Behavioral intrusion detection system based on machine learning

Maxime Lanvin, Pierre-François Gimenez, Ludovic Mé, Yufei Han, Éric Totel, Frédéric Majorczyk
CentraleSupélec, Inria, Télécom SudParis, DGA

Supsec 3rd workshop, September 20th, 2022

CentraleSupélec

Context

Context of this work

- Work on a network intrusion detection system that monitors network packets
- Anomaly detection: we modelize legitimate behavior based on benign training data with no access to attacks
- Based on Sec2graph by a previous PhD (Laetitia Leichtnam)

Goals

- Have good detection performances with limited false positives
- Provide explanations for alarms
- Detect complex APT (Advanced Persistent Threat) attacks

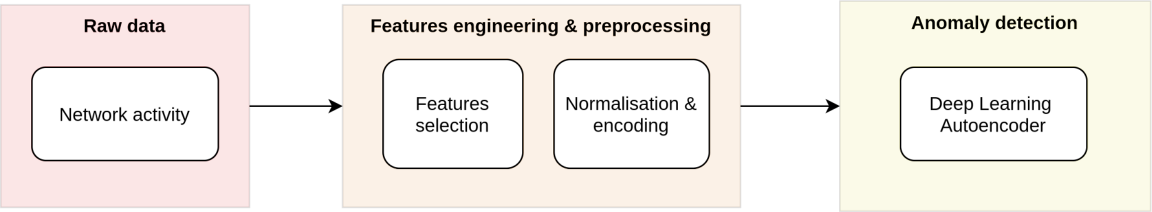
CentraleSupélec

Outline

Overview of the approach

Structure of the approach

- Probes capture the data. For the moment, we only rely on network data
- These data are merged into a graph structure
- The graph is transformed into a format usable with a deep learning model
- The model affects an anomaly score to each data point. From that scores, we can point out what part of the data is anomalous



Data capture

Probe

For now, we rely on public datasets, most notably:

- CIC-IDS2017 (5 days of traffic, 14 machines)
- CSE-CIC-IDS2018 (several weeks, 500 machines)
- DAPT2020 (5 days, 5 machines)

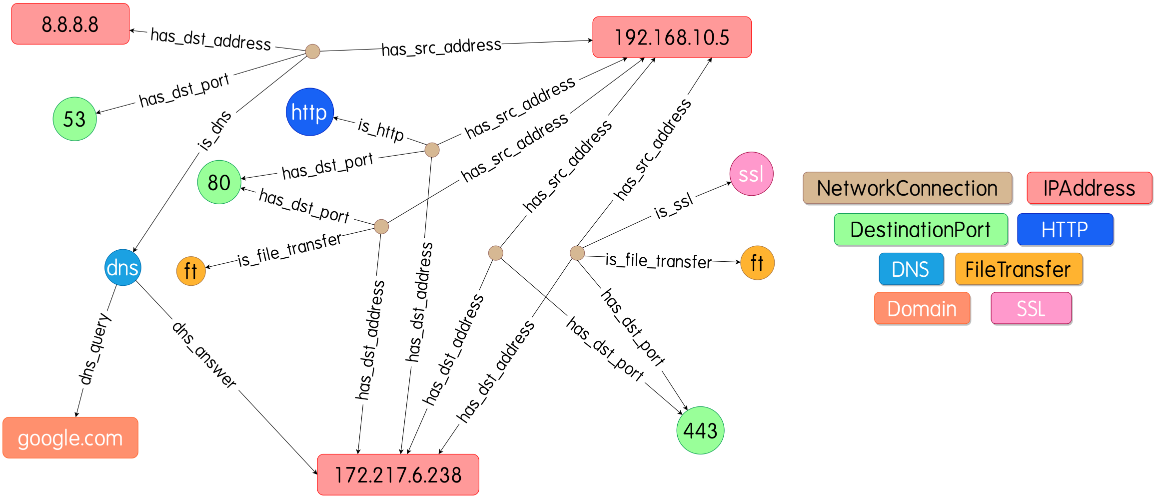
We work directly on the pcap files (the raw capture) and not on the higher levels features

Packet dissector

- We use Zeek (formerly Bro) to dissect the packets
- Zeek creates multiple log files, one for each category of events (network connection, HTTP request, x509 certificates, etc.)
- All events are associated with one network connection

Next step: construct a graph from these logs

Security objects graph built from Zeek's logs



Security objects graph

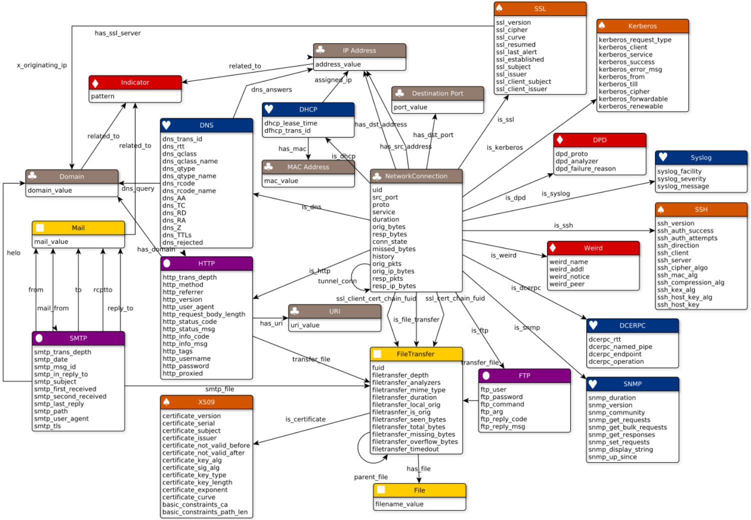
Nodes

- Each node type corresponds to a "security object":
  - protocols: DNS, SSH, DCERPC, SNMP, FTP, DHCP, HTTP, SMTP
  - network data: port, MAC address, IP address, network connection, URI, domain
  - and others
- Nodes contain a set of attributes related to these objects

Edges

- Edges are typed and oriented
- They do not contain attributes
- An edge between two nodes means that these two nodes are found within the same event

All nodes and edge types



Motivation

Why a graph?

- Graph can easily integrate heterogeneous data
- Graph help see the overall structure of the data

Drawbacks

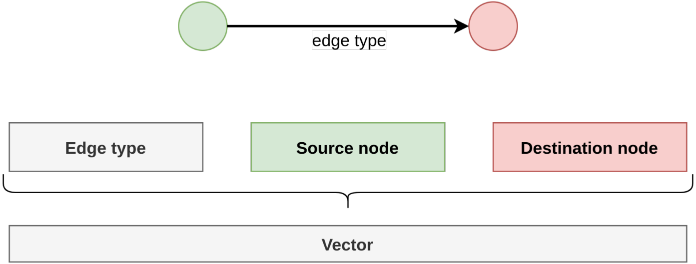
- Graphs hide temporal relations
- Graphs are not straightforward to use with deep learning models
- Even worse with heterogeneous graphs

This structure was mostly designed to help security experts to explore the data and to connect network data with indicators of compromise (IoC)

Graph encoding

Why is the issue?

- We cannot feed the model with the whole graph $\Rightarrow$ we process the graph edge by edge
- Deep learning models generally require a fixed-sized vector with numerical values:
  - To encode discrete values (like port number or protocol), we use one-hot encoding (one feature per value)
  - To encode continuous values (like connection duration), we use a Gaussian mixture model

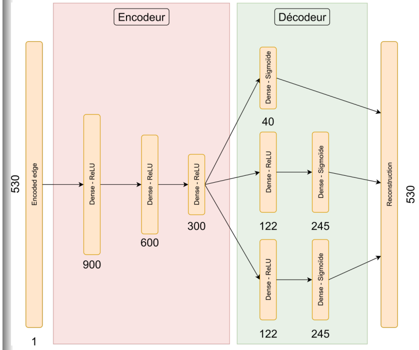


Edge encoding scheme

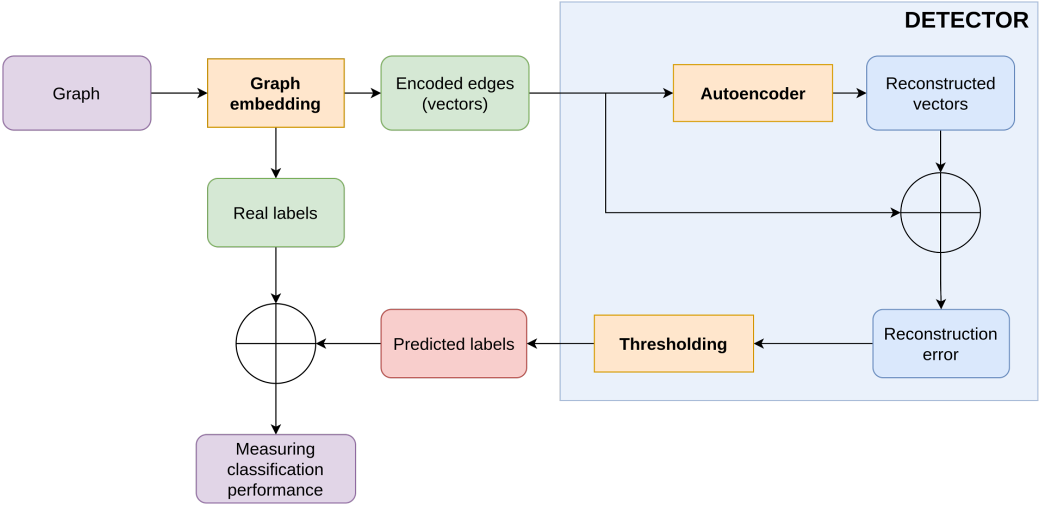
Deep learning model: autoencoder

Autoencoder

- An autoencoder is a deep learning model with the shape of a bow tie
- During the learning phase, the model tries to reconstruct its input data as faithfully as possible
- Due to the bow tie structure, the model needs to find a way to compress the input data by learning the underlying structure of the data
- Once learned, the model is very effective at reconstructing inputs that resemble the training data
- But the compression fails on data too different from the training data!
- We use the reconstruction error as an anomaly score



Summary


DETECTOR
Graph
Graph embedding
Encoded edges (vectors)
Autoencoder
Reconstructed vectors
Real labels
Reconstruction error
Thresholding
Predicted labels
Measuring classification performance

Performances on DAPT2020

Performances

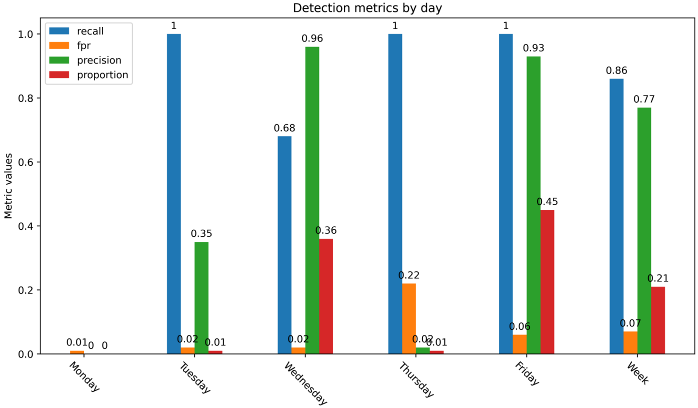
- Experiment on DAPT2020 dataset with APT attacks
- Comparison with the best unsupervised solution proposed by the article (SAE)
- Sec2graph is almost always better
- It has a good recall (it correctly identifies a lot of attacks) and a reasonable false positive rate. However, it's not mature yet for real-world application

	AUC ROC		AUC PR	
APT attack step	*SAE*	*Sec2graph*	*SAE*	*Sec2graph*
Reconnaissance	0.641	**0.888**	0.262	**0.613**
Foothold Establishment	0.846	**0.924**	**0.498**	0.480
Lateral movement	0.634	**0.802**	0.014	**0.603**

Performances on CIC-IDS2017

Performances

Recall is mostly good but we have a very high false positive (22%!) on Thursday



We'll see why later...

How to explain the predictions?

The issue

- Explanations could help us understand the false positives
- There exists a lot of explanation techniques... (LIME, salient maps, counterfactual explanation...)
- ...but little work on explanations for unsupervised learning!

First, naive approach

- We can compute the contribution of each feature to the global reconstruction error
- However, we found out this idea does not produce satisfactory explanations:
  - Some features are always difficult to reconstruct because of their high variance
  - Some features are always very faithfully reconstructed, and even a small reconstruction error may reveal an anomaly

How to explain the predictions?

Our proposal: a statistical explanation

- We split the train data into a training dataset and a calibration dataset
- After learning, we compute reconstruction errors on the calibration dataset
- For each feature, we estimate its distribution of reconstruction error
- During inference, we aggregate the p-value of the reconstruction error for each feature
- The detection threshold is based on this aggregation
- It is easy to isolate the contribution of each feature and output the most influential features to an expert

Evaluation

- We did not perform (yet) a scientific evaluation with experts
- However, we use it to analyze the false positive on CIC-IDS2017

CentraleSupélec

What is the issue with CIC-IDS2017?

Not only one. . .

- Labeling issue: CIC-IDS2017 has a scan attack on Thursday that is not corrected labeled. About 70,000 flows of scan are labeled as "benign"!
- Duplication issue: probably due to a badly configured probe, on average 500,000 packets are duplicated per day. It caused the CSV files to contain bad data
- And a few minors issues

Corrected CIC-IDS2017 files: `https://gitlab.inria.fr/mlanvin/crisis2022`

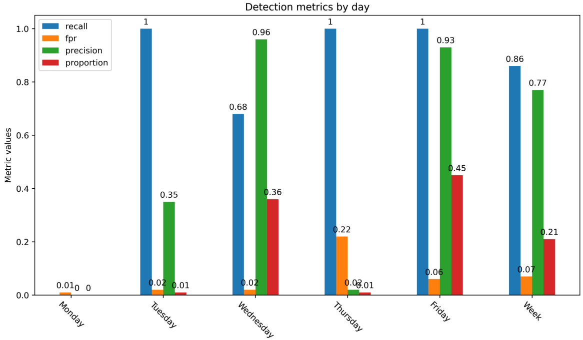
Why wasn't it found before?

Turns out that the missing attack has duplicated packets, so its csv files didn't look like the other scan attacks. Consequence: supervised methods miss this unlabeled attack

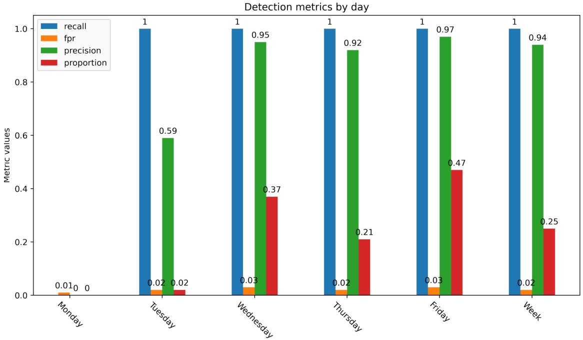
These results make us confident in the usefulness of our explanation method

CentraleSupélec

Updated results on CIC-IDS2017



Before CIC-IDS2017 correction



After CIC-IDS2017 correction

Conclusion and future work

Conclusion

- An anomaly detection approach relying on a security objects graph
- Performances are satisfactory but the false positive rate is still too high

Future work

- Edges should not be processed independently: embeddings and attention mechanisms could help exploit the neighborhood
- Time series analysis is crucial for APT detection: we plan to add new edges between network connections in the security objects graph, with a temporal semantics
- The explanation requires formal evaluation: several evaluation methods are possible, e.g., comparing with other XAI techniques or using experts feedback
- The security graph objects could be extended with other data sources, e.g., application logs