Early Detection of Spam Domains with Passive DNS and SPF

Simon Fernandez, Maciej Korczyński, Andrzej Duda

Early Detection of Spam Domains with Passive DNS and SPF

Simon Fernandez, Maciej Korczyński, and Andrzej Duda

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
{first.last}@univ-grenoble-alpes.fr

Abstract. Spam domains are sources of unsolicited mails and one of the primary vehicles for fraud and malicious activities such as phishing campaigns or malware distribution. Spam domain detection is a race: as soon as the spam mails are sent, taking down the domain or blacklisting it is of relative use, as spammers have to register a new domain for their next campaign. To prevent malicious actors from sending mails, we need to detect them as fast as possible and, ideally, even before the campaign is launched.

In this paper, using near-real-time passive DNS data from Farsight Security, we monitor the DNS traffic of newly registered domains and the contents of their `TXT` records, in particular, the configuration of the Sender Policy Framework, an anti-spoofing protocol for domain names and the first line of defense against devastating Business Email Compromise scams. Because spammers and benign domains have different SPF rules and different traffic profiles, we build a new method to detect spam domains using features collected from passive DNS traffic.

Using the SPF configuration and the traffic to the `TXT` records of a domain, we accurately detect a significant proportion of spam domains with a low false positives rate demonstrating its potential in real-world deployments. Our classification scheme can detect spam domains before they send any mail, using only a single DNS query and later on, it can refine its classification by monitoring more traffic to the domain name.

Keywords: Spam detection · SPF · Passive DNS · Machine Learning

1 Introduction

For years, malicious mails have been representing a significant technical, economic, and social threat. Besides increasing communication costs and clogging up mailboxes, malicious mails may cause considerable harm by luring a user into following links to phishing or malware distribution sites.

Typically, malicious actors run campaigns with instant generation of a large number of mails. Hence, their detection is a race: if we want to prevent their malicious activity, we need to detect spam domain names as soon as possible, blacklist and block them (at the registration level). Once the campaign is over, domain blacklisting is less effective because the recipients have already received mails.

Early detection of spam domains that generate malicious mails is challenging. One of the approaches is to leverage the Domain Name System (DNS) that maps domain names to resource records that contain data like IP addresses. We can use DNS traffic and domain name characteristics to compute features for training and running machine learning detection algorithms, even if malicious actors may try to hide their traces and activities, and avoid domain takedown [12, 30]. The main difference between various algorithms is the set of features used to train and run classifiers. The features mainly belong to four categories: i) lexical: domain names, randomness of characters, or similarity to brand names [1, 3, 5, 19, 22, 23, 34], ii) domain and IP address popularity: reputation systems based on diversity, origin of queries, or past malicious activity [1, 2, 9, 16, 23, 24, 31]), iii) DNS traffic: number of queries, their intensity, burst detection, or behavior changes [5, 24]), and iv) WHOIS: domain registration patterns [9, 23, 27].

In this paper, we propose a scheme for early detection of spam domains, even before they send a single mail to a victim. It is based on the domain SPF (Sender Policy Framework) rules and traffic to the `TXT` records containing them.

SPF rules are means for detecting forged sender addresses—they form the first line of defense in the case of, for instance, Business Email Compromise scams that represented over $1.8 billion USD of losses in 2020 [6]. As malicious actors generally use newly registered domains for sending mails, they also configure the SPF rules for their domains to increase their reputation and thus avoid proactive detection. We have discovered that the content of the SPF rules and traffic to the `TXT` records containing them are different for malicious and benign domains. We have used these features to design a domain classifier algorithm that can quickly detect spam domains based on passive DNS traffic monitoring [8]. With low false positive rate and high true positive rate, our scheme can improve existing real-time systems for detecting and proactively blocking spam domains using passive DNS data.

The rest of the paper is organized as follows. Section 2 provides background on SPF and spam campaigns. Section 3 presents the proposed scheme. Sections 4 and 5 introduce the classification algorithms and present their results. We discuss other related approaches in Section 6 and Section 7 concludes the paper.

2 Background

In this section, we describe the SPF protocol and the mail delivery process, highlighting the steps during which we gather features to detect malicious activity.

2.1 Sender Policy Framework (SPF)

The Sender Policy Framework (SPF) [17] is a protocol used to prevent domain (mail) spoofing. Figure 1 presents the procedure for sending mails and SPF verification. Alice (sender) sends a benign mail to Bob (receiver). Mallory (attacker) wants to send a mail that impersonates Alice to Bob. Mallory and Alice use their respective servers (`mallory.com` and `alice.com`) to send mails.

An effective anti-spoofing mechanism needs to differentiate the Mallory message from the benign Alice mail. The current first lines of defense to protect

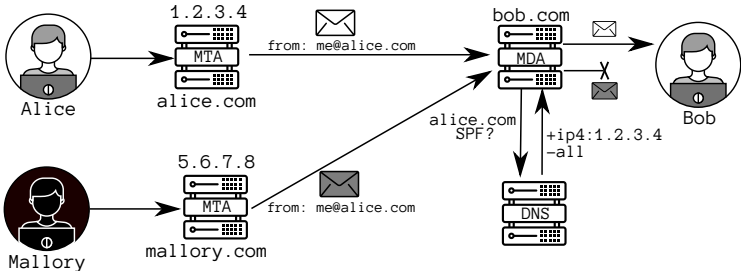


Fig. 1: Sending mails with SPF verification.

users from spoofed mails include SPF [17], DKIM [20], and DMARC [21]. SPF is a set of text-form rules in `TXT` DNS resource records specifying a list of servers allowed to send mails on behalf of a specific domain. During mail delivery over the SMTP protocol, the recipient server authenticates the sender Mail Transfer Agent (MTA) by comparing the given `MAIL FROM` (or `HELO`) identity and the sender IP address with the content of the published SPF record.

In our example, the Mail Delivery Agent (MDA) on the Bob's server queries the DNS for a `TXT` record of the sending domain (`alice.com`). This record contains the SPF rule of `alice.com` and specifies which IP addresses can send mails on behalf of this domain. The mail from Alice comes from a whitelisted server, so it gets delivered. The Mallory's server was not whitelisted, so the (spoofed) mail is rejected.

A valid SPF version 1 record string must begin with `v=spf1` followed by other SPF entries with the following structure: `<qualifier><mechanism>[:<target>]`. The mail sender is matched with the `<mechanism>:<target>` part; the output is determined by the `<qualifier>`. Four types of `<qualifier>` are possible: `PASS (+)` (the default mechanism), `NEUTRAL (~)`, `SOFTFAIL (?)`, `FAIL (-)`. The most common SFP mechanisms are the following:

`ip4, ip6` – the sender IP address matches the predefined IP address or the subnetwork prefix,
`a, mx` – the domain has an `A` (or `MX`) record that resolves to the sender IP address,
`ptr` – a verified reverse DNS query on the sender IP address matches the sending domain (not recommended by RFC 7208 [17] since April 2014),
`exists` – the domain has an `A` record,
`include` – use the rules of another domain,
`all` – the default mechanism that always matches.

To illustrate the operation of SPF rules, let us consider the following configuration for `example.com` domain: `v=spf1 a ip4:192.0.2.0/24 -all` where the `A` record (`example.com A 198.51.100.1`) is stored in DNS. The SPF rule states that only a host with the IP address of `198.51.100.1` (the `a` mechanism) or machines in the `192.0.2.0/24` subnetwork (the `ip4` mechanism) are permitted senders, all others are forbidden (the `-all` mechanism).

2.2 Life Cycle of a Spam Campaign

Most spam campaigns follow the same life cycle presented below.

Domain registration. As most mail hosting companies deploy tools to prevent their users from sending spam, malicious actors need to register their own domains to send spam. To run multiple campaigns, spammers usually register domains in bulk [10]. Once the domains are registered, spammers configure zone files and fill the corresponding resource records in the DNS.

Configuration of anti-spoofing mechanisms. To use SPF, DMARC, or DKIM, each domain must have a `TXT` resource record describing which hosts can send a mail on their behalf and deploying keys to authenticate the sender. Even if DMARC is still not widely used, many benign domains deploy SPF [7, 26, 28]. Thus, a mail from a domain without SPF configuration is likely to be flagged as spam (especially when combined with other indicators of malicious intent). To appear as benign as possible, spammers fill in at least the SPF rule in the `TXT` record. Our scheme extracts most of the features for detecting spam at this step because the SPF records of spam domains are generally different from the configurations of benign domains and even if a given domain has not yet sent a single mail, we can access its SPF rules and detect suspicious configurations. The SPF rules can be actively fetched by sending a `TXT` query to the domain (e.g., newly registered), but to avoid active scanning, we have chosen to use passive DNS to analyze `TXT` requests. In every detected spam campaign, we observe at least one `TXT` query that may originate from a spammer testing its infrastructure.

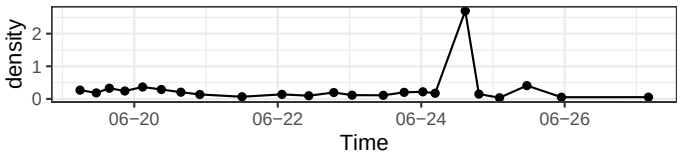


Fig. 2: Density of DNS `TXT` traffic to a spam domain (`promotechmail.online`)

Spam campaign. When a mail server receives a mail, it tries to resolve the `TXT` record of the sending domain to get its SPF rule and checks for possible sender forgery. During a spam campaign, spammers send mails to many servers across the world. At the beginning of a campaign, the (validating) mail servers will all try to retrieve the `TXT` DNS record of the sender domain almost at the same time. Therefore, we expect to observe a surge in queries for `TXT` records. Figure 2 presents traffic density (corresponding to the number of DNS queries over time, defined precisely later) to a spam domain detected during our study. The burst in the number of queries during a time window of less than 24 h, then traffic dropping and never rising again is the typical profile of spammers.

Detection, blacklisting, and cleanup. When spam mails reach the targets, security experts and spam detection algorithms parsing the mail content and its headers flag the sending domain as a spamming source and may report it to domain blacklists like SpamHaus [32] or SURBL [33]. When a domain appears on a blacklist, mail servers will likely drop mails from it. Future spam campaigns from this domain will be unsuccessful, so it becomes useless for spammers. Hosting services may also suspend the sending server whereas domain registrars may

take down the spam domain as it often violates their terms of service and is considered as DNS abuse [4, 19]. Once the domain is blacklisted (or taken down), spammers may just acquire another one and repeat the previous steps.

When looking for spammers, timing is the key: the sooner we detect a spamming domain, the fewer mails it can send, and if an algorithm only detects a spam mail upon reception, it means that the campaign has started and reached some of the targets. This observation was the motivation for our scheme for early detection of spamming domains even before the start of a spam campaign.

3 Scheme for Early Detection of Spam

In this section, we present the proposed scheme. It takes advantage of passive DNS data to obtain the SPF rules for a given domain and the frequency of the queries to retrieve them.

3.1 Data Source: Passive DNS

Passive DNS consists of monitoring DNS traffic by *sensors* usually deployed above recursive resolvers to monitor queries between a local resolver and authoritative name servers [35]. Locally observed queries are aggregated into feeds available for analyses. In this work, we have used the near-real-time Farsight SIE Passive DNS channel 207 [8] to obtain DNS traffic data for the `TXT` records and SPF rules for each domain. We extract the following fields: the queried domain, the record type, the answer from the authoritative server, a time window, and the number of times a given query was observed during the time window.

To be effective, the scheme must analyze unencrypted DNS traffic. Therefore, it is not suitable when using the DNS over TLS (DoT) [13] or DNS over HTTPS (DoH) [11] standards that encrypt user DNS queries to prevent eavesdropping of domain names. To monitor such traffic, the scheme would have to be implemented, e.g., in public recursive resolvers providing DoT or DoH services.

3.2 Features Based on SPF Rules

The SPF configuration for a given domain is stored in the `TXT` record of the domain. Since most mail hosting services provide a default SPF records for their customers, many domains share the same SPF rules. Nevertheless, some domains use custom SPF rules that whitelist specific servers. We have focused on the similarities of domains: two domains that use the same custom SPF rules and whitelist the same IP addresses are likely to be managed by the same entity. Therefore, if one domain starts sending spam, it is reasonable to consider that the domains sharing the same SPF rules are likely to be (future) spammers.

We have analyzed the SPF configuration of spam and benign domains to see if they differ (we later discuss ground truth data in Section 4.1). Figure 3 shows that benign and spam domains do not necessarily use the same rules. For example, benign domains more frequently use the `+include` mechanism while spammers `+ptr`.

We presume that legitimate domains, hosted by major mail hosting providers, are more likely to have default configurations with the `+include` mechanism to

indicate that a particular third party (e.g., a mail server of the provider) is authorized to send mails on behalf of all domains (e.g., in a shared hosting environment). Spam domains may use custom mail servers instead, thus they are more likely to whitelist the IP addresses of their servers with, for instance, the `+ip4` mechanism. We suspect that in some cases spammers may not want to reveal the IP addresses of hosts sending spam. Therefore, they may use the `+all` mechanism (that accepts mails from all hosts) relatively more than legitimate domains whose administrators are concerned about rejecting spam mails from unauthorized host. Finally, the `+ptr` mechanism is marked as "do not use" since April 2014 by RFC 7208 [17]. Major hosting providers seem to follow this recommendation, but individual spammers may not have changed their practices and continue to use this outdated but still supported mechanism.

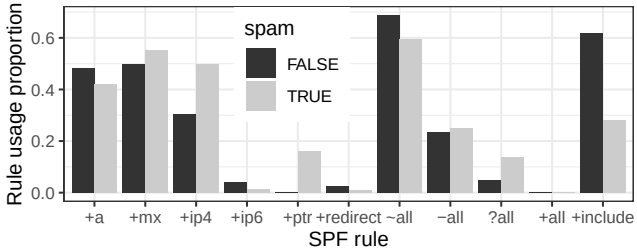


Fig. 3: Usage proportion of SPF rules for benign and spamming domains

For each domain, we compute the number of occurrences of each mechanism in its rule to generate the set of SPF features. Because not all possible combinations of qualifiers and mechanisms are actually used, we have selected the sets of qualifiers and mechanisms that appear in more than 0.1% of domains to avoid overfitting, which leaves us the ones presented in Figure 3.

3.3 Graph Analysis of SPF Rules

Some SPF rules point to an IP address or a subnetwork prefix (like `ip4` and `ip6`) and some point to domain names (like `include` and sometimes `a` and `mx`). We build the relationship graph between domains and IP ranges as shown in Figure 4. For example, the edge between node A (`a.org`) and node B (`b.com`) reflects the fact that node B has an SPF rule that points to node A. The edge between `b.com` and `192.0.2.1` represents the fact that this IP address is used in the `+ip4` rule in the `b.com` SPF configuration.

This graph is built and updated in near real time: nodes and edges are added when domains with SPF data appear in the passive DNS feed, and spam domains (marked in red in Figure 4) are added or deleted from blacklists (SpamHaus and SURBL in our scheme). Thus, over time, the graph becomes more complete, providing more precise relationships and features for domain classification.

We have analyzed different structures in the graph built from our dataset and detected distinctive patterns. Figure 5 shows three examples of the observed

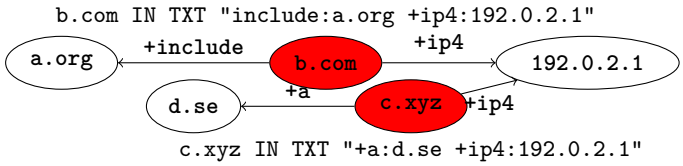
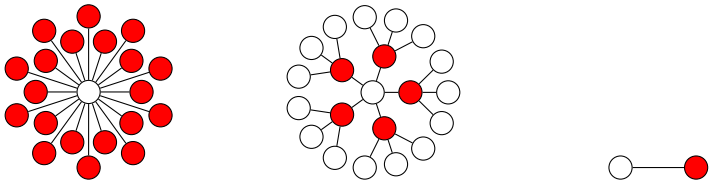


Fig. 4: Example of a relationship graph derived from SPF rules

structure types to illustrate some typical SPF configuration relationship graphs for spam domains.Red nodes represent spamming domains and white nodes correspond to the targets of their SPF rules. Figure 5a shows the pattern in which multiple spam domains share the same configuration: they have a rule targeting the same IPv6 network (these domains are likely to be managed by the same entity). Figure 5b presents spam domains that have an `include` mechanism that points to the same domain and exactly three other custom targets that no other domain uses (this is the case when domains are hosted by a hosting provider that provides an SPF configuration for inclusion by its clients). Finally, many spam domains have rules like in Figure 5c in which a domain has a single target (a custom IP address) that no other domain uses.



(a) Ball shape (b) Tree shape (c) One-to-One

Fig. 5: SPF relation graph for spam domains

The study of these structures can highlight potential spam domains. In our dataset, we found structures like in Figure 5a or Figure 5b in which dozens of domains used the same rule and the majority of them appeared on spam blacklists. As such, it is reasonable to assume that the remaining domains are likely to have not yet been detected or are not yet active spam domains.

To detect the structures indicating spam domains, we have defined two unique features describing the properties of domains in the relationship graph.

Toxicity. We define the *toxicity* of a node as the proportion of its neighbors that are flagged as spam in the graph, or 1 if the domain itself is flagged as spam. With this metric, SPF targets used by known spammers get a high value of *toxicity*. To detect the domains that use rules with high *toxicity* targets, we compute the *Max Neighbor Toxicity*: the maximum *toxicity* amongst all the targets of a domain. This way, if a domain has a target mainly used by spammers, its *Max Neighbor Toxicity* is high.

Neighbor Degree. For each node, we look at the degrees of its neighbors: is it connected to highly used domains and IP addresses? Or, is it using cus-

tom targets that no other domain uses? We expect spamming domains to more likely use custom targets that no other domains use (with a small degree in the graph) like in Figure 5c, compared to benign domains that would use the default configurations of the hosting service and share the same targets as many other domains (with a high degree in the graph).

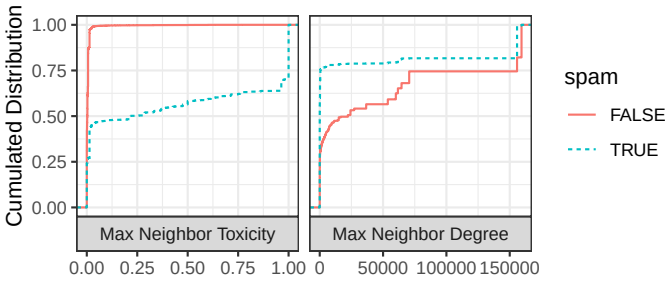


Fig. 6: Cumulative distributions of Max Neighbor Toxicity and Max Neighbor Degree for spamming and benign domains.

Figure 6 shows that the expected differences of *Max Neighbor Toxicity* and *Max Neighbor Degree* between spammers and benign domains match our hypothesis: spammers are more likely to use targets shared by some other spammers and are more likely to use custom targets with low degrees in the graph.

3.4 Time Analysis of Traffic to DNS `TXT` Records

When a domain starts a spam campaign, we expect multiple servers to query DNS for the **`TXT`** record of the sender domain to check its SPF configuration. Therefore, we can observe an unusual number of queries related to the (newly registered) domain. The passive DNS feed we use contains aggregated queries over a given time window: when a DNS query is detected by a sensor, it is inserted in an aggregation buffer with the insertion timestamp. The subsequent identical queries only increase a counter in the buffer. When the buffer is full, the oldest inserted queries are flushed out, yielding an aggregated message with the query, the answer from the authoritative server, and three extra fields: **`time_first`**, **`time_last`**, and **`count`** meaning that the query was seen **`count`** times during the time window from **`time_first`** to **`time_last`**.

From these aggregated messages, we compute the traffic density by dividing the number of queries (in the **`count`** field) by the window duration, and then, dividing this value by the time between the end of the window and the end of the previous window to take into account the time windows in which there is no traffic. The resulting formula is the following:

$$density(i) = \frac{\texttt{count}}{\texttt{time\_last} - \texttt{time\_first}} \times \frac{1}{message\_end(i) - message\_end(i-1)}$$

For a more in-depth definition of the density and an explanation on how we handled overlapping windows, see Appendix A.

Max Variation. To detect large variations in density, we compute the *Max Variation* feature defined as the maximum density variation during 24 h. Domains with a slowly increasing traffic have a low *Max Variation* and those with a spike in the number of `TXT` queries, a high *Max Variation*. We compute two versions of this feature: i) the *Global Max Variation*, using the same time steps to compare all domains and ii) the *Local Max Variation* in which a custom time step is computed for each domain. See Appendix A for more details about the difference between these features.

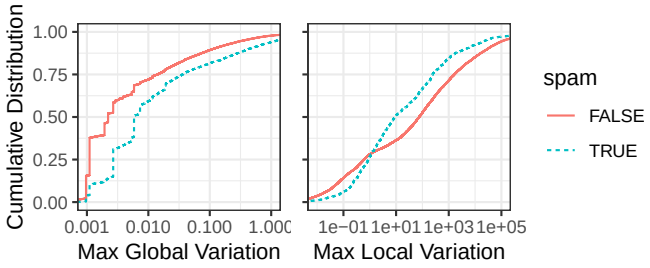


Fig. 7: Cumulative distribution of Max Variation (log scale x-axis)

Figure 7 presents the cumulative distribution of the two features. As expected, we observe that spam domains have a relatively higher *Max Global Variation* when all domains share the same time steps.

However, when we look at the *Max Local Variation*, we observe that benign domains tend to have a higher variation. The distributions are different because this feature is close to the average density variation: domains with a lot of traffic variation and small windows will have a higher *Local Variation*, whereas spam domains with almost no traffic except for a few spikes will have a lower *Local Variation* due to long periods of inactivity before a spike.

4 Classifiers

In this section, we present the classifiers used for the detection of spam based on the proposed features.

4.1 Ground Truth

We have taken the precaution of carefully selecting the domains in our ground truth. We recorded four months (between May and August 2021) of passive DNS traffic to `TXT` records from Farsight Security [8]. Because most spam domains are newly registered and discarded as soon as they are blacklisted, we only considered newly registered domains. From the ICANN Central Zone Data Service (CZDS) [14], we have built a list of new domains by computing the difference between consecutive versions of each generic Top Level Domain (gTLD) zone files. Appendix C provides the general statistics of the collected dataset.

Using SURBL [33] and SpamHaus [32] spam blacklists, we have identified all domains (in near-real time) in our database flagged by one of these sources. Spam blacklists are not perfect and sometimes they may flag benign domains

as spam. Therefore, to obtain reliable ground truth, we added an extra layer of verification: a domain is labeled as

- **benign** if it has not been blacklisted and has been active during the entire period of the study (and has a valid `A` and `NS` records), or
- **malicious** if it was blacklisted by SURLB or SpamHaus and was taken down.

With these criteria, our ground truth dataset contained 37,832 non-spam and 2,392 spam domains.

4.2 Classifier

For spam detection, it is crucial to keep the True Negative[1] Rate (TPR) as high as possible to avoid flagging benign domains as spam. Once a True Negative Rate of at least 99% is achieved, we maximize the True Positive[2] Rate (TPR) to detect as many spam domains as possible. To compare classification results we use true negative and true positive rates, and the F1-score as described in Appendix B. We explored multiple classifiers and parameters with Weka [36], then implemented two of them with the `scikit-learn` [29] Python library, for better benchmarking. Two classifiers that performed the best are:

C4.5 or J48: a decision tree able to describe non-linear relations between features. It highlights complex conditional relations between features.

Random Forest: a set of multiple decision trees with a voting system to combine their results. Its drawback is low explainability.

Table 1: Features used by the classifiers

Category	Feature	Outcome
SPF Rules	Number of...	
	`+all, +mx, +ptr, -all`	Malicious
	`+a, +include, +redirect, ~all`	Benign
	`+ip4, +ip6, ?all`	Mixed[3]
SPF Graph	Max Neighbor Degree	Benign
	Max Neighbor Toxicity	Malicious
Time Analysis	Max Global Variation	Malicious
	Max Local Variation	Benign

We use the k-fold cross-validation technique with k set to 5 (see Appendix B for more information). The number of spam domains in our ground truth dataset represents less than 10% of all domains. The decision tree algorithms are not suitable for classification problems with a skewed class distribution. Therefore, we have used a standard class weight algorithm for processing imbalanced data [37] implemented in the `scikit-learn` Python library [29].

Table 1 summarizes the features used by the classifiers and whether they indicate maliciousness or benignness of the domain.

[1] True Negative: non-spam domain correctly classified as such

[2] True Positive: spam domain correctly classified as malicious

[3] Depends on how many times the rule is present in the configuration

5 Classification Results

We evaluate the efficiency of the classifiers with two sets of features: i) the *static* set without the time analysis features (Max Variation) and ii) the *static + dynamic* set that includes both static and the time analysis features. We have distinguished between the sets because even if the efficiency is lower without the time analysis features, we can get the static features (SPF configuration and graph properties) from a single `TXT` query to the target domain allowing for a rapid detection of most spam domains. Then, we can refine the classification by adding the time based features that are more robust against evasion techniques but require more time to detect spam domains.

5.1 Performance Evaluation

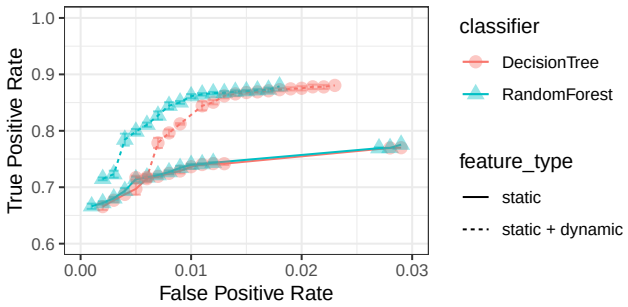


Fig. 8: ROC curve for different classifiers on two sets of features

Figure 8 compares the Receiver Operating Characteristic (ROC) curves of each classifier for two sets of features (to see better the differences in performance, we zoom into high values of TPR). When training the classifiers, we change the weight of the spam class to change the reward of accurately finding a spam domain. If the spam class weight is low, the classifier will be less likely to risk getting a false positive. On the contrary, if the spam class weight is high, the classifier gets higher reward if it accurately flags a spam domain. Therefore, the classifier will "take more risks", reducing its TNR to increase TPR. If we require the False Positive Rate (benign domains flagged as spam) under 1%, the Random Forest is the best algorithm reaching a True Positive Rate of 74% using only the static set and 85% once we add the time analysis features.

Figure 9 illustrates how long we need to monitor a domain so that the classifiers reach their best efficiency. Over time, we observe traffic to each domain and the time analysis features get more precise (until one week), which improves classification. Both classifiers reach almost the best detection performance (computed as the F1-score) after observing a domain for one day.

5.2 Detection Time

The static results (labeled as 0H in Figure 9) show the efficiency of the scheme when a single `TXT` request is observed. In this case, the classifier has no time

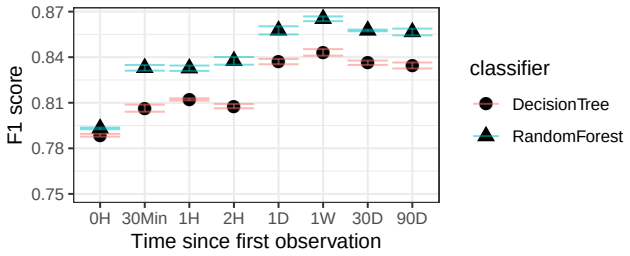


Fig. 9: F1-score of classifiers after the first appearance of each domain

properties of the traffic and only uses the static features (SPF Rules and SPF Graph). We can replace passive detection of SPF Rules with active DNS scans (assuming we have a list of newly registered domain names, which is generally the case for legacy and new gTLDs but not for the vast majority of ccTLD [4, 18]): by actively querying the `TXT` records of new domains and classifying them based on their SPF configuration and formed relationships. Then, over time, as we passively observe traffic to the domain records, the performance of the classifier improves achieving very good results after 30 minutes (F1-score of 0.83) of monitoring (with Random Forest) in comparison with the F1-score of 0.86 after one day.

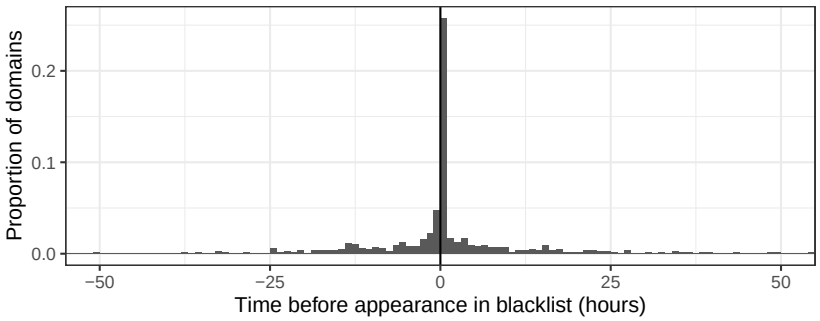


Fig. 10: Time before detected spam domains appear in commercial blacklists

Using only static features, we compared the spam domain detection speed of our scheme with two commercial blacklists (SpamHaus and SURBL). In Figure 10, we plotted the time elapsed between the detection by our scheme and the appearance of domains in the blacklists (with an hourly granularity). We limited the graph to 50 hours, but considerable number of domains only appear in the commercial blacklists weeks after we detect them. Positive values mean that our scheme was faster: for 70% of the detected spam domains, our scheme was faster than the commercial blacklists. However, 26% of the domains detected by our scheme appear in the commercial blacklists in the following hour, whereas 30% of the domains are detected more than 24 hours in advance. The negative values represent domain names where our scheme was slower than the commercial blacklists: 30% of the domains were already in the blacklists when they were observed in our passive DNS feed for the first time and classified as spam.

5.3 Feature Importance

The importance of each feature was computed by looking at how selective the feature was in the Random Forest classifier [29]. The importance of each feature and each category is described in Table 2. It is not a surprise that the Maximum Neighbor Toxicity is by far the most important feature: a domain whitelisting the same IP addresses and domains as a known spamming domain is very likely to be managed by spammers. The most important SPF rule for classification is `+ptr`: as we discussed in Section 3.2, this rule is almost never used by benign domains (following the RFC 7208 recommendations). Lastly, the Global Max Variation is the most important dynamic feature: massive increases in the number of queries to a domain is a distinctive trait of spamming domains, as presented in Section 2.2, but this feature is only useful after the start of the spam campaign.

Table 2: Importance of each feature for the Random Forest classifier

Feature	Importance
SPF Graph features	**0.574515**
`neighbor_max_toxicity`	0.463689
`neighbor_max_degree`	0.110826
SPF Rules features	**0.232846**
`+ptr`	0.100481
`+a`	0.029005
`+ip4`	0.028789
`+mx`	0.021006
`+include`	0.017561
`?all`	0.013728
`~all`	0.011522
Other rules	< 0.01
Time Analysis features	**0.192638**
`global_max_variation_24h`	0.122167
`local_max_variation_24h`	0.036828
`global_max_triggers_24h`	0.022380
`local_max_triggers_24h`	0.011263

6 Related Work

The four main categories of features used to detect malicious domains are the following: i) Lexical: domain name, randomness of characters, or similarity to brand names [1, 5, 22, 23, 27], ii) Domain and IP address popularity: reputation systems based on diversity, origin of queries, or past malicious activity [1, 2, 9, 23, 27, 31], iii) DNS traffic: number of queries, intensity, burst detection, behavior changes [5, 24], iv) WHOIS (domain registration data): who registered a given domain[4], when, and at which registrar [9, 23, 27]. Other methods develop specific

[4] not available after the introduction of the General Data Protection Regulation (GDPR) and the ICANN Temporary Specification [15].

features extracted from the content of mails: size of the mail, links, or redirections [25, 27]. With the selected features, machine learning algorithms classify malicious and benign domains.

With respect to the methods that work on passive data such as Exposure [5] that need some time to detect abnormal or malicious patterns, we focus on early detection of spam domains. Exposure for instance, needs around a week of observation before possible detection, while we achieve a F1-score of 79% based on a single DNS query. Our scheme can be applied at early stages of a domain life cycle: using passive (or active) DNS, we can obtain SPF rules for newly registered domains and classify them immediately, or wait until we detect `TXT` queries to that domain and refine the classification using hard-to-evade temporal features.

Other methods generally try to detect abnormal or malicious patterns at later phases of the domain life cycle. Schemes based on content or long period traffic analysis may reach high efficiency but generally cannot run before or at the beginning of an attack. Schemes using lexical and popularity features can run preemptively but may have reduced efficiency, compared to dynamic schemes.

Our scheme may complement other approaches that aim at detecting spam during other phases in the life cycle of spam campaigns and other algorithms that rely on a variety of different features.

7 Conclusion

In this paper, we have proposed a new scheme for early detection of spam domains based on the content of domain SPF rules and traffic to the `TXT` records containing them. With this set of features, our best classifier detects 85% of spam domains while keeping a False Positive Rate under 1%. The detection results are remarkable given that the classification only uses the content of the domain SPF rules and their relationships, and hard to evade features based on DNS traffic. The performance of the classifiers stays high, even if they are only given the static features that can be gathered from a single `TXT` query (observed passively or actively queried).

With a single request to the `TXT` record, we detect 75% of the spam domains, possibly before the start of the spam campaign. Thus, our scheme brings important speed of reaction: we can detect spammers with good performance even before any mail is sent and before a spike in the DNS traffic. To evaluate the efficiency of the proposed approach based on passive DNS, we did not combine the proposed features with other ones used in previous work like domain registration patterns [9, 23, 27]. In practical deployments, the classification can be improved by adding other features based on, e.g., the content of potentially malicious mails or the lexical patterns of the domain names.

The features used in our scheme yield promising results, so adding them to existing spam detection systems will increase their performance without large computation overhead as SPF data can easily be extracted from near-real-time passive DNS feeds already used in some schemes.

Acknowledgements

We thank Paul Vixie and Joe St Sauver (Farsight Security), the reviewers, our shepherd, and Sourena Maroofi for their valuable and constructive feedback. We thank Farsight Security for providing access to the passive DNS traffic as well as SpamHaus and SURBL for the spam blacklists. This work was partially supported by the Grenoble Alpes Cybersecurity Institute under contract ANR-15-IDEX-02 and by the DiNS project under contract ANR-19-CE25-0009-01.

References

1. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a Dynamic Reputation System for DNS. In: USENIX Security (2010)
2. Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, N., Dagon, D.: Detecting Malware Domains at the Upper DNS Hierarchy. In: USENIX Security (2011)
3. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In: USENIX Security (2012)
4. Bayer, J., Nosyk, Y., Hureau, O., Fernandez, S., Paulovics, I., Duda, A., Korczyński, M.: Study on Domain Name System (DNS) Abuse Appendix 1 – Technical Report. Tech. rep. (2022). https://doi.org/10.2759/473317
5. Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C.: Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. ACM Transactions on Information and System Security **16**(4), 1–28 (2014)
6. Crime Complaint Center (IC3), FBI: Internet Crime Report. `https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf` (2020)
7. Deccio, C.T., Yadav, T., Bennett, N., Hilton, A., Howe, M., Norton, T., Rohde, J., Tan, E., Taylor, B.: Measuring email sender validation in the wild. In: CoNEXT. ACM (2021)
8. Farsight Inc.: Farsight SIE, `https://www.farsightsecurity.com/solutions/security-information-exchange/`
9. Hao, S., Kantchelian, A., Miller, B., Paxson, V., Feamster, N.: PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. In: ACM CCS (2016)
10. Hao, S., Thomas, M., Paxson, V., Feamster, N., Kreibich, C., Grier, C., Hollenbeck, S.: Understanding the Domain Registration Behavior of Spammers. In: IMC (2013)
11. Hoffman, P.E., McManus, P.: DNS Queries over HTTPS (DoH) (2018). https://doi.org/10.17487/RFC8484, `https://datatracker.ietf.org/doc/rfc8484`
12. Holz, T., Gorecki, C., Rieck, K., Freiling, F.C.: Measuring and Detecting Fast-Flux Service Networks. In: NDSS (2008)
13. Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., Hoffman, P.E.: Specification for DNS over Transport Layer Security (TLS) (2016), `https://datatracker.ietf.org/doc/rfc7858`
14. ICANN: ICANN: Centralized Zone Data Service. `https://czds.icann.org`
15. ICANN: Temporary Specification for gTLD Registration Data (May 2018), `https://www.icann.org/resources/pages/gtld-registration-data-specs-en`

16. Kheir, N., Tran, F., Caron, P., Deschamps, N.: Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists. In: ICT Systems Security and Privacy Protection, IFIP Advances in Information and Communication Technology, vol. 428, pp. 1–14. Springer (2014)
17. Kitterman, S.: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1 (2014), https://datatracker.ietf.org/doc/rfc7208
18. Korczyński, M., Tajalizadehkhoob, S., Noroozian, A., Wullink, M., Hesselman, C., Van Eeten, M.: Reputation Metrics Design to Improve Intermediary Incentives for Security of TLDs. In: IEEE EuroS&P (2017)
19. Korczyński, M., Wullink, M., Tajalizadehkhoob, S., Moura, G.C.M., Noroozian, A., Bagley, D., Hesselman, C.: Cybercrime After the Sunrise: A Statistical Analysis of DNS Abuse in New gTLDs. In: ACM AsiaCCS (2018)
20. Kucherawy, M., Crocker, D., Hansen, T.: DomainKeys Identified Mail (DKIM) Signatures (2011). https://doi.org/10.17487/RFC6376, https://datatracker.ietf.org/doc/rfc6376
21. Kucherawy, M., Zwicky, E.: Domain-based Message Authentication, Reporting, and Conformance (DMARC) (2015), https://datatracker.ietf.org/doc/rfc7489
22. Le Pochat, V., Van Goethem, T., Joosen, W.: A Smörgåsbord of Typos: Exploring International Keyboard Layout Typosquatting. In: IEEE SPW (2019)
23. Le Pochat, V., Van hamme, T., Maroofi, S., Van Goethem, T., Preuveneers, D., Duda, A., Joosen, W., Korczyński, M.: A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints. In: NDSS (2020)
24. Lison, P., Mavroeidis, V.: Neural Reputation Models Learned from Passive DNS Data. In: 2017 IEEE International Conference on Big Data (2017)
25. Marchal, S., Armano, G., Grondahl, T., Saari, K., Singh, N., Asokan, N.: Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application. IEEE Transactions on Computers **66**(10), 1717–1733 (2017)
26. Maroofi, S., Korczyński, M., Duda, A.: From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains. In: TMA (2020)
27. Maroofi, S., Korczyński, M., Hesselman, C., Ampeau, B., Duda, A.: COMAR: Classification of Compromised versus Maliciously Registered Domains. In: IEEE EuroS&P. pp. 607–623 (2020)
28. Maroofi, S., Korczyński, M., Hölzel, A., Duda, A.: Adoption of email anti-spoofing schemes: A large scale analysis. IEEE Transactions on Network and Service Management **18**(3), 3184–3196 (2021)
29. Pedregosa, F., *et al.*: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
30. Perdisci, R., Corona, I., Dagon, D., Lee, W.: Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In: ACSAC (2009)
31. Pochat, V.L., van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., Joosen, W.: Tranco: A research-oriented top sites ranking hardened against manipulation. Network and Distributed System Security Symposium, NDSS (2019), https://www.ndss-symposium.org/ndss-paper/tranco-a-research-oriented-top-sites-ranking-hardened-against-manipulation/
32. SpamHaus: The SpamHaus Project. https://www.spamhaus.org
33. SURBL: SURBL - URI reputation data. http://www.surbl.org
34. Wang, W., Shirley, K.: Breaking Bad: Detecting Malicious Domains Using Word Segmentation (2015), http://arxiv.org/abs/1506.04111
35. Weimer, F.: Passive DNS Replication. https://www.first.org/conference/2005/papers/florian-weimer-paper-1.pdf

36. Weka 3: Machine Learning Software in Java, `https://www.cs.waikato.ac.nz/ml/weka/`
37. Zhu, M., Xia, J., Jin, X., Yan, M., Cai, G., Yan, J., Ning, G.: Class weights random forest algorithm for processing class imbalanced medical data. IEEE Access **6**, 4641–4652 (2018)

Appendix

A Density Computation

Comparing the time windows of multiple domains in passive DNS data is a complex task: each window has a different size and we have no information on how the queries are spread inside it.

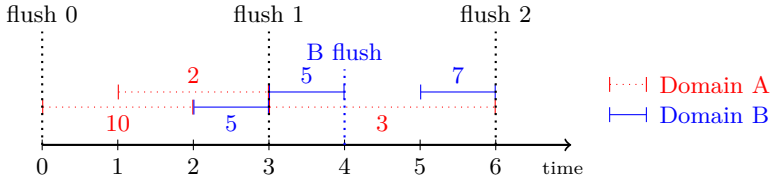


Fig. 11: Computation of traffic density from Passive DNS messages

The query density of multiple domains can only be compared if they are computed the same way, over the same time period. If a period starts or ends in the middle of a domain time window, we need to make an assumption about how the queries are spread inside the time window, to determine how many queries are inside the time period. However, we do not have such information so a period can only start and end at a timestamp that it is not included in any time window. We call those usable timestamps *flushes*. Then, the query density of a domain between two flushes is computed by measuring the time during which the domain was active, the total time between the flushes and the number of queries. For example, in Figure 11, between flush 0 and 1, Domain A has a `count` (total number of queries) of 12 and an `active_time` (total time covered by time windows) of 3, and Domain B has a `count` of 5, and an `active_time` of 1. If $flush(i)$ is the timestamp of the i-th flush, we define the density at time i as:

$$density(i) = \frac{count}{active\_time} \times \frac{1}{flush(i+1) - flush(i)}.$$

The first fraction represents the density of requests in the aggregated time window. The second fraction normalizes this value by the size of the flush window so that all domains have a comparable density, as the flushes are not evenly spread. Therefore, $density(0)$ for domain A is $12/3 \times 1/3 = 4/3$ and $5/1 \times 1/3 = 5/3$ for domain B.

For the *Max Global Variation*, the *flushes* are computed using the time windows of all domains in our ground truth (the numbered flushes in Figure 11). This results in fewer *flushes* but the traffic density between different domains can be compared (as they all use the same time steps). The *Max Local Variation* of a domain is computed using only the time windows of this domain to compute the *flushes* (numbered *flushes* plus domain *flushes* in Figure 11). The *Local Max Variation* uses more time steps so the density is more precise, but these time steps are different for each domain and have a tendency to reduce the detection of sudden bursts following a long inactivity window.

B Classifier Metrics and Algorithms

The performance of each classifier is measured with three metrics:

F1-score: $\frac{2TP}{2TP+FP+FN}$, with TP, FP and FN being respectively the number of True Positives, False Positives, False Negatives

True Positive Rate (TPR): $\frac{TP}{TP+FN}$: proportion of spam domains accurately flagged as spam.

True Negative Rate (TNR): $\frac{TN}{TN+FP}$: proportion of benign domains accurately flagged as benign.

To calculate performance metrics, we use the k-fold technique: the whole ground truth dataset is split in 5 equal parts. We select one fold for testing and train the model using the $k-1$ remaining folds. We repeat this process for each fold. Each metric is the average of the five iterations.

C Dataset Statistics

Table 3 shows the number of queries and unique domains at each data collection and analysis stage. The first step captures DNS `TXT` queries to newly registered domain names observed in the passive DNS feed. The next step retains only the `TXT` queries that contain valid SPF data. Then, we build ground truth with the approach described in Section 4.1.

Table 3: Number of queries and unique domains in the dataset at different stages

Stage	Queries	Unique domains	Spam domains
1. Traffic to new domains	399M	14M	0.8%
2. SPF traffic	36M	1.4M	1.5%
3. Ground truth	26M	40,224	5.9%

Table 4: Classification results for the Random Forest classifier on the ground truth dataset.

Our method \ Blacklists	Spam	Benign	**Total**
Spam	TP = 1 716	FP = 210	**1 926**
Benign	FN = 676	TN = 37 622	**38 298**
Total	**2 392**	**37 832**	**40 224**
	TPR	**TNR**	**F1-score**
	71.7%	99.4%	79.5%

D Classification Results

Table 4 shows the results of the Random Forest classifier using static and dynamic features (SPF Rules, SPF Graph and Time Analysis features). It corresponds to the model from Figure 8 with a TPR of 0.717 and FPR of 0.006. The second and third columns (Spam and Benign) represent how commercial blacklists (SpamHaus and SURBL) classified the domains (ground truth data), whereas the second and third row represent how our system classified the same domains. For example, in the table we can note that 676 domains were classified as Benign by our classifier, but they appear in the commercial blacklists—this represents the number of False Negatives (FN). The second part of the table shows the metrics used to evaluate our classifier (TPR, TNR, and F1-score) as described in Appendix B.