way, the updated models are seamlessly deployed across all the production systems.

3 Continuous improvement process

The threat alerts generated by the deployed models are analyzed by the end users (security analysts or threat researchers working at the serviced environments). As shown in Figure 1, the end users provide feedback, triggering a new model improvement iteration. In the following, we describe the process that takes place when the feedback takes the form of a) new attack or benign examples, b) ideas for new features.

Extending the golden datasets Our threat research team and end users contribute new examples of malicious or benign activities to the existing golden datasets on an ongoing basis. Any time new raw data is contributed, the platform triggers all the steps shown, from left to right, in Figure 1: parsing of the new examples and extension of the appropriate golden dataset, feature extraction, and model retraining and backup in the model repository.

Modifying the modeling strategy We limit the modifications of the modeling to either the addition of new features or the modification of an existing one [2]. As explained in Section 2, in either case the required changes are limited to the edit of configuration files. Any time edits are performed to the feature definition files, the platform triggers the re-extraction of the features for the affected golden datasets, followed by the re-training and distribution of the impacted detection models.

4 Current state of the system

The presented platform currently supports the ingestion of 31 data sources, maintains 27 golden datasets, and counts 70 models readily available for distribution and deployment. As of the day of this writing, the platform has successfully performed more than 1000 model deployments, where each model is updated weekly.

References

[1] Heju Jiang, Jasvir Nagra, and Parvez Ahammad. Sok: Applying machine learning in security-a survey. *arXiv preprint arXiv:1611.03186*, 2016.

[2] MITRE. Adversarial Tactics, Techniques & Common Knowledge. https://attack.mitre.org, 2019.

[3] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li. *AI*^{2}: Training a Big Data Machine to Defend. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud*, pages 49–54, April 2016.

[1] We consider that the details of the modeling strategy are out of the scope of this paper. The interested reader is referred to [3]

[2] Motivation: end users are domain experts that are not well-versed in the advantages and drawbacks of the different ML models and training strategies.