# A Campaign-based Characterization of Spamming Strategies

**Pedro H. Calais, Douglas E. V. Pires**
**Dorgival Olavo Guedes, Wagner Meira Jr.**
Computer Science Department
Federal University of Minas Gerais
Belo Horizonte, MG - Brazil

**Cristine Hoepers,**
**Klaus Steding-Jessen**
Computer Emergency Response Team Brazil
Network Information Center Brazil
São Paulo, SP - Brazil

## Abstract

This paper presents a methodology for the characterization of spamming strategies based on the identification of spam campaigns. To deeply understand how spammers abuse network resources and obfuscate their messages, an aggregated analysis of spam messages is not enough. Grouping spam messages into campaigns is important to unveil behaviors that cannot be noticed when looking at the whole set of spams collected. We propose a spam identification technique based on a frequent pattern tree, which naturally captures the invariants on message content and detect campaigns that differ only due to obfuscated fragments. After that, we characterize these campaigns both in terms of content obfuscation and exploitation of network resources. Our methodology includes the use of attribute association analysis: by applying an association rule mining algorithm, we were able to determine co-occurrence of campaign attributes that unveil different spamming strategies. In particular, we found strong relations between the origin of the spam and how it abused the network, and also between operating systems and types of abuse.

## 1 Introduction

Despite current strategies to minimize the impact of spams, it is necessary a continuous effort to understand in detail how spammers generate, distribute and disseminate their messages in the network, to maintain and even improve the effectiveness of anti-spam mechanisms (Pu & Webb, 2006). The goal of this paper is to characterize different spamming strategies employed by spam senders. We define as an strategy any technique employed by spammers to maximize the effectiveness of their attacks, reducing the probability that the message is blocked by spam filters and preventing their activities of being identified and tracked.

Our approach is to characterize spam campaigns in addition to individual messages. We define a campaign as a set of messages that have the same goal (e.g., advertising a specific product) and employ the same obfuscation strategy, which comprises either content obfuscation and network exploitation strategies. In general, spammers obfuscate and change the content of their messages on a systematic and automated way. They try to avoid sending identical messages, which would make the task of detecting his or her messages easier. Thus, in order to characterize the strategies and traffic generated by different spammers, it is necessary to identify groups of messages that are generated following the same procedure and are part of the same spam campaign. In this paper, we propose a novel and scalable methodology for identifying spam campaigns. After the campaigns have been identified and messages are associated with those campaigns, we can then characterize how each campaign have exploited the network resources and how their contents have been obfuscated.

We consider the identification of spam campaigns a crucial step for identifying spamming strategies and improving our understanding of how spammers abuse network resources for a number of reasons. First, the identification of campaigns creates new dimensions that can be analyzed and correlated. Aggregate analysis on spam data is limited in determining spamming strategies. By grouping spam messages in their associated campaigns, we can characterize how the spammer disseminated his or her messages. Second, the volume of spam messages is huge, and processing such amount of data is costly and sometimes unfeasible. Grouping messages into campaigns provides a summarization criteria, drastically reducing the amount of data to be treated, while maintaining their key characteris-

tics. Finally, the identification of spam campaigns neutralize the effect of the variable volume of messages associated to each spam campaign, which might hide frequent behaviors happening only on smaller campaigns.

After identifying the messages that were generated from the same spam campaign, we propose a methodology for characterizing spam dissemination strategies. The methodology is based on the detection of invariants and co-occurrence of mechanisms for sending messages adopted by a single campaign. These invariants and patterns represent spamming behaviors and may be used for definition of criteria for detection, identification and minimization of the impact of spam.

We applied our characterization methodology to 97 million spam messages captured during 12 months by low-interaction honeypots (Provos & Holz, 2007), which were configured to emulate computers with open relays and open proxies. We were able to find strong relations between the origin of the spam and how it abused the network and also between operating systems and these abuse types.

## 2 Related Work

Many recent works have studied spammers' abuse strategies, both considering network behavior and content obfuscation.

In (Ramachandran & Feamster, 2006) the authors analyze how spammers exploit the Internet infrastructure to send their messages, including the most popular IP ranges exploited for sending spam and the more common abuse types, such as bots and BGP hijacking. In particular, the authors show that spam messages tend to be sent from very restricted IP ranges. Some statistics about the origin of the messages show the most common operating systems originating spams and the autonomous systems (AS) that account for the highest volume of spams. Our paper also characterizes spamming network strategies, but instead of looking at the group of messages as a whole, we group messages into campaigns and then analyze how the groups of IPs that disseminated each campaign have abused network resources. This approach provides insights on how spammers act, which would not be possible on an aggregated analysis. Moreover, we also found relations among operating systems, spam origin and abuse types, which extends the analysis presented on (Ramachandran & Feamster, 2006).

A recent work on characterization of strategies of spam dissemination is presented in (Li & Hsieh, 2006). The authors grouped spams according to the messages' URLs and analyzed the graph representing the relationships between IPs and URLs. Some properties of that graph were analyzed, for example, the identification of large groups of IPs that send spam messages with the same URL. The notion of campaign was implicitly used, by grouping messages by their URLs, but, as URL obfuscation is a common practice, the groups of IPs referencing the same URLs could be even bigger. Our work considers not only URLs, but also other features while identifying campaigns.

SpamScatter (Anderson et al., 2007) is a technique that determines spam campaigns by performing *image shingling*, which looks for similarities between images from different spam web pages. The methodology adopted by the authors is similar to ours: campaigns are first identified and then characterized. However, while their work analyzes the scam hosting infrastructure, our focus is on the characterization of the network infrastructure abuse.

In fact, the idea of identifying spam campaigns is not new. In the literature, most research on grouping near-duplicate spam messages aims to detect campaigns as a strategy for blocking them, based on the fact that an inherent characteristic of unsolicited e-mails is that they are sent in high volumes during short periods of time. Different strategies for grouping messages into campaigns can be mentioned, such as techniques that consider URL information (Yeh & Lin, 2006), signature-based approaches such as I-Match (Kolcz & Chowdhury, 2007) and techniques that compute similarities between spam images (Wang et al., 2007). Our goal is different from those works in the sense that we intend to identify spam campaigns to characterize them in terms of content and network obfuscation. Although our findings may support the development and improvement of anti-spam techniques, this is not our main objective.

Regarding content characterization, (Pu & Webb, 2006) presented some analysis of temporal evolution of spammer strategies regarding the techniques they use to construct their messages. These techniques were extracted from the rules identified by the anti-spam filter *SpamAssassin*. The authors showed that some obfuscation techniques are abandoned over time, possibly due to changes in the environment, such as a bug fix on an e-mail client program. On the other hand, some strategies are able to persist for long periods of time. Our work is complementary to theirs and also provides an interesting framework for trend detection, which would be a future work direction.

Another characteristic of our work is the use of data mining techniques to unveil spamming strategies. Although data mining has been extensively applied on a wide range of contexts such as e-commerce, bioinformatics and industrial applications (Tan et al.,

2005), we are not aware of any work that applied data mining techniques for spam characterization purposes.

## 3 Methodology

In this section, we present our methodology for characterizing spammers' strategies for dissemination of spam messages. The methodology is divided into three distinct phases: data collection, campaign identification, and characterization. These three phases will be detailed in the next subsections.

### 3.1 Data Collection

The data collection architecture comprises a set of sensors based on low-interaction honeypots (Provos & Holz, 2007) to study the spam problem, in particular the abuse of open proxies and open relays. A proxy is a server that acts as an intermediary, making connections on behalf of other clients. An open proxy allows connections to be made from any origin to any destination IP address or port, and is traditionally abused for sending spam. Examples of common proxy protocols are HTTP and SOCKS. Misconfigured SMTP servers, usually called open relays, allow the delivery of messages from any source to any recipient and are also abused by spammers.

We deployed 10 honeypots in 5 Brazilian broadband networks (both cable and ADSL), that captured approximately 525 million spams over 15 months. These spams came from 216,888 different IP addresses, allocated to 165 different countries (Country Codes (CC), as defined in ISO 3166) and would have been delivered to 4.8 billion recipients (Steding-Jessen et al., 2008).

These honeypots were collecting data not at the final spam destination, like in spamtrap accounts or in mail servers. Instead, we measured the abuse of proxies and relays by spammers, and capture the spam at this stage, before reaching its final destination. Also, an advantage of using honeypots was that all messages collected were spams, with no false positives.

We used `Honeyd` (Provos & Holz, 2007) and its SMTP and HTTP server emulation subsystems to capture spam. A SOCKS proxy emulator was developed as part of this work to complement the existing emulators (Steding-Jessen et al., 2008). Although the honeypots were listening on several TCP ports, only the following were abused: 25, 80, 81 1080, 3127, 3128, 3382, 4480, 6588, 8000 and 8080.

All connections to the `Honeyd` modules were logged, including timestamp, client IP, destination IP and TCP port, as well as protocol requested. `Honeyd` also logs the Operating System of the source IP for each TCP connection, using passive fingerprinting techniques (Provos & Holz, 2007). All logs and data captured were then collected by a central server.

Any spammer trying to abuse one of these honeypots to send spam would tend to believe that the emails were delivered successfully. The message, however, was stored locally and never delivered to its recipients. The only exceptions were emails sent by spammers to test if the proxy/relay was delivering messages. Each test message was specially crafted, and contained information about the proxy/relay being tested. The message included IP address, port and protocol, and was intended to a recipient address under the spammer control. The honeypots were configured to deliver these messages only.

### 3.2 Campaign Identification

Campaign identification determines groups of messages that have the same goal and employ the same dissemination strategy. In practice, we want to minimize the effects of the obfuscation techniques employed by spammers, who systematically change the content of the messages they send, either the message body or subject (Sophos.com, 2004). The ultimate goal of obfuscation is to make each message unique, and, for that purpose, spammers use *bulk mailers* developed for sending spams. These tools offer many features for customization and obfuscation, such as the insertion of random pieces of text on the message body or in the URL, making the generation of spam signatures more difficult.

It is important to emphasize that the challenge associated with campaign identification comes not only from the variety of obfuscation strategies employed, but also their constant evolution, since mechanisms stop being used and novel mechanisms arise (Pu & Webb, 2006).

The basic premise of our strategy for campaign identification is that spammers, in general, keep some parts of the message static, while other parts are changed systematically and in an automated fashion. This premise is supported by the fact that the spam messages are generated by tools, which employ the same obfuscation mechanisms for a given campaign. Further, since the duration of the campaigns is relatively short, we assume that the obfuscation mechanisms employed for a given campaign do not change significantly across time. For example, each message from a given campaign may have slightly different terms in the *subject* field, although some keywords are always present, which is based on the intuitive relevance of keeping the message's subject readable, since too much obfuscation in this field may reduce the probability of the

message being read. Other examples are, in the message body, the insertion of greetings that may alternate between "Hello" and "Hi"; the inclusion of random fragments in URLs, which have no meaning and are inserted to make the URL unique, preventing its identification and block.

The problem of identifying spam campaigns may be seen as the determination of a hierarchical clustering of the messages, where messages that are similar w.r.t. a given criterion are clustered together. In order to implement such clustering, it is necessary to both determine the criteria and their hierarchical organization. Notice that our strategy also accounts for various strategies and their evolution.

### 3.3 Characterization of Spammer Behavior

In this phase we analyze both message and campaign information, searching for patterns and invariants that describe spammers' strategies.

Our strategy is based on the premise that we are evaluating machines that are abused for the transmission of spams, and we identify four groups of criteria for performing such characterization: source, destination, type of abuse, and content obfuscation strategy.

Source and destination consider not only the individual IPs/addresses from/to the spams, but also abstractions such as ASes, countries and ISPs. The type of abuse includes basically proxies (HTTP and SOCKS) and relays. The obfuscation strategy already includes several criteria, and we envisage that others may be incorporated as those strategies evolve.

In this context, the characterization may be seen as a two-step procedure. First, we generate profiles for each campaign, determining the characteristics that are shared by the spams in that campaign. We then group the campaigns, obtaining approximations of spammers' behavior. Invariants identified among these groups represent different spamming strategies.

There are several techniques that may be applied in this context, and any correlation analysis techniques are applicable, as we discuss in the next sections.

## 4 Frequent-pattern approach for campaign identification

In this section we describe the implementation of our strategy for identifying spam campaigns. As discussed previously, our strategy identifies the invariant parts of the spam messages and organize them hierarchically. It is divided into two major steps, which are described next.

In the first step, we extract relevant features from each spam message, such as its language, layout, message type (HTML, text, image), URL and subject. The language of each message is extracted using a technique based on the computation of n-grams (Cavnar & Trenkle, 1994). Message layout is a codification that maps the formatting properties of the message to a sequence of characters, based on the proposal of Claudiu Musat (Musat, 2006). For example, the layout of a plain text message containing two blank lines, followed by one URL and two lines of text would be mapped to the layout BBUTT. Message layout is an important invariant to be considered for the purpose of grouping messages from the same campaign, since the general appearance of the messages remain unchanged, although spammers usually insert random pieces of text on their messages. Besides language, type, layout, and subject, URL information is also crucial for clustering messages into campaigns (Yeh & Lin, 2006). We split each URL into tokens (splitting by "/", "." and "?") and they are considered as independent features.

Using the messages' features extracted in the first step, we build a frequent pattern tree, also called FP-Tree (Tan et al., 2005). In this tree, each node after the root represents a feature extracted from the spam messages which is shared by the sub-trees beneath. Each path in the tree represents sets of features that co-occur in messages, in non-increasing order of frequency of occurrences. Thus, two messages that have several frequent features in common (such as the language, type, and layout) and are different just on infrequent features will share a common path on the tree. The root is the only empty node, separating sub-trees which have nothing in common. From our observations, the most common infrequent feature is an URL fragment randomly generated. These random fragments cause the number of children to increase significantly, and are exactly the points in which the campaigns are delimited, that is, all the messages that are in the sub-trees beyond a significant increase in the number of children are grouped into the same campaign.

We should emphasize that our approach is scalable because messages are not compared pairwise, what would lead to a quadratic complexity. The cost of the algorithm is the cost of inserting messages' features in the FP-Tree, which is linear. Among the near-duplicate detection techniques, the FP-Tree would fit in the category of signature-based approaches, as the sequence of features in the tree defines the campaign unique identifier. Our approach, however, is not sensible to random text, which is a common drawback of such techniques (Kolcz & Chowdhury, 2007). Another advantage of the FP-Tree is that it not only detects the

spams that are part of the same campaign, but also describes how the messages were constructed and obfuscated, as it will be detailed in Section 5.1.

## 5 Spam Dissemination Strategies

Table 1 shows details about the data used in our analysis. From the total messages collected, as described in Section 3.1, we decided to consider for analysis a period of 12 months. We also restricted the analysis to messages collected in two honeypots that were deployed on the most stable broadband networks. This reduced the number of messages analysed to approximately 97.5 million. Among those, we identified 6.9 million unique cryptographic hash signatures and 2.1 million unique URLs.

Table 1: Overview of the data analyzed in this paper

| Characteristic | Values |
|---|---|
| Trace Period | 2006-07-08 to 2007-06-23 |
| Spam Messages | 97,511,104 |
| Unique hashes | 6,910,340 |
| Spams with URLs | 88,735,105 (91 %) |
| Unique URLs | 2,110,748 |
| Spam campaigns | 16,115 |

In this Section we present our characterization of spam strategies as observed in our honeypots. We divide this analysis in two major categories: first we identify campaigns and the content obfuscation techniques used in them; after that, using the added insight provided by clustering attacks in campaigns, we analyse their behavior in terms of network activity.

### 5.1 Spam campaigns

After applying our campaign identification technique, 16,115 spam campaigns were identified. Figure 1 shows the cumulative distribution function for the number of messages that are part of each campaign. We can see that while most campaigns are small, there is a significant number of campaigns which comprise more than 100,000 spam messages each.

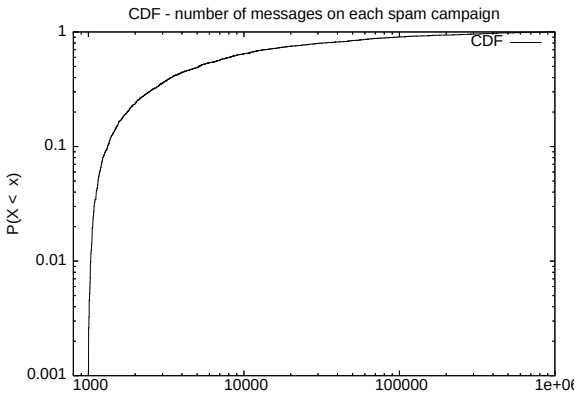


Figure 1: Number of messages in each campaign

Figure 2 shows a small portion of the resulting FP-Tree. When colors are available, each node's color represents a different feature that varied among the messages at that level. The diameter of the node is proportional to the log of the frequency of the characteristic in the campaign. Invariants in the campaigns are detected because they are more frequent than obfuscated features.

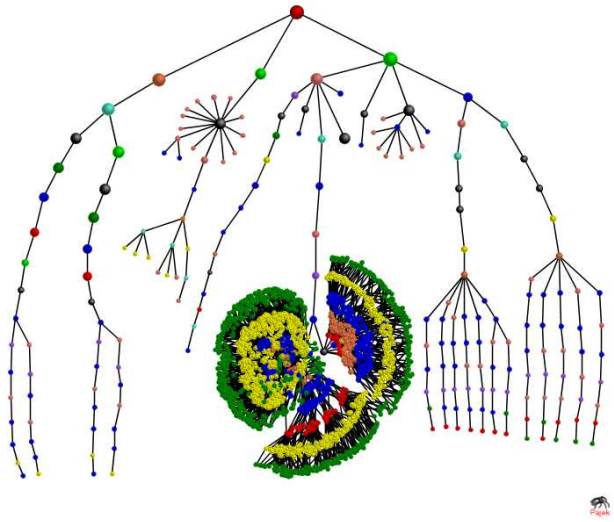

Figure 2: Frequent Pattern Tree showing distinct spam campaigns

For example, if a spammer sends URLs with the format `www.domain.com?parameter=random`, all random fragments would be inserted in the fifth position of the URL, making these messages very distinct from others that obfuscate in one of the remaining positions of the URL, which would characterize a different campaign.

An advantage of our approach is that it is not necessary to specify beforehand where obfuscation should be expected and which exact obfuscation pattern to look for. Spammers that own a web domain can insert obfuscation on virtually any position of the URL and even that would be automatically detected by our technique. In summary, one fundamental aspect of our campaign identification technique is that obfuscation patterns are not defined *a priori*; they are naturally detected. Further, given that the evolution of spamming techniques is an incontestable phenomenon, the FP-Tree is not tied to any currently known pattern. Our technique is extensible in the sense that, as soon as relevant characteristics of new campaigns are detected, they can be readily inserted in the FP-Tree. We also believe that our technique would be useful in other contexts, such as blog spam and spam in online social networks, in which the insertion of random text to avoid fingerprinting is also a common practice.

As an example of this analysis, in Figure 2, the three dense groupings seen at the bottom of the Figure, in

the middle, are three campaigns that shared some attributes (since they had a common path to the root until depth 7). They were then differentiated by the way they instantiated some other features, since the color of the nodes at following levels vary after that. All three campaigns varied some of their attributes randomly over a range of possibilities, yielding the larger number of children at the lower level.

Table 2: Number of Instances per Features

| Feature | Number of instances |
|---|---|
| Message Type | 16 |
| Language | 21 |
| Message Layout | 65,063 |
| URL Fragments | 1,967,160 |

In order to illustrate how the various campaigns exploit the obfuscation of the various features, we determined the number of different instances for each of the four attributes we used in our experiments (language, message type, message layout and URLs) and present a summary in Table 2. Message Type and Language are detected as important invariants, as only 21 types (including combinations, e.g.: spams which contain an HTML part with an image attachment). Message layout is also an important feature for grouping spams, given than only 65,063 distinct message formats where found for the set of 97.5 million messages.

Observing the FP-Tree, we find that language is often the first attribute selected as a classifier, as it would be expected (messages in a campaign are written in the same language). After that, message type and format come next, and URL fragments are usually the last discriminative feature added to the tree. That is expected, since messages in each campaign are likely to have similar URLs. The way those URLs are grouped and how often they are obfuscated in each campaign, however, varies widely among campaigns.

We have identified three types of campaigns in terms of obfuscation of URLs: static campaigns, campaigns with sub-campaigns, and random-obfuscated campaigns. In the FP-Tree sample shown in Figure 2, we can clearly identify those three types. Fixed campaigns are the ones in which the spammer inserts the same URL in all the messages of the campaign. Usually, these URLs correspond to small links with meaningful and readable names, such as *buydvds.com*. Those are the short branches which end at depths 3 or 4, usually. A different strategy frequently observed is the selling of different products in the same campaign, what generates a set of URLs in which each URL correspond to a different product from the same web site. For example, *dvd1.htm, dvd2.htm and dvd3.htm* are different products associated with the same campaign. As long as the spammer keeps other parts of the URL and its layout fixed, these distinct messages will be grouped into the same campaign because the portion of the URL that specifies the product is infrequent compared to the other messages' characteristics. Those are the branches at either side of the tree shown. Finally, the third class of campaign is the one in which spammers constantly obfuscate their URLs inserting random fragments which are different for each message. That refers to the three groupings at the bottom center discussed previously.

### 5.2 Network patterns

To understand how spammers use the network, we combine the analysis of grand totals from the collected data with information derived from the identification of campaigns and other data mining techniques. In the following discussion we highlight the major findings so far.

#### Spammers abuse proxies and relays differently

Table 3 shows a comparison for the three different types of abuse logged by the honeypots (HTTP and SOCKS proxies, and open mail relays). For each abuse we show the number of messages delivered and the number of unique sources of abuse with three different granularities: IP addresses, ASes and Country Codes (CC) of origin. (Percentages do not add up to 100 for IPs, CCs and ASes because of machines using more than one form of abuse.)

It is clear that messages abusing the honeypots as open relays are relatively rare, corresponding to only 2.6 % percent of the total. However, ratios are dramatically different when we look at the distribution of the origins of the abuse. Messages abusing HTTP and SOCKS come from relatively fewer ASes (and countries) that those abusing the open relay, which come from all over the world (142 countries). Despite being responsible for a small volume of messages, open relay abuses account for 98.6 % of the unique ASes abusing the honeypots during the period considered. Moreover, machines abusing the open relay send much fewer messages over time: while an AS abusing the open relay sent less than 2,000 messages over the period, ASes exploiting HTTP and SOCKS sent more than 1,000,000 and 380,000 messages on average, respectively.

This preference for abusing proxies may be explained by the need to better cover the origin of the spam. If spammers contacted mail relays directly, it would be simpler to track messages back to their origin, since even open relays would record the IP address of the previous connection, assuming it came from a valid SMTP server.

Table 3: Observed abuses

| Metric | Abuse: HTTP | Abuse: SOCKS | Abuse: Open Relay |
|---|---|---|---|
| Messages | 67,051,062 (68.8 %) | 27,922,938 (28.6 %) | 2,537,104 (2.6 %) |
| Unique IPs | 41231 (49.3 %) | 16,183 (19.36 %) | 38,252 (45.8 %) |
| Unique ASes | 59 (2.7 %) | 72 (3,3 %) | 2170 (98.6 %) |
| Unique CCs | 14 (9.9 %) | 14 (9.9 %) | 142 (100 %) |
| Messages / IP | 1626.2 | 1725.4 | 66.3 |
| Messages / AS | 1,136,459 | 387,818 | 1,169 |

**Proxies and relays may be abused in a single campaign**

There might seem, at a first glance, that such discrepancies were due to different spamming techniques (and, therefore, different campaigns) using proxies or open relays, but that turned out not to be the case. Indeed, 90 % of the campaigns identified abused only HTTP/SOCKS in the honeypots; however, the other 10 % abused both open relays and proxies. There were no campaigns that abused only open relays.

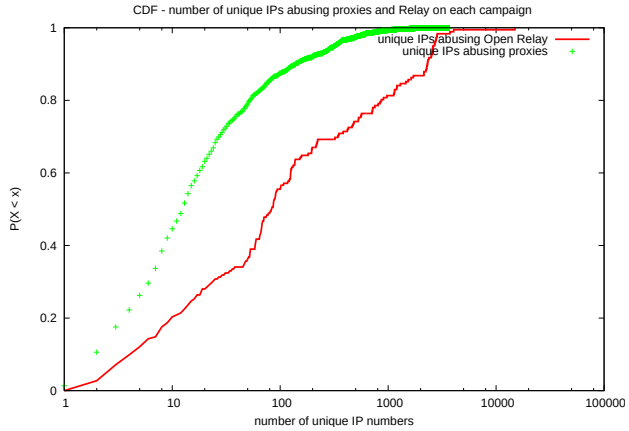


Figure 3: Unique IPs abusing proxies and relay on each campaign

Even on campaigns that included abuse to both proxies and relays, the pattern differed between them. Figure 3 shows, for those campaigns, the cumulative distribution function of the number of IP addresses seen in any campaign abusing proxies or the relay. When abusing proxies, campaigns come from a more concentrated set of addresses. Approximately 50 % of the campaigns come from only 10 sources when abusing HTTP/SOCKS proxies, while 80 % come from more than 10 IPs (and 40 % come from more than 100 addresses) when abuses are directed to the open relay.

**Spammers chain proxies to SMTP servers**

If we look at the next hop intended for the proxy connections, there were 215,719 different IPs targeted, in 189 unique CCs. An interesting fact is that 94 % of all connections were to port 25 (SMTP) of those machines, which could have been legitimate mail servers or open relays. That indicates that the most common behavior of spammers is to use one hop over a SOCKS/HTTP proxy and then use SMTP. If that was not the case, we should see a larger number of connections through the honeypot proxies aimed at other port numbers in the next hop machines.

By inspecting the addresses most popular among those selected by spammers as the next hop for the proxy connections we see most connections are for well established mail servers. For example, machines `mta-v2.mail.vip.tpe.yahoo.com` and `mta-v1.mail.vip.tpe.yahoo.com` are among the most targeted from proxies in both groups of campaigns (the ones that abuse only proxies and those that abuse proxies and relays).

**Spammers chain proxies to open mail relays**

On the other hand, campaigns that were seen abusing the honeypots' mail relays targeted not only well known mail servers, but also mail relays the spammer could find as the next hop for the proxy connections.

Proof of that can be found in the honeypot themselves. Among the campaigns that abused proxies and relays, we found in the honeypots records of approximately 2800 requests for a honeypot to connect the port 25 of another honeypot used during the collection. That is in fact a lower bound, since we considered only the addresses of honeypots with fixed IP addresses; there might be other requests targeted at honeypots in networks using dynamic addresses.

In our continuing work we intend to classify addresses more precisely between relays and servers, and also understand how spammers find those open relays: whether from pre-assembled lists, or through active port scanning.

**Correlations among message features**

In search of evidences that would explain such differences regarding proxy and relay abuses, we correlated four different characteristics present on each campaign: the type of abuse, the message's CC of origin, the intended CC of destination of the spam message and the language used. The Language was derived using the n-grams technique mentioned in Section 4; the CC of origin was obtained from IP allocation tables from the 5 Regional Internet Registries; and the destination country was derived from the domain extracted

Table 4: Association rules - Origin, Destination, Language and Abuse

| rule | antecedent (if) | consequent (then) | support | confidence | lift |
|---|---|---|---|---|---|
| 1 | in Chinese, on HTTP | From TW | 23.8 % | 86.0 % | 1.1 |
| 2 | From BR | in Chinese, on Open Relay, to TW | 0.02 % | 46.7 % | 3.8 |
| 3 | From AR | in Chinese, on Open Relay, to TW | 0.01 % | 76.7 % | 4.5 |
| 4 | From GB | in Chinese, on Open Relay, to TW | 0.02 % | 81.9 % | 3.1 |
| 5 | From PT | in Chinese, on Open Relay, to TW | 0.01 % | 43 % | 2.3 |
| 6 | From AR, on SOCKS | in Spanish | 0.01 % | 95 % | 4.3 |
| 7 | From CN, on HTTP | in Chinese, to TW | 7.5 % | 84 % | 1.3 |
| 8 | From CN, on Open Relay | in Chinese, to TW | 6.3 % | 78 % | 1.1 |
| 9 | From US | in Chinese, on Open Relay, to TW | 0.8 % | 59 % | 1.0 |
| 10 | From US, on HTTP/SOCKS | in English | 3.1 % | 56 % | 1.4 |
| 11 | From US, on HTTP/SOCKS | in Chinese | 1.1 % | 31 % | 0.9 |

from the victims' e-mail addresses. For this analysis, we did not include address in the `.com` domain (e.g., `yahoo.com` or `gmail.com`), since the user behind such accounts could be virtually on any part of the globe.

Using that information, we applied an association rule mining algorithm (Tan et al., 2005) to each campaign's data. Some of the most interesting association rules found on our analysis are shown in Table 4.

In that table, rule 1 shows the most frequent abuse observed on our dataset: 23.8 % of the abuses are related to messages written in Chinese abusing HTTP. Moreover, 86 % of the messages with those characteristics are sent from an IP address in the TW Country Code. Rules 2 to 5 indicate that spams written in Chinese are also observed being sent from BR, AR, GB and PT abusing open relays, with high confidence. In the case of AR, on the other hand, rule 6 shows that 95 % of the spams sent from that country abusing SOCKS were written in Spanish. Rules 7 and 8 indicate that CN sends spams in Chinese both through proxies and open relays to TW. Rules 9 to 11 show the most common abuses related to messages being sent from US. While US IP addresses are detected sending spams in Chinese through open relays (rule 9) and in English through Proxies (rule 10), US is also seen sending spams in Chinese through Proxies and SOCKS (rule 11), which is different from what is observed for the other country codes.

From these results, we conclude that there is a strong relation between abuse type, origin and destination of spams. While TW primarily sends spams through HTTP and SOCKS aiming TW recipients (writing in Chinese), most of the other country codes (BR, AR, PT and other 139 CCs) abuse open relays sending messages in Chinese. The only exception is US: it also sends Chinese spams through Proxies and SOCKS, what might indicate a different spamming strategy.

**Observed behavior**

The association rules, analyzed in conjunction with Table 3, suggest that HTTP and SOCKS are exploited directly by spammers, i.e., at the origin of the attack. That is supported by the concentrated IP addresses that originate such abuses, the coincidence between the language used in the message and the language associated to the CC. On the other hand, open relay abuses come from machines all over the world. In Table 4, we have presented only some few rules showing countries sending spams in Chinese through open relays, but, actually, this is seen for all the other countries in our dataset. Those may be HTTP/SOCKS proxies also abused, or may have a more organized structure, kept indirectly under control of spammers, known as *spam bots*, like machines in a botnet (Cooke et al., 2005). Prior work have reported that botnets usually send very low volumes of spams over longs periods (Ramachandran & Feamster, 2006), which is a strong evidence that, in fact, these open relay abuses spread worldwide are compromised infected machines.

The results about abuses from AR illustrate that clearly. 76.7 % of all spam coming from AR carry messages in Chinese, addressed to machines in TW and abuse open relays. On the other hand, 95 % of the spam coming from AR hosts abusing SOCKS are in Spanish.

**The use of probe messages**

Section 3.1 mentioned that we observed some messages that could be identified as probe messages sent by spammers to assert the correct behavior of the machines abused by them. They often have a nearly empty body with some information about the machine probed and should be recognized and dealt with properly, otherwise spammers would leave and would not try to abuse the honeypot infrastructure (Andreolini et al., 2005). We identified two major types of probe messages based on their format and programmed our honeypots to respond only to those messages. Table 5 summarizes the results in terms of the country codes and IP addresses of origin of those messages.

It seems the two kinds of messages are associated with two different spamming tools. Messages of type 1 came mostly from TW, from a relatively large number of IP addresses. Type 2 probes came from both US and TW machines, with somewhat different behaviors: from US

Table 5: Probe Messages

| CC | Messages | Unique IPs |
|---|---|---|
| *Probe Message Type 1* | | |
| TW | 251,228 (99.8 %) | 979 (97.8 %) |
| US | 23 (0.15 %) | 16 (1.6 %) |
| KR | 7 (0.05 %) | 6 (0.06 %) |
| *Probe Message Type 2* | | |
| US | 1836 (51.0 %) | 19 (2.5 %) |
| TW | 1473 (40.1 %) | 672 (89.4 %) |
| CN | 47 (1.3 %) | 31 (4.2 %) |
| CA | 34 (1.2 %) | 13 (1.8 %) |
| KR | 34 (1.2 %) | 5 (0.9 %) |
| other | 12 (0.07 %) | 6 (0.9 %) |
| unknown | 163 (4.5 %) | 2 (0.3 %) |

only 19 hosts (2.5 % of the type 2 probing machines) were responsible for more than 50 % of the probes, while a similar number of probes originated from 672 hosts (89.4 % of the machines). Probe messages seem to be related to the real origin of the message, as they come from the same CCs identified as those responsible for the abuses on proxies. That suggests a dedicated set of high-throughput machines in the first case, and a more widely distributed infra-structure in the second case.

**Behavior varies for different operating systems**

Finally, Table 6 unveils some strong patterns correlating the operating systems of the machines which abused the honeypots during the period of this analysis and the type of abuse associated to each one. Association rules 1 to 3 show that machines configured with Linux, FreeBSD and Solaris abuse the honeypots mainly as open relays, in the vast majority of the abuses observed. The high value for lift[1] in all cases (higher than 8) indicates that chances of observing abuses on open relays is much higher when the messages come from Linux, FreeBSD and Solaris computers, although these operating systems account for less than 3 % of the total flows observed. Correlating these results with our prior observations, we believe that these rules are due to the fact that *bulk mailers*, in general, are developed for Windows platforms, and not Linux or Solaris. On the other hand, it is relatively common to find in these operating systems some sort of HTTP server, possibly poorly configured.

On the other hand, rules 4 to 6 show that Windows is usually used to exploit SOCKS (with 31 % of confidence) and proxies (with 62 % of confidence). In our dataset, over half of the operating systems of the machines which abused our honeypots could not be identified (rules 7 and 8). As the proportions of HTTP/SOCKS abuses are similar to those observed for Windows, and very different from machines configured with Linux-based systems, we believe that those connections may be in fact associated with Windows Vista, which was a new OS on 2006 and might not have a proper signature yet.

[1]lift: ratio between the calculated probability and the expected probability, if the events of the association rule were independent.

## 6 Conclusions and Future Work

In this paper we have presented a methodology for characterizing spamming strategies based on the grouping of spams into spam campaigns and then detecting invariant an co-occurrence patterns among them.

Our technique builds a frequent pattern tree (FP-Tree) using relevant features extracted from spam messages (*e.g.*, layout, language, URL fragments). Based on that, messages that share a common frequent path in the tree and differ only on infrequent features are grouped into campaigns. We have tested our technique on a dataset of approximately 97.5 million spam messages collected on low-interaction honeypots deployed on Brazilian networks acting as HTTP and SOCKS proxies and open mail relays.

After identifying the distinct spam campaigns, we showed that data mining techniques (such as clustering and association rule mining) can be useful to unveil relevant spamming behavior patterns. We found that HTTP and SOCKS abuses originate from few nodes and show strongly correlations between language and CC of origin, suggesting that they are close to the origin of the campaign. On the other hand, Open Relay abuses are more disperse, originating from many different sources and exhibiting no correlation between language and CC of origin, probably being triggered by *spam bots*. We also determined some relations between operating systems and abuse types, indicating that Linux and Solaris systems are rarely used as the origin of abuses to HTTP and SOCKS proxies.

As future work, we will compare our campaign identification technique with other approaches available on the literature. We will also study more deeply how spam campaigns abuse the network infrastructure, specially in the case when connections to proxies are relayed to an intermediary open relay machine before being delivered to a official mail server. A temporal analysis of campaigns evolution is also being considered.

We intend to deploy honeypots in other countries. With that we can have a more global view of how spammers abuse network resources around the Internet.

Table 6: Association rules – Operating Systems and Abuse Types

| rule | antecedent (if) | consequent (then) | support | confidence | lift |
|---|---|---|---|---|---|
| 1 | OS: Linux | Abuse: Open Relay | 1.3 % | 97.0 % | 8.0 |
| 2 | OS: FreeBSD | Abuse: Open Relay | 0.7 % | 100 % | 8.2 |
| 3 | OS: Solaris | Abuse: Open Relay | 0.6 % | 100 % | 8.2 |
| 4 | OS: Windows | Abuse: Open Relay | 4.1 % | 7 % | 0.6 |
| 5 | OS: Windows | Abuse: HTTP | 7.1 % | 62 % | 0.9 |
| 6 | OS: Windows | Abuse: SOCKS | 15.3 % | 31 % | 1.2 |
| 7 | OS: Unknown | Abuse: HTTP | 49.8 % | 72 % | 1.0 |
| 8 | OS: Unknown | Abuse: SOCKS | 16.1 % | 26 % | 1.0 |

**Acknowledgements**

This work was partially supported by NIC.br, CNPq, CAPES, Finep and Fapemig.

## References

Anderson, D. S., Fleizach, C., Savage, S., & Voelker, G. M. (2007). Spamscatter: Characterizing internet scam hosting infrastructure. *USENIX Security*.

Andreolini, M., Bulgarelli, A., Colajanni, M., & Mazzoni, F. (2005). Honeyspam: honeypots fighting spam at the source. *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop* (pp. 11–11). Berkeley, CA, USA: USENIX Association.

Cavnar, W., & Trenkle, J. (1994). N-gram-based text categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (pp. 161–175). Las Vegas, US.

Cooke, E., Jahanian, F., & McPherson, D. (2005). The zombie roundup: understanding, detecting, and disrupting botnets. *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop* (pp. 6–6). Berkeley, CA, USA: USENIX Association.

Kolcz, A., & Chowdhury, A. (2007). Hardening fingerprinting by context. *Proceedings of the 4th Conference on Email and Anti-Spam (CEAS). Mountain View, CA.*. Mountain View, CA.

Li, F., & Hsieh, M.-H. (2006). An empirical study of clustering behavior of spammers and group-based anti-spam strategies. *Proceedings of the Third Conference on Email and Anti-Spam (CEAS). Mountain View, CA.*

Musat, C. (2006). Layout based spam filtering. *Transactions on Engineering, Computing and Technology*.

Provos, N., & Holz, T. (2007). *Virtual honeypots: From botnet tracking to intrusion detection*. Addison-Wesley Professional. 1st edition, ISBN-13: 978-0321336323.

Pu, C., & Webb, S. (2006). Observed trends in spam construction techniques: A case study of spam evolution. *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS). Mountain View, CA.*

Ramachandran, A., & Feamster, N. (2006). Understanding the network-level behavior of spammers. *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (pp. 291–302). New York, NY, USA: ACM.

Sophos.com (2004). The Spam Economy: The Convergent Spam and Virus Threats. August 2004. `http://www.sophos.com/whitepapers/Sophos_spam-economy_wpus.pdf`.

Steding-Jessen, K., Vijaykumar, N. L., & Montes, A. (2008). Using low-interaction honeypots to study the abuse of open proxies to send spam. *To appear in: INFOCOMP Journal of Computer Science*.

Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining, (first edition)*. Addison-Wesley Longman Publishing Co.

Wang, Z., Josephson, W., Lv, Q., Charikar, M., & Li, K. (2007). Filtering image spam with near-duplicate detection. *Proceedings of the Fourth Conference on Email and Anti-Spam (CEAS). Mountain View, CA.*. Mountain View, CA.

Yeh, C.-C., & Lin, C.-H. (2006). Near-duplicate mail detection based on url information for spam filtering. *Information Networking. Advances in Data Communications and Wireless Networks* (pp. 842–851). Springer Berlin / Heidelberg.