Machine Learning Engineer Nanodegree
Capstone Proposal
Abdelrhman Magdy
February 7st, 2019

Domain Background

Malicious Web sites largely promote the growth of Internet criminal activities and constrain the development of Web services.

As a result, there has been strong motivation to develop systemic solution to stopping the user from visiting such Web sites.

A malicious website is a site that attempts to install malware (a general term for anything that will disrupt computer operation, gather your personal information or, in a worst-case scenario, gain total access to your machine) onto your device. This usually requires some action on your part, however, in the case of a drive-by download.

There are many techniques that malicious site often follows, and nowadays our browsers uses these techniques to detect these malicious sites and alert the users.

One of the techniques that malicious sites follows:

- The website asks you to download software, save a file, or run a program
- Visiting the website automatically launches a download window
- You are asked to download an invoice or receipt, such as a PDF file, .zip or .rar, or an executable file or .scr screensaver file

So building an early detector for the malicious web content will be very useful tool for the browsers to alert the users and help them browsing safely.

The idea inspired by this paper: Learning to Detect Malicious URLs

Problem Statement

Confidentiality and Privacy is the biggest challenge faced. If a user accesses the malicious website through search engine with no perception, malicious scripts usually launch attacks to install rogue program, steal personal identities and credentials, so our Problem is to develop an algorithm which accurately predicts the malicious and the benign URLs.

This is a binary classification problem that takes the URL as an input and returns whether it contains malicious content or not as an output.

Datasets and Inputs

Dataset source:

Kaggle platform: Malicious and Benign Websites

- Data set characteristics: Multivariate, Time-series
- Attribute characteristics: Integer, real, characters
- Associated tasks: Classification
- Number of instances: 1782
- Number of attributes: 21
- Data format: csv

The data is imbalanced and contains 1782 labeled URL:

- 216 malicious URLs (12 %)
- 1566 benign URLs (88 %)

The targets are the "TYPE” attribute:

- One means a malicious URL
- Zero means a benign URL

The data attributes:

- URL: it is the anonymous identification of the URL analyzed in the study
- URL_LENGTH: it is the number of characters in the URL
- NUMBER_SPECIAL_CHARACTERS: it is a number of special characters identified in the URL, such as, “/”, “%”, “#”, “&”, “. “, “=”
- CHARSET: it is a categorical value and its meaning is the character encoding standard (also called character set).
- SERVER: it is a categorical value and its meaning is the operative system of the server got from the packet response.
- CONTENT_LENGTH: it represents the content size of the HTTP header.
- WHOIS_COUNTRY: it is a categorical variable, its values are the countries we got from the server response (specifically, our script used the API of Whois).
- WHOIS_STATEPRO: it is a categorical variable, its values are the states we got from the server response (specifically, our script used the API of Whois).
- WHOIS_REGDATE: Whois provides the server registration date, so, this variable has date values with format DD/MM/YYY HH:MM
- WHOIS_UPDATED_DATE: Through the Whois we got the last update date from the server analyzed
- TCP_CONVERSATION_EXCHANGE: This variable is the number of TCP packets exchanged between the server and our honeypot client
- DIST_REMOTE_TCP_PORT: it is the number of the ports detected and different to TCP
- REMOTE_IPS: this variable has the total number of IPs connected to the honeypot
- APP_BYTES: this is the number of bytes transferred
- SOURCE_APP_PACKETS: packets sent from the honeypot to the server
- REMOTE_APP_PACKETS: packets received from the server
- APP_PACKETS: this is the total number of IP packets generated during the communication between the honeypot and the server
- DNS_QUERY_TIMES: this is the number of DNS packets generated during the communication between the honeypot and the server
- TYPE: this is a categorical variable, its values represent the type of web page analyzed, specifically, 1 is for malicious websites and 0 is for benign websites

Solution Statement

The solution is to try different supervised learning algorithms with different hyper parameters using grid search in order to

record the best accuracy.

The supervised algorithm that will included:

- Decision tree algorithms (Random Forest, CART)
- SVM.
- One-Nearest Neighbor

Benchmark Model

I'd like to use the Dummy Classifier from Sklearn as our benchmark model.

Comparing with Dummy Classifier, we can figure out how the prediction model works well.

Evaluation Metrics

As malicious URLs wrongly classified as benign is more important than the benign that classified as malicious and because of the data is imbalanced

We will consider precision as our metric besides the Confusion Matrix and AUC ROC curves.

Project Design

1- preprocessing the data:

- Encode the attributes to integers.
- Using under-sampling and over-sampling techniques to increase the number of positive examples and try to balance the data
- Analyze the data and decide what to do with the N/A values

  Whether to ditch them or replace them with the mean value
- Split data into a train and test data

2- implement the different algorithm mentioned before and start training

3- evaluate and validate the results

4- visualize the different algorithms result

5- comparing the results with the benchmarks

6- optimizing the error, update the parameters and re-train

References

- Learning to Detect Malicious URLs
- Tactics to Combat Imbalanced Classes
- Oversampling and undersampling in data analysis
- metrics should be used for evaluating a model on an imbalanced data set