Metrics-based Outlier Detection for 5G security

Athanasios Priovolos, Dimitris Lioprasitis, Georgios Gardikis, Socrates Costicoglou
R&D Department
Space Hellas S.A.
Athens, Greece
{apriovolos;dlioprasitis;ggar;scostic}@space.gr

***Abstract*—5G (and future 6G) networks bring unprecedented benefits, such as softwarisation and edge processing capabilities, which yet also widen the attack surface, making the network more prone to cyberattacks and calling for more sophisticated security controls. We demonstrate an approach for combined collection and processing of monitoring metrics from the RAN and the edge. Processing and analytics are based on Deep Learning, with the aim of detecting anomalies and identifying attacks both to the edge application as well as the infrastructure. Our approach uses bidirectional LSTMs to yield quite promising results, operating in real time in a full-stack 5G testbed.**

***Keywords*—5G, Edge Node, Security, Cybersecurity, Anomaly Detection, Supervised Learning, LSTM, Deep Learning**

I. Introduction

5G comes with a new rich set of features and capabilities[1], which are very important, but also they generate new vulnerabilities in 5G networks. A lot of 5G applications track personal data, or support critical operations[2]. A lot of work has been done about security [3][5] and existing vulnerabilities [4] in 5G networks. The concept of 5G Security Analytics, which is addressed in this paper, is to collect and join data from all structures in a 5G network and analyses them as time series metrics, using AI techniques. In such way zero-day attacks and attacks in very early stages can also be identified. In this context, this paper presents a framework to detect anomalies and identify attacks using Biderectional LSTMs implemented and evaluated in a full-stack 5G testbed

II. Background and Related Work

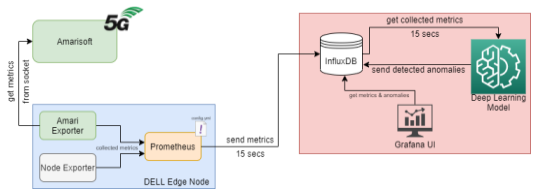

Fig. 1.Testbed architecture and components

A. Background

The Autoencoder is a popular approach for detecting, among others, anomalies in time-series data. Autoencoders are usually trained in normal data. Autoencoders compress taken input data into a smaller representation and then they decompress them into their original form with an error. If this error is above a threshold then the given input can be considered as an anomaly. Long Short-Term Memory (LSTM) networks are a type of recurrent neural networks (RNN) capable of learning order dependence in sequence prediction problems, such as the ones involving time-series data [5].

III. Testbed Architecture and Components

A. Overview

The architecture of our testbed consists of three nodes, as it is shown in Fig.1. The main node of 5G infrastructure is the 5G gNB. There is an Edge compute node, where 5G edge-based applications are deployed and running. Finally, there is a third compute node where the storage, the visualization and the anomaly detection functions are deployed. Our anomaly detection pipeline collects metrics from two different data sources (RAN/gNB and Edge node), predicting which of these metrics are part of an anomaly and visualizes the detected anomalies in a web interface. Prometheus is used for collecting data and InfluxDB for storing them.

B. Data Collection

The Data collection component consists of two data collectors, Amari Exporter, which retrieves metrics from the Amarisoft RAN (gNB) periodically through sockets and the

Fig. 2.Grafana User Interface for monitoring

Node Exporter, which is responsible for collecting system metrics from the Edge node. Collected metrics from both exporters are saved in Prometheus in Edge node. The Node Exporter is a Prometheus component for collecting Linux system metrics from Edge node. The Amari Exporter is also running in Edge node. It was developed specifically for the needs of the present work, to collect gNB metrics. Both the Amari Exporter and the Node Exporter are written in Go programming language.

C. Data Storage & Visualization

The Data storage & visualization component is responsible for fetching collected data from collectors, transforming and store them. In addition, it provides a UI for visualization of the collected metrics and alerts. Prometheus is responsible for collecting and transforming data. They stored to an InfluxDB. For the visualization of collected metrics and detected anomalies, a web user interface has been created using Grafana (Fig.2), which monitors the usage of edge node and shows all detected anomalies.

D. Anomaly Detection Component

For the Anomaly Detection Component, the Autoencoder architecture has been selected. It consists of three parts: an Encoder with 5 Bidirectional LSTMs, a Decoder with 4 Bidirectional LSTMs and a Fully Connected layer. For our model, the following features have been selected: CPU and memory usage from edge node, gNB RX/TX processes CPU usage, edge node Tx/Rx rate in each network interface and gNB radio Tx/Rx rates. For training this model Stochastic Gradient Descent (SGD) with Nesterov momentum has been selected as optimizer, with learning rate equal to 0.01 and ReLU as activation function for 50 epochs. For training, we have created a "normal" dataset, for which we have used two UEs (smartphones) with a mixed Internet usage pattern, running over approx. 30 hours. If the RMSE between the predicted and actual value is above the set threshold, this record is considered as an anomaly and will be shown in UI along with a proposal about what metric may cause this anomaly. The Anomaly Detection Component has also a feature to propose deviation thresholds for each metric, based on 99th percentile of evaluation data. The Autoencoder is developed in Python using Keras with Tensorflow backend.

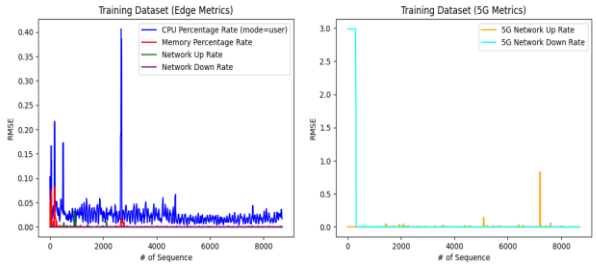

Fig. 3.Prediction Error in normal Data: Edge metrics (left) and 5G metrics (right)

IV. Evaluation

For evaluating the algorithm and the pipeline as a whole, in real time under actual network operation, three different evaluation scenarios are considered. The first scenario corresponds to the normal network behaviour, with two UEs under normal user activity. The second and third scenario correspond to two different types of attacks to the 5G infrastructure and services. More specifically, the second scenario corresponds to a CPU overload (Infrastructure compromise). The third scenario emulates an eavesdropping incident (Service compromise). Fig.3 shows the results from the first scenario, with normal traffic. In left figure the RMSE for edge's metrics is visualized. Right figure shows the same error for 5G gNB metrics. Fig.4 shows the prediction error in second scenario, which is a CPU overload attack, emulating an infrastructure compromise scenario. The two peaks in the plot show the start and the end of the CPU overload attack. So, the trained Autoencoder correctly detects the attack and the corresponding anomaly entries are inserted in Influx DB and shown in Grafana UI. Finally, in Fig.5 the results of the third scenario are shown, emulating a service compromise scenario (eavesdropping). It can be seen in these two plots that the anomaly has been correctly detected and is temporally aligned with the actual incident. As shown in all figures above, in most cases Autoencoder can detect normal traffic with small error and very few false positives, producing very few false alarms.

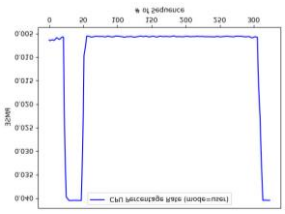

Fig. 4.Prediction Error in CPU Overload Scenario

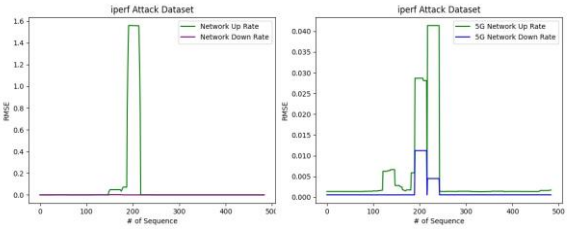

Fig. 5.Prediction Error in Eavesdropping Scenario: Edge metrics (left) and 5G metrics(right)

V. Conclusion And Future Work

This paper describes a proposal for an anomaly detection pipeline for 5G infrastructures, which has as its basis LSTMs following the Autoencoder architecture. This pipeline was evaluated in real time in a fully functional 5G network with over-the-air tests. The data collection, anomaly detection and visualisation modules have been released as open-source (https://github.com/5genesis/Security-Framework) as part of the 5G experimentation enabler framework ("Open5GENESIS"). As future steps, we plan to include more metrics from more network elements (including the 5G Core functions), as well as to conduct tests with more types of attacks.

Acknowledgment

The work described in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 815178 (5GENESIS) and No 883335 (PALANTIR).

References

[1] E. Hajlaoui, A. Zaier, A. Khlifi, J. Ghodhbane, M. B. Hamed and L. Sbita, "4G and 5G technologies: A Comparative Study," 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 2020, pp. 1-6.

[2] Szalay, Z., Ficzere, D., Tihanyi, V., Magyar, F., Soós, G., & Varga, P. (2020). 5G-enabled autonomous driving demonstration with a V2X scenario-in-the-loop approach. Sensors, 20(24), 7344.

[3] 5G Security Landscape, June 2017, 5G-PPP, https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP_White-Paper_Phase-1-Security-Landscape_June-2017.pdf

[4] ENISA Threat Landscape for 5G Networks [Online], Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-5g-networks

[5] Gers, Felix & Eck, Douglas & Schmidhuber, Jürgen. (2001). Applying LSTM to Time Series Predictable through Time-Window Approaches. Recent Advancements, and Future Directions," in IEEE Communications Surveys & Tutorials, vol.22, no.1, pp. 196-248, 2020.