Exploring the Spam Arms Race to Characterize Spam Evolution

Pedro H. Calais Guerra
Universidade Federal de Minas Gerais (UFMG), Brazil
pcalais@dcc.ufmg.br

Dorgival Guedes
Universidade Federal de Minas Gerais (UFMG), Brazil
dorgival@dcc.ufmg.br

Wagner Meira Jr.
Universidade Federal de Minas Gerais (UFMG), Brazil
meira@dcc.ufmg.br

Cristine Hoepers
Brazilian Network Information Center (NIC.br)
cristine@cert.br

Marcelo H. P. C. Chaves
Brazilian Network Information Center (NIC.br)
mhp@cert.br

Klaus Steding-Jessen
Brazilian Network Information Center (NIC.br)
jessen@cert.br

ABSTRACT

Current studies on spam evolution usually extract evolution patterns and trends by analyzing historical spam message corpora. In this paper, we propose a novel methodology that incorporates spam filters to the spam trend analysis, as they are the agents that may force spammers to change their tactics. Moreover, filters also evolve over time and different filter releases present different characteristics, providing different views of the spams. We considered both outdated and recent releases of the Open Source SpamAssassin filter and applied their criteria on spams collected from the Spam Archive dataset, a dataset that contains spams collected from 1998 to 2010. When we compare the effectiveness of old and recent filters over old and recent spams, spam trends naturally emerge. Our results give a general picture of the dynamic nature of spam over the last 12 years and indicate that, on any given year, spams from different generations are observed. Moreover, we investigated how the popularity of spam construction techniques changes when filters start to detect them. We also determined automatically techniques that seemed more resistant than others and thus subsidize studies on improving anti-spam mechanisms.

1. INTRODUCTION

Spam fighting is an "arms race" characterized by an increase in the sophistication adopted by both spammers and spam filters [8]. The co-evolution of spammers and anti-spammers is a remarkable aspect of the anti-spam battle and has motivated a variety of works that devise *adversarial* strategies to tackle the spam problem as a moving target [4, 1, 3].

Characterization and measurement studies also have been developed, describing spam trends regarding both content generation techniques [15, 7] and also the evolution of the infrastructure used by spammers to disseminate spams over the network, which migrated from direct spamming to complex chains of open proxies, open relays [2] and compromised machines organized in botnets [14].

However, current studies on spam evolution try to extract trends from spam based only on spam data, and do not consider that filters themselves also evolve to detect spammers' latest tricks and thus affect much of the behavior of spammers. In this paper, we take a different approach to investigate and measure spam evolution: instead of looking for trends and changes of spamming strategies based solely on historical logs of spam messages, we consider both spams *and* filters from different moments in time. Each filter release provides a different view in relation to spams, and confronting and comparing each view yields interesting information about spam evolution. We analyzed how spams from different generations are interpreted by the same filter, and how the different releases of a spam filter process the same spam. We employ this methodology to characterize how spam has evolved over the last 12 years and, additionally, we point out new aspects of spam evolution which have remained unnoticed by previous studies that focused solely on the evolution observed on the spammer's side.

Our analysis is based on the meta-features extracted from spams using the Open Source *SpamAssassin* [16] filter. Each meta-feature is the validity or not of a *spamicity test*, which checks for a specific spam construction technique. As on [15], we assume that SpamAssassin spamicity tests capture the most relevant aspects of spam message content and structure.

We considered 6 main releases of the SpamAssassin filter (from 2002, 2003, 2004, 2005, 2007 and 2009) and executed each filter against each spam message from the Spam Archive Corpus [10], which collected spams from 1998 to 2010. Our main contributions are:

- we demonstrate that confronting spam messages and filters from different generations and releases is a simple and elegant strategy to characterize spam evolution;
- we show that, in any year, spams from different generations compose the spam flow observed on that year;
- we show that the popularity of spam construction techniques varies diversely after such techniques began to be detected by filters;
- we group spam messages into clusters according to the filters' reaction to them. We found that while some

CEAS 2010 - Seventh Collaboration, Electronic messaging, Anti-Abuse and Spam Conference July 13-14, 2010, Redmond, Washington USA

"classical" types of spam are detected by all releases of the filter, other messages manage to evade from all releases.

The remainder of the paper is organized as follows. Section 2 discusses related work. In Section 3 we discuss SpamAssassin and how its ruleset changed over six major releases. In Section 4 we use the different views provided by each of those releases to assess spam evolution. Finally, Section 5 concludes the paper.

2. RELATED WORK

The dynamic behavior of spammers has been discussed in different venues. New spam techniques are documented in periodical reports generated by security companies which point out statistics about the most recent spam innovations and trends [18, 11, 13, 9]. Some of those reports show the changes in the coverage of spams across categories such as Adult, Phishing, Political and Health. Dangerous types of spam disseminating malware have been reported as an increasing threat [11]. The volumes of spam from different countries also change over time and are closely monitored by those companies, as well as the overall increase of the volume of spam over legitimate messages over time.

The academic field also has devoted some effort to understanding the spam arms race phenomenon. Researches like the Fawcett [7] pointed out some strategies that spammers began to adopt in 2002, such as word obfuscations intended to defeat bayesian filters. He also found that some spam terms exhibit a very bursty nature.

Sullivan used SpamAssassin to measure spam volatility over time [17]. SpamAssassin rules were applied to a small dataset comprising 2,500 spams and Principal Component Analysis (PCA) was used to map the original space of SpamAssassin meta-features in two composites variables (dimensions) that kept 86% of the same information. The two dimensions were then mapped as one time-invariant dimension (capturing 70% of the information) and one time-changing dimension (capturing just 16% of the information). The conclusion, then, was that spam evolves slowly and the assumption that spam is highly volatile and changes randomly is a myth.

The work which is the closest to ours is that of Webb and Pu [15]. The authors investigated the evolution of the adoption rate of spam techniques detected by SpamAssassin on the same dataset we consider in this work, but over a shorter period of time (as their research is from 2006) and evaluating just one release of the filter (3.1.0, released in 2005). They establish an analogy to biological evolution and classify the adoption of spam construction techniques over time in two patterns: *extinction*, when an obfuscation strategy is abandoned by spammers after some time, and *co-existence*, when they are still adopted despite being detected by filters. They raise hypotheses that may explain those trends. Our work is complementary to theirs and we use the different views provided by multiple filters to explain some of their observations. Furthermore, we propose and apply a systematic methodology to find interesting trends, going beyond an exploratory analysis.

3. CHARACTERIZING FILTER EVOLUTION

SpamAssassin [16] is an open source spam filter which employs a variety of spam detection techniques, such as bayesian filters, and queries to DNS blacklists and to spam signatures databases. SpamAssassin also counts with a set of rules, usually represented as regular expressions, that are matched against the body or header fields of each message. For example, the rule `/V(?:agira|igara|iaggra|iaegra)/i` looks for spams that obfuscate the word Viagra. Each spam detection rule has an associated score value that will be assigned to a message if it matches the rule; scores are summed up and a message is classified as a spam if the total score goes above a given threshold (usually, 5.0).

From 2002 to 2010, 25 SpamAssassin versions were released[1]. Each version included new features, bug fixes and an updated ruleset. Each ruleset included new rules to keep up to date with the most recent spamming techniques, removed rules that were no longer considered useful, and kept other rules that were still considered effective.

Table 1: SpamAssassin's releases description

Release	Year	Rules	Added	Removed
2.4.3	Oct 2002	655	655	–
2.6.4	Jul 2003	591	283	347
3.0.0	Jun 2004	463	187	315
3.1.0	Jun 2005	500	159	122
3.2.0	Jan 2007	636	367	231
3.3.0	Jun 2009	475	92	253

Table 1 shows a summary of the SpamAssassin releases we have considered in this work. We considered two releases from the 2.x.x cycle (2.4.3 and 2.6.4) and each 3.x.0 version: 3.0.0, 3.1.0, 3.2.0 and 3.3.0. We discarded all 3.x.x intermediary releases because their rulesets had very few changes. Figure 1 shows the composition of each filter's ruleset when compared to the others, i.e., the proportion of rules first found in each version. For example, the second SpamAssassin release we considered (release 2.6.4) kept 308 rules which were present on release 2.4.3 and introduced 283 new ones. The recent 3.3.0 ruleset is composed of rules which were introduced in all previous 5 versions considered. We can observe that the size of the ruleset which is kept from a given release usually is reduced after each new release, which is an evidence that some of those rules have become outdated and no longer capture spammer behavior. Furthermore, the introduction of new rules is an evidence of the new strategies adopted by spammers. The evolution of filters, therefore, act as an indirect vantage point on spammer evolution. On the next section, we exploit this fact to characterize spam evolution over time.

4. CHARACTERIZING SPAM EVOLUTION

In this section we characterize how the spamming strategies evolve from the perspective of the spam filters using data from the Spam Archive dataset. The Spam Archive dataset [10] is composed by 2,223,353 spam messages collected using spam traps since 1998. Each of those spams were used as an input for each of the 6 filter releases we described in Section 3, what resulted in approximately 14.4 million evaluations. 61.7 million rules were matched — we considered just content rules, disabling all rules that checked

[1]available at `http://archive.apache.org/dist/spamassassin/` and `http://svn.apache.org/viewvc/spamassassin/branches/`

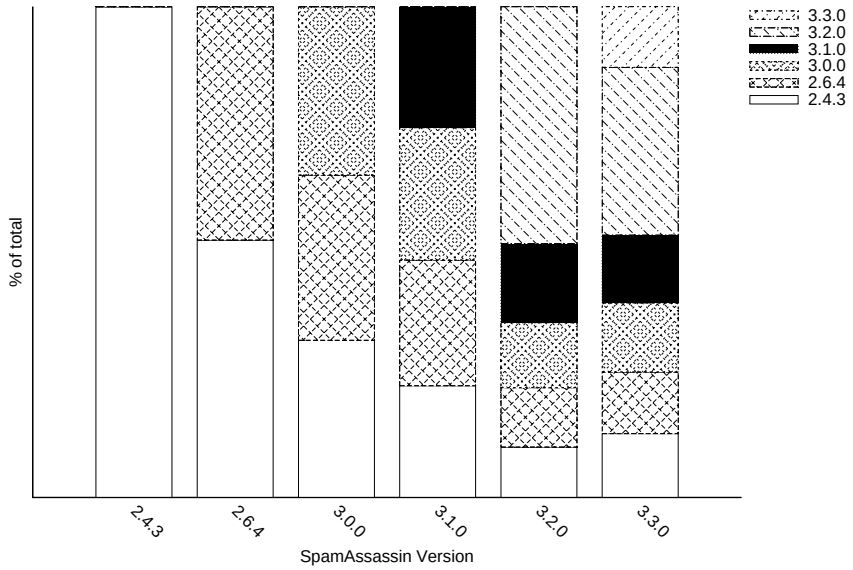

Figure 1: Ruleset composition for each SpamAssassin Release

blacklists and required external information, since that information changed over the years and previous snapshots are not available in all cases.

The set of rules matched by each SpamAssassin release to each message represents how that release classified each spam in terms of the strategies adopted by spammers to build the spam content. Although we have not experimented with other filters, we believe that SpamAssassin provides a significant view of the anti-spam arsenal available in each period, due to its widespread adoption.

Our methodology to characterize spam evolution is to cross filters and spams from different ages and then compare how each filter identified each spam (Sections 4.1 and 4.2), and how each spam was identified by all filters (Section 4.3).

4.1 Using filter evolution to characterize spam trends

We start by evaluating the effectiveness of the various filters in detecting spams. Figure 2 shows how each SpamAssassin release identified spam characteristics in messages collected on a monthly basis in the period from March 1998 to February 2010. For each SpamAssassin release, we computed the average number of rules matched for each spam in each year as a measure of the effectiveness of the ruleset for a group of spams from a given period. For each version, we considered just the rules which had been created in that version. For example, the rule FB_CIALIS_LEO3, which looks for obfuscations in spams selling Cialis pills, was introduced in SpamAssassin release 3.2.0 (2007). Although that rule was still present on release 3.3.0 (2009), it was not considered to produce the line for the later version, since we wanted to assess how a detecting strategy first introduced by a certain release would work on spams created both before and after that release.

The first hypothesis we wanted to confirm was that the rules from the release from a given year would have difficulty in identifying spams that came after its time. In fact, for any of the 6 releases we considered, the effectiveness of the release decreases after its release date. That suggests that spammers notice when their tactics have been detected and at least reduce their dependence on those tactics. The only exception is for release 3.3.0, probably because that version was released too recently for spammers to react to its updated ruleset. Figure 2, in certain aspects, captures the cyclical evolution of spam since 1998: on any given period, new strategies were being created while other spam construction techniques were being abandoned. As soon as a new ruleset was released, spammers stopped using some of the techniques detected by that new release and start adopting new strategies which would come to be detected by the filters that came afterwards. We can see that the curve for each release spikes at a different moment, which is always coincident with the year it became available. It is interesting to notice, though, that rules found in the first release we considered were successful for a longer period and had a slower drop than others, probably indicating that some old tactics persisted for a long time.

Although Figure 2 shows that the release from a given year is usually the best tool to detect spams from that year, it provides an aggregate view. Were there spams that would be better detected by rulesets released before or after their time? To assess this hypothesis, we identified the best filter for each spam in the SpamArchive dataset and plotted the results in Figure 3. We considered the best ruleset to detect a spam message the one that matched more rules against it. In this analysis, we considered all the rules present in each release and not just the rules introduced by that release. We also experimented with the total score computed by each filter as the comparison criterion, and results were similar.

One thing that can be easily noticed is the prevalence of the oldest release considered (2.4.3) for the period from 1998 to 2001, actually before its release. The main reason for that is that we did not have older releases to compare against, so that release is the best to match that period *among the ones*

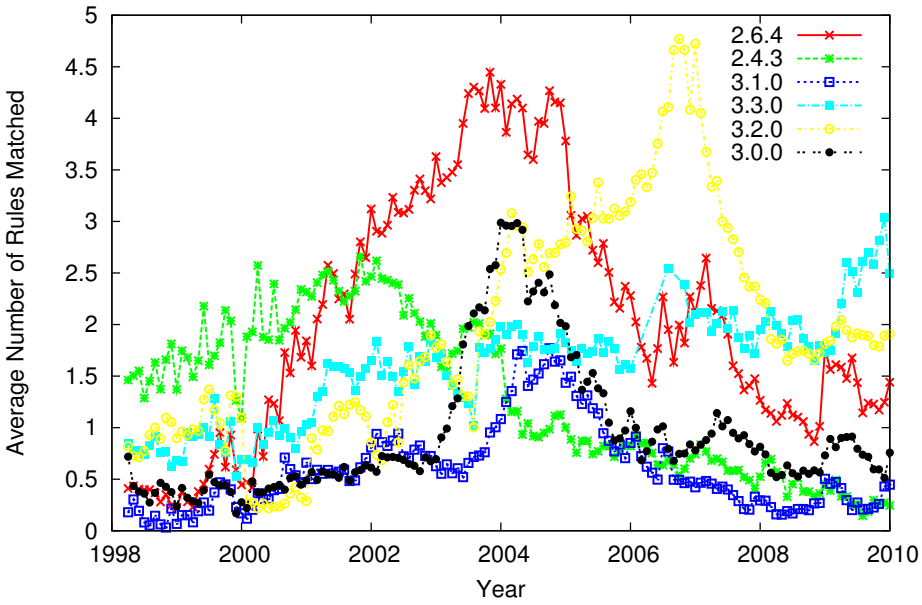

Figure 2: Average number of rules from each filter release matched against spams by year

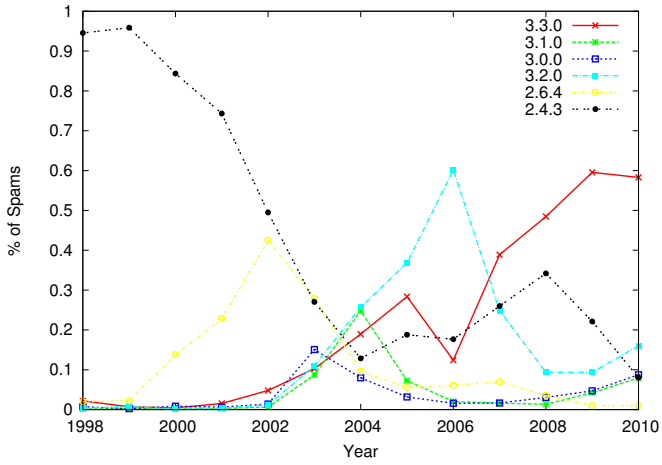

Figure 3: Best ruleset in each period (in % of spams detected by each ruleset)

considered. Nevertheless, that suggests that spam during that period may have changed more slowly, and several of the tactics that were addressed by the rules in the release from 2002 were already present in the years before.

However, even on recent years, there is a non-negligible fraction of spams whose construction techniques are better captured by older releases. In 2010, although 60% of the spam messages were better detected by the most recent SpamAssassin version (3.3.0, from 2009), more than 15% were better detected by version 3.2.0 and 10% by each of the older versions. The main idea that those results indicates to us is that, in any given period, spams from different "generations" are present. Although there is a general trend in which new strategies are created and older strategies are abandoned (as pointed out by Figure 2), there are groups of spams which are at different stages of development. Some are produced by the early adopters of the most recent spam construction techniques, while others still keep older techniques, either because they do not know the new techniques (i.e., they rely on old bulk mail software) or they still believe on their effectiveness. Those "outdated spammers" generate spams which match more rules in a filter which was released *earlier*. As we discuss later on this paper, this is an evidence of the multiple-enemy adversarial scenario of spam fighting.

Another interesting aspect that can be observed at close inspection is that, except for the older release (which has no release earlier than itself to be compared against), all releases hit their peak of effectiveness for the spam from the year *before* their release (e.g., release 2.6.4, from july 2003, performs best in 2002; release 3.2.0, from january 2007, hits its peak in 2006). That seems a clear indication that a certain version is developed based on the spam seen so far and, as soon as it is released, spammers start to adapt and drop the tactics used until then for new ones.

Specifically in relation to release 3.2.0, it was the best for more than 65% of the spam observed in 2006, just before its release, but that had dropped to less than 30% in the year of its release. On the other hand, we see that the ruleset used in release 3.3.0 would be the best even for a few years before, except exactly for 2006. That suggests, first, that the ruleset in 3.3.0 was a substantial improvement over the two previous releases; second, that the spam observed in 2006 had some peculiarities that were quickly abandoned when 3.2.0 was released. Nevertheless, some of it has been present ever since, and the increase in the fraction of spam in 2010 that would be best detected by that release suggests that spammers may have realized that and are resorting to the same tactic again, since it is not well detected by the newer release. In the next Section we study this kind of effect in greater detail.

4.2 Effect of Filter Detection on Strategy Adoption

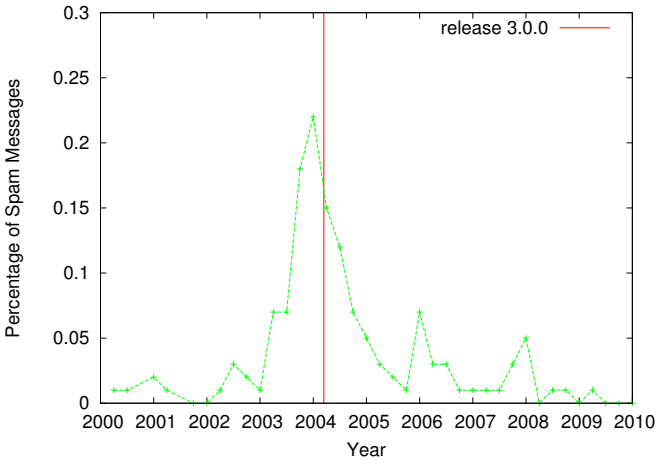

Figure 4: Evolution of HTML-based obfuscation techniques

In this section we investigate the correlation between the rules that compose a filter release and the changes in the spamming strategies. We start by considering the analysis of Pu and Webb [15]. The authors selected some rules from SpamAssassin release 3.1.0 (from 2004) and analyzed how the adoption rate of the strategies addressed by those rules by spammers varied over time. We chose some of the rules considered by them to see if we could explain the behavior they observed by incorporating the evolution of releases over time in the analysis.

One of the spam generation techniques analyzed by Pu and Webb was the rise of HTML-based obfuscation techniques in 2003, and the slow decrease in the use of such strategy after that. Figure 4 shows that clearly as the percentage of spam messages from each year that were matched against the rules that detect such obfuscation.

The authors argued that filters were able to detect those HTML obfuscation techniques, what lead to the slow drop in the use of the strategy since the end of 2003. We found that HTML_OBFUSCATE rules first appeared on SpamAssassin on version 3.0.0, released on 2004 (vertical lin in Figure 4). The analysis of the evolution of releases, then, matches the authors' hypothesis that this strategy was abandoned by spammers due the effective detection by spam filters.

We selected another construction technique analyzed by Pu and Webb, the use of illegal characters on the subject header, and observed that this strategy, although it had an initial drop, actually *increased* after being identified by SpamAssassin 2.6.4, as shown on Figure 5. This reinforce the observation from Pu and Weber that this construction technique has, in fact, succeded despite of filters' actions and that the adoption of this strategy may be conditioned to an environmental condition that might explain why the adoption rate of this strategy have dropped only after 2008.

We generalized this analysis by investigating how the popularity of a spam construction technique changed as it started being detected by spam filters. Figures 6, 7, 8 and 9 show the relative frequency of the obfuscation techniques adopted by spammers *before* they started being detected by each filter (on X axis) and the absolute change in this percentage *after* the inclusion of the rule in that SpamAssassin version. Each dot represents a different spam construction technique (matched by a SpamAssassin rule). We can notice that each release leads to a reduction in the frequency of most spamming techniques detected by that release's ruleset. However, some isolated points in each scatter plot indicate rules associated with spammer tactics that did not reduce — or even increased — their presence in spam messages. In particular, several strategies detected by the 3.3.0 release (Figure 9) are still present on spams, due to the fact that the filter was released recently.

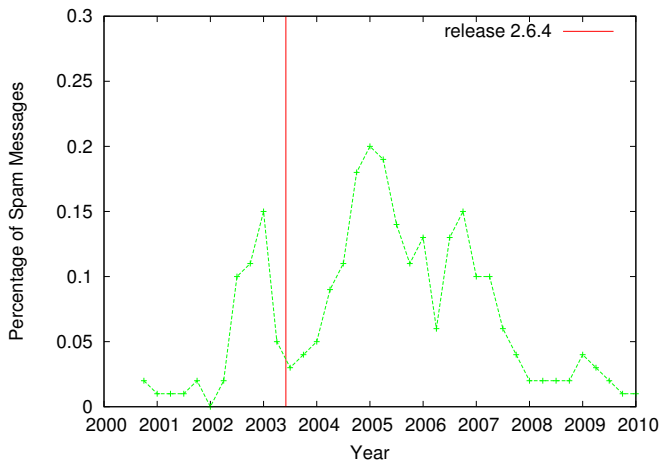

Figure 5: Evolution of SUBJ_ILLEGAL_CHARS spam construction technique

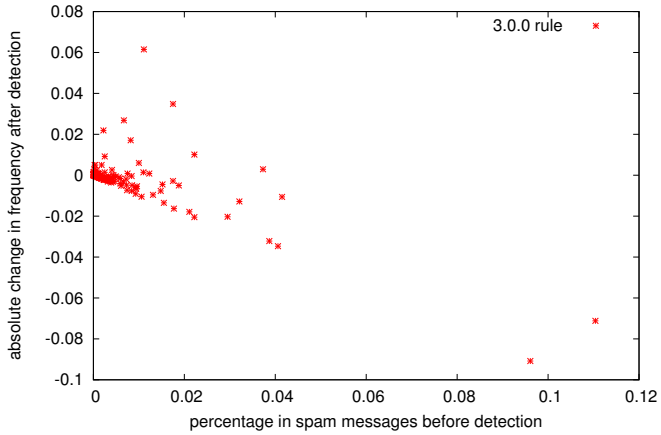

Figure 6: Effect of filter 3.0.0 (2004)

By investigating the spam construction techniques whose frequency among spams were kept high or increased after detection by a given SpamAssassin filter, we can automatically find rules which fall in the *co-existence* category defined by Pu and Webb. Some of those rules are shown in Table 2. When we look into rules whose frequency substantially decreased after being detected by a given SpamAssassin version, *extinction* patterns arise; some examples are shown in Table 3. Notice that the co-existence pattern found by those

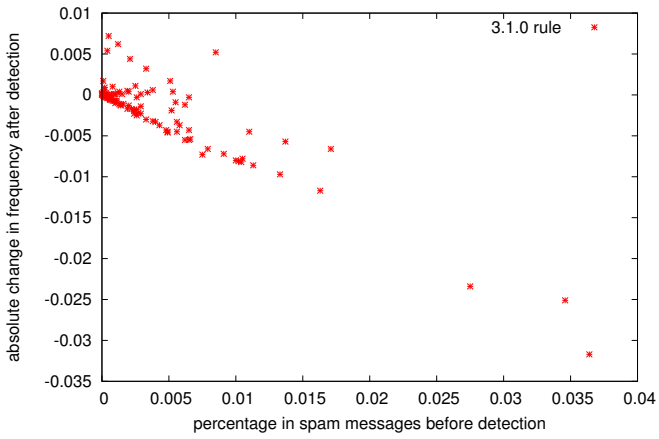

Figure 7: Effect of filter 3.1.0 (2005)

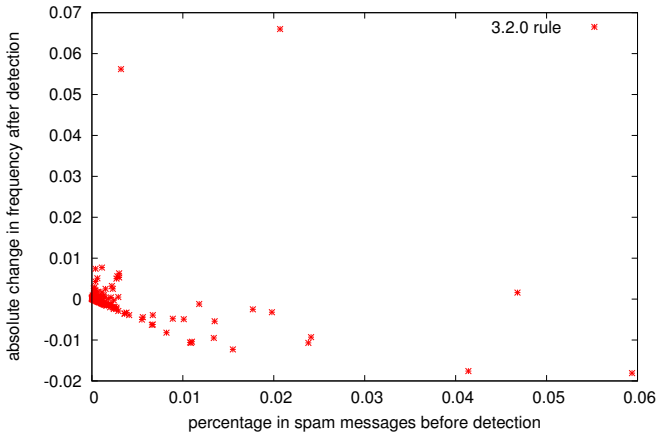

Figure 8: Effect of filter 3.2.0 (2007)

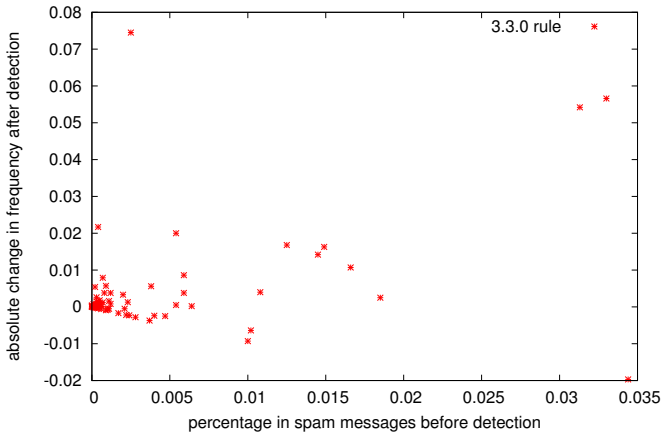

Figure 9: Effect of filter 3.3.0 (2010)

authors for the use of illegal characters was automatically found and is listed in Table 2.

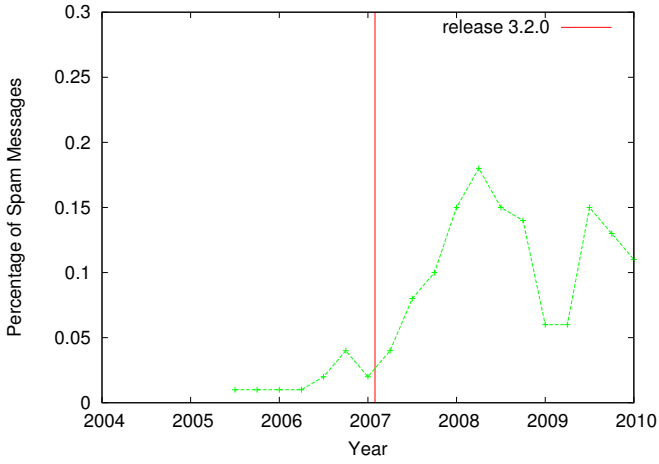

Figure 10: Evolution of STOX_REPLY_TYPE spam construction technique

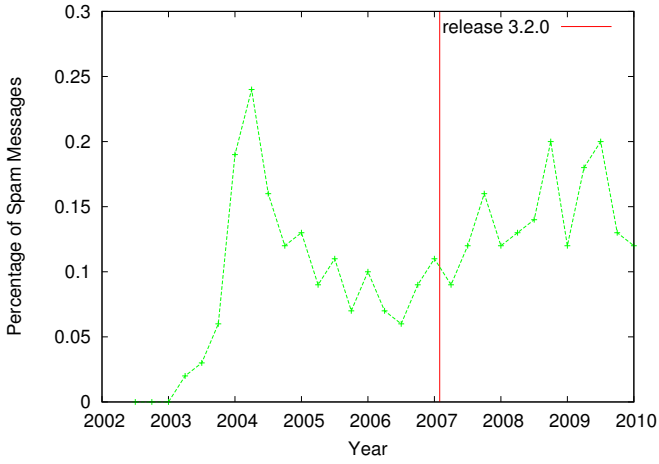

Figure 11: Evolution of FH_HELO_EQ_D_D_D_D spam construction technique

By automatically determining techniques that were more resistant than others to the releases' detection capacity, we can subsidize studies on improving anti-spam mechanisms. We have chosen two of those rules, STOX_REPLY_TYPE and FH_HELO_EQ_D_D_D_D, to investigate their prevalence in spam over time. The evolution of the adoption rate of those strategies are shown on Figures 10 and 11. Both spam construction techniques started being detected by SpamAssassin 3.2.0 (released in 2007) and the frequency of those strategies on spams have not decreased after that year; in some months, the frequency of those rules were even greater than before the release of the filter. In this case, whenever we detect such behavior we may work towards enhancing the detection of such spams.

4.3 Clustering Spam Messages According to Filters' View

So far, we have investigated how different spams are matched by a SpamAssassin release at the ruleset level (Section 4.1) and at the rule level (Section 4.2). Now, we focus on a dif-

Table 2: Spam construction techniques that co-exist with rules

Name	Release	Observed presence (percentage)		Difference
		Before release	After release	
SPAM_PHRASE_00_01	2.4.3	0.02	0.11	0.09
HK_NAME_DRUGS	3.3.0	0.01	0.08	0.07
FH_HELO_EQ_D_D_D_D	3.2.0	0.02	0.09	0.07
HELO_DYNAMIC_IPADDR2	3.0.0	0.01	0.07	0.06
FSL_HELO_NON_FQDN_1	3.3.0	0.03	0.09	0.06
STOX_REPLY_TYPE	3.2.0	0.01	0.06	0.06
HELO_NO_DOMAIN	3.3.0	0.03	0.09	0.05
USER_AGENT_OE	2.4.3	0.01	0.05	0.04
SUBJ_ILLEGAL_CHARS	2.6.3	0.01	0.05	0.04
HELO_DYNAMIC_IPADDR	3.0.0	0.02	0.05	0.03

Table 3: spam construction techniques that become extinct

Name	Release	Observed presence (percentage)		Difference
		Before release	After release	
MSGID_OUTLOOK_INVALID	2.6.4	0.23	0.01	-0.22
INVALID_DATE	2.4.3	0.23	0.04	-0.19
MSGID_SPAM_ZEROES	2.6.4	0.15	0.00	-0.15
FROM_ENDS_IN_NUMS	2.4.3	0.19	0.06	-0.14
INVALID_MSGID	2.4.3	0.18	0.05	-0.13
EXCUSE_3	2.4.3	0.13	0.00	-0.13
SUBJ_ALL_CAPS	2.4.3	0.13	0.01	-0.12
REMOVE_SUBJ	2.4.3	0.11	0.00	-0.11
MAILTO_TO_REMOVE	2.4.3	0.11	0.00	-0.10
MIME_HTML_ONLY	2.6.3	0.28	0.18	-0.10

ferent question: how the same spam message is handled by different rulesets?

Our approach was to identify groups of spams that were systematically treated in a similar way by all 6 releases. For each spam message, we built a tuple with 6 binary values that indicate whether the message has been detected as a spam or not by the ruleset from release i (using the default threshold 5.0[2]). For example, vector (0, 0, 1, 1, 1, 1) represents a message that has not been identified as spam by the 2 older releases (2.4.3 and 2.6.4) but has been detected by the newer ones (3.0.0, 3.1.0, 3.2.0 and 3.3.0).

We applied the clustering algorithm XMeans [5], an extended version of the distance-based algorithm KMeans which automatically determines the best number of clusters. We chose XMeans because it is suitable for low-dimensional numerical data [19]. We found four clusters, and show their details in Table 4.

Table 4: Clusters of Spam Messages

Cluster	% msgs	6-filter vector
1	12%	(0.18, 0.20, 0.01, 0.02, 0.00, 1.00)
2	11%	(0.17, 0.15, 0.02, 0.02, 0.33, 0.00)
3	41%	(0.57, 0.79, 0.83, 0.88, 0.98, 0.97)
4	36%	(0.07, 0.06, 0.02, 0.22, 1.00, 0.99)

Each cluster defines distinct classes of spam, determined by the different ways they are viewed by the rulesets. Next, we interpret each of the clusters we found.

- **Cluster 1**: this cluster groups messages that have successfuly beaten all versions of the SpamAssassin ruleset, except the latest version, 3.3.0.
- **Cluster 2**: messages from this cluster have created difficulties to all SpamAssassin rulesets. The best ruleset for this group, 3.2.0, could detect no more than one third of the messages as spams. Those messages adopt strategies that consistently beat the rulesets, and thus deserve further investigation.
- **Cluster 3**: this cluster represents spams that exhibit high detection rates for all releases, regardless of the ruleset age. We hypothesize those messages consist in "classical" spams, i.e., spams which employ widely known spam construction techniques such as mentioning words like "Viagra" and "Pill" without any obfuscation.
- **Cluster 4**: spams from this cluster were weakly recognized by old filters (no more than 7% of those messages were recognized as spams by SpamAssassin versions 2.4.3 to 3.0.0), but they were very accurately identified by the two newer rulesets, releases 3.2.0 and 3.3.0, which identified more than 99% of those messages as spams.

Clusters 1 and 4, given their characteristic of being identified by some clusters and not others, are most likely associated with the forces that caused the rulesets to evolve to

[2]Although SpamAssassin sets this threshold while considering other spam evidences (e.g., blacklist information), we think this approximation do not affect our results qualitatively.

a new release once they became prevalent. To confirm that, we must consider the evolution of clusters over time.

We then look at the amount of spam belonging to each cluster in each year, as depicted in Figure 12, some patterns become clear. Although in every period messages from each of the four clusters were present, the ratio of messages on each cluster over time vary a lot for some clusters. Spams that were not easily detected by any of the releases (cluster 2) concentrate at the early years, before the first release considered in the analysis. Probably, they were well detected by a previous release, which led to their near extinction before 2002. They are not easily detected by the releases we used because they are not considered significant for the current rulesets, since their occurrence has been low. Nevertheless, our analysis shows a slight increase in recent years, which may suggest some attention is needed for that group, in case they begin to grow again.

The ammount of "classical" spam (cluster 3) grew with the release of version 2.4.3, and was prevalent until version 3.0.0 was released, when it reduced its presence, although it has not gone completely: about 20% of current spam is in that group. Its growth prior to 2002 is clearly associated with the reduced presence of cluster 2 messages, just discussed. The prevalent kind of spam changed again around 2004, to those messages in cluster 4. Their rise was countered by the development of version 3.2, the first one to detect them properly. Nevertheless, their share of the total spam traffic is still high at almost 50% despite their detection, what leads us to say that they are the "new classical" spam.

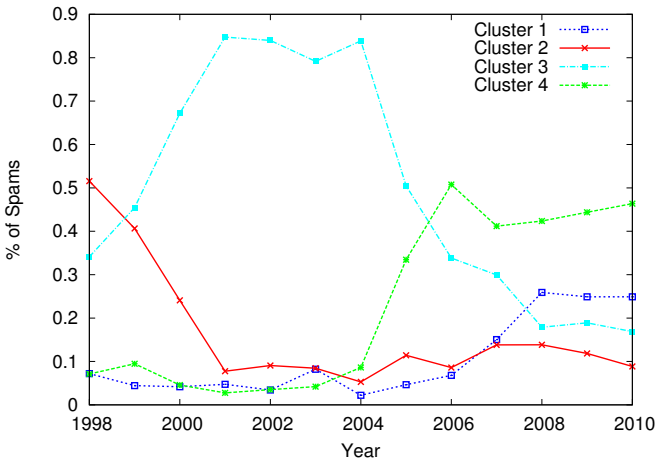

Figure 12: Variability in size of clusters over time

Finally, messages in cluster 1 accounted for less than 10% of all messages each year until 2007, when they started to grow in volume. Clearly that growth lead to the development of the rules in release 3.3.0 that started to detect them in 2009. From that behavior and that of clusters 2 and 4, we can say that the development of rules for SpamAssassin releases focus mainly on kinds of spam that account for at least 10% of the overall volume: rules for cluster 2 messages became absent from all releases after that group dropped below 10% in 2001 and route for the two other clusters first appeared in the first release after the year when their presence grew to more than 10%.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we describe spamming evolution patterns based on the cross view of the mutual evolution of spammers and spam filters. We used 6 SpamAssassin versions released on different years (2002, 2003, 2004, 2005, 2007 and 2009), evaluated them on 2.2 million spams from Spam Archive and analyzed the interactions between spammers and filters over time.

We showed that old filters can be useful to provide a general picture of the evolutionary process spam has been through over the last 12 years. By computing the prevalence of spam construction techniques before and after they started being detected by filters, we automatically found techniques that were not detected by any filter and thus may deserve more attention from the anti-spam community.

Our approach also may serve to generate a dataset to be tested against trend and novelty detection algorithms: filters ahead of their time detect new trends that should be detected by algorithms looking for spam evolution patterns.

Current studies on the spam arms race have modeled the problem as a single-enemy scenario and they focus on detecting future spam [4, 12]; our results suggest that in the real scenario there are multiple adversaries to be handled, as [4] already pointed out. The assumption that spam is generated by a single evolving entity may lead the spam filter to evolve to detect new types of spam, and, in a real scenario, end up allowing old spams to evade the new filter more easily.

Therefore, combining old and new filters (for example, using ensemble classifiers [6]) may be an interesting strategy to deal with the diversity of spams. Our next step is to assess the applicability of a combination of old and recent spam filters to improve spam filter detection, using the observation that a spam stream observed in any period is composed of spams which have been build using strategies created in different periods.

Acknowledgments

This work was partially supported by UOL (`www.uol.com.br`) through its Bolsa Pesquisa Program, Process Number 20100214193900, by NIC.br, CNPq, CAPES, FAPEMIG, and FINEP.

6. REFERENCES

[1] Biggio, B., Fumera, G., and Roli, F. *Evade Hard Multiple Classifier Systems*, vol. 245. Springer Berlin / Heidelberg, 2008, pp. 15–38.

[2] Calais, P. H., Guedes, D., Wagner Meira, J., Hoepers, C., Chaves, M. H. P. C., and Steding-Jessen, K. Spamming chains: A new way of understanding spammer behavior. In *Proceedings of the 6th Conference on e-mail and anti-spam (CEAS)* (Mountain View, CA, 2009).

[3] Chinavle, D., Kolari, P., Oates, T., and Finin, T. Ensembles in adversarial classification for spam. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management* (New York, NY, USA, 2009), ACM, pp. 2015–2018.

[4] Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D. Adversarial classification. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and*

data mining (New York, NY, USA, 2004), ACM, pp. 99–108.

[5] Dan Pelleg, A. M. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning* (San Francisco, 2000), Morgan Kaufmann, pp. 727–734.

[6] Dietterich, T. G. Ensemble methods in machine learning. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems* (London, UK, 2000), Springer-Verlag, pp. 1–15.

[7] Fawcett, T. "in vivo" spam filtering: a challenge problem for kdd. *SIGKDD Explor. Newsl. 5*, 2 (2003), 140–148.

[8] Goodman, J., Cormack, G. V., and Heckerman, D. Spam and the ongoing battle for the inbox. *Commun. ACM 50*, 2 (2007), 24–33.

[9] Graham-Cumming, J. The spammers' compendium. `http://www.jgc.org/tsc.html`, 2010.

[10] Guenter, B. Spam Archive, 2010. `http://untroubled.org/spam/`.

[11] IronPort. Internet security trends. `http://www.ironport.com/securitytrends/`, 2008.

[12] Kantarcioglu, M. A game theoretical framework for adversarial learning.

[13] Kapersky. Spam evolution 2006: Executive summary. `http://www.kaspersky.com/spam_evolution_2006_summary`, 2006.

[14] McCarty, B. Botnets: Big and bigger. *IEEE Security and Privacy 1*, 4 (2003), 87–90.

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

[16] SpamAssassin, 2010. `http://spamassassin.apache.org`.

[17] Sullivan, T. The myth of spam volatility, 2004. `http://www.qaqd.com/research/mit04sum.html`.

[18] Symantec. The state of spam – home of the monthly report. `http://www.symantec.com/business/theme.jsp?themeid=state_of_spam`, 2010.

[19] Tan, P., Steinbach, M., and Kumar, V. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., 2005.