Privacy concerns in IoT systems with the use of AI techniques

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

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

Biagio Boi
Gigi Jr Del Monaco
b.boi@studenti.unisa.it
g.delmonaco1@studenti.unisa.it
Universitá 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

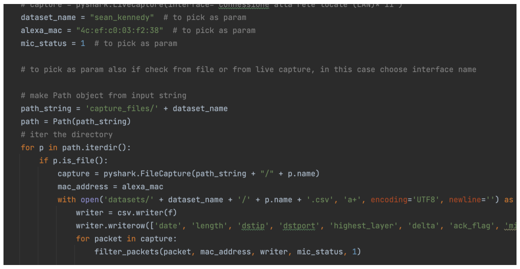
4.1 First comparison between projects

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; the model created 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 second 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.



```
# capture = pyshark.LiveCapture(interface='Connessione alla rete locale (LAN)* 11')
dataset_name = "sean_kennedy"  # to pick as param
alexa_mac = "4c:ef:c0:03:f2:38"  # to pick as param
mic_status = 1  # to pick as param

# to pick as param also if check from file or from live capture, in this case choose interface name

# make Path object from input string
path_string = 'capture_files/' + dataset_name
path = Path(path_string)
# iter the directory
for p in path.iterdir():
    if p.is_file():
        capture = pyshark.FileCapture(path_string + "/" + p.name)
        mac_address = alexa_mac
        with open('datasets/' + dataset_name + '/' + p.name + '.csv', 'a+', encoding='UTF8', newline='') as
            writer = csv.writer(f)
            writer.writerow(['date', 'length', 'dstip', 'dstport', 'highest_layer', 'delta', 'ack_flag', 'mi
            for packet in capture:
                filter_packets(packet, mac_address, writer, mic_status, 1)
```

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 Dataset checks for integrity and compatibility

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.