HARVEY MUDD COLLEGE Computer Science Clinic

Statement of Work for
Proofpoint, Inc.

Predicting Malicious URLs

October 2, 2016

Team Members

Vidushi Ojha (Project Manager)
James Best
Aidan Cheng
Kevin Herrerra
Carli Lessard

Advisor

Elizabeth Sweedyk

Liaisons

Thomas Lynam
Mike Morris

Contents

1 Project Motivation **2**

2 Problem Statement **3**

3 Goals **3**

4 Classifier Types **4**

4.1 Naive Bayes 4

4.2 Support Vector Machines 4

4.3 Clustering 4

5 Feature Selection **4**

6 Architecture **5**

7 Deployment Strategy **5**

8 Schedule **5**

8.1 Phases Overview 5

8.2 Workflow and Deliverables 5

9 Tools **5**

10 References **5**

1 Project Motivation

Proofpoint is a cybersecurity company that provides security and data protection solutions to other companies. Amongst their many products, they provide an inbound email URL screening service that scans URLs embedded in clients' emails, and determines whether or not they lead to sites containing malware.

Determining the maliciousness of URLs is a critical component of Proofpoint's security suite because of the ease with which malware can affect clients' machines. Malware can covertly install itself when a user clicks on a URL, and compromise personal and sensitive information without the victim knowing it. Indeed, attackers can send emails containing these malicious URLs from seemingly benign sources, like someone in the victim's address book, making such emails hard to detect for the client. Thus, Proofpoint would like to to block URLs before they even get clicked.

Proofpoint's solution currently redirects every URL embedded in an email through their servers, where they employ a filter to distinguish between URLs that should and should not be blocked. Their current filtration technique has approximately 70% accuracy in determining whether a URL is malicious. This method checks how many times the URL appears in a certain time period and context, and how many domains it goes through.[1] If a URL reaches a certain threshold with regards to these two test metrics, it will be sent to Proofpoint's *sandboxing environment* for further testing. Sandboxing, the practice of opening a URL on a virtual machine and simulating its effects, is currently the most accurate method of determining whether a URL is malicious. If the sandbox becomes infected with malware, Proofpoint will block that URL in the future.

However, sandboxing is slow, which makes it expensive in both time and money. Given the billions of emails Proofpoint's security suite sees every day, it is unfeasible to sandbox every single one. It would therefore be useful to have a more effective way of determining which URLs are malicious, as this would significantly reduce the number of URLs that have to go through the expensive sandboxing process.

[1] These heuristics are useful because they provide characteristics common to malicious URLs. Many will be sent through multiple domains to try to hide where they came from, and they will be sent multiple times to try to get through to the client.

Although their current method of predicting malicious URLs is relatively effective, there is much room for improvement. Proofpoint is interested in improving the number of URLs blocked overall, but also the number of URLs blocked before the client has a chance to click on them. Our team believes that methods of machine learning are well suited for this problem: given the vast amounts of data, classification and pattern matching are exactly the kinds of solutions needed for this problem. Our hypothesis is that there are common characteristics shared by malicious URLs, and indeed, others before us have investigated such characteristics (see section 5). An effective learning technique could determine these characteristics and use them as facets of a learning model. For this reason, we plan on employing a number of different machine learning techniques to the malicious URL detection problem.

2 Problem Statement

Proofpoint processes billions of URLs a day to determine if they are malicious. However, their sandboxing method is much too expensive a process for it to be attempted for every URL they see. Thus, there is a need for better predictive model that reduces the number of expensive sandbox tests that must be performed. This system should learn from existing metadata about URLs. The ideal solution for this problem would be able to learn from its predictions. For example, if it predicts a URL to be malicious, and that URL is deemed safe by the sandboxing environment, the predictor should refine its model to account for this data. The problem, then, is to construct a model with these characteristics that can make these predictions for the vast number of URLs being processed by Proofpoint on a daily basis.

3 Goals

Over the course of this academic year, our team intends to design and implement a system that uses machine learning to detect malicious URLs before they are clicked. The primary aim of this systems is to block these URLs before Proofpoint's clients view them, thereby avoiding any opportunity for the URL to be clicked. However, while the primary goal is to block URLs before they are clicked, there are three metrics total that will be measured in order to evaluate the success of our model. These are:

- The proportion of malicious URLs tagged as malicious before they are clicked
- The proportion of malicious URLs tagged as malicious total
- The proportion of malicious URLs submitted to the sandbox for additional analysis

We are aiming to improve upon the current proportion of URLs blocked before clicking, which is currently at around 70%. Although we have no precise numbers to outperform with regards to the other two metrics, our aim is to accurately block as many URLs as possible while minimizing inaccurate predictions.

The system we aim to build must have the following features:

1. For any given input URL that is given to it, the system returns a score between 0 and 1, where the score indicates the probability of the URL being malicious.
2. Using the above score, each sample will be classified as either dangerous or not, based on some cutoff. For instance, if we decide on a cutoff of 0.7, then anything with a score of 0.7 or above will be considered dangerous.
3. Our model will explain how the score was assigned, for instance by pointing to characteristics of the URL that make it more likely that it is malicious.

4 Classifier Types

The following is a summary of the classifiers our team will investigate over the course of this project.

4.1 Naive Bayes

4.2 Support Vector Machines

4.3 Clustering

5 Feature Selection

Stuff

6 Architecture

Stuff

7 Deployment Strategy

Stuff

8 Schedule

8.1 Phases Overview

8.2 Workflow and Deliverables

9 Tools

Stuff

10 References

Stuff