PALANTIR

Newsletter Issue #4

Editorial

Welcome to the 4th PALANTIR newsletter, an EU-funded Innovation Project. In this issue, we describe in great detail the Hybrid Threat Intelligence component whose main goal is to develop a set of tools and algorithms to apply Machine Learning (ML) and Deep Learning (DL) for the detection and mitigation of cybersecurity threats.

Introduction

The main goal of the Hybrid Threat Intelligence (HTI) component is to complement the protection provided by the Security Capabilities (SCs) part of the Secure Services Ecosystem (e.g. IDS, DPI and firewalls) with advanced analytics mechanisms based on ML and DL and to provide automatically generated remediations to address the detected threats. This component is organised in a pipeline of four main subcomponents: Data pre-processing, Multi-Modal Anomaly Detection, Threat Classification & Alarm Management and Recommendation & Remediation.

Key benefits

- Provide novel hybrid incident detection with live threat intelligence sharing.
- Scalable data ingestion and pre-processing
- Muti-Modal Machine Learning on heterogeneous sources of data
- Automatic generation of remediation recipes for the detected threats

Data pre-processing

Data from heterogeneous sources (netflow and system logs) are collected, processed and stored in near real-time through a distributed pipeline by the Distributed Collection and Data Processing (DCP) module. The pipeline is scalable, to ingest and process a big volume of data. Intra-communication and inter-communication with other components happens through Kafka. All collected data are ingested through Kafka to Spark, where the preprocessing functions (including the anonymization of sensitive data using the Cryptopan algorithm) are applied. The preprocessed data are written back to Kafka and then stored to OpenSearch for visualizing the data, enriched with extra information (e.g. IP geolocation). An additional dashboard enables the monitoring of the health and resources usage of all deployed components in a K8s cluster.

Anomaly detection

The preprocessed data is then fed to the Multi-Modal Anomaly Detection (MAD) module which implements a set of anomaly detection modules to detect abnormal behaviours from heterogeneous sources of data. It includes two main pipelines running in parallel and associated to two data modalities: network traffic in netflow format and system logs collected from end hosts. The set of algorithms developed so far includes approaches based on DL, ML and dynamic graphs and have been tailored to address a wide range of attack types (e.g. malware, botnet and volumetric attacks). In the first PALANTIR release, 4 types of models have been considered: Isolation Forest, AutoEncoder, GANomaly and MIDAS.

Threat classification

The Threat Classification & Alarm Management (TCAM) module receives the outliers detected by the MAD and is responsible for classifying them either as false positives or as threats, providing a corresponding confidence score for each predicted label. It supports the two data modalities and implements the corresponding modules as independently trained ML models which can be run in parallel. In the first PALANTIR release, 2 Random Forest ML models have been trained: their algorithmic design is similar but each model uses a different data modality as input and is relevant to classify complementary attack scenarios (i.e., network-based threats and endpoint-based threats). The models have been developed using the Apache Spark distributed computing framework to accommodate efficient big data processing. In the second phase of the project, emphasis will be given to the implementation of the threat findings aggregation and the threat sharing functionalities using the STIX format.

Recommendation & Remediation

The Recommendation & Remediation (RR) module tool is able to evaluate the correct strategy to react to network attacks and malware host infections. The RR suggests the strategy to the network administrator, which can either revise and manually enforce it or automatically deploy it. Starting from the alerts received from the MAD, the RR employs the following steps:
(1) it searches in an internal database the best “recipe” (a generic sequence of operations to handle the specific attack type reported in the alert) which can be actually deployed in the managed network. (2) it interprets the recipe and customizes its operations for the specific network topology. (3) in case of automatic deployment, it concretizes the operations as specific reconfiguration commands written in the languages of the specific SC (e.g. iptables) involved in the recipe. (4) it sends the command to the SCs and informs the network administrator through the dashboard on how the risk has been remediated. The tool is written in Python and has been deployed as K8s pod. Recipes are instead written in an ad-hoc highly customizable language.

Follow us:

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 883335. The information contained in this newsletter reflects only the authors’ view. EC is not responsible for any use that may be made of this information.