Proposal Report

Contents

- 1. Executive Summary
- 2. Introduction
- 3. Data Science Techniques
- 4. Timeline
- Citation

An Analytical Framework for Quantifying API Risks

Contributors

- Anupriya Srivastava
- Harry Chan
- Jacqueline Chong
- Son Chau

1. Executive Summary

Application Programming Interface (API) is a set of definitions and protocols for building and integrating application software [1]. Due to the heightened focus on data privacy and security, there is an increased risk of using APIs. TeejLab is a research-driven cybersecurity company, aimed at helping organisations with the evolving challenges of the API economy.

Presently, risk is quantified via manual inspection or a rules-based approach, which is ineffective in and incapable of capturing a multitude of aspects of risk. As such, we aim to create a machine learning model to quantify risk at each API endpoint based on *security* and *data sovereignty* markers. Security refers to how vulnerable the endpoint is to attack, such as the exposure of personal identifiable information (PII) or financial identifiable information (FII), and how stringent the hosting countries' bylaws are for privacy policy. Data sovereignty refers to governance of data, which is at risk if they fail certain security tests, such as injections, or if the endpoints are accessible to users.

More tangibly, we seek to create a data pipeline, in the form of a python script, along with a risk assessment report. This will allow Teejlab to iterate on the present code, add additional security aspects into the machine learning pipeline, and aid in their long-term goal of incorporating a "API risk" column on their platform to provide their clientele an additional metric on the API.

2. Introduction

APIs have been present for decades and are set to grow exponentially, in part due to regulations (in public health or finance) or by industry interoperability (in telecommunications) or disruption (in media or retail) [2]. As a cybersecurity company focused on API management at a global scale, TeejLab aims to tackle this key industry challenge of ***quantifying business risk at the API level***. A manual inspection or a rules-based approach is insufficient to accurately capture various aspects of risk, such as *security, legal, similarity, and data sovereignty*. In addition to those listed previously, the legal aspect is related to the level of protection for users for data use and distribution. Finally, similarity refers to how much user data APIs in the same category are requesting, where one that is requesting for too much data is deemed as less secure.

Based on the dataset provided and time limitations, our team has narrowed the scope to focus on creating a machine learning pipeline to ***quantify the risk of the endpoints of each API based on security and data sovereignty markers***. Our final data product that we will present to TeejLab is a well-annotated python script for each stage - (1) data pre-processing and feature engineering, and (2) machine learning. As this is an unchartered field of research, our team also aims to create a framework for API risk assessment.

3. Data Science Techniques

3.1 Datasets

The dataset provided by TeejLab contains 2,000 observations of Hypertext Transfer Protocol (HTTP) Requests via third-party APIs. Each row of data represents the full HTTP request made by TeejLab Services to the third-party API endpoint, and all HTTP requests are annotated by the level of severity (i.e. High, Medium, Low, No). The overview of the datasets is shown in Table 1 and the detailed description of the dataset is introduced in Table 2.

Label	Number of the samples
High	682
Medium	52
Low	1,211
No	55
Total	2,000

Table 1 : The statistical summary of the Data Endpoints

Column	Type	Description
api_endpoint_id	Categorical	Unique id of API Endpoint
api_id	Categorical	Unique id of API Service
api_vendor_id	Categorical	Unique id of API Vendor
api_vendor	Categorical	Name of API Vendor
api	Categorical	Name of API
category	Categorical	Category of API
usage_base	Categorical	Type of the pricing model of API
sample_response	Text	Sample HTTP Response in JSON format
authentication	Categorical	Authentication method used (e.g. OAuth2.0, Path, None)
security_test_category	Categorical	Category of security vulnerability test
security_test_result	Binary	Result of security vulnerability test
server_location	Categorical	Location of server host
hosting_isp	Categorical	Internet service provider (ISP) that runs website
server_name	Categorical	Name and the version of web server used in API
response_metadata	Categorical	API Response Header
hosting_city	Text	Location of web hosting
Risk label	Ordinal	Severity level of risk (target label)

Table 2 : The detailed description of the columns in the dataset

3.2 Data Pre-processing

During our EDA, we found out that multiple entries were identical with the exception of the request header field “Date” (i.e. timestamp of message) in the “metadata_response” column. However, this information is not useful for our analysis. Moreover, there were insufficient samples for certain risk labels after data wrangling (See Figure 1). More data points are required to make any further statistical conclusions.

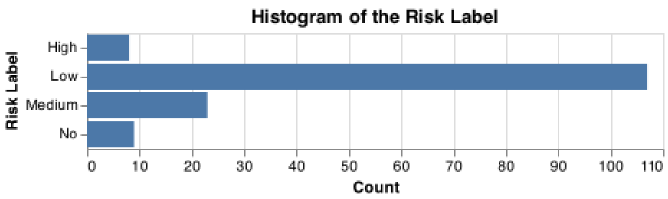

Fig. 1 Distribution of the Risk Label

3.3 Feature Engineering

While there are multiple ways for an API to be exploited, we focused on the features that are available to the attacker, which will help us quantify the risk of a possible attack. This information can be found in the “sample_response” and “response_metadata” column.

In the “sample_response” column, there are two key pieces of information that are critical - PII and FII. These features include individuals’ names, ID, and bank number, which can be extracted via a NLP package. The greater the number of such information is exposed, the greater the risk.

In the “response_metadata” column, there are several keys to be extracted. Information, such as server software, and X-rate-limits, are vital. While it is difficult to concisely elaborate on the importance of each key that we are going to extract, the intuition is that with more information exposed to the attacker, they will be better able to exploit its vulnerabilities.

These transformations can be visualised in Figure 2.

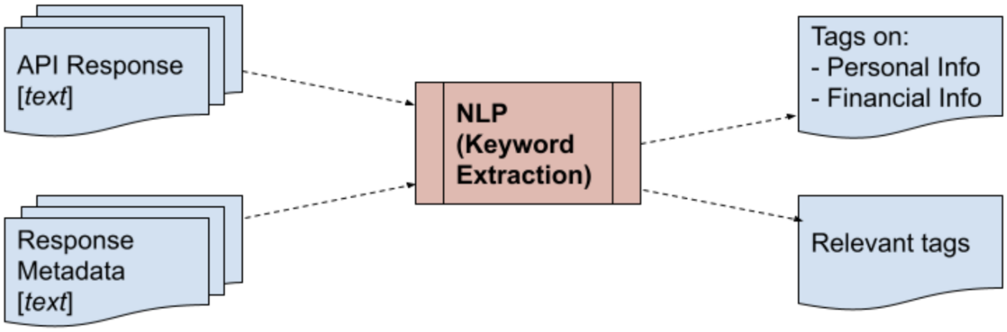

Fig. 2 Feature engineering on API responses

Currently, most rows in “sample_response” contain missing data. As such, while waiting to acquire more data, we will test pre-trained ML models or find a similar corpus to train the model.

3.4 Machine Learning Pipeline

Before embarking on machine learning, we will use a 80:20 train-test split. This is to ensure that we do not influence the test data while training the model.

One challenge is the validity of the provided risk labels. We want to be certain that the labels accurately capture the underlying pattern. Thus, we will first employ unsupervised clustering with three components (to mirror the three risk classes) to evaluate if (a) there are three distinct clusters and (b) if they correspond to the risk labels provided by TeejLab (Figure 3). If a huge discrepancy is observed, we will feedback to TeejLab to request for a relook of labels, as they are the subject-matter experts.

Once we have the final input features, we will train supervised classification algorithms. Presently, we are going to train and optimise Random Forest, CatBoot, XGBoost, and Ensemble Models. However, lest the accuracy is less than 80%, we will look out for other Machine Learning or Deep Learning Models.

In this problem, it is very critical to identify the high risk APIs accurately. Hence, we are using Recall as the primary evaluation metric. However, as we want to avoid low risk APIs to show up as high risk, we will also look at f1-score and accuracy.

Based on metrics scores, we will select the best performing model and build an integrated Machine Learning pipeline for predicting risk classification on new data. This pipeline will be used for evaluating performance on the test data.

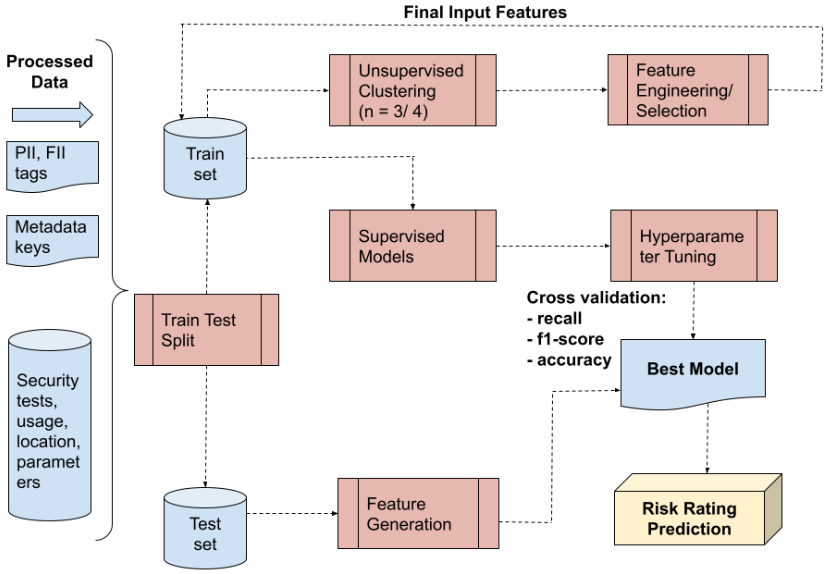

Fig. 3 Machine Learning pipeline

4. Timeline

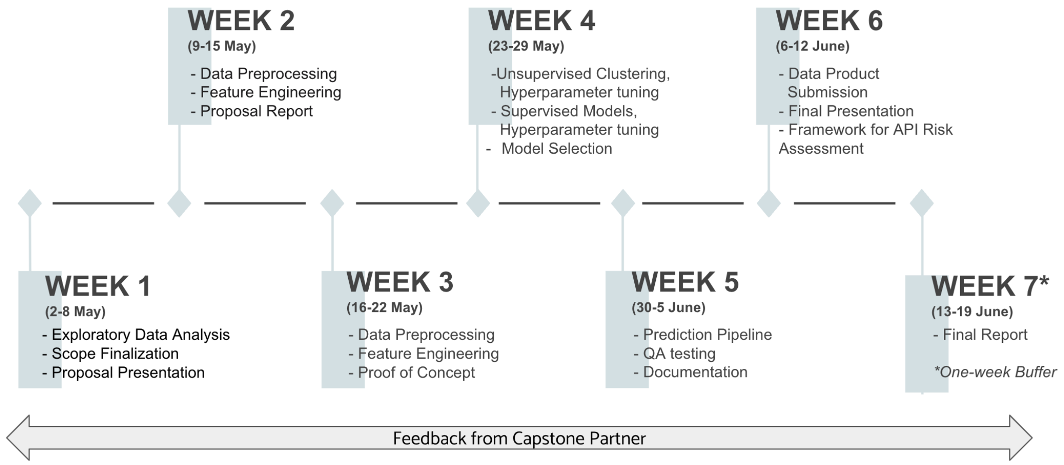

Fig. 4 The proposed timeline

We aim to complete all code components and models by Week 5, and will focus on the documentation, report and final packaging of the data pipeline, ready for submission by Week 8.

Citation

1 What is an api? Oct 2017. URL: https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces.

2 Tiffany Xingyu Wang and Matt McLarty. Apis aren't just for tech companies. Apr 2021. URL: https://hbr.org/2021/04/apis-arent-just-for-tech-companies.

By Team API Risks