Melting the Snow: Using Active DNS Measurements to Detect Snowshoe Spam Domains

Olivier van der Toorn*, Roland van Rijswijk-Deij*, Bart Geesink†, Anna Sperotto*
*University of Twente
{o.i.vandertoorn, r.m.vanrijswijk, a.sperotto}@utwente.nl
†SURFnet
bart.geesink@surfnet.nl

***Abstract*—Snowshoe spam is a type of spam that is notoriously hard to detect. Anti-abuse vendors estimate that 15% of spam can be classified as snowshoe spam. Differently from regular spam, snowshoe spammers distribute sending of spam over many hosts, in order to evade detection by spam reputation systems (blacklists). To be successful spammers need to appear as legitimate as possible, for example, by adopting email best practices, such as the Sender Policy Framework (SPF). This requires spammers to register and configure legitimate DNS domains. Many previous studies have relied on DNS data to detect spam. However, this often happens based on *passive* DNS data. This limits detection to domains that have actually been used and have been observed on passive DNS sensors. To overcome this limitation, we take a different approach. We make use of *active* DNS measurements, covering more than 60% of the global DNS namespace, in combination with machine learning to identify malicious domains crafted for snowshoe spam. Our results show that we are able to detect snowshoe spam domains with a precision of over 93%. More importantly, we are able to detect a significant fraction of the malicious domains up to 100 days earlier than existing blacklists, which suggests our method can give us a time advantage in the fight against spam. In addition to testing the efficacy of our approach in comparison to existing blacklists, we validated our approach over a 3-month period in an actual mail filter system at a major Dutch network operator. Not only did this demonstrate that our approach works in practice, the operator has actually decided to deploy our method in production, based on the results obtained.**

***Index Terms*—<Spam Detection, Active DNS Measurements, Blacklisting.>**

I. Introduction

Spam is a major problem on the Internet. In particular, the kind of spam containing URLs to malicious content, or viruses, is troublesome. Pfleeger and Bloom [1] have reported that the sending of one million spam emails costs around US$250 for the sending party, but that on average, the time spent deleting them costs the receiving party about US$2,800 in lost wages. Spam is a cat-and-mouse game between the spammers and email providers. Spammers often try to bypass mail filters by developing new methods of distributing spam. Snowshoe spam is one of these methods.

In 'normal' types of spam the entire burden of transmitting the spam messages is often put on only a few hosts. In contrast, in the case of snowshoe spam the sending is spread out over many hosts, to avoid detection by spam reputation systems (blacklists). A second characteristic of snowshoe spam is that spammers want to appear as legitimate as possible, by adopting email best practices. An example of such a best practice is Sender Policy Framework (SPF), a technique to ensure only authorized email servers can send email for specific domains. However, SPF requires spammers to also register and configure a legitimate Domain Name System (DNS) domain. Additionally, it requires them to create a DNS record for every host that should be able to send email for that domain. This results in a domain with a large number of records. The creation of such domains is often called *crafting*. Cisco [2] reported that 15% of spam in 2014 was classified as snowshoe spam.

Spam detection has been studied intensely by the security research and anti-abuse communities. A number of studies link the use of data in the DNS to spam detection. However, this usually happens in a passive manner. The goal of this paper, on the other hand, is to detect *crafted* snowshoe spam domains using active DNS measurements. Our approach combines active DNS measurements with supervised machine learning. The active DNS measurements are retrieved from the unique OpenINTEL platform[1], which actively queries more than 60% of all registered domain names worldwide. We verify our results by comparing them to well-known blacklists.

The main contributions of this paper are that we:

- perform detection of domains crafted for snowshoe spam, using active DNS measurements;
- show that our method can identify domains earlier than existing blacklists, which allows us to block spam that would otherwise bypass a mailfilter;
- make the resulting blacklist available for researchers and spam filter operators, for further study and to improve detection of spam.

The remainder of this paper is structured as follows. Section II discusses related work. In Section III we present our methodology. The datasets we use are introduced in Section IV. Section V discusses our results. A real-world deployment of our work is presented in Section VI. We analyse the ethical impact of our work in Section VII. Finally, Section VIII details our conclusions.

978-1-5386-3416-5/18/$31.00

[1] https://openintel.nl/

II. Related work

A. Passive and active DNS monitoring

Many security-related studies have looked at passively monitoring DNS. Especially when done at a large scale, passive monitoring of DNS can yield important information about the use and security of DNS. A notable approach is *passive DNS* (pDNS) [3], a system that monitors DNS queries and responses issued from a recursive resolver towards authoritative name servers. pDNS is used to investigate DNS anomalies [4]–[7], such as domains used for spam campaigns and malware. Perdisci et al. [8] used passive DNS to measure the growth of IP addresses in order to determine if a domain would be used in flux service networks. The biggest advantage of pDNS is that it reflects live use of the DNS. However, this also means that pDNS is in general usage-biased and that only anomalous behavior in the monitored network can be detected. In this paper, we take a different approach. We use actively collected DNS data, which allow us to detect anomalous domains at a global scale and independently from their being accessed by users.

A few studies have already looked at how active DNS measurements can be used to identify malicious activities. Konte et al. [9] monitor changes in DNS records of known spam domains to investigate at which rate and to which extent malicious domains change their characteristics, e.g., in relation to fast-flux domains. Hao et al. [10] use zone transfer records to obtain DNS data to characterize, among others, the time between registration of a malicious domain and its appearance on a spam blacklist and the location of the name servers used for the domain. Felegyhazi et al. [11] investigate the use of DNS in proactive blacklisting of malicious domains. Hao et al. [12] also look at the history of a domain name and the details of new registrations to single out malicious domains. While these studies share our same intuition, that is that malicious domains need to be registered and configured before they can be used, our contribution differs in the following aspects. First, while several other contributions are limited to analyzing only a handful of zones, our work covers more than 60% of currently registered domain names. Secondly, most of the previous studies start from a set of known malicious domains, and use this for inferring general characteristics. We focus instead on building a model of malicious behavior using a machine learning approach.

B. Spam

Syed et al. [13] and Moura et al. [14] report that spam sources can be identified using only network-based characteristics. Their works are based on the observation that spam sources tend to be clustered in relative address proximity, e.g. in the same subnet or autonomous system. Yamakawa et al. [15] show that this address clustering also exists geographically, since large volumes of spam comes from the same countries. To be effective these approaches need to observe large volumes of spam to identify *bad neighborhoods* (*tainted* address space), and they can only perform just-in-time spam identification, namely at the time spam has already been sent.

There exists very little related work specifically focusing on snowshoe spam. Bhowmick et al. [16] mention snowshoe spam as an emerging threat. In this work, we focus on snowshoe spam in particular.

C. Machine learning

Supervised machine learning is a way to build a predictive model (classifier) based on labeled data. It is often used to detect malicious activity on networks [17], [18]. Several studies combine the subjects of spam detection and machine learning. Youn et al. [19] provide an overview of classifier types and their performance. Clustering and decision trees are techniques frequently used [5]–[8]. Drucker et al. [20] use a Support Vector Machine (SVM) to classify email, based on the content, as spam or ham. In the work of Sakkis et al. [21], classifiers are used sequentially to increase the accuracy of the classification. Bhowmick et al. [16] look at how spam evolves and what tools emerge, or change, to combat these new types of spam. All techniques presented focus on the headers of an email and/or the content of the email.

While we also use a machine learning approach, we differ from the state of the art in the fact that our methods is independent from the content of an email, but it relies only on domain names configurations.

III. Methodology

In this section, we present our methodology. Figure 1 shows a high-level overview of our approach. From left to right, it displays four parts (A)-(D) that together make up our detection process. In addition to this, a fifth part, (E), is shown in the gray rectangle, which represents the training of the machine learning classifier that our detection relies on.

At a high-level, our method for detection does the following. Every day, based on data from the OpenINTEL platform (A), we perform a filtering step, called the long tail analysis (B), to extract candidate domains. We then use a machine learning classifier (C) to perform a binary prediction of domains to blacklist, which are then added to our Real-time Blackhole List (RBL) (D).

In the section 'Building and training a classifier' (E) we explain the parts of Figure 1 with a gray background. These parts concern the training of the classifier.

A. DNS data collection

Both our training and detection make extensive use of data from the OpenINTEL platform. Since February 2015, the OpenINTEL[1] large scale active DNS measurement platform collects daily snapshots of the data in the DNS [22]. The measurement currently queries 60% of the global DNS namespace. At the time of writing, the measurement covers the zones `.com`, `.net`, `.org`, `.info`, `.mobi`, the new gTLDs defined by ICANN, and a set of ccTLDs such as `.nl`, `.se`, `.ca`, `.fi`, `.at`, `.dk`, `.nu` and `.ru`. For each domain name, the measurement performs a fixed set of queries including `A`, `AAAA`, `MX`, `TXT`, etc. The result from these queries forms the basis for the features we use. These features are described in Section III-E

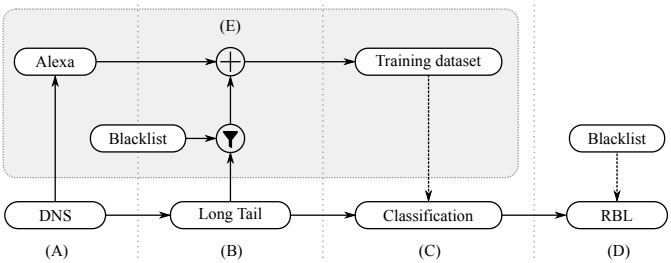


Figure 1. High level overview of our approach

B. Long Tail Analysis

Snowshoe spammers aim at spreading the sending load among a large number of hosts. At the same time, they are likely to use SPF in order to make their domains appear legitimate. Therefore, we expect that snowshoe spam domains will have a large number of `A` or `MX` records and large (in terms of number of characters) `TXT` records. Domains with these characteristics will likely show up in a long tail analysis of DNS domains. The long tail typically refers to the outliers of a distribution. In our case, the majority of domains will have only a few DNS records of a given type, whereas snowshoe spam domains will exhibit many records of the certain types. Thus, these domains appear far away from the mean, in the long tail of the DNS. In this paper we look at two types of long tail of the DNS. The first tail holds domains with a large number of records. The second type holds domains with exceptionally large `TXT` records. We have defined four thresholds for what we consider to be the long tail: 99.9%, 99%, 98% and 97%. We have chosen these thresholds to range from very conservative to more permissive selections. We stopped at 97% to limit the number of domains to we need to analyze, so we can perform daily detections in a timely manner.

C. Classification

To cope with the dynamic nature of spam, we have opted to use machine learning to do the detection. The reason for this is that a classifier can easily be retrained on new data if spam trends change. In addition to this, the vast amount of data makes a manual creation of signatures unfeasible. In this step, we match the domains selected from the long tail analysis against a machine learning classifier. The classifier has been chosen as described in Section III-E. The classifier takes into account a set of features derived from the DNS records for the candidate domains (see Table II). The output of the classifier is a binary decision detailing if a domain should be considered as a snowshoe domain.

D. Realtime Blackhole List (RBL)

To make our results easily available and usable, we store them in the form of an RBL. In this section, we explain how the RBL is kept up to date. As said, every day we run our detection process. The classifier outputs a list of domains that it considers to be snowshoe spam domains. Domains from this list, which are not already present on the RBL, are added to it. For validation purposes, all domains on the RBL are then checked against existing public blacklists (Table I). We mark it as soon as a domain on our RBL also appears on a public blacklist. This allows us to do time analysis of our detections. The blacklists from Table I were selected based on their popularity among operators.

E. Building and training a classifier

In this section we describe the building of the training dataset and the training of the classifier. Figure 1 shows these steps with a grey background.

1) Dataset: To build a training dataset for our classifier, we label the dataset of candidate domains extracted from the long tail of the DNS. We do this by checking the domains against public blacklists (Table I). Depending on the nature of the public blacklist, we either check the domain name of a candidate domain against the blacklist or an Internet Protocol (IP) address from one of the DNS records (`A` and `MX`) for the domain. If the domain is listed, we label the domain as a positive, otherwise, it is labeled as a negative. In order to increase the accuracy of the training dataset, we filter the positives from the dataset and balance them with and equal number of negatives from the Alexa top one million list. While domains on the Alexa list are not guaranteed to be benign, the probability of them being benign is much higher than the negative instances extracted from the long tail.

Additionally, we created an evaluation dataset which does not perform this extra filtering step. This evaluation dataset is used to compare different classifier types.

Both the training and evaluation dataset consist of 35 features. The features we have used and their sources are listed in Table II. Most of these features measure how many records of a certain type a domain contains. Some features are more complex and rely on evaluating regular expressions. For example, the 'spfv1_ip_count' feature uses a regular expression to count the number of IP addresses in an SPF record. The output of all of the features is numerical, because all the evaluated classifiers are able to make predictions based on numeric features and only a few (special) classifiers are able to process raw strings [23]. Thus to reach maximum compatibility we make sure all features are in numeric form.

Not all features are equally important. Following the output from a trained 'Decision Tree', the 'response_name_matches' feature is the most important, since it has the highest Gini index. This feature details if the query name in the response is the same as in the request. The 'ip4_count' and 'mx_count' features are, after the 'response_name_matches', equally important.

2) Classifier: In order to perform optimal detection, we first needed to select a suitable classifier. Below, we explain our methodology for finding the 'best' classifier for our problem. We also explain what we mean by 'best'.

We evaluated classifiers in a number of categories. In the Naive Bayes category, we looked at the 'BernoulliNB', 'GaussianNB' and 'MultinomialNB' classifiers. For Decision

Tree-type classifiers, we tested the 'DecisionTreeClassifier' and the 'RandomForestClassifier'. Of the Nearest Neighbor variant, we evaluated the 'KNeighborsClassifier' and 'RadiusNeighborsClassifier'. From the Gradient Descend type we took the 'GradientBoostingClassifier' and 'SGDClassifier'. Finally we also looked at the 'Support Vector Classifier (SVC)', 'MLPClassifier' and the meta-classifier 'AdaBoostClassifier'[2]. Our selection of classifiers was primarily motivated by the combination of classifiers used in related work [6]–[8], [13], [16], [19]–[21], and the availability of classifiers in the 'sklearn' [24] Python library.

Selection of the 'best' classifier is done in two steps. First, we establish the optimal parameters for each of the 13 classifiers selected. This step aims at understanding what the optimal performance of each classifier is, given our training set. This also allows us to compare the classifiers later on. This was done as follows. The training set is split in training data and test data. The classifier is then trained on the training data following the K-Fold Cross Validation [25] method, which is visualized in Figure 2. The training part of the dataset is split into K folds. A classifier is trained K times on $K-1$ folds of the training part, for example, in the figure parts $k(1)$ through $k(4)$. During the training process the chosen algorithm builds a model of the (labeled) data, in particular the boundaries between the positive and negative entries. Based on such a model predictions can be made on new, unseen, data. Then the performance of the classifier is validated in the K-th fold, in our example $k(5)$. This is done K times, where the validation fold is a different fold each time. The performance of the classifier is the average over each fold. Based on this performance, we select the parameters for each classifier that lead to the highest precision, where precision is expressed as the number of True Positives (TP) relative to the total amount of positives, which also includes False Positives (FP) (Equation 1). We repeat this procedure for every type of classifier.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

The second step consists of comparing the optimal performance of the different classifiers. The performance of each classifier is measured on the evaluation dataset. The one with the best precision on this dataset is selected as the classifier that will be used for our daily detection.

[2]Documentation on these classifiers is available at http://scikit-learn.org/stable/modules/classes.html

TABLE I
THE USED BLACKLISTS AND THEIR PURPOSE

Name	Domain	IP address
multi.uribl.com	✓	
dbl.spamhaus.org	✓	
rbl.rbldns.ru		✓
zen.spamhaus.org		✓

TABLE II
USED FEATURES AND THEIR DATA SOURCES

Data source	Feature
as	as_count
cname_name	cname_count, cname_in_domain, cname_out_domain
country	country_codes
ip4_address	ip4_count, ip4_prefixes
ip6_address	ip6_count, ip6_prefixes
mx_address	mx_cloud, mx_count
ns_address	ns_count, ns_domain_count
query_name	p_numeric
query_name & response_name	response_name_matches
soa_minimum	soa_minimum
txt_text	p_txt_numeric, spfv1_{a,cidr,include,ip}_count, spfv1_{a,cidr,include,ip}_ratio, spfv1_{a,cidr,include,ip}_unique_count, txt_length,verification_{globalsign,google}_count, verification_{globalsign,google}_ratio, verification_{globalsign,google}_unique_count

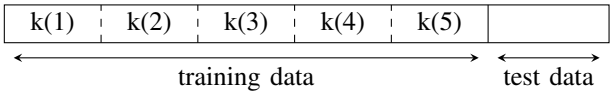


Figure 2. Visualization of split in training dataset

IV. DATASETS

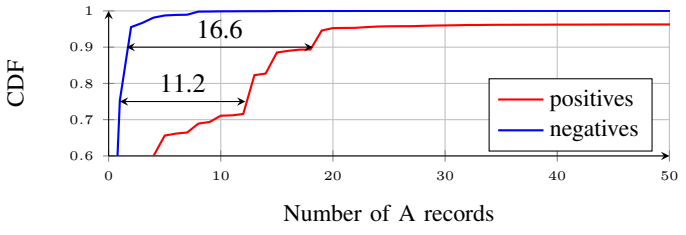
Based on the approach discussed in Section III, we have performed daily detection from May 24, 2017 till September 5, 2017. This section discusses the details on the datasets used, either in the training, validation or during the daily detections.

A. Distinction positives & negatives

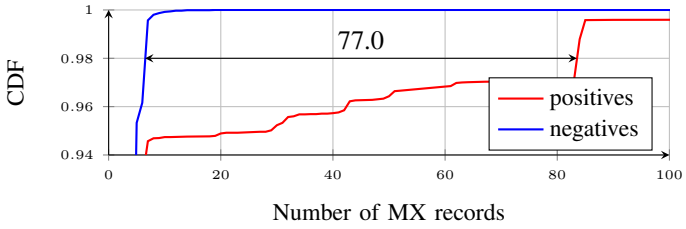
Before we dive into the results of our method, we verify that there is a clear difference between the positives (spam) and negatives (ham). For this goal we have made a dataset from April 2017. We have selected domains above the 99 percentile, since this percentile threshold gave a clear distinction between positives and negatives. After labeling the dataset we filtered this dataset following the same method as for the training dataset. This resulted in a dataset with both 136441 positives and negatives. We do so by plotting the Cumulative Distribution Function (CDF) for two features, these plots are visible in Figure 3. This analysis indicates that at the 90th percentile for the `A` record distribution, spam domains have on average 16.2 records more than regular domains. Similarly, at the 98th percentile of the `MX` record distribution, spam domains have 77 records more than regular domains. The fact that not all domains show this clear distinction motivated us to make use of the many features available to us.

B. Training and evaluation dataset

For the selection of the 'best' classifier we have made two datasets. The first is the training dataset, the classifier is trained upon this dataset, it consists of data from April 18, 2017 till April 24, 2017. The second dataset, the evaluation dataset, consists of data from April 25, 2017. Table III lists how many



(a) Comparison of the number of A records



(b) Comparison of the number of MX records

Figure 3. CDF of two features in the test data set

domains there are in both datasets, along with how many positives and negatives. As intended the training dataset is balanced.

C. Daily detection datasets

Since May 24, 2017, we have been doing daily detections of possible snowshoe spam domains. The basis of these detections is a dataset of that day containing domains exceeding the 99.9, 99, 98 or 97 percentile. On average there are about 2.7K domains in the dataset of the 99.9 percentile. This figure grows to 57.3K domain names in the dataset of the 97 percentile. Table VI shows the average size of each of the daily datasets.

V. Results

This section has been split into two parts. First, we discuss the results from selecting the 'best' classifier. Secondly, we discuss the daily detections made for the RBL.

A. Selecting the 'best' classifier

As discussed in Section III-C, the selection of the best classifier is a two step process. First, select the optimal parameters for each classifier. Then, we select the 'best' classifier, as the one with the best performance. For brevity sakes we omit the results of the first step and directly present a comparison of the classifiers, in Table IV.

TABLE III
Statistics of Training and Evaluation Dataset

%-tile	#Domains (total, positive, negative)	
	Training dataset	Evaluation dataset
99.9	2018 (1009 – 1009)	1407 (1261 – 146)
99	3540 (1770 – 1770)	5453 (5199 – 254)
98	4806 (2403 – 2403)	20534 (20177 – 357)
97	5526 (2763 – 2763)	25381 (24968 – 413)

TABLE IV
Classifier Performance on the 'Real' Data Set

Classifier Type	TP	FN	FP	TN	Accuracy	Precision
AdaBoost Improved	6688	7842	110	10741	68.69%	98.38%
AdaBoost	5971	8559	164	10687	65.63%	97.32%
MLP	7273	7257	707	10144	68.62%	91.14%
DecisionTree	6279	8251	695	10156	64.75%	90.03%
MultinomialNB	12179	2351	1397	9454	85.23%	89.70%
RandomForest	11156	3374	1488	9363	80.84%	88.23%
KNeighbors	4562	9968	676	10175	58.06%	87.09%
GaussianNB	13330	1200	2075	8776	87.10%	86.53%
SVC	13449	1081	2339	8512	86.53%	85.18%
RadiusNeighbors	13318	1212	2367	8484	85.90%	84.90%
SGD	3599	10931	674	10177	54.28%	84.22%
BernoulliNB	12995	1535	2507	8344	84.07%	83.82%
GradientBoosting	12645	1885	9605	1246	54.73%	56.83%

We have decided to look for a classifier with a low number of FP. This is because is spam detection it is far more costly to make an FP, a ham domain marked as spam, than any other error. The cost of making a FP outweighs making a correct classification, a TP. The reasoning can be put in perspective by an example; the cost of marking an important email as spam, or discarding the email, is much higher than receiving a spam message. This is the reason we have chosen to rank our classifiers on their precision metric (Eq 1), since it is more closely related to the number of FPs made by the classifier than other metrics.

The performances from the second step are listed in Table IV. The 'AdaBoost' classifier has the highest precision on our evaluation dataset, and it has the lowest number of false positives. However, it does not have the highest number of true positives. We improve this classifier by taking a closer look at the parameters of the classifier. We managed to increase the number of TPs by 717 and reduce the number of FPs by 54. The resulting classifier is labeled as 'AdaBoost Improved' in Table IV. The 'AdaBoostClassifier' is a meta-classifier. "It begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases" [26]. To improve our classifier we changed the base estimator from the 'DecisionTreeClassifier' to the 'MultinomialNB' and set the number of estimators to 1. The additional parameters are in Table V.

Table IV makes clear why we rank the classifiers based on their precision metric rather than, for example, the accuracy. If we compare the classifier with the highest accuracy, 'GaussianNB', with the improved 'AdaBoostClassifier', we see about double the TPs but there are more than 18 times as many FPs. The cost of making a FP is much higher than the gain of a TP, since it may mean important benign email is discarded. Thus, for our goal the 'AdaBoostClassifier' is 'better' than the 'GaussianNB' classifier.

B. Detection results

In this section we discuss the general detection results. During our measurement period, our detection method marked 35,004 domains as snowshoe spam domains. 32,677 of these

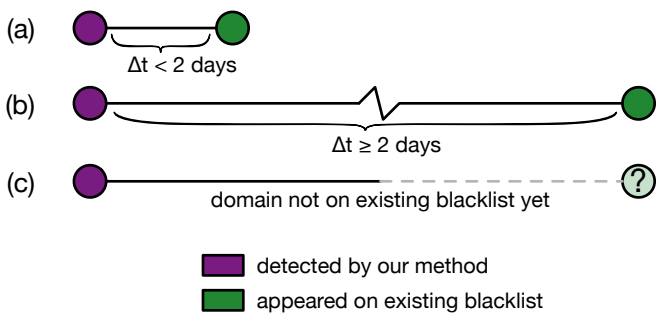


Figure 4. Early detection categories

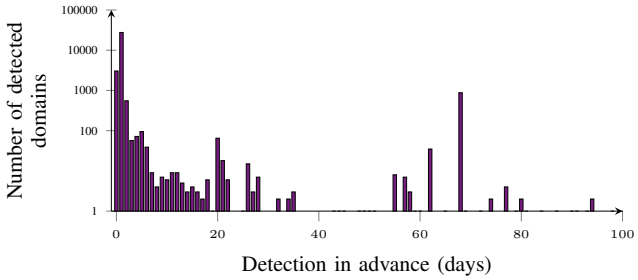


Figure 5. Early detection of domains

domains (93.35%) appeared on an existing blacklist at some point during the measurement period. This indicates that our method is highly effective at detecting snowshoe spam domains. The remaining 2,327 domains (6.65%) are either false positives or they have not yet appeared in one of the existing blacklists. This second case occurs when our detection mechanism reports snowshoe domains (much) earlier than blacklists. We analyze this case in the next section.

Table VI lists how many domains per day on average are in the long tail dataset (per percentile), how many are detected by the classifier and how many are newly added to the RBL.

C. Early detection

In this section we analyze if our approach has a time advantage over regular, existing blacklists, such as the Spamhaus blacklist. By time advantage we mean the window between detection by our method and the time at which the same domains appears on one of the existing blacklists we considered (see Table I).

In the context of early detection, we distinguish three categories of domains. Figure 4 depicts these categories, and they are described in more detail below:

(a) domains that are already on a blacklist at the time of detection, or have only a day difference. There can be a one day difference since the daily data is of the previous day, while the blacklist query happens in real-time.

TABLE V
PARAMETERS OF THE IMPROVED ADABOOSTCLASSIFIER

Name of parameter	Value
base_estimator	MultinomialNB
n_estimators	1
learning_rate	1.
algorithm	SAMME.R

TABLE VI
PER-DAY AVERAGES OF THE DATASETS AND DETECTIONS

Percentile	Avg. domains in dataset	Avg. domains detected	Avg. added to the RBL
99.9	2728.07	243.96	18.99
99	19179.59	3228.75	149.37
98	37202.64	5226.31	205.72
97	57250.48	6805.55	239.37

(b) domains with a detection difference of at least two days or more.

(c) domains that – on the day of writing – have not (yet) been blacklisted.

Figure 5 shows how many domains have been detected, with how much of a time difference before being blacklisted. The y-axis is log-scaled to make the spread more visible.

In total 35,004 domains have been detected. The majority of domains by far falls in the first category (a), 30,705 domains (87.72%) appear on a blacklist less than two days after detection via our method. In the second category (b), where our detection is at least two days in advance, contains 1,972 domains (5.63%). Of these 1,972 domains, 1,154 domains (3.30%) were detected at least a week in advance, 1,105 domains (3.16%) were detected more than two weeks in advance, and 971 domains (2.77%) were detected at least a month in advance. There are even 949 domains (2.71%) which were detected at least 60 days before they appeared on a blacklist. The maximum time difference we observed so far is 104 days. 2,327 domains (6.65%) fall in the last category (c), and have not (yet) been blacklisted. While these numbers may seem small percentage-wise, it should be noted that this type of email often makes it past an email filter.

VI. OPERATIONAL DEPLOYMENT

To validate our method in a real-world scenario, we deployed the RBL (Section III-D) in an operational mail filtering service. This allows us to measure how effective our detections are. This deployment was done in collaboration with SURFnet, the National Research and Education Network in The Netherlands. The email of most of their connected universities and colleges is handled by SURFmailfilter. Hence, this is an excellent vantage point for evaluating if the domains we detect are in use for sending spam. In this section we discuss the results of this case study.

A. Method

First we describe the setup of this case-study. SURFmailfilter works, like many mail filters, with a scoring system; the higher the score, the more likely it is that the email is spam. The operators of SURFmailfilter have set the defaults for tagging an email as spam to a score of 5, and discard any email with a score higher than 10. While these thresholds

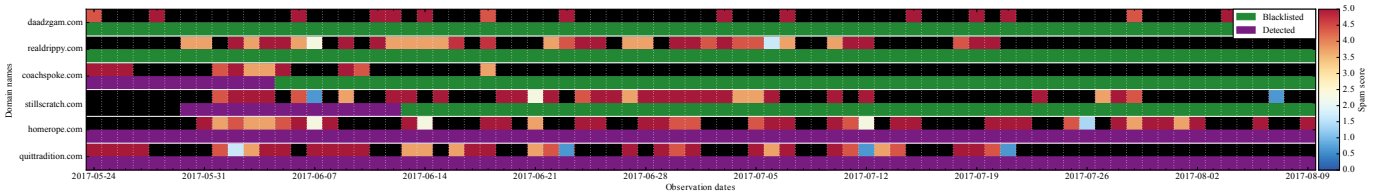


Figure 6. SURFmailfilter Detections

are configurable, in this paper we follow the thresholds as set by the SURFmailfilter operators. To test our approach, we configured our RBL as an extra source for blacklisted domains in SURFmailfilter. To not influence the normal spam score an email would get by too much, we have given the RBL a minimum score (0.1). This has the effect that the mail filter will not ignore the RBL, but at the same time our detection system will not accidentally turn ham into spam in case a benign domain happens to be on our RBL.

Then, to assess the effectiveness of our method, we retrieve the email IDs which have hit the RBL, and we extract the domains that have triggered the RBL. Of these emails we record the triggering domain, the spam score, the date the domain was detected, if the domain was blacklisted, and if so, when.

B. Results

We discuss the results from SURFmailfilter in two ways. Firstly, via the domains which have been seen by SURFmailfilter. The initial goal was to confirm that the detected domains are in use, but these results can also be used to confirm that the domains are actually spam domains. And secondly, via the emails themselves which have hit the RBL. With these results we can answer how much extra spam could be tagged or blocked by using our approach.

1) Domains: The domains which have been seen by SURFmailfilter can, roughly, be categorized into the same three categories as used in Section V-C. The first category (a) consists of domains which have appeared on a blacklist shortly after detection by our method (one day or less of a difference). The second category (b) contains domains which have appeared on a blacklist some time after detection by our method (two days or more of a difference). Finally, the last category (c) is for domains which have, during our measurement period, never appeared on a blacklist.

Figure 6 exemplifies 6 domains from these 3 categories. This graph is built by looking at each domain separately. The upper row displays their maximum score per day. A black color means no email containing the domain was observed that day. The score visualization ranges from a low score, in blue, to a high score, in red. The score is cut off at five. This means that while emails may have scored higher than five, these are all displayed in red.

In overlay, we have the status of the domain. A domain is either detected only by our system (purple) or it is detected and appears in one of the blacklists (green). The visualization in Figure 6 summarizes the various possible cases we face when comparing our method with blacklists.

In total, 130 domains that appear on our RBL have been seen by SURFmailfilter in the body of an email. These domains can roughly be categorized in three ways:

1) The first category, where the detection difference is one day or less, contains 23 domains (17.69%). Of these, 16 have an average score above five. The other four domains appear in emails scoring both below and above the five point mark, but on average score below five. The reason many domains in this category have a high spam score can be explained by the fact that the blacklist status causes an increase in spam score, and thus it exceeds the threshold of five more easily. This also means that in this category many of the emails are already marked as spam, because of their high score, and that our approach does not offer much gain for this category.
2) In the second category, where the detection difference is two days or more, there are 38 domains (29.23%). Of these domains, 22 have an average score above five. Four domains have only been seen in an email once, scoring below five. Two domains have been seen in multiple emails, all scoring less than five points. The remaining 10 domains, appear in emails with scores above and below the five point mark, but do not make the five point average. Percentage wise there are fewer domains with an average score above five compared to the first category, thus our approach may make difference in this category of domains.
3) The last category, where detected domains have not appeared on a blacklist in the measurement interval, contains 69 domains (53.08%). Of these 69 domains, there are 38 with an average spam score above five. 12 domains have appeared in emails which have all scored below the five point mark. However, seven of these domains have only been seen in a single email. The remaining 19 domains were seen in emails scoring both below and above the five point mark, but with an average score of below five. Our approach is most beneficial in this category. About half of the domains in this category score, on average, below five, this means that emails containing these domains are able to bypass the mail filter. However, since the domains do appear on our RBL the score of those emails can be increased by assigning a higher score of hitting the RBL.

A large portion of detected domains have an average score

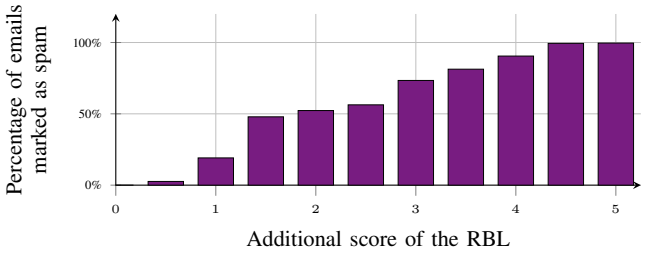


Figure 7. Amount of extra email marked as spam

above five, this gives confidence that our method is effective in detecting domains associated with spam.

2) Emails: Over our measurement period, SURFmailfilter processed 3,773 emails that triggered the RBL. Of these emails, 1695 come from the latter two categories presented in Section V-C at the time of receiving the emails. We only evaluate emails containing domains which are not blacklisted. 560 emails have a score equal to, or above five. While this means that the email would have been marked as spam with or without our method, it also gives confidence that our method is effective at detecting spam. In the pool of 1,135 emails scoring below five, 77 emails contain domains in the body which have not appeared in emails scoring higher than five.

This pool of 1,135 emails, which have scored below the five point mark, has been used to evaluate how many emails could additionally be blocked, at what assigned score for the RBL. Figure 7 visualizes the results of this analysis. As a conservative measure the RBL could be awarded a single point. In our situation this would have marked 19.1% of those 1,135 emails as spam. If the score is increased to two, 52.3% of emails would have been marked as spam.

While we have strong reasons to assume that all domains on our RBL are linked to spam, this approach lets mail filter operators control how much they trust these results.

C. Uptake

SURFnet has used our RBL for three months as discussed above. At first glance, the amount of additional spam that could potentially be filtered seems small. Typically, spam filtering systems catch a large percentage of spam messages [2], and very few actually end up in a user's inbox. SURFnet has indicated that the emails detected by our method are actually those that currently slip through the cracks and end up in users' inboxes. This makes our approach valuable to operators. In fact, SURFnet has decided to start using our method in production, and will assign a score of two to the RBL.

VII. Ethics

The SURFmailfilter case study raises obvious privacy concerns, as the system processes actual private email. Therefore, the operators at SURFnet protected the privacy of their customers by only giving us enough information to do our research. We did not have access to the actual body of emails. Occasionally we were given access to the subject line, in order to get a better idea if a message could be spam or not.

Another concern is that the RBL resulting from our method may contain benign entries (false positives). This is true for all blacklists. While blacklist operators try to ensure that only malicious domains end up on their list, sometimes false positives slip through the cracks. This problem is doubled for our RBL since our classifier is only as good as the training set is. The training set is labeled by looking up the domains on existing blacklists, if their accuracy is not one hundred percent, the predictions from the classifier are not going to be perfect. We therefore caution against treating our RBL as 'absolute truth', and instead advocate that it is treated as circumstantial evidence that supports the suspicion of a message being spam.

VIII. Conclusions

In this paper we investigate how domains crafted for sending snowshoe spam can be detected using active DNS measurements. Using the unique large-scale OpenINTEL dataset of the DNS and by applying machine learning techniques, we are able to detect malicious domains. 93.25% of domains we have detected have appeared on an existing blacklist at some point during the measurement period. Additionally, we have shown that our method is able to detect domains from 2 to 104 days in advance, when compared to regular blacklists, such as the Spamhaus blacklist.

In the operator case-study at SURFnet, we demonstrated that the time advantage translates into additional emails being marked as spam. In addition to this, we verified that these emails actually contain domains known to be associated with spam. These emails would otherwise bypass the email filter.

A. Future work

This paper has shown the potential of active DNS measurements in the search for snowshoe spam domains. We realise, however, that this is just a starting point. First, the next obvious step is to collaborate with spam filter operators in order to have more measurement points. Since spam is highly targeted it is reasonable to assume that SURFnet, the Internet service provider for academia in the Netherlands, receives a different kind of spam than, say, for example, an email provider in the United States. Since SURFnet is planning to add two points to emails hitting the RBL we will follow up with SURFnet after some time to learn from their experience. Secondly, the optimal period for obtaining a fresh training set and retraining the classifier needs further investigation.

Acknowledgements

Special thanks go to SURFnet for allowing us to test our method in practice. Their contribution proved invaluable to testing our approach in an operational environment.

The research leading to the results presented in this paper was made possible by OpenINTEL, a joint project of SURFnet, the University of Twente and SIDN. This research was partly funded by SIDN Fonds.

References

[1] S. L. Pfleeger and G. Bloom, "Canning Spam: Proposed Solutions to Unwanted Email," *IEEE Security Privacy*, vol. 3, no. 2, 2005.

[2] J. Schultz, *Walking in a Winter Wonderland*, 2014. [Online]. Available: https://blogs.cisco.com/security/walking-in-a-winter-wonderland.

[3] F. Weimer, "Passive DNS Replication," in *Proc. of FIRST 2005*, 2005.

[4] B. Zdrnja, N. Brownlee, and D. Wessels, "Passive Monitoring of DNS Anomalies," in *Proc. of DIMVA 2007*, 2007.

[5] Bilge, Leyla and Kirda, Engin and Kruegel, Christopher and Balduzzi, Marco, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in *NDSS 2011*, 2011.

[6] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a Dynamic Reputation System for DNS," in *Proc. of the 19th USENIX Security*, 2010.

[7] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou II, and D. Dagon, "Detecting Malware Domains at the Upper DNS Hierarchy," in *Proc. of the 20th USENIX Security*, 2011.

[8] R. Perdisci, I. Corona, D. Dagon, and W. Lee, "Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces," in *2009 Annual Computer Security Applications Conference (ACSAC '09)*, 2009, pp. 311–320.

[9] M. Konte, N. Feamster, and J. Jung, "Dynamics of Online Scam Hosting Infrastructure," in *Proc. of PAM 2009*. 2009.

[10] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the Initial DNS Behavior of Malicious Domains," in *Proc. of the 2011 ACM IMC*, 2011.

[11] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the Potential of Proactive Domain Blacklisting," in *LEET 2010*, 2010.

[12] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration," in *Proc. of the 2016 ACM CCS*, 2016.

[13] N. A. Syed, A. G. Gray, N. Feamster, and S. Krasser, "Snare: Spatio-temporal Network-level Automatic Reputation Engine," Georgia Institute of Technology - CSE Technical Reports - GT-CSE-08-02, Tech. Rep., 2008.

[14] G. C. M. Moura, A. Sperotto, R. Sadre, and A. Pras, "Evaluating third-party Bad Neighborhood blacklists for Spam detection," in *2013 IFIP/IEEE Int. Symp. on Integrated Network Management (IM 2013)*, May 2013.

[15] D. Yamakawa and N. Yoshiura, "Analysis of spam mail sent to Japanese mail addresses in the long term," in *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, Apr. 2010.

[16] A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1606.01042.

[17] P. Owezarski, "Unsupervised Classification and Characterization of Honeypot Attacks," in *CNSM 2014*, 2014, pp. 10–18.

[18] J. J. Santanna, R. d. O. Schmidt, D. Tuncer, J. de Vries, L. Z. Granville, and A. Pras, "Booter Blacklist: Unveiling DDoS-for-hire Websites," in *CNSM 2016*, 2016, pp. 144–152.

[19] S. Youn and D. McLeod, "A Comparative Study for Email Classification," in *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, K. Elleithy, Ed. Springer Netherlands, 2007.

[20] H. Drucker, D. Wu, and V. N. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, 1999.

[21] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, "Stacking classifiers for anti-spam filtering of e-mail," *CoRR*, 2001. [Online]. Available: http://arxiv.org/abs/cs.CL/0106040.

[22] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, 2016.

[23] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification Using String Kernels," *J. Mach. Learn. Res.*, 2002.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, 2011.

[25] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in *Int. Joint Conf. on Artificial Intelligence*, 1995.

[26] Scikit-Learn, *sklearn.ensemble.AdaBoostClassifier*. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html.