Technica™

GraphHunt

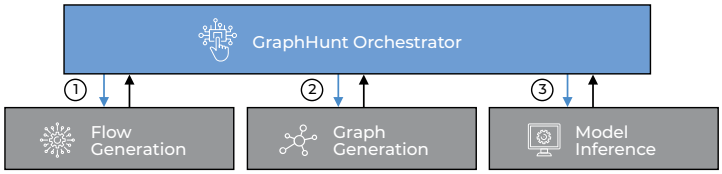
GRAPH NEURAL NETWORKS TO SUPPORT THE CYBER THREAT HUNT MISSION

SOLUTION VALUE

Technica's GraphHunt solution offers an end-to-end system for malicious network traffic classification. Data is ingested as Raw PCAP, encoded into a standard representation, then organized as a graph – predictions are made at a network flow level of granularity. Predictions indicate whether the flow is malicious or benign — if malicious, further classification using the Mitre ATT&CK framework reveals which device is the attacker and what kind of attack was mounted.

SOLUTION

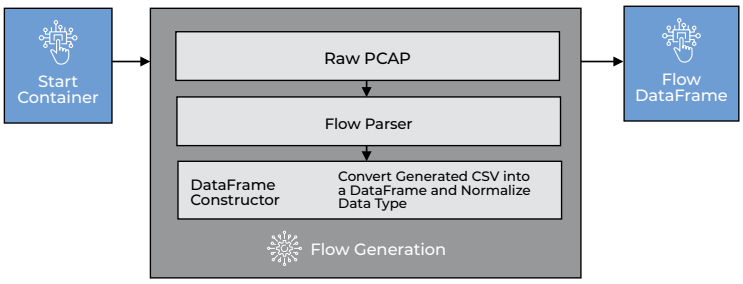
Technica's GraphHunt solution is organized as a three-step pipeline: **Flow Generation**, **Graph Generation**, and **Model Inference**. Each stage of the pipeline is containerized for modularity. Additionally, these containers are coordinated using the GraphHunt Orchestrator. Data is passed between containers as Apache Feather files, a high-performance data file format that supports efficient compression and fast read-write speeds.



GraphHunt Architecture

Flow Generation

The Flow Generation step of the pipeline is responsible for the bulk of the pre-processing. It is the only step that deals directly with the raw PCAPs. As shown next, the raw PCAP file is converted into dataframes with information about the packets and their associated flow, which are passed to the second step of the pipeline.



GraphHunt Flow Generation

Data is encoded to CSV format utilizing Technica's PCAP Parser. Modern machine learning techniques (e.g., Decision Trees, K-means, KNN, Graph Neural Networks) require inputs to be presented as fixed-length vectors; therefore, the data encoder must convert input PCAP data to a fixed-length vector, preserving as much semantic content as possible. Consider a TCP packet and a UDP packet. These packets have different headers and carry different semantic content, but both must be encoded into vectors of the same length, ensuring that corresponding elements in the resulting vector have the same meaning.

WE LISTEN. **WE APPLY. WE SOLVE.**

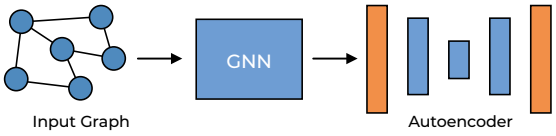
GRAPH GENERATION

Packets from the same flow are grouped together, and the features are computed for each flow. A windowing technique is used to split the PCAP to avoid exceeding available memory, and to support a potential enhancement for predictions on streaming data. Then a graph is constructed from each window, where IP:Port pairs are represented by nodes, the flows between nodes as edges, and the features of each flow as a vector on the corresponding edge.

MODEL INFERENCE AND ARCHITECTURE

During inference, nodes are classified as potentially malicious (anomalous) or potentially benign (normal). If a flow is determined to be anomalous, it is classified according to the attack type; if the attack type cannot be identified, the attack type is classified as "unknown", which may indicate that the attack type was not observed during training. Only nodes identified as anomalous produce an output.

GraphHunt uses separate models to identify and classify flows. The identification model is a Graph Neural Network (GNN) comprised of an E-GraphSAGE[1](EGS) layer followed by an autoencoder. The EGS layer represents each flow as a function of the features of that flow and neighboring flows. The autoencoder detects anomalies in those representation. By combining these layers, the model can perform anomaly detection on graphs.



GraphHunt Model Architecture

Lastly, the custom loss function in E-GraphSage determines whether the flow is malicious, and if so, the type of attack. The loss function determines that the model is wrong and penalizes it only in the case when the attack type is incorrect and the flow is determined to be malicious.

The classification model is an ensemble of models, one for each class, selected and trained by AutoGluon.

SUMMARY

GraphHunt detects and classifies malicious events at a flow level when given a PCAP file for analysis. GraphHunt includes the following:

- A self-contained model that is run from a single script to manage the components and produce the expected output results. This can be done from a central point of entry and further controlled through script orchestration.
- Capability to train the model on provided training and testing data. This allows various models to be used depending on different architectures or training data.
- Final output that categorizes each detected flow with an ID and groups related flows.
- Output that is easily understood by a cyber analyst.
- Ability to detect malicious activity, determine its related flow IDs, and classify them with the appropriate label as described by the MITRE ATT&CK framework. The MITRE ATT&CK framework is the industry standard for descriptors of malicious activity.

[1] Egraphsage: A graph neural network based intrusion detection system. arXiv preprint arXiv:2103.16329, 2021.