Measuring the Role of Greylisting and Nolisting in Fighting Spam

Fabio Pagani
Eurecom
pagani@eurecom.fr

Matteo De Astis
Università degli Studi di Milano
matteo.deastis@studenti.unimi.it

Mariano Graziano
Eurecom and Cisco Systems, Inc.
magrazia@cisco.com

Andrea Lanzi
Università degli Studi di Milano
andrea.lanzi@unimi.it

Davide Balzarotti
Eurecom
davide.balzarotti@eurecom.fr

***Abstract*—Spam has been largely studied in the past years from different perspectives but, unfortunately, it is still an open problem and a lucrative and active business for criminals and bot herders. While several countermeasures have been proposed and deployed in the past decade, their impact and effectiveness is not always clear. In particular, on top of the most common content- and sender-based anti-spam techniques, two minor approaches are popular among system administrators to cope with this annoying problem: *greylisting* and *nolisting*. These techniques exploit known features of the Simple Mail Transfer Protocol (SMTP) protocol that are not often respected by spam bots. This assumption makes these two countermeasures really simple to adopt and, at least in theory, quite effective.**

In this paper we present the first comprehensive study of nolisting and greylisting, in which we analyze these spam countermeasures from different perspectives. First, we measure their world-wide deployment and provide insights from their distribution. Second, we measure their effectiveness against a real dataset of malware samples responsible to generate over 70% of the global spam traffic. Finally, we measure the impact of these two defensive mechanisms on the delivery of normal emails.

Our study provides a unique and valuable perspective on two of the most innovative and atypical anti-spam systems. Our findings may guide system administrators and security experts to better assess their anti-spam infrastructure and shed some light on myths about *greylisting* and *nolisting*.

***Keywords*—Spam, Greylisting, Nolisting, Botnet**

I. Introduction

According to the annual Symantec Threat Report [12] the overall number of spam messages decreased by 3% in 2014, but spam still accounted for 60% of the global email traffic on the Internet. This recent decrease in the spam volume is due to many reasons, including the increasing effectiveness of the numerous solutions that exist to either prevent or detect spam messages. These solutions can be broadly grouped in two major families: sender-based filtering and content-based filtering. The two approaches are applied at different stages (the pre-acceptance and post-acceptance tests, respectively) by the receiving SMTP server. Pre-acceptance tests typically require to keep some senders status and/or retrieving special Domain Name System (DNS) records in order to verify an existing trust relationships or a certain reputation before accepting an email. Post-acceptance tests involve instead a number of local tests, such as the use of a classifier on the email body.

While the most popular techniques have been widely studied in previous research [3]–[5], [11], [18], [22], [23], [28], [29], [35], [36], [38], the effectiveness of other minor approaches have never been properly measured. In particular, in this paper we focus on two of them, *nolisting* and *greylisting*, to try to shed light on their practical advantages and disadvantages. Nolisting and greylisting are both based on the assumption that nowadays most of the spam is sent by machines infected by malware. As a consequence, the malicious executable responsible to send spam do not use full-fledged Mail Transfer Agents (MTA), but small routines that often implement part of the message delivery protocol in custom ways – not compliant with the RFCs. The existence of these SMTP "dialects" has been experimentally confirmed in 2012 by Stringhini et al. [34], who noticed that details about the protocol can also be used to fingerprint botnets and tell them apart from benign MTA agents.

The main idea of nolisting and greylisting is that the lack of compliance to standards can be used to prevent malware from delivering the spam messages in the first place. Nolisting does that by mimicking a malfunctioning primary mail server, while greylisting achieve a similar result by temporarily rejecting emails coming from unknown senders. In the first case, the defender hopes that the malware is not designed to retrieve and contact the secondary email server, while in the second case the defender hopes that the malware would simply move to the next target without retrying to deliver the message a second time. Because of this behavior that privileges volume over the accurate delivery of single messages, malware samples that fall in these categories are often called *fire-and-forget* spammers.

Despite the fact that today nolisting and greylisting are already in use to protect thousands of email servers, their core assumptions have never been confirmed by real experiments.

In fact, to the best of our knowledge there is no previous study on the effectiveness of nolisting, and greylisting has been the focus of some previous works [31]–[33] which only measured the impact of the parameters and thresholds on the number of messages blocked by greylisting services deployed in real email servers. While this is useful, the *fraction* of spam messages that is blocked by greylisting is still not clear. To better answer this question, in this paper we experiment with a number of real malware samples that in 2014 were responsible for the delivery of over 93% of the spam generated from botnets, which in turn accounts for over 70% of the worldwide spam traffic. This gives us a privileged view to understand whether these malware families are able to cope with nolisting and greylisting services.

Our study clearly shows that malware is indeed adapting to these techniques, but not as quickly and not as effectively as many people say. Therefore, in 2015 both nolisting and greylisting can still play an important role in the fight against spam.

To summarize, this paper makes the following contributions:

- We present the first study of nolisting, measuring both its effectiveness to protect against spam and its world-wide deployment.
- We test the efficacy of nolisting and greylisting techniques by using binaries belonging to the four malware families which are known to produce over 93% of the botnet-generated spam traffic.
- We study how the greylisting threshold affects the delivery of normal emails and also how botnets react to the different greylisting thresholds in terms of number of attempts and time to delivery.
- Our experimental results enable network administrators to make a more systematic and informed decision about which technique to adopt for their email infrastructure to reduce the impact of spam.

The rest of the paper is structured as follows: In Section II we introduce in more details the nolisting and greylisting technologies, presenting the open questions and the current debate on their effectiveness. In Section III we detail the experimental setup and the tests performed in our study. Section IV and Section V report the results for the nolisting and greylisting experiments. Section VI presents some overall summary and discussion. Finally, Section VII discusses related studies, and Section VIII concludes the paper.

II. Nolisting & Greylisting

In this section we introduce in more details the two techniques we study in this paper. For each of them we present the approach, the way in which the defense is typically deployed on a mail server, and the main criticisms about its advantages and disadvantages.

Nolisting

Nolisting is a very simple anti-spam mechanism that consists of registering to the DNS a non-existent primary mail server (i.e., MX record) and a full-functioning secondary server. The primary record still needs to point to an address with a proper A record, so the common suggestion is to use a real machine that has port 25 closed. Theoretically, this configuration should be indistinguishable from the case in which the primary server is malfunctioning or it is temporarily offline for maintenance. According to the RFC 5321 [21] each email client MUST try to contact all Mail eXchanger (MX) addresses in order of priority (see Figure 1 for an example of the protocol). As a result, the main advantage of nolisting is that it should not affect the delivery of benign emails, and it should not introduce any delay in the delivery of the messages. However, the authors of nolisting noticed that a large percentage of fire-and-forget spam sources (which follow by definition a very simple logic) is not able to deal with this case and would therefore fail to deliver their messages.

Criticisms: Nolisting is a controversial technique. The advocates say that it is a very simple and yet effective solution to reduce spam, as most of the botnets responsible for it are not RFC compliant and they limit their attempts to the main MX server.

Other people have instead a more skeptical position, objecting that malware authors already switched their tools to contact directly the secondary MX server (skipping the first altogether) and therefore the beneficial effects of nolisting are nowadays very limited. On top of that, it is possible (even though extremely rare) that this technique can prevent some legitimate email client (especially small programs used to send automated notifications) from delivering legitimate messages. These people conclude that nolisting is practically useless, and therefore should not be used to protect email servers. As it often happens when there is no real data to support either claims, the final decision is left to folks wisdom or the "feelings" of network administrators.

Open Questions: There are a number of open questions we want to answer with our study. First of all, we want to measure *How widespread the use of nolisting is on the Internet*. We will then move to the fundamental question: *Is it true that modern malware is not affected by this technique?* If this is the case, *is it because malware samples only focus on the secondary email server or is it because they can properly contact all of them based on their priority?*

Greylisting

Greylisting is more sophisticated and more popular than nolisting, and is fully implemented in the SMTP server without requiring any modification to the DNS records. A server protected by greylisting accepts incoming messages delivered from known senders – based on the triplet <*sender_address, sender_IP, recipient*>. Whenever a triplet is unknown, the server answers with an error, asking the client to try again later. While RFC-compliant clients would attempt again at regular time intervals, fire-and-forget software do not always support this feature. The server keeps sending back the same error for a configurable time interval, after which a new delivery attempt results in accepting the email and adding the sender to a white-list for future connections.

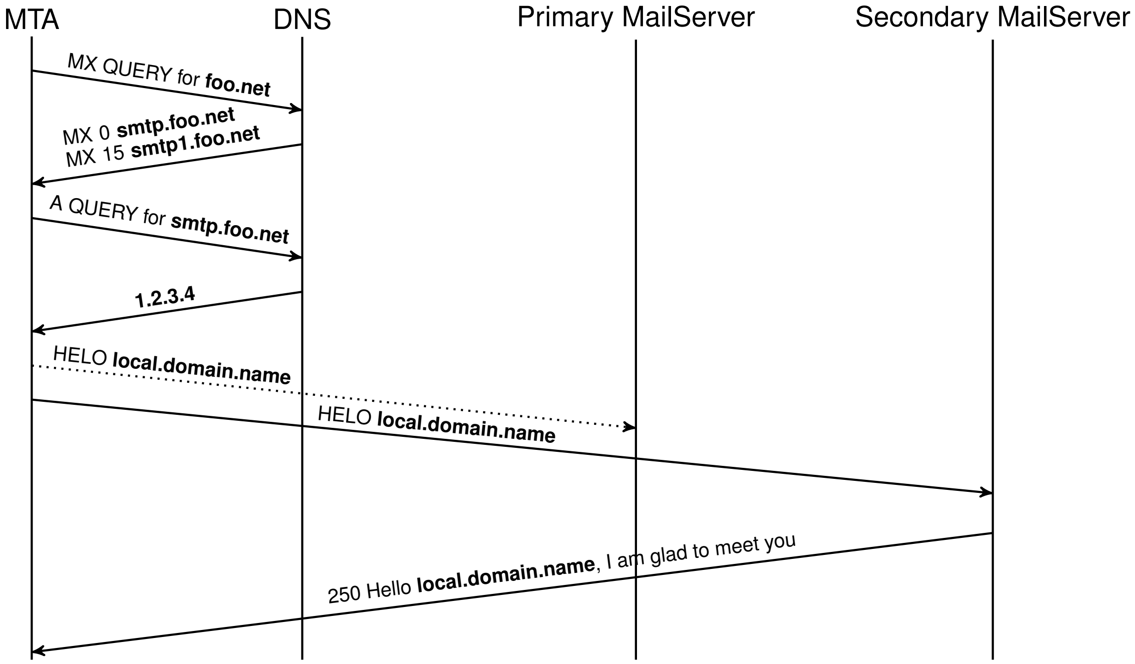

MTA
DNS
Primary MailServer
Secondary MailServer
MX QUERY for **foo.net**
MX 0 **smtp.foo.net**
MX 15 **smtp1.foo.net**
A QUERY for **smtp.foo.net**
1.2.3.4
HELO **local.domain.name**
HELO **local.domain.name**
250 Hello **local.domain.name**, I am glad to meet you

Fig. 1: DNS Communication in presence of Nolisting

Criticisms: Greylisting has two obvious problems. First, it introduces a noticeable delay in the delivery of certain legitimate emails. Second, it only works if the client retries to delivery the message always with the same IP address. This is a reasonable assumption for normal email servers, but it is not the case for large and redundant infrastructures (such as the one used by popular web-mail providers). To solve this problem, greylisting implementations typically need to resort to white-list popular email providers.

In exchange for this limitation, critics say that greylisting offers limited protection, due to two main reasons. First, recent malware families are sophisticated enough to retry the delivery of messages. Second, even when they do not support this feature, it is likely that the same address will be used for sending several spam messages, with the side-effect that the spammer would be eventually whitelisted by mistake (remember that the message itself is irrelevant, and only the sender and the recipient needs to be the same).

At this point, the supporters of greylisting rebut that, even when ineffective, greylisting would still be useful because the delay introduced in the delivery of spam messages can be enough for the sender (especially if it is a mass-spammer) to be detected and added into popular spammer blacklists - therefore still helping to prevent the final delivery of the spam message.

Open Questions: Unlike nolisting, measuring the adoption on a large scale of greylisting is very difficult. In fact, email servers are typically configured to refuse messages for non-existing recipients **before** applying greylisting. Therefore, the only way to test a particular server is to know in advance a valid address of one of its users (and, unfortunately, common addresses such as `postmaster` are not covered by the greylisting protection). Even if we cannot measure its worldwide use, there are still many important questions that we can answer. In particular, *Is greylisting still effective to stop spam?* and *How does the choice of the greylisting threshold affect the delivery of normal emails?*. *Is it true that greylisting causes more harm than good?* and finally, according to our experiments *is there a way to use greylisting to maximize its advantages and reduce its negative impact?*

III. Datasets and Experimental Setup

In order to measure the efficacy of the two main anti-spamming techniques analyzed in our study we collected four different datasets. The first two of them are assembled from the zmap *scans.io* project [39] and are used to evaluate the prevalence of nolisting servers. In particular, we used the *DNS Records (ANY)* dataset and the *Daily Full IPv4 SMTP Banner Grab and StartTLS* zmap results. The first dataset contains a DNS lookup for all the domain names as a results of other types of scans, such as reverse DNS scan or HTTP requests. This dataset contains 135 millions of resolved domains. As reported in the official documentation, the data was collected by issuing a DNS query of type `ALL`, even though for our purposes we limited the analysis to the `A` and `MX` records (i.e., mail server query). Since the original dataset contained several `MX` records that were not properly resolved we implemented a parallel scanner to resolve the missing entries. These missing entries can be found in case a DNS query is issued to resolve a mail server IP address (i.e., a MX record) and the reply for this query only contains the domain name of the mail server but not its IP address. Consequently the DNS needs to look up again the mail server

Malware Family	Percentage of Botnet Spam in 2014 [12]	Number of Samples
Cutwail	46.90%	3
Kelihos	36.33%	6
Darkmailer	7.21%	1
Darkmailer(v3)	2.58%	1
Total Botnet Spam	93.02%	11
Total Global Spam	70.69%	

TABLE I: Malware samples used in our experiments

domain for obtaining the final IP address. The second dataset for the nolisting analysis contains instead the list of all IPv4 hosts which responded to a `SYN` packet on port 25. We labeled those IP hosts as SMTP servers.

For the second set of experiments on the greylisting technique we built a different dataset. This dataset is used to test the impact of the threshold of the waiting time set by the greylisting method. In particular, this dataset contains data collected for over four months (from January to April 2015) and represents the anonymized log entries of the mail server of the Computer Science department of Univeristà degli Studi di Milano, where greylisting protection is in use. More in details, this dataset contains, for each greylisted message, the time of each attempted delivery from the client. The time between consecutive attempts depends on the particular configuration of the client MTA software. The values contained in this dataset depend on the threshold time of Greylisting mechanism. In our case we considered the threshold of the mail server of our university, that was set to 300 seconds.

The last dataset is used to measure the effectiveness of greylisting and nolisting and contains a set of malware samples, in form of executable binaries, that belong to the top malware families responsible for the generation of the majority of the SPAM on the Internet. A similar distribution was also reported in 2009 by John et al. [20] in their study of spamming botnets. In order to collect the malware samples we proceed in the following way. First of all, based on the regular reports published by antivirus companies (such as the Symantec Threat Report 2014 [12]), we identified the top four families that contribute to over the 90% of the botnet spam traffic. Since 76% of the world spam was sent from botnets, the chosen families account for more than the 70% of the global spam sent in 2014.

Afterward, we collected the hashes of the malware samples which belong to those four families from security public reports and we downloaded the corresponding executable binaries from several public repositories, including VirusTotal [7], VirusShare [6] and malwr [8]. When possible, we retrieved several samples of the same family, to account for possible variations or parallel versions that were active during the same time period.

After the collection phase, we performed a malware analysis phase in which each sample was analyzed in an instrumented environment composed by two VirtualBox virtual machines: a *Mail Server* VM and an *Infected* VM. The first machine run Postfix (and Postgrey for the greylisting tests), on top of the latest `Debian` stable, while the second machine run a vanilla `Windows 7` installation equipped with few custom modifications to block the most common anti-virtualization techniques adopted by current malware [9], [26]. All the SMTP traffic produced by each sample was routed from the infected VM to the Mail Server VM by intercepting all the DNS MX requests and replying with some bogus MX entries which resolved to our virtual machine.

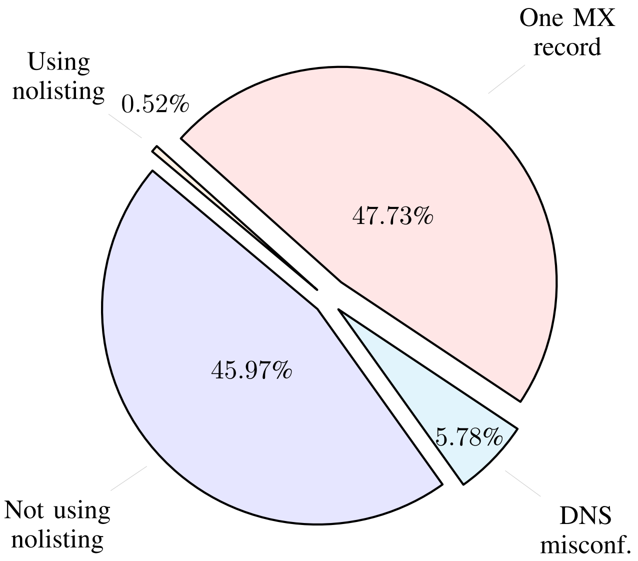


Fig. 2: Nolisting mail server statistics

Each sample was run in a controlled environment for 30 minutes, a sufficient time that permitted us to monitor the behavior of the malware sample and observe the generated spam traffic. After this test, we populated our dataset with only those samples that generated spam traffic in the assigned analysis period. In Table I we report the four families that we were able to analyze, and the number of samples we collected for each family. Even though the number of samples collected by our analysis is limited, it is important to note that according to the Symantec report the spam-related functionality of each sample in our dataset is representative for the whole family the malware belong to.

IV. Nolisting

We start our analysis of nolisting by measuring its global adoption, i.e., how many email servers on the Internet use nolisting as a spam protection mechanism. We then move to answer the most important question: *"Is it really effective to stop spam?"*.

A. Worldwide Adoption of Nolisting

First of all, it is important to note that while we often refer to an individual *server* protected by nolisting, the protection is actually applied at the DNS level, and therefore at the domain granularity. Therefore, as described in Section III, in order to estimate the number of domains which adopt nolisting, we combined two datasets obtained by using the zmap tool [13]: the *DNS Records (ANY)* (hereinafter simply

DNS scan) and the *IPv4 SMTP Banner Grab* (hereinafter SMTP scan).

Evaluating whether a domain implements nolisting is a three step process. First of all, we retrieve from the DNS scan all the `MX` records associated with all existing domains and check their correctness. We then resolve the IP address of each record, ordered by their priority. Finally, we look-up the IPs in the SMTP scan dataset to verify whether the IP in question was accepting `SMTP` connections at the time the scan was performed. If the primary `MX` server is not present in our list and the secondary is, the domain is a possible candidate for implementing nolisting. However, it is also possible that the primary email server was simply malfunctioning at the time the dataset was collected. To address this possibility and minimize the errors in our results, we repeated the same measurement twice, two months apart, on February 28 and April 25, 2015. If one domain had the primary email server operational in *at least one* of the two datasets, we concluded that it was not using nolisting. If the primary server was not responding in both cases but the secondary did, we assumed that the domain was protected by nolisting (or it had a persistent problem with its primary record, which is in practice equivalent to nolisting).

Overall, our approach covered 42.6 million email servers, which resolved to over 49.2 million non-unique IP addresses. As one would expect, the difference between the two experiments was very small, with a change of only 0.01% in the number of domains adopting nolisting.

The final results are summarized in Figure 2. The pie chart shows that nearly half (47.8%) of the domains are configured with only one MX record and the other half (45.9%) used more than one record but did not use nolisting. For over 5% of the domains we encountered a DNS misconfiguration, e.g., we were not able resolve any MX record. Finally, only 0.52% of the domains had (in both scans) a non-responding email server with the highest priority and a responding one as a secondary email server. While this percentage may seem small, it still accounts for over 133 thousand domains. Moreover, these domains are not necessarily associated to small companies. In fact, by crosschecking our results with the domain popularity reported by Alexa [1], we found that nolisting is adopted by one domain in the top-15 worldwide ranking, by two in the top-500 and by other two in the top-1000.

B. Impact on Spam Delivery

With few very large companies and hundred of thousands of other installations worldwide, we can certainly conclude that the adoption of nolisting is not negligible. Therefore, with no previous studies on this subject, it becomes extremely important to assess its effectiveness in reducing spam, and to study whether malware writers adapted to this technique.

As explained in Section III, we conducted a number of experiments using the four families that according to Symantec were responsible for the vast majority of spam messages in 2014. All the SMTP traffic generated by the samples was redirected towards our server, whose DNS server was configured for this experiment to use *nolisting*. In particular, for every `MX` request it provided an answer containing two records with different priorities. The primary record resolved to a machine without a `SMTP` server, while the secondary record pointed to a working `SMTP` machine.

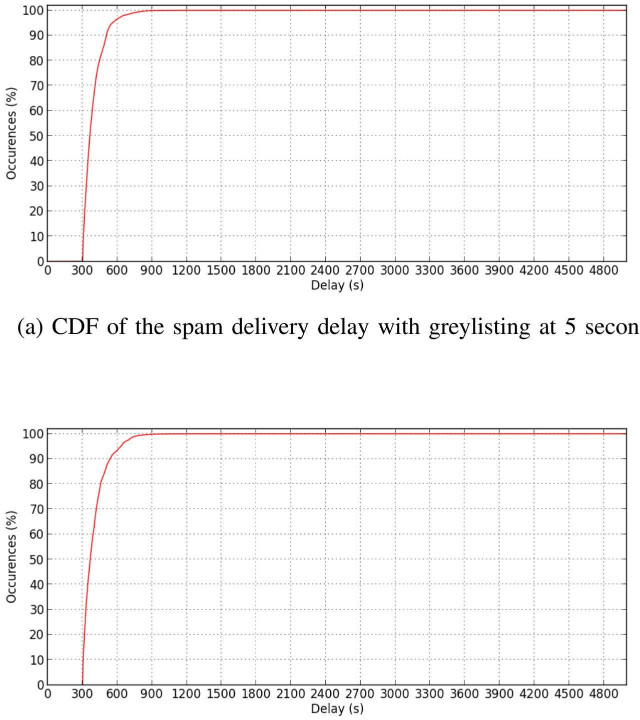
SAMPLE	GRAYLISTING	NOLISTING
Cutwail:		
sample1	✓	✗
sample2	✓	✗
sample3	✓	✗
Kelihos:		
sample1	✗	✓
sample2	✗	✓
sample3	✗	✓
sample4	✗	✓
sample5	✗	✓
sample6	✗	✓
Darkmailer:		
sample1	✓	✗
Darkmailer(v3):		
sample1	✓	✗

TABLE II: Effect of nolisting and greylisting on popular malware families. A ✓ sign indicates that the technique was effective in preventing the spam messages from being delivered. A ✗ sign means instead that the technique was ineffective against that malware family.

Each malware sample was executed in isolation, and after each execution we inspected the network traces and the email server logs to correctly identify the behavior of the family under analysis. As a first result of our experiments we noticed that all malware samples belonging to the same family shared the same behavior with respect to nolisting. In other words, we did not encounter any variations inside the same family that suggested that the authors modified the email message delivery implementation of the malware.

The results of the experiments are summarized in Table II. The table shows that nolisting is effective against one family (`Kelihos`) but it is currently ineffective against `Cutwail` and `Darkmailer`. Since `Kelihos` alone is responsible for over 36% of the botnet-generated spam messages on the Internet, this already shows that in 2015 nolisting still has a positive impact in reducing spam. However, knowing whether a technique works is not sufficient if we do not understand the reason. In fact, it is possible to classify a spam bot with respect to its `MX` behavior [2] in four categories:

- **RFC Compliant**: the malware sample targets all email servers following their priority order from the higher to the lowest, according to the RFC 5321 [21].
- **Primary Only**: the malware sample targets only the mail server with the highest priority (this is the fundamental assumption of nolisting).
- **Secondary Only**: the malware sample targets only the mail server with the lowest priority, skipping the primary server altogether. People who criticize nolisting often say that this was the natural reaction of malware writers to nolisting.



(a) CDF of the spam delivery delay with greylisting at 5 seconds

(b) CDF of the spam delivery delay with greylisting at 300 seconds

Fig. 3: Effect of greylisting on Kelihos

- **All MX**: the malware sample targets all the email servers of the target domain, in a random or systematic order.

By inspecting the communication of the malware sample with our DNS and our email server, we were able to establish to which category each family belongs to. As we already mentioned, `Kelihos` only targets the primary server – thus failing to deliver the spam messages. `Cutwail` is not affected by nolisting because it targets immediately the lowest priority mail server, ignoring the first one. On the contrary, the two `Darkmailer` versions we tried were RFC-compliant, and they contacted the email servers in order of priority.

V. Greylisting

While greylisting is much more popular than nolisting, measuring its precise adoption on the Internet is not possible because it would require to know in advance a valid recipient for each server. For this reason, we focus our analysis on the impact of this technique on the delivery of spam and benign messages.

A. *Impact on Spam Delivery*

Similarly to what we described to measure the impact of nolisting, we run each malware sample in our contained environment – forwarding all the SMTP traffic to a local email server protected by greylisting. Every sample was run three consecutive times, the first using a significantly low greylisting threshold (5 seconds), the second using the default Postgrey threshold of 300 seconds, and finally using a very high threshold set to 21,600 seconds.

The results of the experiments are summarized in Table II. First of all, our tests show that greylisting is still very effective in practice. In fact, it was able to stop `Cutwail` and `Darkmailer` (together responsible for over 43% of the world spam) from delivering any spam message. Unfortunately, `Kelihos` was able to cope with *greylisting*, making this countermeasure ineffective against this particular malware family.

The cumulative distribution of the delivery attempts time of `Kelihos` are presented in Figure 3. The similarity between the two curves clearly shows that the malware is *not* able to take advantage of a shorter greylisting threshold (e.g., by trying to resend more often in a short window time), but instead it is designed to retry again to delivered a message after a minimum delay of 300 seconds – which is, in fact, the default threshold used by popular greylisting software.

The picture is quite different when the greylisting threshold is set at 21600 seconds (i.e., six hours). In this case, it is possible to observe the complete behavior of `Kelihos`, while it attempted to delivery the failed messages multiple times. Figure 4 shows a plot of all the delivery attempts made by the malware: we can clearly identify a number of peaks, such as the one we already mentioned between 300-600 seconds, a second around 5000 seconds, and a third between 80,000 and 90,000 seconds. As shown by the red dots on the right side of the graph, after several attempts `Kelihos` was able to deliver its messages.

To confirm these results, we needed to rule out a possible subtle side-effect of how spam bots could interact with a service implementing greylisting with a very high threshold. Suppose for instance that `Kelihos` only tried to deliver each message three times before abandoning the task. However, one hour later the bot could have received from its bot master a new job to deliver a second spam message, different from the first one but directed again to the same list of recipients. Since greylisting does not keep track of the message itself, the server would consider the incoming connection as an attempt to delivery the same message that had been greylisted before. In this case, the first spam message would have been dropped, but the second (and all the following ones after that) would successfully pass through the spam filter. To rule out this hypothesis, we left few email addresses unprotected (e.g., `postmaster`), allowing those spam messages to be delivered without greylisting. Since all email messages directed to the unprotected email addresses were the same of the ones filtered by the greylisting filter, we can conclude that the there was only one spam task during the entire experiment.

It is now interesting to compare the retransmission behavior of `Kelihos` with the one of benign email transfer agents.

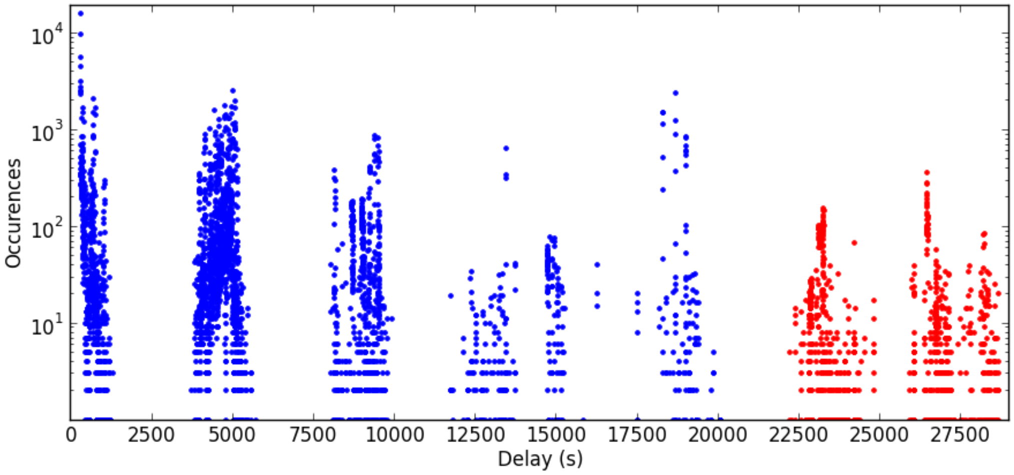


Fig. 4: Retransmission delays of Kelihos with a greylisting threshold of 21600 seconds. In blue the failed attempts (below the threshold) and in red the delay of delivered emails (above the threshold).

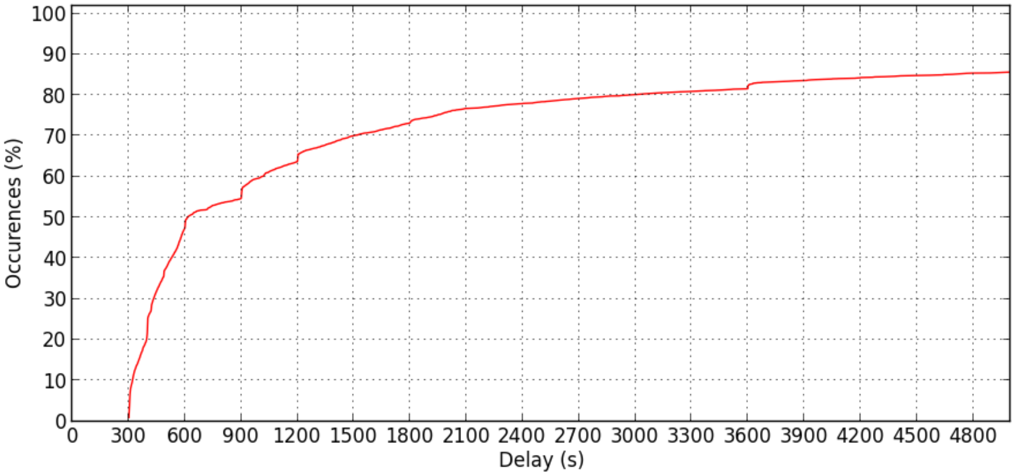


Fig. 5: CDF of the email delivery delay of a real-world mailbox with greylisting at 300 seconds

B. Greylisting on a Real Deployment

The experiments described in the previous section shows that greylisting is still able to block many popular malware families, thus preventing over 50% of the spam to be delivered to the users inboxes. However, the picture would not be completed without looking at a real installation.

Greylisting is currently deployed to protect the mail server of our University department. By looking at the anonymized logs (containing only the timestamps of greylisted attempts) we found two interesting things. First, Figure 5 shows the distribution of the delays introduced in the message delivery of greylisted messages. The cumulative distribution of the delays increases much slower than the curve we observed for malware in Figure 3. This is a surprising, and quite negative, result. In fact, even with greylisting configured on 300 seconds (5 minutes), only half of the messages get delivered in less than 10 minutes. Even worse, some messages are delivered with over 50 minutes of delay, and some even beyond that. This seems to suggest that, as often mentioned by the critics of this technique, the impact of greylisting on benign messages is not negligible.

C. Impact on the Delivery of Benign Emails

Figure 5 showed that, in presence of greylisting configured with a 5-minute threshold, some emails get delivered with very long delays. However, it is hard to tell precisely which fraction of those were legitimate and which were spam or commercial advertisements. Therefore, we decided to verify ourselves the re-transmission delays of popular MTA clients and web-mail providers.

In order to perform our experiments we created an account on the top ten web-mail providers, as reported in Table III and we used them to send an email to a test account on our mail server. For this experiment, we removed the default white-list of Postgrey, since by default it contains all the web-mail providers used in our tests. To measure the reliability of web-mail servers and to collect a sufficient amount of data, we set the greylisting delay to the excessively large value of 6 hours. Table III shows the results of this experiment. More precisely, the table contains five columns showing the web-mail providers chosen for our experiments, the number of unique IP addresses used to deliver the same email message, the number of delivery attempts in the six hours time window, whether or not the web-mail provider was able to deliver the email message, and the timestamps of all performed attempts.

By analyzing the results, we can notice that the retry policy of each service is quite different, often without a clear pattern. In few cases the server clearly used a linear approach, in others it seems that the delay is doubled between each attempt. Moreover, also the total number of attempts changes very substantially between services – ranging from 9 in case of `gmail` to 94 for `hotmail` in the same time window of six hours.

In most cases, the servers tried to deliver the email long enough to successfully go over the 6h greylisting threshold. However, two of them abandoned the task earlier and therefore were not able to deliver the message. Quite surprisingly, `aol.com` stopped its delivery attempts after only 30 minutes. This is strange since the RFC-822 [30] clearly states that *"retries continue until the message is transmitted or the sender gives up; the give-up time generally needs to be at least 4-5 days"*.

The second column of Table III reports another information that is crucial for testing the efficiency of greylisting technique: the number of distinct IP addresses that were used to deliver the same message. Since greylisting authorizes emails based on a triplet containing the sending IP address, if the sender changes address between consecutive attempts the message risks to be greylisted again. This situation occurs for five out of ten web-mail providers. Even though all of them were able to eventually deliver the message because the same IP was reused in different connections, this behavior increases the delivery time and potentially results in a failed delivery.

To complement this result, we looked at the default configuration of the seven most popular MTA servers used on the Internet [16]. Table IV shows the retransmission times extracted from the software documentation, for the first 10 hours. The second column of the table represents the interval time for each delivery attempt performed by the MTA server. The third column shows instead the maximum time-to-live of an email before the server stops retrying and bounces the message back.

While the exact intervals are often controlled by a combination of multiple parameters, as one can see from the table, by default some MTA servers (e.g., Sendmail and Qmail) are very regular regarding the time interval between consecutive attempts. Exchange was the only MTA not RFC-822 complaint with respect to the time-to-live. Sendmail, Exim and postfix follow the same time to live suggested in RFC-822, while Qmail and Courier are even more conservative and use a threshold of 7 days.

VI. Discussion

In the previous sections we presented a number of experiments we conducted to assess the effect of nolisting and greylisting. Based on our results, we can finally answer a number of important questions.

First of all, the good news is that nolisting and greylisting are still effective in 2015. In fact, over 70% of the world spam is prevented by using either one or the other technique. Quite surprisingly, the most common families of malware responsible for sending spam are now able to cope with either nolisting (e.g., by skipping the primary mail server and contacting directly the secondary) or greylisting (by continuously re-trying to deliver the messages) but not with both. Between the two, greylisting seems to be more effective, but it also introduces negative side effects that need to be properly evaluated before deployment. On the other hand, nolisting is very simple to deploy and does not have many disadvantages. While it is certainly less famous and widespread than greylisting, according to our measures some popular companies and over 130 thousand domains already use it as an additional layer of defense against spam. If possible, our experiments show that using both techniques together is a very effective way to protect against the majority of spam.

Our study also confirms that the possible negative side-effect of greylisting are not a myth. To begin with, it is fundamental for greylisting services to white-list web-mail providers. The fact that many of them use multiple IP addresses and that some stop retrying after only 30 minutes could otherwise be problematic. Finally, our experiments also help to answer the question of which threshold should be used to maximize the protection and reduce disadvantages. In fact, on the one hand malware that supports the retransmission of failed messages is able to do that multiple times and to successfully deliver spam also with very high thresholds. On the other hand, a low threshold help reducing the delay of normal emails and to minimize the possibility of losing messages. Therefore, the use of a very short threshold is probably the best way to maximize both aspects (stopping spam and reducing unwanted delays).

Results Validity

This study presents a snapshot of the advantages and disadvantages of the use of greylisting and nolisting in 2015. It is very difficult to say when our results will be outdated: both techniques have been known for more than ten years, but our results shows that today they are still quite effective in practice. On the other hand, greylisting and nolisting have a cost for the system (for example in terms of disk space and computation resources) and for the Internet community at large (because of the increased traffic and bandwidth). The effectiveness of these two techniques can change in the future and it is important to know when they will become obsolete because at that moment it will not be worth paying the price anymore.

PROVIDER	SAME IP	ATTEMPTS	DELIVER	DELAYS (min:sec)
gmail.com	✗ (7)	9	✓	6:02, 29:02, 56:36, 98:44, 162:03, 229:44 309:05, 434:46
yahoo.co.uk	✓	9	✓	2:07, 5:39, 12:58, 27:16, 55:13, 109:35 216:47, 430:36
hotmail.com	✓	94	✓	1:01, 2:03, 3:04, 5:06, 8:07, 12:08, 16:10 . . . every 4 minutes . . . , 362:11
qq.com	✗ (2)	12	✗	5:05, 5:11, 5:17, 6:19, 8:22, 12:25, 20:29, 52:31, 84:35, 144:42, 204:56
mail.ru	✗ (7)	13	✓	1:18, 19:15, 49:14, 79:49, 113:20, 154:18, 187:53, 235:20, 271:03, 305:50, 340:38, 373:45
yandex.com	✓	28	✓	1:05, 2:58, 6:53, 14:55, 30:28, 45:41, 61:01, ...every 15:30 minutes..., 369:21
mail.com	✗ (2)	10	✓	5:02, 12:37, 23:59, 041:03, 66:38, 105:01, 162:35, 248:56, 378:28
gmx.com	✗ (3)	10	✓	5:01, 12:33, 23:50, 40:46, 66:09, 104:14, 161:22, 247:04, 375:36
aol.com	✓	5	✗	5:32, 11:32, 21:32, 31:32
india.com	✓	10	✓	6:21, 16:21, 36:21, 76:21, 146:22, 216:21, 286:21, 356:21, 426:21

TABLE III: Webmail delivery attempts with a 360-minute (6h) greylisting threshold.

MTA	RETRANSMISSION TIME (min.)	MAX QUEUE TIME (days)
sendmail	10, 20, 30, 40, 50, 60, . . . , 600	5
exim	15, 30,. . . , 120, 180, 270, 405, 607.5	4
postfix	5, 10, 15, 20, 25, 30, 45, . . . , 600	5
qmail	6.6, 26.6, 60, 106.6, 166.6, 240, 326.6, 426.6, 540, 666.6	7
courier	5, 10, 15, 30, 35, 40, 70, 75, 80, 140, 145, 150, 270, 275, 280, 400, 405, 410, 530, 535, 540, 660,	7
exchange	15, 30, 45, 60, 75, 90, . . . , 600	2

TABLE IV: Retransmission time of popular MTA servers

We hope our paper can bring attention to this problem, and that AV companies will start mentioning gray and nolisting support when they report the top malware families sending spam in their yearly reports. This additional information would require little additional work on their side, but it can provide a real benefit for the entire community.

VII. Related Work

Spam detection has been an active research topic for decades. A number of approaches have been proposed to mitigate the impact of unsolicited messages. These approaches can be categorized into sender-based filtering and content-based filtering methods, based on whether they detect and block spam before or after accepting the email. Examples of sender-based filtering methods are blacklists [11], [23], [28] and graylists [17], [19]. Sender-based filtering approaches based on server authentication [3] and IP reputation [4], [5], [14] have also been recently proposed. Examples of content-based filtering methods include bayesian filters [29], [36], collaborative filtering [18], [22] and email prioritization [35], [38].

Greylisting was initially introduced by Harris [17] in 2003 as a simple and effective method to filter out spam emails. One of the first experiments that shows the efficacy of such technique was performed by Levine [24], where the author pointed out that non-RFC-complaint clients (i.e. the clients that does not retry after the first failed attempt) are rare enough to be handled manually with a white-list. A first insight about the effectiveness of such detection mechanism was reported by Sochor [31], who evaluated the performance of greylisting combined with some Postfix restrictions during a long time period, spanning from the beginning of 2007 to the end of 2008. The author noticed that the effectiveness of greylisting remained constant over the two years of experiments. However, he also suggested that greylisting is not enough as a standalone spam defense mechanism because of the automatic administration of the automated white listing. The same author in [32] discusses different variants of greylisting and makes empirical suggestions about efficient values of the greylisting parameters. He also recommends to perform additional tests in a controlled environment due to the unstable intrinsic nature of real email messages. Even though these works can provide an insight about the effectiveness of such technique, the experiments were performed only on a small set of email traffic, for a limited amount of time, and only on a few servers.

In recent years, botnets have emerged as a major tool for sending spam from end-hosts. Methods to identify the spamming bots have been explored in a number of studies [10], [15], [20], [25], [27], [37]. Note that our analysis uses such results to select samples and test the efficacy of the greylisting and nolisting techniques. Thus, previous botnet studies provided a starting point for our deeper analysis of the role of greylisting and nolisting as anti-spam techniques.

VIII. Conclusions

In this paper we presented a comprehensive study of the advantages and disadvantages of using two not very popular spam filtering techniques: nolisting and greylisting. In particular, to the best of our knowledge we are the first to measure the effectiveness of nolisting, and the first to test how real malware behaves in the presence of these two defenses.

Before this work, both supporters and opponents of nolisting and greylisting had no concrete values to base their hypothesis and were therefore engaged in a futile battle in which one side was supporting their adoption and the other one claimed that these techniques may have worked in the past but are now useless against recent malware.

We hope that this paper can help system administrators to decide if they need nolisting or greylisting, and how to properly configure these solutions in their networks.

IX. Acknowledgment

We would like to deeply thank Giorgio Biacchi from the Computer Science department of Univeristà degli Studi di Milano, for his help in performing the experiments on the server of the university.

References

[1] Alexa. http://www.alexa.com/.

[2] Poor Man's Nolisting. http://nolisting.org/.

[3] Sender policy framework. http://www.openspf.org/.

[4] Senderbase. http://www.senderbase.org/.

[5] Spamhaus. http://www.spamhaus.org/.

[6] ViruShare. http://virusshare.com/.

[7] VirusTotal. https://www.virustotal.

[8] Malwr. https://malwr.com, 2010.

[9] Chen, X., Andersen, J., Mao, Z. M., Bailey, M., and Nazario, J. Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware. In *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on* (2008), IEEE, pp. 177–186.

[10] Chiang, K., and Lloyd, L. A case study of the rustock rootkit and spam bot. In *The First Workshop in Understanding Botnets* (2007), vol. 20.

[11] Chiou, P.-R., Lin, P.-C., and Li, C.-T. Blocking spam sessions with greylisting and block listing based on client behavior. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on* (2013), IEEE, pp. 184–189.

[12] Corporation, S. Internet security threat report. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_appendices_v19_221284438.en-us.pdf, 2014.

[13] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-wide scanning and its security applications. In *Proceedings of the 22nd USENIX Security Symposium* (Aug. 2013).

[14] Esquivel, H., Akella, A., and Mori, T. On the effectiveness of ip reputation for spam filtering. In *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on* (2010), IEEE, pp. 1–10.

[15] Esquivel, H., Mori, T., and Akella, A. Router-level spam filtering using tcp fingerprints: Architecture and measurement-based evaluation. In *Proceedings of the Sixth Conference on Email and Anti-Spam (CEAS)* (2009).

[16] Hafiz, M., Johnson, R., and Afandi, R. The security architecture of qmail. In *Proceedings of the 11th Conference on Patterns Language of Programming (PLoP04)* (2004), Citeseer.

[17] Harris, E. The next step in the spam control war: Greylisting, 2003.

[18] Heckerman, D. E., Breese, J. S., Horvitz, E., and Chickering, D. M. Collaborative filtering utilizing a belief network, Dec. 30 1997. US Patent 5,704,017.

[19] Islam, M. R., Zhou, W., Guo, M., and Xiang, Y. An innovative analyser for multi-classifier e-mail classification based on grey list analysis. *Journal of network and computer applications 32*, 2 (2009), 357–366.

[20] John, J. P., Moshchuk, A., Gribble, S. D., and Krishnamurthy, A. Studying spamming botnets using botlab. In *NSDI* (2009), vol. 9, pp. 291–306.

[21] Klensin, J. C. Rfc5321: Simple mail transfer protocol.

[22] Koren, Y. Collaborative filtering with temporal dynamics. *Communications of the ACM 53*, 4 (2010), 89–97.

[23] Levine, J. Dns blacklists and whitelists (internet draft irtf anti-spam research), 2008.

[24] Levine, J. R. Experiences with greylisting. In *CEAS* (2005).

[25] Mori, T., Esquivel, H., Akella, A., Shimoda, A., and Goto, S. Understanding the worlds worst spamming botnet. *University of Wisconsin Madison Tech Report TR1660* (2009).

[26] Quist, D., Smith, V., and Computing, O. Detecting the presence of virtual machines using the local data table. *Offensive Computing* (2006).

[27] Ramachandran, A., and Feamster, N. Understanding the network-level behavior of spammers. *ACM SIGCOMM Computer Communication Review 36*, 4 (2006), 291–302.

[28] Ramachandran, A., Feamster, N., and Vempala, S. Filtering spam with behavioral blacklisting. In *Proceedings of the 14th ACM conference on Computer and communications security* (2007), ACM, pp. 342–351.

[29] Ramachandran, A., Feamster, N., and Vempala, S. Filtering spam with behavioral blacklisting. In *Proceedings of the 14th ACM conference on Computer and communications security* (2007), ACM, pp. 342–351.

[30] Rfc822. urlhttp://cpansearch.perl.org/src/PDWARREN/Mail-RFC822-Address-0.3/Address.pm.

[31] Sochor, T. Greylisting long term analysis of anti-spam effect. In *Risks and Security of Internet and Systems (CRiSIS), 2009 Fourth International Conference on* (2009), IEEE, pp. 98–104.

[32] Sochor, T. Greylisting method analysis in real smtp server environment–case-study. In *Innovations and Advances in Computer Sciences and Engineering*. Springer, 2010, pp. 423–427.

[33] Sochor, T. Efficiency comparison of greylisting at several smtp servers. *Procedia Computer Science 3* (2011), 930–934.

[34] Stringhini, G., Egele, M., Zarras, A., Holz, T., Kruegel, C., and Vigna, G. B@ bel: Leveraging email delivery for spam mitigation. In *USENIX Security Symposium* (2012), pp. 16–32.

[35] Twining, D., Williamson, M. M., Mowbray, M., and Rahmouni, M. Email prioritization: Reducing delays on legitimate mail caused by junk mail. In *USENIX Annual Technical Conference, General Track* (2004), pp. 45–58.

[36] Wu, C.-T., Cheng, K.-T., Zhu, Q., and Wu, Y.-L. Using visual features for anti-spam filtering. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (2005), vol. 3, IEEE, pp. III–509.

[37] Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., and Osipkov, I. Spamming botnets: signatures and characteristics. In *ACM SIGCOMM Computer Communication Review* (2008), vol. 38, ACM, pp. 171–182.

[38] Yoo, S. *Machine learning methods for personalized email prioritization*. PhD thesis, Carnegie Mellon University, 2010.

[39] Zmap Team. Internet-Wide Scan Data Repository. https://scans.io/.