Privacy concerns in IoT systems with the use of AI techniques

Fabio Palomba
fpalomba@unisa.it
Università degli studi di Salerno
Salerno, Italy

Giammaria Giordano
ggiordano@unisa.it
Università degli studi di Salerno
Salerno, Italy

Biagio Boi
Gigi Jr Del Monaco
b.boi@studenti.unisa.it
g.delmonaco1@studenti.unisa.it
Università degli studi di Salerno
Salerno, Italy

1 ABSTRACT

The work tries to discover problems related to privacy in IoT systems; in particular those which make use of smart assistances like Alexa or Google Home. The project studies all the existing works in order to understand if these systems have a good level of reliability and integration between each other. A complete refactoring of a system has been proposed; in particular by thinking about the MLOps paradigm, which guarantee a good degree of control.

2 INTRODUCTION

The evolution of smart devices over last years has increased exponentially and the introduction of these devices in the house is progressively growing. The major problem related to these devices is that usually the privacy is not considered, although there are a lot of regulations (just see the GDPR) that describe how the user data have to be stored and who can access to these data. Starting from these two points we've decided to understand what happens within the context of smart assistance. Various projects have been developed and all these projects try to discover a correlation between packets and possible patterns to discover the conversations and the presence of someone inside the home; which can be seen as serious privacy violation.

3 GOAL OF THE PROJECT

The goal of the project is to assess the reliability of developed projects; in particular, an initial possible integration has been evaluated and consequentially; since this integration has not been possible, a comparison between dataset and pipeline automation has been proposed.

4 METHODOLOGICAL STEPS CONDUCTED TO ADDRESS THE GOALS

4.1 First comparison between projects

As introduced, the project started with the comparison among already developed projects and datasets in order to extract important feature from each project. The first considered project has been that one developed by Kennedy et al. [?], which examine a passive attack on home smart speaker able to infer users' voice commands. The project focuses on a particular metric, the so called semantic distance. Anche se questo progetto si è focalizzato su un punto importante della privacy; il ragionamento che sta dietro al calcolo fatto per trovare la distanza semantica sembra essere abbastanza complicato; per questo motivo l'idea di estendere tale progetto è stata trascurata. Il secondo progetto considerato è quello di Alexa real time analyzer che ci offre una visione più ampia sui dati catturati dalla rete e sulle metriche considerate; il modello creato from this project seems to be powerful on data retrieved by the creator; for this reason we want to assess if the model works good also on other data.

4.2 Dataset Creation

Since the Kennedy project offers a good set of captured packets we've decided to use these files to produce a dataset which is compatible with the format requested from the Boi's project. In particular, since the existing script was able to perform live capturing from a live scenario, we have modified this script to guarantee both capture: from files and from live context; in this way we have automated the creation of dataset starting from both contexts. The decision of creating a dataset from the captured files has been taken for three main reasons:

- To check if the model created from the second project works well on unseen data;
- To create a new model (using various ML technique, that we will see below) based on these new data;
- To assess the possibility of automatically collect new data.

Figure 1: Parser able to do live or files capture

In this phase no check mechanisms has been take in place; in the following subsection we will see these mechanisms.

4.3 Data Debt & Solution

As introduced in the previous subsection, the data are pushed into a dataset by transforming well-structured data, such as Wireshark packet, into a row of a csv file. In particular, by following the documentation related to the Boi's project, the dataset has the following feature:

- **date**: the date in which the packet has been collected, in the format YYYY-MM-DD hh:mm:ss:ms;

- **length**: the length of the packet, it includes just the payload (in case of ack packest it's equal to zero);
- **dstip**: the destination ip, collected to analyze the owner of the server to which the packet is directed;
- **dstport**: the destination port;
- **highest_layer**: the protocol used, in order to parse the protocol into an integer we will use the following mapping:
  – 0 - SSL
  – 1 - TCP
  – 2 - DATA
  – 3 - HTTP

  Notice that all the packets that use other protocol are discarded since they have no meaning for our purpose;
- **delta**: the time occured from the previous packet of the same stream;
- **ack_flag**: the acknowledge flag; it is equal to 1 if the packet contains an ack;
- **microphone**: the status of the microphone, it is equal to 1 if the microphone is active, 0 otherwise;
- **content_type**: the type of content sent, it is valorized only if the packet is sent over SSL;
- **synchronized**: status of synchronization of the device, it is equal to 1 if Echo has been previously associated to an account, 0 otherwise.

Different problems have been issued here; in order to guarantee a well-structured dataset, that avoid data debt, we have decided to do some checks on dataset created from capture files. Following problems have been faced up and solved by applying different techniques:

- When the script tried to analyze packets from the Haipeng and Sean dataset, it was unable to classify application data packets since the highest layer used from the version of Echo Dot was TLS instead of SSL. For this reason, a modification to the parser has been applied in order to correctly classify these packets. Furthermore, other modifications have been introduced to notify the Data Engineer that the parser wasn't able to assign a class to the packet.
- During the dataset overview various None values have been retrieved on the class column (which is the target class) and on content type feature; we've decided to delete the rows which contains None values on target class and notificate the Data Engineer in order to check the behavior of the parser, and consequentially modify it in order to optimize the classification; while we have decided to put 0 to content type since the packets that have null content type are related to synchronization and acknowledge packets. We avoided the possibility to use Data Imputation techniques to substitute the value of this target class column, since it is the most important column of the dataset. Anyway, a final check has been introduced if some new technical modification happens to Echo dot communication system.

4.4 Feature Engineering

After solved problem on structure, quality and integrity of data captured; we will now discuss about the feature engineering process achieved. As first step we checked if data contains feature that are not relevant for the aim of our project - in particular, the date of packet capture and the destination ip seems to be strictly related to the particular instance of the packet; for this reason we removed these feature. Furthermore, after conducting an analysis that considers the feature distribution, we've decided to remove the synchronization feature since there is no packet captured without synchronization. This clear behavior of *univariate feature removal* is caused by the Echo Dot architecture, which does not admin any data transferring without Amazon account association. This last consideration, let us think that this feature might be directly removed from the feature captured at parsing time since it is totally not relevant.

5 METHODOLOGY

In order to implement the tool, we've decided to follow each of these steps:

(1) Consider the current state of art in order to retrieve useful informations. This step is important to produce knowledge to better perform the next steps;
(2) Analyze the existing datasets to achieve feature engineering;
(3) Apply normalization techniques (data cleaning, data balancing);
(4) Implementation and training of a ML model by considering different approaches to find the best fit model for our problem. Looking to the context related projects is most likely that we're going to focus on a Neural Network by using Keras;
(5) Analyze, monitor and compare the results of each model by using ML Flow tool;
(6) Develop a real time tool based on this model.

Clearly, all these steps will be conducted using a MLOps approach.