Classifying URLs
A Machine Learning Approach

Springboard Data Science Program
Capstone II Project - Milestone Report
Author: Helga Wilde

Background

Malicious URL Content

Malicious web content is a key resource leveraged by threat actors in spamming, phishing and malware attacks. This malicious web content may assist a threat actor in achieving objectives such as credential harvesting, data exfiltration, or data destruction. To deliver malicious content, threat actors may utilize their own web-based infrastructure or use compromised, reputable websites.

How do users determine if a url presented in an email, on social media, in a text or on a website, is safe? Users rely heavily on the security industry which has developed systems to protect users from making an uninformed choice. For the security industry, url analysis has historically been based on 1. Reputation of a url and its underlying domain infrastructure and 2. Analysis of the underlying web resource: connection and evaluation of the web page's source code, evaluation of the communication between web client and web server, and heuristic analysis of the follow-on activity on the client system. This type of analysis can and does support a multitude of detection and prevention-based security tools, either by identifying the integrity of specific domains and urls, or by providing details on specific malicious tactics, techniques and procedures (TTPs) used by threat actors.

Machine Learning Techniques

Machine learning techniques are being applied to cyber security problems, and there are several machine learning models which analyze (and thus classify) urls. These studies utilize one or more of the following sources to develop data feature sets used to train and test new models:

1. The URL itself. Lexical, aka textual properties, of the URL link.
2. Host-based characteristics
    a. Reputation lists for url, domain and/or hosting IP address.
    b. Domain name registration information. WHOIS properties such as name servers, associated IP addresses, registrant and registrar records, and dates like domain creation, update and expiration.
    c. Domain name resolution information. DNS records pertaining to the hosting infrastructure, including A, MX, NS, PTR records, IP addresses across all records. Geographic location may be utilized as well.
    d. Connection speed to/from web client and host
    e. Link popularity. A measure of traffic to/from url resource as compared to established, benign web resources.
    f. URL resource code. Assessment of web content such as links, tags, scripts.

Project Goals

The goal of this project is to build a feature set and machine learning model that:

- Accurately classifies a url as benign, phishing or malicious
- Is self-reliant and not dependent on the existence of url reputation data, connection to a potentially short-lived url link, domain registration or DNS information. Analysis that is not compromised by the use of anti-forensic techniques.
- May be incorporated into an existing security operations and automation tool, to quickly assess the risk of the activity surrounding a url.

Project Approach

Feature Set

The following limitations were considered with regard to historical url reputation and analysis techniques:

- A threat actor's infrastructure and TTPs change over time
- Detonation infrastructure resources or subscriptions are valuable, yet costly, when used to analyze malicious web content. Anti-forensic measures may curtail proper analysis.
- The inherent risk in connecting to malicious web links for analysis purposes
- Reputable websites with clean records are often compromised and leveraged to host malicious content. Features created from domain registration records may be useless in these cases. Blacklists alone are not a 100% reliable resource for url analysis.

Given these limitations, this project will attempt to focus solely on a url's lexical features to build a safe, efficient machine learning model.

Two approaches will be used. The first approach is to build out a large feature set based on the url string and url components (such as domain name, path, etc.).

The second approach is to rely solely on a feature set of url tokens and use natural language processing techniques on these tokens prior to model training and testing.

Classification Model

This project's goal is to build one multiclass classification model and accurately classify a url as benign, phishing or malicious. However, if accuracy scores vary considerably between categories, separate models will be built for phishing and malicious url classification. For either scenario, our goal is to understand the underlying features important for each classification.

Data Sources

Phishing URLs - Over 17,000 phishing url links were retrieved from PhishTank[1]. PhishTank is a collaborative clearing house for data and information about phishing on the Web. It's url lists are available to developers for integration into tools and applications.

Malicious URLs – Over 600,000 malicious url links were retrieved from abuse.ch[2]. Abuse.ch operates the URLHAUS project, which collects and shares malware URLs to assist network administrators and security analysts in protecting their networks from cyber threats.

Benign URLs - Over 25,000 urls were collected by crawling Alexa's list of the top 2500 websites[3]. Internal and external links were captured. In order to validate that each url was 'benign', each url's reputation was checked via Virus Total's reputation service[4]. VirusTotal inspects urls with over 70 antivirus scanners and URL/domain blacklisting services, as well as other tools. Virus scans were requested in those instances where a url had no previous scans or reporting available.

Data Wrangling

A sample of 10,000 records was taken from both the phishing and malicious url lists, and subsequently all url records were labeled with a category, then merged together. The resulting dataset consisted of 25,077 benign, 10,000 phishing and 10,000 malicious urls, with each record consisting of a url and corresponding category.

Feature Creation

Six new features were created by parsing the url into scheme, netloc, path, params, query, fragment.

Domains were extracted from the netloc section. E.g. example.com.

Finally, additional features were created to reflect the lexical characteristics of the entire url link and the parsed out registered domain, netloc, paths, parameters, queries and fragments[5].

[1] http://phishtank.org/developer_info.php
[2] https://urlhaus.abuse.ch
[3] https://www.alexa.com/topsites
[4] https://www.virustotal.com
[5] See Table 1 for feature descriptions

Table 1: Feature Descriptions and Applicable URL Sections

Feature Type	Description	URL Section						
		URL	Domain	NetLoc	Path	Param	Query	Frag
Length	length	✓	✓	✓	✓	✓	✓	✓
	avg section/token length	✓	✓	✓	✓			
	shortest path length				✓			
	longest path length				✓			
Composition	list of all tokens	✓						
	number of sections		✓	✓	✓			
	number of letters	✓	✓	✓	✓	✓	✓	✓
	number of numbers	✓	✓	✓	✓	✓	✓	✓
	number of special characters	✓	✓	✓	✓	✓	✓	✓
	percent of letters	✓	✓	✓	✓	✓	✓	✓
	percent of numbers	✓	✓	✓	✓	✓	✓	✓
	percent of special char	✓	✓	✓	✓	✓	✓	✓
	percent of uppercase letters	✓			✓			
	percent of lowercase letters	✓			✓			
	location of last //	✓						
	location of last slashes as %	✓						
	number of @ signs	✓						
	number of underscores	✓						
	number of question marks	✓						
	number of %20				✓			
	entropy	✓	✓	✓	✓	✓	✓	✓
	number of masques	✓	✓	✓	✓	✓	✓	✓
	character continuity rate	✓						
	is domain an ip address		✓					
	is domain in Alexa top 500		✓					
	number of subdomains		✓					
	number of domain suffixes		✓					
	number of single character paths				✓			

Following is a brief description of several key features.

List of all tokens: This is a list of parsed elements from the url, specifically each substring of the same character type. For example www.google.com would be reflected as [www, ., google, ., com].
Entropy: A Shannon entropy score is calculated, reflecting the string's character distribution. Larger character distributions equate to a higher score.
Character continuity rate: Reflects the total of: length of the longest alphabetic string + length of the longest digit string + length of the longest special character string. This total is then divided by the length of the url.

Number of masques: A masque reflects a letter + digit + letter combination. This characteristic may reflect masquerading, an attempt to spoof a legitimate string with a deceptive replacement. E.g. goog1e.
Percent of lowercase | Percent of upper case: These two features reflect the percent of these characters compared to all alphabet characters in the referenced string.
Is domain an ip address: It's not that uncommon for a url to have an IP address in lieu of a domain, but malicious links have a *much higher* occurrence.
Is domain in Alexa top 500: Each domain was checked against Alexa's list of the current top 500 websites[6].

Final Data Wrangling Steps

Categorical features containing strings, such as url and other url elements, were dropped. Features with Boolean data types were changed to integer. We have no missing values. The dataset now consists of ninety-six predictor variables and one target variable. Many of the predictor features are likely correlated with one another, and feature selection will be necessary before training a model.

Exploratory Data Analysis

Exploratory data analysis techniques were used to investigate, analyze and summarize characteristics of the url string and its components. This section covers key findings.

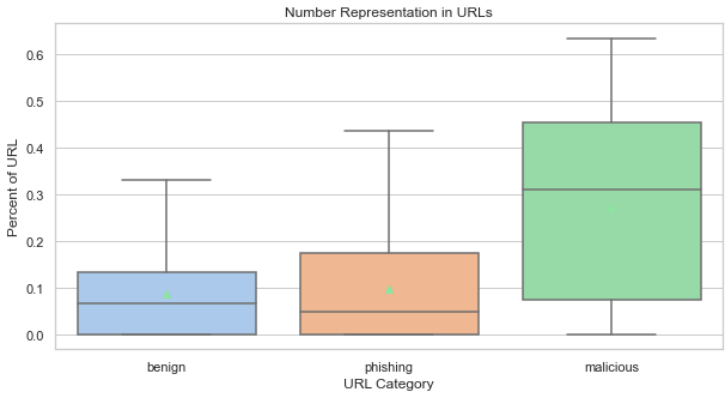
Url String Analysis

Malicious URLS - Approximately 50% of the malicious urls in our dataset have an IP address in lieu of a domain name. This accounts for some of the notable differences in malicious url statistics, in comparison to the other categories. For example, on average, only 50% of the malicious url consists of letters, versus 71% and 73% for benign and phishing urls[7]. Malicious urls are, on average, shorter in length and have shorter token lengths.

Phishing URLs - The mean url length score is 89.1 in comparison to 57.4 for benign urls, and 44.9 for malicious urls. Thus, it's not surprising that phishing urls have the highest average number of tokens (20.8 versus 17.4 and 14.8) and longest average token length of 4.1 (versus 3.4 and 3.0). They also have the highest average entropy score and highest average masque count.

[6] https://www.alexa.com/topsites
[7] See Chart 1: Number Representation in URLs

Chart 1: Number Representation in URLs



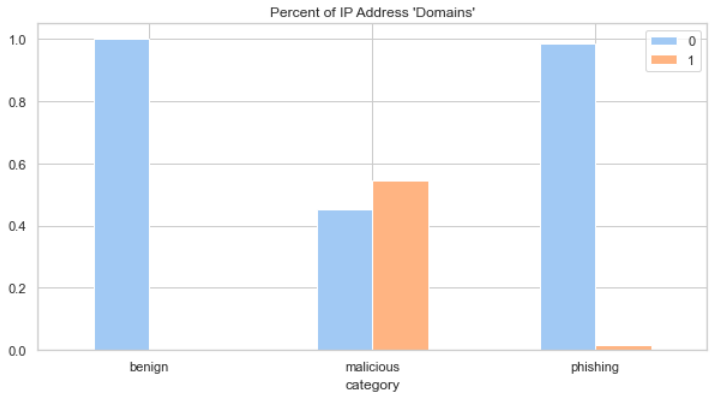
Registered Domain Analysis

Malicious URLs - 54% of the records in this category have an IP address, not a registered domain name[8]. Malicious domains are on average shorter than phishing domains, but longer than benign domains. This category had the highest entropy score of 1.072, in comparison to .576 for benign and .778 for phishing domains[9].

Phishing URLS - Phishing domains are longer on average, with a longer token length.

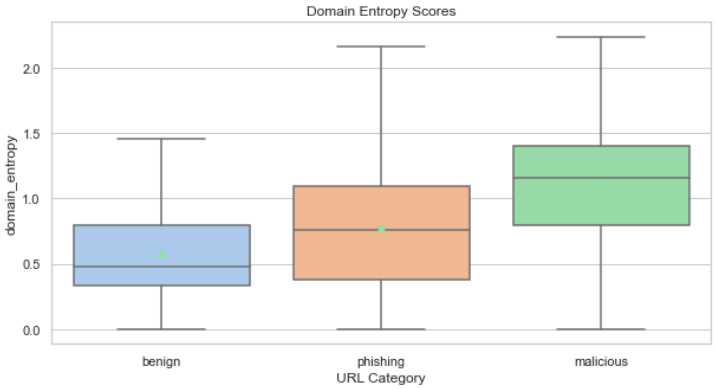
Benign URLs - In comparison with the phishing and malicious domains, benign domains have shorter lengths overall, the lowest entropy score, a greater propensity to be on the Alexa Top 500 website list. They are the least likely url type to contain an IP address.

Chart 2: IP Addresses per Category



[8] Chart 2: IP Addresses per Category
[9] Chart 3: Domain Entropy Scores

Chart 3: Domain Entropy Scores



Netloc String Analysis

Malicious URLS - For the malicious url category, this section is on average longer than benign urls and shorter than phishing urls (18.07 characters versus 13.55 for benign and 22.59 for phishing). The entropy score of 1.32 is similar to phishing's score of 1.34, but greater than benign urls' score of .85.

Phishing URLs - The phishing category trumps the others again with its high length scores. This category is also the most likely to have one or more subdomains.

Benign URLs - The benign netloc section has the lowest mean score for length (13.55 in comparison to phishing's 22.59 and malicious' 18.07). It has the lowest mean entropy and masque scores, lowest mean number of tokens, and shortest average token length. Subdomains are more common in benign urls than malicious urls.

Path String Analysis

Malicious Category. In comparison to benign and phishing urls, urls in this category have shorter path sections on average (mean of 18.29 characters versus 30.26 for phishing and 36.46 for benign), and a smaller amount of path items (1.81 versus 2.47 for phishing and 2.26 for benign). They also have lower entropy scores (mean of .722 versus 1.33 and 1.52). In comparison to benign urls, malicious urls have a greater percentage of letters and special characters.

Phishing Category. In comparison to benign and malicious urls, these urls have a greater percentage of special characters (29.3% versus 16.2% and 23.2%) and masques (.468 versus .257 and .109). The path sections are a little shorter than benign url and have a greater propensity for single character paths.

Benign Category. These urls have the greatest tendency for numbers within path sections. Numbers comprise 14.3% of the path section, versus 8.7% and 7.6% for phishing and malware. This may be the reason for the highest average entropy score of all categories (1.529 versus 0.722 and 1.334). Its average path item length, 17.34, is much higher than the other categories which score 12.59 and 9.29.

Inferential Statistics

It is important to identify strong correlations between pairs of predictor variables and between predictor variables and category, our target variable.

Pairwise Correlation Analysis: We identified a considerable amount of highly correlated features. 73 pairs of features have correlation scores of 80% or higher.

Predictor v. Target Correlation Analysis: Numeric-based domain/netloc features are the most highly correlated with the target feature. The features with the top eight scores may be reflective of the fact that 50+% of the malicious urls in our dataset have numeric IP addresses instead of domain names. Since the domain and netloc features overlap, features may need to be tailored down before machine learning work.

Next Steps

We have identified a number of features highly correlated with the target variable, however the top scorers relate to the presence of numbers in the url domain. We will consider approaches to tailor down the feature set to ensure we have independent features to train a machine learning model. We will utilize the pairwise feature correlation scores in deciding which features may be eliminated from the data set.