Design and Evaluation of a Real-Time URL Spam Filtering Service

Geraldo Franciscani

15 de Maio de 2012

Teacher: Ponnurangam K (PK)
Introduction

Initial Presentation

- Monarch is a real-time system for filtering scam, phishing, and malware URLs as they are submitted to web services.
Introduction

General Considerations

- An alternative for account-based system (ineficient because can incur delays between a fraudulent account's creation and its subsequent detection).
- Development and evaluation of a real-time, scalable system for detecting spam content in web services.
- Exposure of fundamental differences between email and Twitter spam, showing that spam targeting one web service does not generalize to other web services.
Introduction

General Considerations

- Presentation of a novel feature collection and classification architecture that employs an instrumented browser and a new distributed classifier that scales to tens of millions of features.

- Examination of the salience of each feature used for detecting spam and evaluate their performance over time.
Architecture

Overview

1. URL Stream
2. Spam Decisions

Web Services
Social Networks,
Webmail, Blogs, Reviews

Monarch
Architecture

Design Goals

- Real-time results.
- Readily scalable to required throughput (applied to big social networks like Twitter).
- Accurate decisions (emphasize low false positives).
- Tolerant to feature evolution (easily adapt to new features).
- Context-independent classification (applied to different types of web services).
Architecture

System Flow
Architecture

Design Goals

• URL Aggregation.
• Feature Collection (crawler).
• Feature Extraction (raw data into a sparse feature vector understood by the classification engine).
• Classification.
Feature Collection

Overview

- Expand upon the idea of considering just lexical properties of URLs, page content, and hosting properties of domains.
- The system collects its own sources of features based on one of three components: web browser, DNS resolver and IP address analysis.
Web Browser

Features

- Initial URL and Landing URL.
- Redirects (monitoring each redirected page).
- Sources and Frames (analysis of pages with spam content embedded within a non-spam page).
- HTML Content (similar layout and content across spam webpages).
- Page Links (search and analyze each link block - ex. "href"s).
Web Browser

Features

- JavaScript Events (webpages that require user action).
- Pop-up Windows (saves the total number and associate them with the parent URL).
- Plugins (requests that leads to an outgoing HTTP request, causes a page to redirect).
- HTTP Headers (languages and versions of spam hosts).
DNS Resolver

Overview

- The system collects features from the initial, final, and redirect URLs submitted to a DNS resolver.
- Features: hostnames, nameservers, mailservers, and IP addresses associated with each domain.
- Each feature provides a means for potentially identifying common hosting infrastructure across spam.
IP Address Analysis

Overview

- Extract geolocation and routing information.
- Identify portions of the Internet with a higher prevalence of spam.
Feature Extraction

Overview

- Preparation for the classification.
- Transform the unprocessed features gathered during feature collection into a meaningful feature vector.
- Provide processed sparse maps to the classifier for training and decision making.
# Distributed Classifier Design

## Overview

- Design as a parallel online learner for quick training and memory fit.
- Combination of strategies of iterative parameter mixing and subgradient L1-regularization.
Implementation

Overview

- The system operates on Amazon Web Services (AWS).
- Each of the four components of Monarch was implemented as an independent system.
Evaluation

Overview

- Evaluation of the classifier’s accuracy and its run-time performance.
- 90.78% accuracy (0.87% false positives).
- Median feature collection and classification time of 5.54 seconds.
- Overlap between email and tweet spam features, requiring the classifier to learn two distinct sets of rules.
Classifier Performance

Overall Accuracy

<table>
<thead>
<tr>
<th>Training Ratio</th>
<th>Accuracy</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1</td>
<td>94.14%</td>
<td>4.23%</td>
<td>7.50%</td>
</tr>
<tr>
<td>4:1</td>
<td>90.78%</td>
<td>0.87%</td>
<td>17.60%</td>
</tr>
<tr>
<td>10:1</td>
<td>86.61%</td>
<td>0.29%</td>
<td>26.54%</td>
</tr>
</tbody>
</table>

- Trained with all the features.
- Non-spam to spam ratio.
- 4:1 ratio adoption because of its low false positives and reasonable false negatives.
Classifier Performance

Accuracy of Individual Components

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Accuracy</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source URLs</td>
<td>89.74%</td>
<td>1.17%</td>
<td>19.38%</td>
</tr>
<tr>
<td>HTTP Headers</td>
<td>85.37%</td>
<td>1.23%</td>
<td>28.07%</td>
</tr>
<tr>
<td>HTML Content</td>
<td>85.32%</td>
<td>1.36%</td>
<td>28.04%</td>
</tr>
<tr>
<td>Initial URL</td>
<td>84.01%</td>
<td>1.14%</td>
<td>30.88%</td>
</tr>
<tr>
<td>Final URL</td>
<td>83.59%</td>
<td>2.34%</td>
<td>30.53%</td>
</tr>
<tr>
<td>IP (Geo/ASN)</td>
<td>81.52%</td>
<td>2.33%</td>
<td>34.66%</td>
</tr>
<tr>
<td>Page Links</td>
<td>75.72%</td>
<td>15.46%</td>
<td>37.68%</td>
</tr>
<tr>
<td>Redirects</td>
<td>71.93%</td>
<td>0.85%</td>
<td>55.37%</td>
</tr>
<tr>
<td>DNS</td>
<td>72.40%</td>
<td>25.77%</td>
<td>29.44%</td>
</tr>
<tr>
<td>Frame URLs</td>
<td>60.17%</td>
<td>0.33%</td>
<td>79.45%</td>
</tr>
</tbody>
</table>

- Accuracy of classifier when trained on a single type of feature.
Classifier Performance

Accuracy Over Time

- Determine how often the classifier needs to be retrained and how long it takes to become out of date.
- Two types: one classifier trained every four days, and another trained just at the beginning of the test.
- Conclusion: the classifier needs to be retrained on a continual basis to achieve a good classification.
Classifier Performance

Accuracy Over Time - Chart

![Accuracy Over Time Chart](chart.png)
Classifier Performance

Training Across Input Sources

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Testing Set</th>
<th>Accuracy</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tweet spam</td>
<td>Tweet spam</td>
<td>94.01%</td>
<td>1.92%</td>
<td>22.25%</td>
</tr>
<tr>
<td>Tweet spam</td>
<td>Email spam</td>
<td>80.78%</td>
<td>1.92%</td>
<td>88.14%</td>
</tr>
<tr>
<td>Email spam</td>
<td>Tweet spam</td>
<td>79.78%</td>
<td>0.55%</td>
<td>98.89%</td>
</tr>
<tr>
<td>Email spam</td>
<td>Email spam</td>
<td>98.64%</td>
<td>0.58%</td>
<td>4.47%</td>
</tr>
</tbody>
</table>

- Effects of training and testing on matching and mismatching data sets.
- Email and tweet spam are independent.
- Low cross classification accuracy.
Classifier Performance

Context vs. Context Free Training

<table>
<thead>
<tr>
<th>Training Method</th>
<th>Accuracy</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>With Tweet Features</td>
<td>94.15%</td>
<td>1.81%</td>
<td>22.11%</td>
</tr>
<tr>
<td>Without Tweet Features</td>
<td>94.16%</td>
<td>1.95%</td>
<td>21.38%</td>
</tr>
</tbody>
</table>

- Omitting account and tweet properties from classification has no statistically significant effect on accuracy.
Run Time Performance

Overview Analysis

- Latency: median time of 5.54 seconds per URL.
- Throughput: 638,000 URLs per day.
- Cost: $1,600.00 on cloud machinery per month.
- Twitter example: total cost for filtering 15.3 million URLs per day to $22,751 per month.
Comparing Email and Tweet Spam

Overview

- Email: short-lived hosting infrastructure and campaigns.
- Twitter: longer lasting campaigns that push quite different content.
- Email and Twitter spam share only 10% of features in common.
- The classifier must learn two separate sets of rules to identify both spam types.
- Persistence: email spam is marked by much shorter lived features compared to tweet spam and non-spam samples.
Comparing Email and Tweet Spam

Overview - Chart
Spam Infrastructure

Redirecting to spam

- Both Twitter and email spammers use redirects to deliver victims to spam content.
- Twitter: 67% of spam URLs use redirects, with a median path length of 3.
- Email: only 20% of spam URLs use redirects, with a median path length of 2.
Spam Infrastructure

Page Content

• Motivation: observation of popular news sites with non-spam content displaying ads that cause a variety of spam popups, sounds, and video to play.

• Conclusion: it is necessary to analyze all of a webpage’s content to not overlook spam with dynamic page behavior or mash-up content that includes known spam domains.
Discussion

Overview

- Attackers can tune features to fall below the spam classification (feature evasion), modify content after classification (Time Based Evasion) and block the crawler (Crawler Evasion).
- Future work: in depth study of each attack and potential solutions.
Conclusion

Overview

- Presented a real-time system for filtering scam, phishing, and malware URLs as they are submitted to web services.
- Explored the distinctions between email and Twitter spam.
- Demonstrated that a modest deployment of Monarch on cloud infrastructure can achieve a throughput of 638,000 URLs per day with an overall accuracy of 91% with 0.81% false positives.
- Estimated it would cost $22,751 a month to run a deployment of Monarch capable of processing 15 million URLs per day.