Observed Trends in Spam Construction Techniques: A Case Study of Spam Evolution

Calton Pu
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
calton@cc.gatech.edu

Steve Webb
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
webb@cc.gatech.edu

ABSTRACT

We collected monthly data from SpamArchive over a three year period (from January 2003 through December 2005), accumulating more than 1.4 million messages. Then, we conducted an evolutionary study by running 495 spamicity tests from SpamAssassin on each month. The population of messages testing positive for each spamicity test indicates the adoption of the spam construction technique associated with that spamicity test. This paper focuses on two evolutionary trends in our population study: *extinction*, where the population dwindles to zero or near zero, and *co-existence*, where the population maintains a consistent level or even grows, despite attempts by spamicity tests to eliminate it. We divide the factors that lead to extinction or co-existence into three groups: environmental changes, individual filtering, and collaborative filtering. We observed evidence of extinction (e.g., HTML-based obfuscation techniques), and somewhat unexpectedly, we observed evidence of co-existence between spam messages containing construction techniques and spamicity tests in filters (e.g., illegal characters in the "Subject" header and block list collaborative filtering).

1. INTRODUCTION

The goal of spam producers is to create spam messages that reach their intended receivers (called victims or simply users). In response to the increasing amount of spam, many victims have adopted statistical learning filters [1, 10, 18, 21, 22] with the goal of finding and "killing" spam before it reaches their mailboxes. These frontally opposing goals have been modeled as an arms race [7, 13, 17], with the evolution of spam construction techniques and the increasing sophistication of spam filtering techniques. Our study on spam evolution is inspired by an analogy between the spam arms race and the biological arms race, where new drugs (e.g., antibiotics) are created to kill existing bacteria as well as the subsequent evolution of new bacteria variants that are capable of resisting these new antibiotics.

In this paper, we describe a population evolution study of spam construction techniques based on their "genetic markers" in spam messages. Specifically, we adopt the detailed analysis of spam message content and structure developed and maintained by the SpamAssassin Project [20]. In SpamAssassin 3.1.0, 495 *spamicity tests* are used to characterize spam. These tests reflect specific spam construction techniques that are used by spam producers. Typically, these spam construction techniques are syntactically correct features that are rarely used in legitimate email but frequently abused by spam producers in the construction of many spam messages. Like genetic markers, the spamicity tests help characterize spam through a detailed analysis of message content and structure. However, unlike genetic markers that deterministically characterize a strain of bacteria, the spamicity tests are statistical in nature, only indicating a probability of whether the message is spam. We observe at the outset that we are not evaluating the effectiveness of SpamAssassin's approach (as a spam filtering technique). We simply use SpamAssassin's tests as a type of "genome mapping" in our study of spam evolution. Concretely, we will look at the prevalence of spam messages that employ specific spam construction techniques (i.e., they test positive for specific spamicity tests) and analyze the changes in their popularity over a three-year period. This is analogous to a population study of bacteria strains using specific genetic markers. As a result, we sometimes refer to spam messages that test positive for a spamicity test as a "strain of spam."

In this paper, we study two trends in the analysis of spam construction techniques. The first trend of interest is *extinction*, indicated by the population of a strain of spam declining to zero or near zero during the study period. We will attempt to find a causal explanation for the spamicity tests that show extinction of spam messages employing those spam construction techniques. The second trend of interest is *co-existence*, indicated by a sustained population of a strain of spam, particularly through the end of the study period. Co-existence shows the survival of some spam construction techniques, even though the presence of spamicity tests shows a clear ability to identify those strains of spam messages. We found that explaining co-existence was usually quite speculative at this stage of study.

In our trend analysis of spam construction techniques, we classify the spamicity tests (and our explanations) into three groups of significant factors in our study of spam evolution. These groups are: (1) significant environmental changes, (2) individual filtering techniques, and (3) collaborative filtering techniques. For the cases of extinction, our hypothesis is that the identification of that spam construction technique (i.e., the definition of that spamicity test) was the cause of extinction. Conversely, the long term persistence of a strain of spam would indicate that the corresponding spamicity

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

test did not cause the extinction of the strain, even though that spam construction technique is clearly identifiable. The co-existence of a persistently surviving spam construction technique and its spamicity test in spam filters indicates an equilibrium similar to the co-existence of a pray and its predator. The co-existence does not necessarily mean the predator is ineffective in killing some of the pray. It simply indicates some concrete limitations in the predator's killing capability that allows the pray to continue to survive and perhaps even thrive. This study does not evaluate quantitatively the "amount of killing" for each spamicity test, which is a subject of future research.

The main contribution of the paper is the large-scale experimental evaluation of the prevalence of representative spam construction techniques over a three-year period. Concretely, we study the evolution of more than 1.4 million spam messages that were collected from January 2003 through December 2005. Through this study, we have found convincing evidence that some factors have been effective in causing the extinction of specific strains of spam. As an example of significant environmental changes in the extinction category, the removal of USERPASS support by Internet Explorer and Mozilla in 2004 seems to have effectively eliminated that feature from spam messages. Prior to this environmental change, spammers (and phishers, in particular) were exploiting a syntactic feature of URLs (i.e., the ability to include arbitrary text in the <user>:<password> field of a URL) that allowed them to deceive users.

Perhaps more intriguingly, we failed to find conclusive evidence of extinction where some was expected. For example, URL block lists are considered an effective method for identifying spam-related URLs. When they are adopted by collaborative filtering, they are a powerful technique to identify spam [5, 16]. However, Figure 10 shows that despite the deployment of block lists by many sites, a significant percentage of spam messages persist in containing URLs that are listed on the block lists. Therefore, we include the URL block lists in the co-existence category. We note the coarse granularity of our study, which is only concerned with the extinction or co-existence of a particular spam strain. Consequently, URL block lists could be effective in distinguishing a number of spam messages, but they have not been as strong a deterrent as the removal of USERPASS support by browsers.

Our study based on spamicity tests has goals and methods that are qualitatively different from most previous and current reports on spam evolution [3, 4, 8, 11, 19], which focus primarily on the evolution of spam content (e.g., the emergence and popularity of certain topics such as drugs and stocks). By focusing on spamicity tests, our goal is to learn more about what allows some spam messages to pass through the filters to reach their victims and what prevents others. This is in contrast to topical analysis, which is primarily a reflection of the expected economic gains of spam producers.

The rest of the paper is organized as follows. Section 2 describes our spam corpus and experimental setup. Section 3 shows illustrative examples of spamicity tests that became extinct over time. Section 4 summarizes examples of spamicity tests that show unexpected resilience to filtering techniques. Section 5 summarizes related work, and Section 6 concludes the paper.

2. EXPERIMENTAL EVALUATION METHOD

2.1 Spam Corpus Collection and Preparation

Since January 2003, we have been collecting spam messages systematically. For each period (e.g., monthly), we copy the new spam messages from the SpamArchive spam corpora[1]. As of January 2006, our accumulated spam corpus contained more than 1.4 million spam messages. SpamArchive's spam messages are stored in two collections: submit and submitautomated. The messages in the submit collection were submitted individually by users, and the messages in the submitautomated collection were submitted by automated tools on behalf of their users. Each of these collections contains close to a thousand archives that are stored as gzipped mbox folders. The spam messages within these mbox folders contain most of their original headers; however, some information has been removed to protect the privacy of the users that submitted the messages. Specifically, the recipient of the message (the "To" header) has been replaced by "submit@spamarchive.org," and the IP address of the relay recipient in the first "Received" header (i.e., the relay used to deliver the message to the submitting user) has been omitted.

Since the SpamArchive spam corpora are updated daily, our system is fully automated to update concurrently with those corpora. Every day, the system performs a number of activities. First, it downloads the latest archives from SpamArchive's two spam collections (i.e., submit and submitautomated), and these archives are gunzipped to obtain the corresponding mbox folders. Next, to facilitate tracking the evolution of various spam characteristics over time, the spam messages in these mbox folders are sorted based on the month and year they were received by the users that submitted them to SpamArchive.

The email messages stored in an mbox folder typically have three fields that store date information: the "From " line that delimits each message in the mbox folder (not to be confused with the "From" email header), the "Date" email header, and the "Received" email header(s). To sort the spam messages, we used the date information found in their "Received" email header(s) because that information is the most reliable indication of when the messages were delivered to the submitting users. Specifically, we used the first "Received" header because it is attached to the message by the relay that is responsible for delivering the message to the submitting user (i.e., the most trustworthy relay between the spammer and the end user). We rejected the date information in the "From " line because it represents the date that the message was placed in the mbox folder by SpamArchive and not the date that the message was received by the submitting user. We rejected the "Date" email header date information because it can be spoofed easily by spammers. Figure 1 shows the distribution of spam messages that were received from January 2003 through December 2005, based on our sorting algorithm.

2.2 Testing Infrastructure

As previously mentioned, the actual characteristics of spam messages that are used in our evaluation have been adopted from SpamAssassin 3.1.0. SpamAssassin is an open source

[1] SpamArchive's spam corpora can be found at ftp://spamarchive.org/pub/archives/.

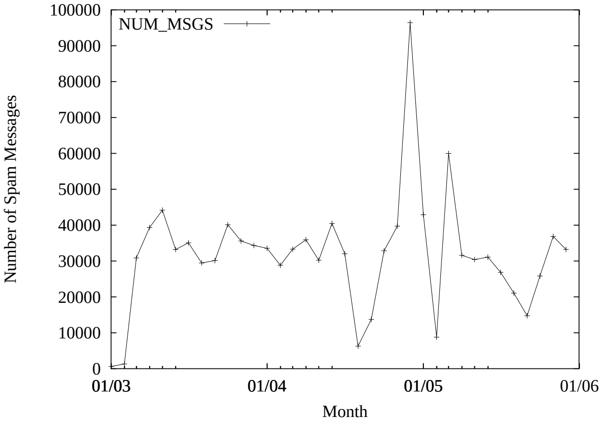

Figure 1: Month-by-month break-down of the number of spam messages in our spam corpus

spam filter that identifies spam messages by combining various spam detection techniques. These techniques include textual analysis of a message's headers and body, querying DNS block lists, and querying collaborative filtering databases.

Each of SpamAssassin's spam detection techniques is composed of a variety of spamicity tests. For example, SpamAssassin's textual analysis component contains tests such as OBFUSCATING_COMMENT and INTERRUPTUS, which identify examples of HTML-based obfuscation techniques. All of these tests have a user-specified score associated with them. When SpamAssassin analyzes a given message, it runs all of the spamicity tests. When the message satisfies one of the tests (i.e., it tests positive), that test's score is added to an overall spamicity score. The message is classified as spam if its accumulated overall spamicity score is above a user-specified threshold. In our experiments, we are only interested in the results of each test on each message, and we ignore the overall score since we are not using SpamAssassin as a spam filter. Specifically, we ran the spamicity tests on the messages (grouped by month), and for each test, we counted the number of messages (population) that tested positive. To compensate for the variations of new spam messages each month (Figure 1), we normalize the population count, dividing it by the total number of messages in that month.

2.3 Overview of the Spam Evolution Study

In this section, we summarize the results of evaluating all 495 spamicity tests from SpamAssassin 3.1.0 on our three-year spam collection from SpamArchive. We have divided the spamicity test graphs into three groups: extinction, co-existence, and complex. The extinction group (discussed in Section 3) consists of graphs that show a downward trend, starting from a significant number of messages testing positive and ending with a relatively negligible number during the last three months. The co-existence group (discussed in Section 4) consists of graphs that show a persistently high number of messages testing positive at least for the last three months (regardless of the starting point). The complex graphs combine different trends or contain high variability, and their analysis is the subject of future research.

The extinction group includes 236 spamicity tests, and the co-existence group includes 64 spamicity tests. We studied the distribution of test popularity within each group. Table 1 contains the distribution of the number of tests (divided by group), according to the average percentages calculated over the three-year period. The overwhelming majority of tests averaged less than 10% of the messages, with only 22 tests in the co-existence group averaging between 10% and 60%. Since the co-existence group consists of curves that remain flat (to be considered surviving), it is expected that this group contains the highest average numbers. However, the average does not represent the extinction group since it blurs the downward trend that characterizes the group. Table 2 contains the distribution of the number of tests according to the maximum percentages achieved during that three-year period. The table shows that the extinction group contains a significant number of tests (35) that started with more than 10% of messages containing that spam construction technique.

Table 1: Distribution of average results

Average Range	Number of Spamicity Tests		
	Extinction	Co-existence	Complex
[0.0 – 0.1)	230	42	195
[0.1 – 0.2)	6	11	0
[0.2 – 0.6)	0	11	0

Table 2: Distribution of maximum results

Maximum Range	Number of Spamicity Tests		
	Extinction	Co-existence	Complex
[0.0 – 0.1)	201	26	180
[0.1 – 0.2)	22	12	14
[0.2 – 0.3)	8	8	1
[0.3 – 0.4)	4	4	0
[0.4 – 0.5)	1	5	0
[0.5 – 0.9)	0	9	0

The remaining graphs (Figure 2 through Figure 11) shown in the paper are population evolution graphs, with the x-axis representing time (from January 2003 through December 2005) and the y-axis representing the percentage of messages (in a given month) that tested positive for the various spamicity tests being shown in the graph.

3. EVIDENCE OF EXTINCTION

Of the 236 graphs in the extinction group, we selected a few of the most interesting ones for discussion. In a sense, these are the "success stories" for spam filtering or other techniques that combat spam since spam producers are completely avoiding these markers.

3.1 Significant Environmental Changes

The evolution of the USERPASS spam signature is an example of the extinction category. According to RFC 1738 [2], URLs with the following format are syntactically valid:

```
<scheme>://<user>:<password>@<host>:<port>/<url-path>
```

However, spam producers (particularly, phishing message producers) began to exploit this URL format to deceive

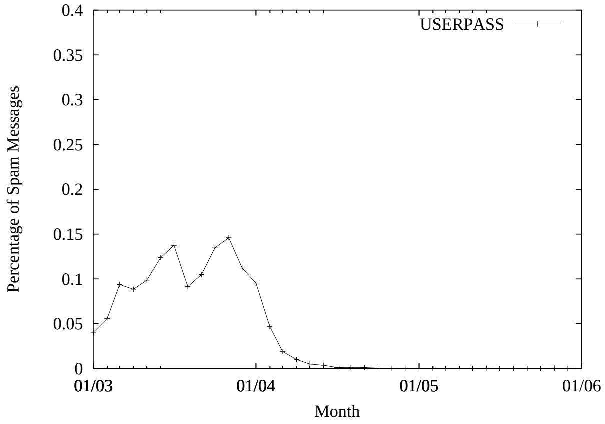

Figure 2: Evolution of the presence of Username:Password URLs

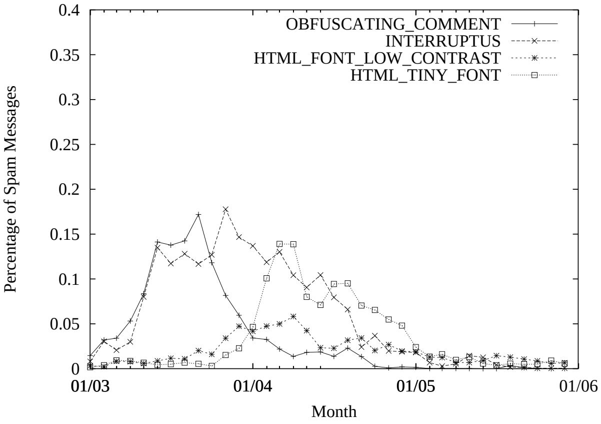

Figure 3: Evolution of specific HTML-based spam obfuscation techniques

users, disguising their malicious sites by inserting a popular site in the <user>:<password> field. For example, a phisher might use http://www.ebay.com@badsite.com to trick users into believing they're accessing ebay.com, when they're actually accessing badsite.com. A more extensive discussion of this technique (and its many variations) is provided in [14]. The USERPASS spam signature tracks messages that contain URLs with the <user>:<password> format. As the figure shows, more than 10% of the spam messages in almost every month from May 2003 through January 2004 contained at least one USERPASS URL. Then, rather abruptly, only 4.7% of the spam messages in February 2004 contained a USERPASS URL. In the following month, this percentage fell to 1.9%, and by May 2004, almost none of the spam messages contained a USERPASS URL. Upon observing this trend, the most obvious question is, "What forced phishers to abandon this technique?"

On February 2, 2004, Microsoft issued Microsoft Security Bulletin MS04-004[2] along with a security update that removed support for USERPASS URLs from Internet Explorer. The Mozilla Project quickly followed suit by removing USERPASS support from Mozilla (as described in Mozilla's Bugzilla Bug 232567[3]). Both of these actions are environmental changes that made the USERPASS option useless to spam/phishing producers. As Figure 2 shows, by mid-2004, spam producers eliminated all USERPASS URLs from their messages.

3.2 Individual Filtering

One of the earliest defenses against spam was keyword-based filters [6]. Unfortunately, spam producers defeated keyword filters by replacing keywords with randomized misspellings. In response, victims began to use statistical learning filters (e.g., Naïve Bayesian, Support Vector Machines – SVM, and LogitBoost) [1, 10, 18, 21, 22] that are capable of learning and identifying a large number of unpredictable misspellings. These filters operate individually (i.e., they are trained by each user), and it appears that some spam strain extinctions are due to the effectiveness of these individual filters. An example is HTML-based obfuscation techniques.

We first discuss the evolution of four spam construction techniques involving HTML-based obfuscations:

- OBFUSCATING_**COMMENT**
- **INTERRUPTUS**
- HTML_FONT_**LOW_CONTRAST**
- HTML_**TINY_FONT**

These techniques are used to disguise keywords that indicate spam (have high spamicity scores). Each one of the techniques can be used to invisibly divide spam keywords into randomized components. The result is the avoidance of keyword filters and low spamicity scores by statistical learning filters since the customary spam keywords are never present in their entirety. For example, **LOW_CONTRAST** and **TINY_FONT** were used to introduce virtually invisible fragments so the visual presentation becomes quite different from the underlying parsed text. Similarly, **COMMENT** and **INTERRUPTUS** are used to insert HTML tags in the middle of keywords, making the keywords unrecognizable by learning filters. For example, spam producers might obfuscate the word *Viagra* using Vi<xxx>ag<yyy>ra or V<!−−x−−>iagr<!−−y−−>a.

Figure 3 shows the evolution of these four spam construction techniques. Initially, the **COMMENT** and **INTERRUPTUS** techniques were the most popular. Then, as the popularity of the **COMMENT** technique steadily declined, spammers focused their attention on the INTERRUPTUS technique. When the INTERRUPTUS technique began to decline (after November 2003), the **LOW_CONTRAST** and **TINY_FONT** techniques were already rising in popularity. These phase differences suggest an arms race between spam producers and individual filters. As spam producers adopt an HTML-based obfuscation technique, spam filters (e.g., SpamAssassin and statistical learning filters) begin to associate the technique with high spamicity scores during the continuous retraining of the filter. The effectiveness of filter retraining [7, 17, 21] forces spam producers to migrate to a new obfuscation technique. Figure 3 shows four specific examples of these arms race cycles.

Fortunately, the spamicity tests also give us a tool to analyze all HTML-based obfuscation techniques as a group.

[2] http://www.microsoft.com/technet/security/bulletin/MS04-004.mspx

[3] https://bugzilla.mozilla.org/show_bug.cgi?id=232567

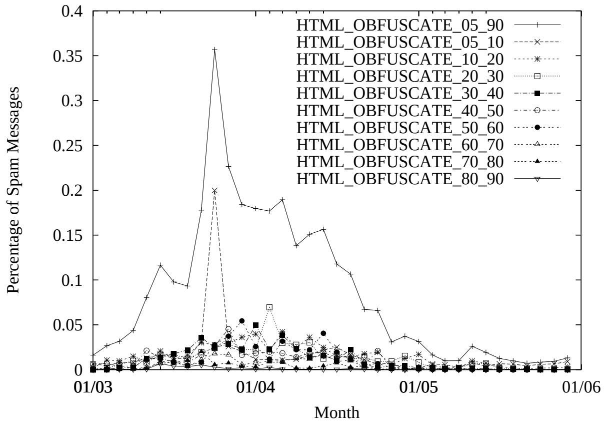

Figure 4: Evolution of all HTML-based spam obfuscation techniques

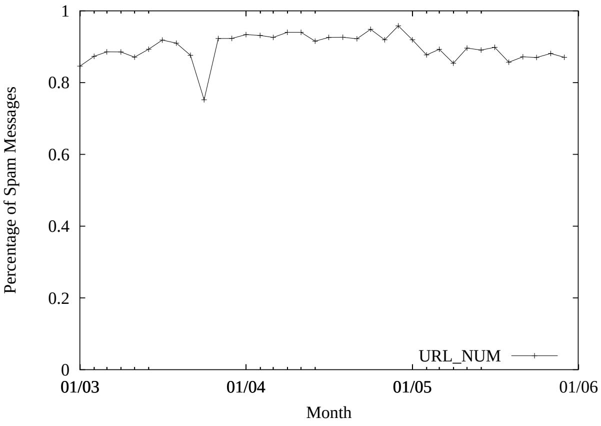

Figure 5: Evolution of the presence of URLs in spam messages

Figure 4 shows the population of spam messages that employ any HTML-based obfuscation techniques, classified by the percentage of HTML obfuscation content in each message. For example, the line marked as HTML_OBFUSCATE_05_10 represents the percentage of spam messages with a message body consisting of between 5% to 10% HTML obfuscation content. Similarly, the line for HTML_OBFUSCATE_05_90 represents almost all of the messages that contain HTML obfuscation content. In Figure 4, we can observe that the line for HTML_OBFUSCATE_05_90 gradually increases, indicating a possible learning curve. Then, after the peak in October 2003, spam producers began to slowly move away from HTML-based obfuscations, and by March 2005, the number of messages containing HTML obfuscation techniques became vanishingly small.

Figures 3 and 4 suggest that individual filters won a battle against spam producers in the HTML-based obfuscation arms race. Although new obfuscation techniques (e.g., camouflage attacks [17, 21]) continue to plague learning filters, the individual filters were able to successfully detect HTML-based obfuscation techniques. As a result, by the end of 2005, this filtering ability forced the spam producers to virtually abandon spam construction techniques using HTML-based obfuscation.

3.3 Collaborative Filtering

Our efforts to find a clear example of extinction for spamicity tests in the collaborative filtering category failed to yield good results. Instead, we found some evidence of co-existence (Section 4.3), where clearly effective collaborative filtering techniques did not cause those spam construction techniques to become extinct. Whether collaborative filtering techniques have inherent limitations that prevent them from "killing off" some strains of spam is an interesting area of future research.

4. SURVIVAL AND CO-EXISTENCE

In this section, we discuss examples of spam construction techniques that exhibit unexpected and persistent resiliency. These examples are interesting since they seem to work well for the spam producers, despite explicit identification tests and attempts to filter them out. This phenomenon is contrary to our expectations since we would normally expect the spamicity tests to be effective in filtering out these targeted spam messages. The resilient nature of these spam construction techniques makes them good candidates for further research since these strains of spam are surviving and perhaps even thriving.

4.1 Significant Environmental Changes

First, we discuss a spamicity test that, by definition, cannot be used to extinguish spam messages: the presence of URLs in an email message. In contrast to USERPASS, which was rarely used in general, URLs are often present in both spam and legitimate messages. Figure 5 shows that at least one URL appeared in between 85% and 95% of spam messages in every month except for October 2003 (when only 75% of the spam messages contained at least one URL).

Although a single data point could be the result of data collection problems or random statistical fluctuations, we have a conjecture for the dip shown in October 2003. On September 23, 2003, California Governor Gray Davis signed into law an anti-spam bill, Senate Bill No. 186, that made each email advertisement fineable up to $1 million [15]. This could explain the removal of URLs from some spam messages and the dip shown in October 2003. Unfortunately, Congress passed the CAN-SPAM Act at the end of October, which replaced the strict penalties detailed in the California anti-spam bill [12]. This could explain the "back to business as usual" mindset of spam producers and the restoration of URLs to their normal level.

A refinement of the URL-presence spamicity test consists of the tests for URLs with specific top level domains (TLDs). While COM and NET are commonly used TLDs, other TLDs such as BIZ and INFO have also been used frequently by email marketers. Figure 6 shows the evolution of spam messages containing URLs with four specific TLDs: COM, NET, BIZ, and INFO (the four most popular TLDs found during the three-year period). We observe a dip in the COM curve around October 2003 (lasting four months). This dip seems anti-correlated with a peak in BIZ around the same time. Whether this valley and peak are correlated with the dip in Figure 5 is open for debate. Another observation from Figure 6 is that the spam producers' favorite TLD choice (other than COM) has changed from NET (January 2003 to June 2003) to BIZ (July 2003 to July 2004) and

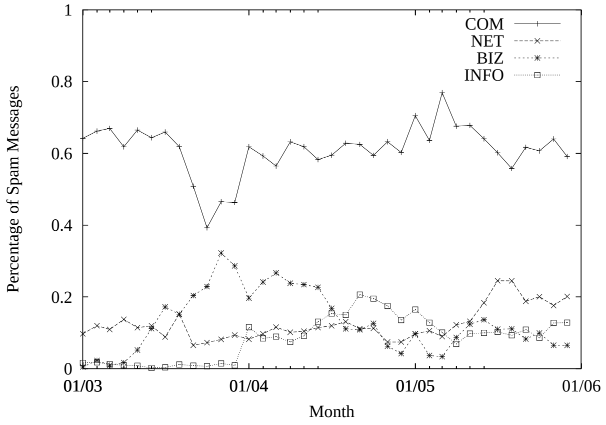

Figure 6: Evolution of the presence of URLs with specific TLDs

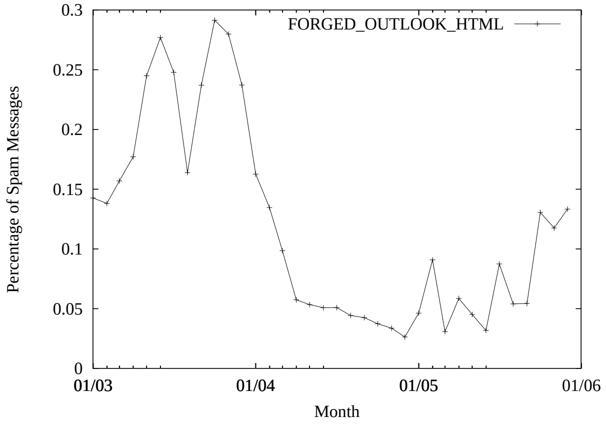

Figure 7: Evolution of messages that pretend to be sent using Outlook

INFO (August 2004 to March 2005), with NET eventually returning to the top position (April 2005 to December 2005).

4.2 Individual Filtering

In this section, we analyze three cases of individual filtering that were unsuccessful, despite seemingly having the capability to extinguish spam messages of a particular strain. The first case concerns spam messages that pretend to be sent by Outlook (i.e., the messages contain a forged "X-Mailer" header with "Microsoft Outlook"). When the real Microsoft Outlook application sends an HTML email message, two versions of the message are sent: the HTML version and an automatically generated plain text version. Since the forged messages only contain HTML content, they could not have been sent by Outlook; thus, this spamicity test is a fairly reliable indicator of spam. Figure 7 shows the survival and co-existence of spam messages containing the forged Outlook header with this spamicity test. In 2003, the forged header was consistently identified in over 15% of the spam messages, but this value dropped to around 5% in 2004. Surprisingly, in 2005, the technique began to grow in popularity towards 15%, despite the spamicity test.

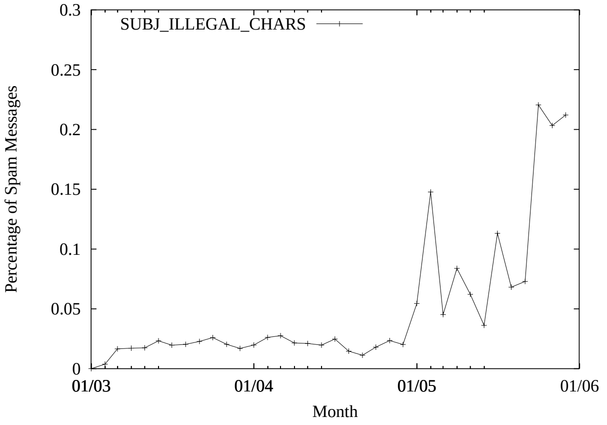

Figure 8: Evolution of messages that use at least 2 illegal characters in their "Subject" headers

The second case consists of messages that use at least 2 illegal characters in their "Subject" headers. An illegal character is defined as a character that should be MIME encoded (as per RFC 2045 [9]) but is not. Figure 8 shows the evolution of the number of messages that employ this spam construction technique. This figure shows that despite the identification of this spamicity test and attempts to filter it out, the number of spam messages containing such illegal characters actually grew from about 2% before 2005 to about 20% at the end of 2005. The reasons for this thriving spam construction technique are a subject of future research.

The third case consists of messages that contain a specific pattern in their "Message-ID" headers. The pattern is defined by the following Perl regular expression:

```
<[0-9a-f]{4,}\$[0-9a-f]{4,}\$[0-9a-f]{4,}\@\S+>
```

This pattern is legitimately used by mail clients that place "Produced By Microsoft MimeOLE" in their "X-MIMEOLE" headers. Thus, if a message contains the pattern without this value in its "X-MIMEOLE" header, the "Message-ID" header is considered forged, and the message is considered spam. Figure 9 shows the evolution of the number of messages that have the above pattern in their "Message-ID" headers. This feature has gained and lost popularity over the years, with a low of 2% in early 2005, followed by a sudden growth during 2005, and another low of 2% at the end of 2005. Due to the ups and downs in the figure, despite the ease of executing this spamicity test, whether this strain of spam has become extinct is unclear. Depending on the interpretation of the curve during 2005, Figure 9 can be interpreted as extinct (if you only look at the last three months), co-existence (if you take the average for the year), or inconclusive and therefore in the complex category.

4.3 Collaborative Filtering

Another example of an obviously effective spamicity test concerns "URL block lists" that enumerate URLs that are known to be spam-related through reliable sources. These block lists are typically constructed and maintained by collaborative filtering (i.e., contributions by many trusted participating users). For a given block list, the spamicity test finds the spam messages that contain at least one URL that appears on that block list. Figure 10 shows the evolution of the number of messages that contain at least one

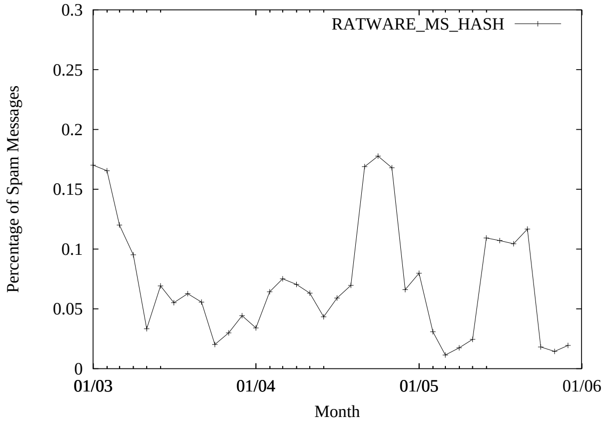

Figure 9: Evolution of messages that have a specific pattern in their "Message-ID" headers

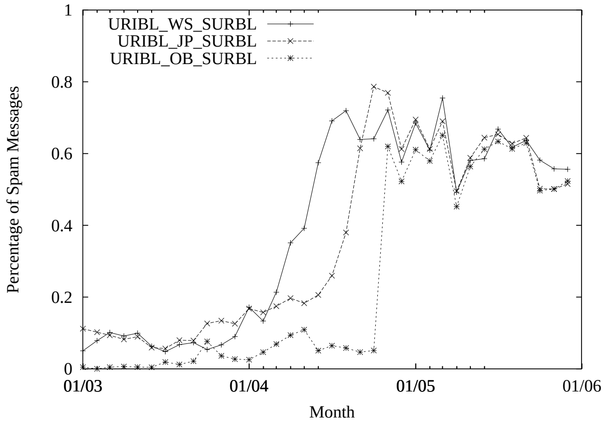

Figure 10: Evolution of URLs appearing on block lists

URL that appears on one of three block lists: ws.surbl.org, jp.surbl.org, and ob.surbl.org.

As the figure shows, the percentage of spam messages that contained at least one URL on any of the three block lists remained below 20% up until March 2004. Then, from March 2004 through August 2004, the percentage of spam messages that contained at least one URL on ws.surbl.org grew from 21.4% to 71.9%. It took jp.surbl.org slightly longer to gain this level of popularity, but from March 2004 through October 2004, the percentage of spam messages that contained at least one URL on this block list grew from 17.5% to 78.6%. The ob.surbl.org block list was the slowest to obtain popularity, but from October 2004 through November 2004, the percentage of spam messages that contained at least one URL on this block list skyrocketed from 5.1% to 62.0%.

An alternative explanation for the population gains in Figure 10 is the improvement in collaborative filtering performance. Suppose that the effectiveness of collaborative filtering is directly related to the participation of effective collaborators. It is reasonable to assume that at the beginning of any collaborative effort, only a limited number of effective collaborators participate, with a limited coverage. In the case of block lists, this effect would translate to a partial coverage of known spam-related URLs. As more and more people contribute suspicious URLs, the block list becomes more comprehensive, and the spamicity test becomes more effective. This may explain the phase differences among the three lists, if we assume that ws.surbl.org achieved full effectiveness first, followed by jp.surbl.org and ob.surbl.org. Figure 10 shows that the block lists are clearly effective in identifying spam messages (finding more than 50% of the spam messages after November 2004) that contain spam-related URLs.

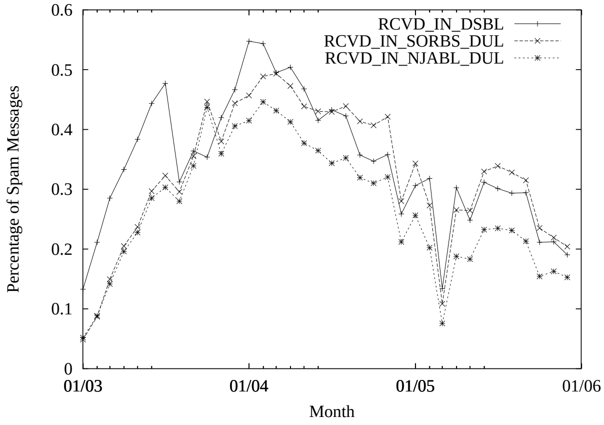

Figure 11: Evolution of relays appearing on block lists

As we acknowledge the success of URL block lists, the continued and constant presence of spam-related URLs in spam messages also shows that the effectiveness of block lists is somehow limited. Unlike USERPASS and HTML-based obfuscation, which became extinct by a change in browser technology and effective individual filtering, the presence of spam-related URLs on block lists did not completely "kill" this strain of spam. The survival and co-existence of spam-related URLs on block lists implies that the benefits for spam producers to continue including spam-related URLs in their messages may outweigh the costs of being identified and filtered by the block list spamicity tests.

One possible explanation for this phenomenon involves a cost/benefit analysis from the spam producer's point of view. It is obvious that having a convenient URL directly pointing to the spam producer's web site is very valuable. If the spamicity test is able to completely and immediately filter out all such spam messages, this strain of spam would probably become extinct. Since we assume the block list coverage is very good, the main question is the length of the time lag between the creation of a spam-related URL and its detection and inclusion on a block list. Despite the presence of effective collaborators, it is reasonable to assume a non-negligible time lag (e.g., on the order of hours or days) exists between the creation of a new URL and its discovery and inclusion on a block list. This time lag could be a fundamental limitation of the collaborative filtering approach, which is based on effective human participation.

Another analogous collaborative approach is DNS block lists. Unlike the URL block lists described above, DNS block lists attempt to filter messages based on the servers that were used to deliver the messages. Figure 11 shows the evolution of the number of spam messages that were delivered through at least one relay that appears on a DNS block list.

Three block lists are represented: The Distributed Sender Blackhole List (DSBL), The Spam and Open Relay Blocking System (SORBS), and NJABL.ORG. We observe both growth (probably due to the stabilization process described above for collaborative filters in general) and decline of this spamicity test in Figure 11. However, as of December 2005, the spam messages that pass through at least one of the relays on these lists did not become extinct (at around 20%). We believe the time lag explanation above also applies here.

5. RELATED WORK

Previous studies that investigated the evolution of spam were primarily concerned with the content of spam messages. Fawcett [8] discovered a few interesting spam trends that occurred in 2002. In his study, he found a great deal of variation in the traffic patterns of spam and legitimate email messages, and using these variations, he illustrated the time variation of the class prior p(spam). He also investigated the evolution of spam message content over time, finding a number of complex trends. Specifically, he found that spam terms (i.e., words that appear in spam messages) fall into a combination of three categories: constant, periodic, and episodic occurrences. Finally, Fawcett mentioned the early stages of the "spam arms race" [17], primarily focusing on simple techniques that were created to defeat keyword filters.

On two separate occasions [3, 4], Brightmail published statistics about the evolution of spam traffic and spam content. In the first set of statistics [3], they showed that from January 2003 through December 2003, the percentage of all email that was spam grew from 42% to 58%. Additionally, in December 2003, they found that most spam messages were categorized as PRODUCTS (21%) and ADULT (18%). In the second set of statistics [4], they showed that from January 2004 through March 2004, the percentage of all email that was spam grew from 60% to 63%. Additionally, in March 2004, they found that most spam messages were categorized as PRODUCTS (25%) and FINANCIAL (20%). In the middle of 2005, Sophos also released statistics that provided spam content categorizations. Specifically, they found that from January 2005 through June 2005, Medication/pills was the top spam category (41.4% of all spam messages during that period), followed by Mortgage (11.1% of all spam messages during that period).

Hulten et al. [11] "hand-examined" 200 spam messages from a one month period in 2003 and 1000 spam messages from a one month period in 2004. The purpose of this examination was to identify the types of products being promoted and the types of exploits being used by spam messages. Their main observations were that non-graphic porn/sex content was the most prevalent spam category and that "text chaff" (i.e., textual obfuscation techniques) was the most prevalent exploit in their data.

Our study differs from the previous work on spam evolution in several ways. First, we study the evolution of spam construction techniques in spam messages, instead of spam content. Second, our study uses large corpora (over 1.4 million spam messages over a three year period) to produce concrete and clear evidence of evolution. Third, we focus on two clear trends (extinction and co-existence) that give us a quantitatively supported evaluation of spamicity test effectiveness in "killing" spam messages (either completely for the extinction group or partially for the co-existence group).

6. CONCLUSIONS

In this paper, we studied the evolution of spam, focusing on a trend analysis of spam construction and filtering techniques. The study used over 1.4 million spam messages that were collected from SpamArchive between January 2003 and January 2006. The spam constructions and filtering techniques were adopted from the spamicity tests found in SpamAssassin 3.1.0. The study ran all of the messages through all of the spamicity tests, and it plotted the percentage of messages for which the test result was positive through the three-year period.

We consider only two trends in this study: the spam construction techniques that became "extinct" (zero or near zero spam messages for that spamicity test) and the spam construction techniques that survived and co-exist with a well-defined spamicity test. We divide the explanations of these trends into three groups of spamicity tests: significant environmental changes, individual filtering, and collaborative filtering. Extinction of a spam construction technique means complete filter effectiveness (e.g., individual filtering of HTML-based obfuscation techniques) or environmental changes (e.g., the elimination of USERPASS functionality in browsers). In contrast, co-existence indicates the existence of concrete limitations in the spam filters. Identified examples include forged Outlook "X-Mailer" headers and illegal characters in "Subject" headers for individual filters and block lists for collaborative filtering.

This paper opens the door to many interesting future research opportunities using the same population evolution method. For example, each one of the spamicity tests showing co-existence is a challenge to be explained in more detail since those filters were unable to "kill off" that particular spam construction technique. More concretely, the several conjectures and potential explanations for the interactions between a spamicity test and its associated spam construction technique should be verified quantitatively. Another interesting research question is the lack of extinction examples for collaborative filtering, despite the large number of extinctions. Is it possible that collaborative filtering approaches have some inherent limitations (e.g., time lag) that prevent them from causing any strain of spam to become extinct?

7. ACKNOWLEDGMENTS

This work was partially supported by NSF under the ITR (grants CCR-0121643 and CCR-0219902) and CyberTrust/DAS (grant IIS-0242397) programs, an IBM SUR grant, and Hewlett-Packard.

8. REFERENCES

[1] I. Androutsopoulos, G. Paliouras, and E. Michelakis. Learning to filter unsolicited commercial e-mail. Technical Report 2004/2, National Center for Scientific Research "Demokritos", 2004.

[2] T. Berners-Lee, L. Masinter, and M. McCahill. RFC 1738 - uniform resource locators (url). `http://www.faqs.org/rfcs/rfc1738.html`, 1994.

[3] Brightmail. Spam percentages and spam categories. `http://www.nospam-pl.net/pub/brightmail.com/spamstats_Dec2003.html`, 2003.

[4] Brightmail. Spam percentages and spam categories. `http://www.nospam-pl.net/pub/brightmail.com/`

spamstats_March2004.html, 2004.

[5] J. Chan. Surbl - spam uri realtime blocklists. http://www.surbl.org/, 2006.

[6] W. W. Cohen. Learning rules that classify e-mail. In *Proceedings of the AAAI Spring Symposium on Machine Learning and Information Access*, pages 18–25, 1996.

[7] N. Dalvi et al. Adversarial classification. In *Proceedings of the 10th Int'l Conf. on Knowledge Discovery and Data Mining (KDD '04)*, pages 99–108, 2004.

[8] T. Fawcett. "In vivo" spam filtering: a challenge problem for KDD. *SIGKDD Explorations Newsletter*, 5(2):140–148, 2003.

[9] N. Freed and N. Borenstein. RFC 2045 - multipurpose internet mail extensions (mime) part one. http://www.ietf.org/rfc/rfc2045.txt, 1996.

[10] P. Graham. A plan for spam. http://www.paulgraham.com/spam.html, 2002.

[11] G. Hulten et al. Trends in spam products and methods. In *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS '04)*, 2004.

[12] JLCom Publishing Co. Senate unanimously approves creation of do-not-spam list. http://www.lawpublish.com/spam.html, 2003.

[13] D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS '05)*, 2005.

[14] G. Ollmann. The phishing guide: Understanding & preventing phishing attacks. http://www.ngssoftware.com/papers/NISR-WP-Phishing.pdf, 2004.

[15] PCWorld.com. California anti-spam law. http://www.pcworld.com/downloads/file_description/0,fid,23113,tfg,tfg,0%0.asp, 2006.

[16] V. V. Prakash. Vipul's razor. http://razor.sourceforge.net/, 2006.

[17] C. Pu et al. A case study of learning filters in the spam arms race: Resistance to camouflage attacks. 2006. submitted to Journal of Machine Learning Research.

[18] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*, pages 55–62, 1998.

[19] Sophos. Sophos identifies the most prevalent spam categories of 2005. http://www.sophos.com/pressoffice/news/articles/2005/08/pr_us_20050803t%opfive-cats.html, 2005.

[20] SpamAssassin Development Team. The apache spamassassin project. http://spamassassin.apache.org/, 2005.

[21] S. Webb, S. Chitti, and C. Pu. An experimental evaluation of spam filter performance and robustness against attack. In *Proceedings of the 1st International Conference on Collaborative Computing (CollaborateCom '05)*, 2005.

[22] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing*, 3(4):243–269, 2004.