Looking Beyond the Horizon: Thoughts on Proactive Detection of Threats

OLIVIER VAN DER TOORN and ANNA SPEROTTO, University of Twente

The Internet exposes us to cyberthreats attacking information, services, and the Internet infrastructure itself. Such attacks are typically detected in a reactive fashion. The downside of this approach is that alerts of an attack are issued as it is happening. In this article, we advocate that the security community could benefit by complementing traditional reactive solutions with a proactive threat detection approach, as this would enable us to provide early warnings by analyzing and detecting threat indicators in actively collected data. By describing three use cases from the DNS domain, we highlight the strengths and limitations of proactive threat detection and discuss how we could integrate those with existing solutions.

CCS Concepts: • **Networks** → *Network services*; • **Security and privacy** → **Network security**; **Intrusion/anomaly detection and malware mitigation**;

Additional Key Words and Phrases: Threat detection, DNS, machine learning, big data

ACM Reference format:
Olivier van der Toorn and Anna Sperotto. 2020. Looking Beyond the Horizon: Thoughts on Proactive Detection of Threats. *Digit. Threat.: Res. Pract.* 1, 1, Article 4 (March 2020), 13 pages.
https://doi.org/10.1145/3373639

1 INTRODUCTION

The Internet was born as an open, decentralized, and scalable infrastructure supporting the visionary dream of an interconnected world of data, information, and services. Never like in the past decade have we have witnessed a more booming expansion of the Internet in terms of infrastructure (e.g., available bandwidth), services (e.g., online social networks, online streaming, online banking, health care), and user-generated content. The Internet is considered a means for ensuring human rights such as freedom of speech and expression, and a structural way for educating people on democracy [4]. Although the debate is raging as to whether the Internet should be called a *utility* by virtue of being a fundamental human right, the Internet runs the risk of becoming our biggest liability.

The benefits of an interconnected society are immediately clear to everybody; however, it has recently become more and more evident, even to laypeople, that the Internet exposes us to cyberthreats attacking information, services, and even the Internet infrastructure itself (e.g., attacks against the Domain Name System). Chief examples are (Distributed) Denial of Service (DDoS) attacks, an old threat that has recently taken new shapes and

This work was funded by SIDN-fonds, an independent fund on the initiative of SIDN, the registrar for ".nl" domains, and partially funded by the EU H2020 projects CONCORDIA (#830927).
Authors' addresses: O. van der Toorn and A. Sperotto, University of Twente, Drienerlolaan 5, Enschede, Overijssel, 7522NB; emails: {o.i.vandertoorn, a.sperotto}@utwente.nl.

2576-5337/2020/03-ART4 $15.00
https://doi.org/10.1145/3373639

proportions (e.g., the 2016 attacks against the hosting company OVH (1 Tbps) [20] and the attack against the service and DNS provider Dyn [11]) but also ever more advanced phishing attacks (e.g., CEO fraud [16]), advanced spam campaigns (e.g., snowshoe spam [22]), and other forms of insidious activities.

When a cyberthreat makes its appearance on the Internet, a mitigation strategy follows right after. The rise of DDoS attacks, for example, has paved the way to a new market for DDoS protection systems, such as appliances and services aiming at stopping malicious traffic from hindering a certain service [redacted due to anonymization]. In practice, there exists a plethora of mechanisms to protect against cyberthreats (e.g., firewalls, blacklists, ACL, IDS, DDoS protection services). What all of these solutions have in common, however, is that they are inherently reactive. In other words, they only come into play when an attack is already ongoing.

In this article, we investigate the possibility of addressing cyberthreats in a different yet possibly complementary way. We argue that a shift to a proactive form of threat detection, in which we are able to provide indicators that an attack is *in the making*, might be beneficial for enhancing security.

The rationale for this research is that sophisticated attacks are no longer a matter of running a script but require careful preparation. For example, attack phases need to be staged, and additional infrastructure needs to set up. In preparing an attack, attackers expose characteristics of their infrastructure that can be used as threat indicators. Our intuition tells us that if we are successful in identifying these threat indicators, we also have a chance to act on this information before the threat morphs into an actual attack. An example that we will investigate in more detail later is registering a domain name with an anomalously high number of `A` and `MX` records to be used in snowshoe spam campaigns. In the window between crafting such a domain and performing an attack, a fundamental piece of the attack infrastructure is exposed and vulnerable, providing us with a chance to perform threat detection in a proactive manner.

The remainder of this article is organized as follows. In Section 2, we reason about what is needed for the idea of proactive threat detection to be feasible. In Sections 3 through 5, we present three case studies, whereas in Section 6, we discuss about the strengths and limitation of our approach. We conclude the article with related work (Section 7) and conclusions (Section 8).

2 ENABLING PROACTIVE THREAT DETECTION

The main difference between reactive and proactive threat detection is that the second aims at detecting indicators of an attack before the actual attack takes place. To enable a proactive approach, two main aspects are key: data and domain knowledge.

In our experience, proactive threat detection is best supported by using active measurements. Data sources in reactive threat detection typically are passive—sensors within the network collecting metrics upon which triggers may fire. A textbook example in this respect would be an intrusion detection system. In active measurements, however, rather than passively collecting what a sensor observes, one requests the necessary data directly. Active measurements require a seed, which is dependent on the context of the measurement. For example, in the case of an active Domain Name System (DNS) measurement, the seed could be zone files. Examples of large-scale active scanning projects are, among others, Shodan[1] and the ANT censuses of the Internet Address Space [10]. These projects actively probe the entire IPv4 address space (large scale) for open ports, in the case of Shodan, or for testing Internet connectivity, in the case of ANT census. The first advantage of an active measurement is that, since one has control on what data is requested, we can ensure a relatively high degree of completeness. A second advantage is that the measurement can be repeated at a regular interval of time, thus building a longitudinal view of the phenomenon that is investigated. In proactive threat detection, therefore, *large-scale* and *longitudinal* measurements play a major role.

When dealing with large-scale, longitudinal datasets, we have learned that the amount of data is at times of such a proportion that finding how to characterize a specific threat indicator becomes like finding the proverbial

[1] https://www.shodan.io/.

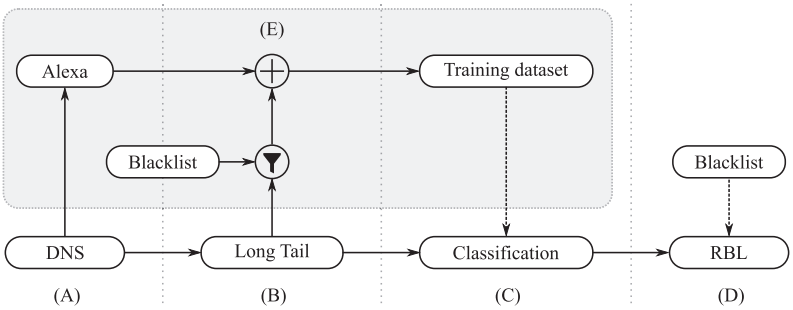


Fig. 1. High-level overview of our approach.

needle in the haystack. We therefore argue that domain knowledge about the preparation steps related to the attack we are trying to predict is fundamental.

Our research focuses on using DNS for proactive security. We have access to a large-scale active DNS measurement dataset from the OpenINTEL project [23]. This dataset consists of daily snapshots of around 65% of the global DNS name space since March 2015. For every domain in the measurement dataset, responses to 'A', 'AAAA', 'NS,' and 'MX' queries are available, among others.

Additionally, in our research, we frequently use blacklists as our ground truth. We realize that blacklists are rarely perfect and blacklist operators rarely reveal their method of obtaining information. This may mean that blacklist operators create their blacklist in either a passive or even in an active way, or that they have their own predictive registration methods of malicious activity. In any case, blacklists are currently considered the *de facto* standard to verify malicious behavior (e.g., in spam filters), and we therefore adopted them in our validation.

In Sections 3 through 5, we discuss use cases where this dataset plays a central role, highlighting how threat indicators for each use case can be identified and characterized. The use cases will also help us reason about the strengths and limitations of a proactive approach.

3 SNOWSHOE SPAM

In snowshoe spam,[2] the sending of spam is spread out over a large number of hosts, each one used to send a relatively small amount of emails, to avoid detection by spam reputation systems (blacklists). We have also observed that snowshoe 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 that only authorized email servers can send email for specific domains. However, SPF requires spammers to register and configure a legitimate DNS domain, and to create a DNS record for every host that should be able to send email for that domain. This enables us to proactively search for domains related to snowshoe spam in the active DNS dataset used in our research.

3.1 Methodology and Dataset

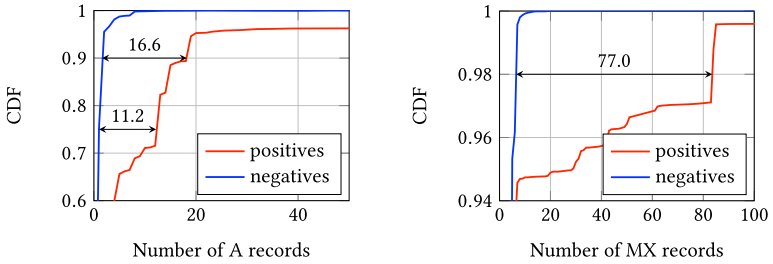
Figure 1 shows a high-level overview of our approach. From left to right, the figure displays parts (A) through (D), which together makes up our detection process. In addition to this, part (E) is shown in the gray rectangle, which represents the training of the machine learning classifier on which our detection relies.

At a high level, our method for detection does the following. Every day, we collect the most recent snapshot from our dataset (A). Since 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, we then perform a filtering step to only focus on the long tail of the domain size distribution (B). At this point, we have extracted candidate domains. We then use a

[2]This section is a summary of a previously published work of the authors [22].

Table 1. Used Blacklists and Their Purpose

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



(a) Comparison of the number of A records (b) Comparison of the number of MX records

Fig. 2. CDF of two features in the test dataset.

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).

For the explanation on how we have built and trained the classifier, we refer the reader to our original work [22].

For validation purposes, all domains on the RBL are then checked against existing public blacklists (Table 1). We monitor these blacklists to identify the first moment in time in which one of our candidate snowshoe domains is blacklisted. This allows us to quantify the time advantage of our detection method.

We performed daily detection from May 24, 2017, until September 5, 2017. This section discusses the details on the datasets used during the daily detections. The basis of these detections is a dataset of that day containing domains exceeding the 99.9th, 99th, 98th, or 97th percentile. On average, there are about 2.7K domains in the dataset of the 99.9th percentile. This figure grows to 57.3K domain names in the dataset of the 97th percentile.

3.2 Results

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 made a dataset from April 2017. We selected domains above the 99th percentile, as this percentile threshold gave a clear distinction between positives and negatives. We filtered out the positives and matched it with an equal number of negatives from the Alexa top million. This resulted in a dataset with 136,441 positives and the same number of negatives. We visualize the difference by plotting the Cumulative Distribution Function (CDF) for two most distinctive features: the number of `A` and `MX` records per domain (Figure 2). 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, indicating that there is a clear distinction between benign domains and domains crafted for snowshoe spam.

During our measurement period, our detection method marked 35,004 domains as snowshoe spam domains, and 32,677 of these 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

Table 2. Per-Day Averages of the Datasets and Detections

Percentile	Avg. Domains in the Dataset	Avg. Domains Detected	Avg. Added to the RBL
99.9th	2,728.07	243.96	18.99
99th	19,179.59	3,228.75	149.37
98th	37,202.64	5,226.31	205.72
97th	57,250.48	6,805.55	239.37

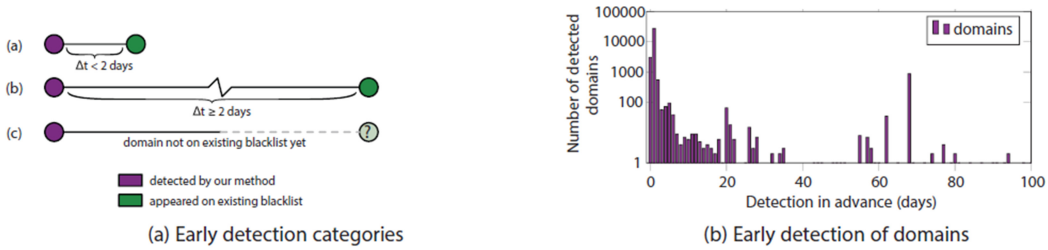


Fig. 3. Early detection of spam domains.

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.

Table 2 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.

We finally 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 appear on one of the existing blacklists we considered (see Table 1).

In the context of early detection, we distinguish three categories of domains. Figure 3(a) depicts these categories, and they are described in more detail in the following:

(a) Domains that are already on a blacklist at the time of detection or have only a day difference. There can be a 1 day difference since the daily data is of the previous day, whereas the blacklist query happens in real time.
(b) Domains with a detection difference of at least 2 days or more.
(c) Domains that—during the measurement period—have not (yet) been blacklisted.

Figure 3(b) 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 2 days after detection via our method. The second category (b), where our detection is at least 2 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 2 weeks in advance, and 971 domains (2.77%) were detected at least a month in advance. There are even 949 domains (2.71%) that were detected at least 60 days before they appeared on a blacklist. The maximum time difference we observed so far is 104 days. In addition, 2,327 domains (6.65%) fall in the last category (c) and have not been blacklisted during the measurement period. Although these numbers may seem small percentage wise, it should be noted that the impact of this approach is in reality quite valuable, as this is typically the type of email that makes it past an email filter.

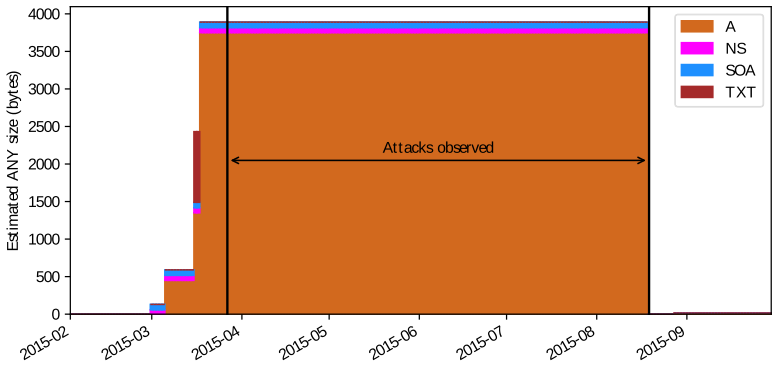


Fig. 4. Lifetime of a DDoS domain.

3.3 Summary of the Snowshoe Spam Use Case

In this use case, we investigate how domains crafted for sending snowshoe spam can be detected using active DNS measurements. 93.25% of domains we 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.

4 DDOS DOMAINS

DNS amplification is a form of DDos attack in which an attacker will prompt the DNS to answer fake queries seemingly generated by the target (spoofing). The attacker will typically send a query for which he knows the response will be very large, to maximize the amplification effect. The attacker can do this in two ways. The first option is to use an existing domain for which the attacker knows the answer to certain queries to be large, such as a domain that is DNSSEC signed [redacted due to anonymization]. The second option is to, instead, *craft* a domain under their control, populated in such fashion to guarantee a large response.

In this use case, we focus on attackers who choose the second option. We present here an example of the behavior over time of a domain crafted and misused for DDoS attacks, as well as the outline of a possible proactive detection strategy. Albeit anecdotal, we believe that this example points to a valuable research direction for the approach we are outlining in this article. We plan to investigate this further as future work.

We expect domains crafted for DDoS to stand out because an attacker has an incentive to maximize the response size for a domain to achieve high amplification. Since the attacker needs to rely on the DNS to achieve high amplification, our intuition tells us that these type of domains are likely to be visible in the active DNS dataset we use. Figure 4 presents an example of a domain that shows a behavior compatible with the one described earlier. In particular, Figure 4 shows that although the domain was registered well in advance, the number of records and the estimated amplification size (as carried, e.g.,, in a query of type ANY) were modest until March 2015. Starting from March 2015, we observe that the domain has been inflated, specifically by adding more than 200 A records, reaching an estimated `ANY` size of 3,500 bytes. This coincides with the time windows in which the domain is used in DDoS attack (based on data from the AmpPot project [15]). After the attack window ends, in September 2015 the domain deflates. We believe that this growth pattern is indicative of domains crafted for DDoS attacks. However, it remains future work to investigate if our hypothesis holds.

A possible proactive approach to detect these kinds of domains at scale could comprise the following steps:

(1) Filter domains with more than an average number of records or domains with a longer than average `TXT` record.
(2) Gather the records for the past X days.

Table 3. Types of Domain Squatting Using the Domain 'utwente.nl' as an Example (Adapted from Kintis et al. [13])

Type	Example
Typosquatting	utwent.nl
Combosquatting	utwente-login.nl
Bitsquatting	utwenpe.nl
Homophone-based squatting	utwentie.nl
Homograph-based squatting	utvvente.nl
Abbrevsquatting	ut.nl

(3) Determine the trend lines in terms of growth (number of records, TXT length).
(4) Predict the size of the domain for the next Y days.
(5) Flag the domain if the predicted size is above a certain threshold.

Future work consists of evaluating if this method is effective and what proper values for X and Y are.

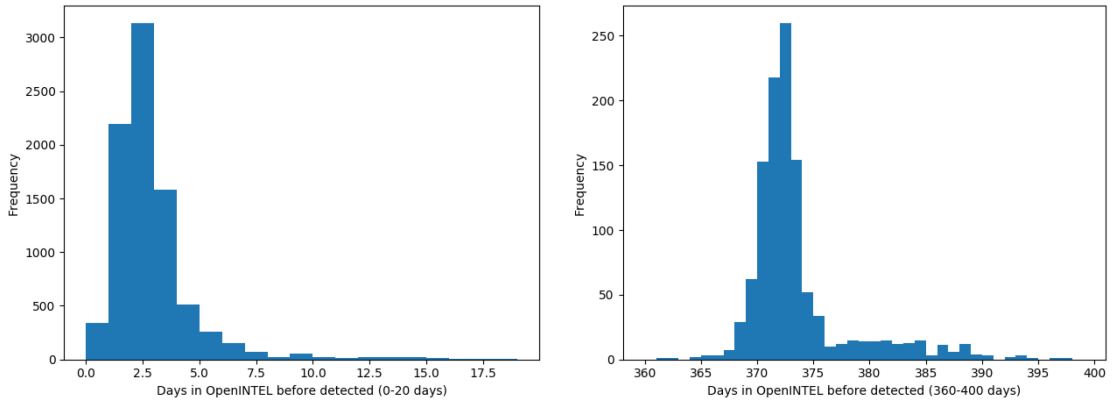
This case study strongly suggest that the estimated amplification size for a domain and its trend over time could be used as treat indicators for amplification DDoS attacks. Additionally, the behavior of the example domain we presented seems to indicate that a certain window of time exists between the moment of domain registration and the moment when it is misused. Estimating and exploiting this time window would effectively allow us to flag a crafted domain before it is misused.

5 COMBOSQUATTING DOMAINS

Domain squatting is typically seen in phishing attacks. Attackers mimic every aspect of their target web page to trick a victim into entering credentials, for example. This mimicking includes the domain name. Since attackers cannot use the original domain name, they need to come up with a name closely resembling the original one. Types of domain squatting are listed in Table 3. Our focus in this use case lies with combosquatting, a type of squatting where the trademark identifying the target is left intact and another word is either prepended or appended, resulting in a domain name that appears to be owned by the trademark holder but actually leads to a malicious endpoint. However, the practice of prepending or appending word to trademarks is by no means restricted to malicious use. For example, the domain 'youtubego.com' contains the trademark YouTube but is not malicious. For this reason, we do not base our approach on domain name analysis but instead focus on active DNS data. The additional DNS data could be used to increase the certainty of a prediction of maliciousness.

We started this work with a similar approach as in the snowshoe spam use case. We used a machine learning classifier to distinguish between regular domains and combosquatting domains. Originally, we aimed for a generic detection model, one where a list of trademarks is not required, thus relieving us from the need to keep such a list up to date, as well as the need to know each brand name. Our training dataset consisted of an equal number of positives (combosquat domains) and negatives (benign domain). A domain was considered a combosquat domain when it contained a trademark and was listed on a blacklist. This ensured that only combosquat domains with malicious intent were present in our training data, to avoid false positives (e.g., the previously mentioned 'youtubego.com') from becoming part of the training set. The negatives were selected from the Alexa top million list under the assumption that it is unlikely for a combosquat domain to gain enough popularity to be listed in Alexa.

However, the classifications of a classifier trained on such data were disappointing. The main distinguishing feature used by our classifier, a Gaussian Naive Bayes classifier, was the length of the domain name. This is understandable from the perspective that a combosquat is naturally longer than the target it is squatting. Yet



(a) First peak at 2 days

(b) Second peak after a year

Fig. 5. Difference between detection and blacklisting in days.

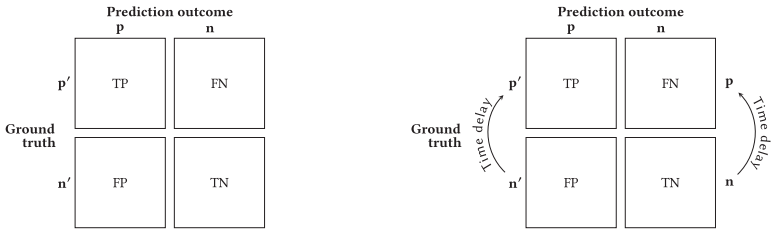
this does not automatically mean that a long domain is a combosquat domain. We therefore modified our dataset by adding a feature that specified if the domain under consideration contained a trademark from a predefined list of trademarks. This list of trademarks was based on manual inspection of the Alexa top 500 list. Ambiguous domain names were excluded, as well as short names (less than four characters).

Due to this change in the classifier, we were able to detect combosquat domains, as each of the detected domains contained a trademark. We analyzed the performance of our detection method by comparing them to historic blacklist data. This allowed us to go back in time to observe when the domain was registered, when we would have detected it, and when the domain was blacklisted. Analyzing the difference between detection and blacklisting resulted in two major peaks, which are visible in Figure 5(a) and Figure 5(b). The first was at 2 days after detection, and the second peak was at 372 days after detection. The first peak suggests that these domains were actively used and therefore quickly blacklisted because there was enough evidence. The second peak might be explained when we observe the ICANN Domain lifecycle diagram [12]. Assuming a domain with an initial expiration date of 1 year from registration, the 372 days correspond with the 5-day "Add Grace Period" and the 2 days of detection time. This could mean that the domain switched ownership after a year and was then used in malicious practice. We performed active HTTP measurements against these combosquat domains. These domains typically go through the following stages:

(1) The domain is registered/parked. At this stage, the domain is not actively misused.
(2) First signs of activity. A web server is set up, and a folder containing malicious content is uploaded.
(3) The combosquat domain is up and running with malicious intentions.
(4) Either the domain is blacklisted or serves an error page.

In addition to the combosquat domains that Kintis et al. [13] observed—domains with a long lifetime—we also observed a class of domains that have a lifetime of only a couple of hours in which they go through these stages.

In this use case, although we succeeded in detecting combosquat domains, we also observe that the time advantage is not as large as we saw in the snowshoe spam use case. This is due to two factors. First, there seems to be a fast reaction time from the moment a domain with a name suggesting combosquatting is registered (and therefore it appear in our database) and the moment this is put on a blacklist. Second, since combosquatting domains might aim at a fast completion of their lifecycle, we cannot exclude that a number of domains complete their lifecycle in less than a day, after which the domain remains registered but is no longer used. Depending on when the domain registration happens, our dataset might take notice of the new suspicious domains with up to a



(a) Traditional confusion matrix (b) Proactive threat detection

Fig. 6. Comparision between the traditional confusion matrix and the confusion matrix with time uncertainty in proactive threat detection.

day delay, at which point the combosquat campaign is likely to have ended. This suggests that the measurement frequency of our dataset, once per day, is too low to effectively combat combosquat domains.

6 DISCUSSION

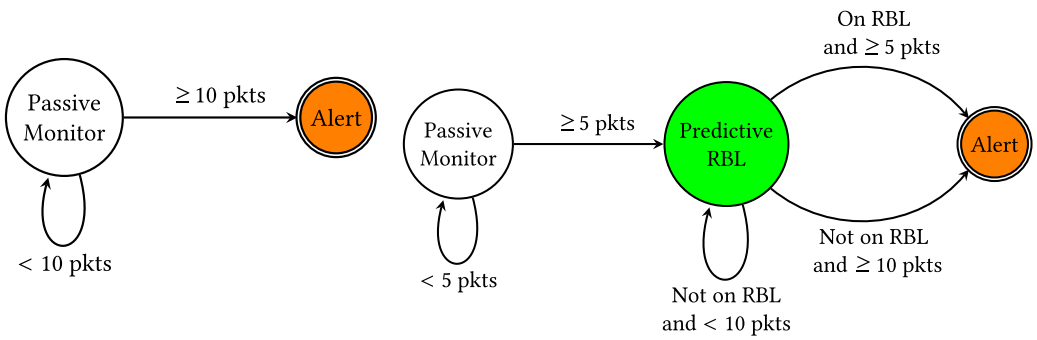
Section 3 through 5 presented three examples of the application of a proactive threat detection approach. We chose these use cases because they allow us to reason about both the strengths and limitations of proactive threat detection.

The use cases highlight that the major strengths of a proactive threat detection approach is twofold. First, proactive threat detection has the potential to identify a threat indicator earlier than the attack takes place, thus allowing security experts to make the most of a time advantage that can be up to months. We strongly believe that this time advantage is key to deploy more targeted security measures that can overall improve the security of the Internet. Second, since proactive threat detection strongly depends on large-scale, longitudinal data and is not linked to a specific target but to the attacker infrastructure, the scope in terms of security impact of a proactive approach is larger than a reactive system. By acting on the attacker infrastructure, we therefore have a larger security benefit.

A proactive approach does not come without risks, however. First and foremost, it is important to keep in mind that proactive threat detection is in essence a prediction approach, since rather than reacting when an attack is happening (certainty), we use threat indicators to warn that an attack might happen in the future. Therefore, there is no hard proof, unlike the case of passive measurements, of malicious activity related to a certain domain, nor a clear indication of how much time will pass from the prediction to the attack. Although validation based, for example, on later appearance of malicious domains in blacklists helps in building confidence that a proactive threat detection approach works (e.g., see the snowshoe spam use case in Section 3), we cannot, of course, ensure that a prediction will result in an attack. The *conversion rate*, meaning how many predicted threats convert to attacks, is therefore a good indication of the likelihood that an attack will strike, but it gives, by design, no certainty of imminent attacks.

The traditional confusion matrix typically used to determine if a classification approach (Figure 6(a)) is effective with respect to a certain ground truth no longer applies *as is* due to the time component that is introduced. The domains that our approach predicts as malicious may, at the time of the prediction, be a true positive or a false positive. With normal classification problems, these labels are static. However, since we are dealing with a prediction, in our case these may change over time. Once an attack strikes, triggering an update in the ground truth, what we flagged as a false positive becomes a true positive. The same applies to the domains that we do not mark as malicious. At prediction time, these are either false negatives or true negatives. If an attack is witnessed by the ground truth, a true negative becomes a false negative. We depict this phenomenon in Figure 6(b).

The traditional false positives and true negatives in a proactive setting are therefore uncertain and possibly dynamic. The reasons for this phenomenon are the conversion rate and the possible time delay. Ideally, an effective



(a) Current reactive approach (b) Example of complementary approaches

Fig. 7. Architectural example.

proactive approach will almost never trigger the conversion from true negatives to false negatives. The conversion from false positives to true positives, however, are desired because they improve the overall performance of the system.

In practice, this means that proactive threat detection entails a *trade-off* between the risk of acting on a prediction that might not convert to an attack and the risk of not acting on it and at a later moment dealing with a miss. We argue that this trade-off is application and situation specific, as it might involve different risk factors.

With this in mind, we suggest that proactive threat detection should be used as an enhancement to the current reactive methods (shown in Figure 7(a)). For example, the reports from the proactive methods could lower the reaction thresholds in reactive methods, as shown in Figure 7(b) where an alert is triggered after 5 packets coming from a certain domain instead of the normal trigger of, say, 10 packets.

In addition, it is important to keep in mind how the available data impact the applicability of proactive methods. In Section 2, we already stressed the importance of having large-scale, longitudinal data for this type of research. We argue here that the type of data will also influence which type of threats we are able to detect. In the case of combosquat domains, for example, having the platform behind our dataset measure more frequently than once per day is unfeasible because of the number of domains that need to be measured. However, proactive threat detection could be used as a filter for these measurements. Suppose that the combosquat detection model is fed a list of newly registered domains from a zone file; the model filters out candidates that conform to the combosquat definition. At that point, measurements with high frequencies, every hour or every minute, can be started.

We realize that developing proactive threat detection strategies and the related blacklists can possibly affect the behavior of the attackers. From the perspective of the attackers, an effective proactive RBL may mean a higher number of 'unusable' domains for their attack. As a consequence, they might change their preparation tactics, such as by publishing their domain just before performing the attack. What we believe will not change, however, is the need attackers have to rely on established infrastructure (e.g., the DNS) to support their activities. This would mean that the proposed methodology might become less effective with respect to the detection time advantage, but it would still be able to complement reactive approaches.

7 RELATED WORK

Proactive security is closely related to threat detection, which, in itself, is a large field of research. Because this work draws its inspiration from the DNS, we focus on research using the DNS for threat detection. We first introduce literature on passive and active DNS measurements. We then focus on how DNS misuse can be detected and finally focus on proactive threat detection.

DNS data can be acquired in a passive or active manner. The most well known method for passive DNS measurements is Passive DNS (pDNS) [27]. pDNS is mostly used to capture DNS traffic between a recursive caching name server (resolver) and the authoritative name servers, thus ensuring the privacy of the end users. Notable examples of large-scale pDNS deployment are Farsight Security's DNSDB[3] and the pDNS infrastructure operated by CERT.at.[4] The drawback of pDNS data is that it is inherently reactive, as a domain becomes part of this dataset only if a user accesses it. In contrast, active measurements work by sending targeted queries to the DNS. There are several examples of active DNS measurements in the literature. Schomp et al. [21] used active scans to investigate the client-side DNS infrastructure. Kountoras et al. [14] proposed a DNS data collection platform for the study of short-lived disposable domains. Rijswijk-Deij et al. [24, 25] used active DNS measurements to study aspects of the DNSSEC protocol.

Several contribution in the literature focus on detecting malicious activities using DNS measurements. Fukuda et al. [6] used reverse DNS queries to detect network-wide activity, among which large scans at times were related to malicious activity (e.g., Heartbleed and SSH scanning). For their work, the authors used passive DNS data collected from DNS servers at the root or at country level. Consequently, their work, although valuable, cannot be used in a proactive manner. DNS queries are commonly used in the detection of botnets [3, 17, 26, 28]. Also in this case, the common ground of these works is the use of passive DNS traces, making them unable to predict new command and control domains.

An important topic of research, in the context of threat detection using the DNS, is malicious domain detection—that is, focusing specifically on the characteristics of the domain name. The EXPOSURE system [1], for example, used features from a combination of domain characteristics and passively obtained DNS answers to classify domains based on the likelihood of being malicious. As before, the passive collection of DNS answers limits this approach to a reactive response to emerging threats. On the contrary, the work by He et al. [9] is closer to the approach we propose in this article. The authors aim at predicting, at registration time, if a domain will be used maliciously or not. They largely base this prediction on features derived from the analyzed name. This works supports our argument that the data in the DNS is a valuable source of information for proactively identifying malicious activities in the making.

We argue that using proactive threat detection techniques and blacklists allows us to react quicker to emerging threats posed to us from the Internet. In line with our proposed approach, several works in the literature focus on predicting undiscovered malicious activity. Felegyhazi et al. [5] used the notion of perpetrators registering malicious domains in bulk to predict additional malicious domains. Using name server properties and registration information, the authors clustered domains into confirmed bad domains, unknown domains, and suspected bad domains. Fukushima et al. [7] took a bad neighborhoods approach [19], namely focusing on the idea that malicious activities tend to cluster together in the network space. The authors clustered domains based on their CIDR /24 prefix and registrar. Through this information, the authors calculated a score that increased as the number of IP addresses from a specific prefix were listed. The authors assumed that new domains matching a high scoring prefix and registrar combination were malicious as well. Lee et al. [18] posed the idea of tracking phishing domains through their redirections and forms. In their work, the authors proposed to submit URLs resulting from the tracking automatically to PhishTank to develop a proactive blacklist. For the initial seed, the authors took 3,916 phishing URLs from PhishTank and discovered an additional 2,345 phishing domains. Hao et al. [8] developed PREDATOR, a proactive system for recognition and elimination of domain abuse at the time of registration. This contribution is an additional example of how proactive security allows for a faster response against abuse. Finally, the work by Bui [2] comes closest to our proposal. The authors argued that cybersecurity can capitalize on the *Germination Period*, which is "the time lag between hacker communities discussing software flaw types and flaws actually being exploited" Bui [2]. They consequently proposed to crawl hacker forums

[3] https://www.dnsdb.info/.
[4] The Austrian National CERT team: http://www.cert.at/index_en.html.

for 0-days and vulnerabilities to feed a proactive security system. Similarly to our proposal, the data gathering methodology of Bui [2] is active; however, whereas Bui focuses on vulnerabilities and 0-days described in textual sources (fora and public databases), we focus on DNS data.

8 CONCLUSION

In this work, we argue that proactive threat detection has the potential to become a novel and complementary approach to traditional reactive security. We presented two case studies (i.e., reflection and amplification DDoS attacks, and snowshoe spam) for which evidence shows that we can gain a time advantage on the actual attack and larger coverage. Yet we also argue that proactive threat detection needs to be handled with care. First, the appropriate type of data is a must for enabling the observation of threats, as we show in the use case of combosquatting, for which our dataset is not able to provide the necessary information for a proactive approach. Second, the actual conversion rate from threats to attacks, combined with the predictive nature of our methods, needs to be evaluated carefully depending to the application. Based on this observation, we advocate the use of proactive threat detection *in combination* with reactive detection methods, such as by lowering thresholds for suspicious activity.

We aim to extend this work by studying other areas of attacks. In our case, our main data source will remain the DNS—an area in which we plan to extend our active DNS measurement with data from passive DNS monitors to study how active and passive measurements can optimally be combined for threat detection.

ACKNOWLEDGMENTS

We would like to thank Roland van Rijswijk for providing the graph in the DDoS section and Joost Jansen for performing work in the combosquat use case.

REFERENCES

[1] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding malicious domains using passive DNS analysis. In *Proceedings of the 2011 NDSS Symposium (NDSS'11)*. ACM, New York, NY, Article 14, 28 pages.

[2] Tung Bui (Ed.). 2017. *Can Cybersecurity Be Proactive? A Big Data Approach and Challenges*. IBM. DOI: https://doi.org/10.24251/hicss.2017.725

[3] Hyunsang Choi and Heejo Lee. 2012. Identifying botnets by capturing group activities in DNS traffic. *Computer Networks* 56, 1 (2012), 20–33. http://www.sciencedirect.com/science/article/pii/S1389128611002787

[4] H. Clinton. 2010. Remarks on Internet Freedom. Retrieved February 1, 2020 from https://2009-2017.state.gov/secretary/20092013clinton/rm/2010/01/135519.htm.

[5] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. 2010. On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET'10)*. 6. http://dl.acm.org/citation.cfm?id=1855686.1855692

[6] Kensuke Fukuda, John Heidemann, and Abdul Qadeer. 2017. Detecting malicious activity with DNS backscatter over time. *IEEE/ACM Transactions on Networking* 25, 5 (2017), 3203–3218. DOI: https://doi.org/10.1109/TNET.2017.2724506

[7] Y. Fukushima, Y. Hori, and K. Sakurai. 2011. Proactive blacklisting for malicious web sites by reputation evaluation based on domain and IP address registration. In *Proceedings of the 2011 IEEE 10th International Conference on Trust, Security, and Privacy in Computing and Communications*. IEEE, Los Alamitos, CA, 352–361. DOI: https://doi.org/10.1109/TrustCom.2011.46

[8] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 2016. PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, New York, NY, 1568–1579.

[9] Yuanchen He, Zhenyu Zhong, Sven Krasser, and Yuchun Tang. 2010. Mining DNS for malicious domain registrations. In *Proceedings of the 6th International Conference on Collaborative Computing: Networking, Applications, and Worksharing (CollaborateCom'10)*. 6. DOI: https://doi.org/10.4108/icst.collaboratecom.2010.5

[10] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. 2008. Census and survey of the visible Internet. In *Proceedings of the ACM Internet Measurement Conference*. ACM, New York, NY, 169–182. http://www.isi.edu/%7ejohnh/PAPERS/Heidemann08c.html.

[11] Scott Hilton. 2016. Dyn Analysis Summary of Friday October 21 Attack. Retrieved February 1, 2020 from http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

[12] ICANN. 2012. Life Cycle of a Typical gTLD Domain Name. Retrieved February 1, 2020 from https://www.icann.org/resources/pages/gtld-lifecycle-2012-02-25-en.

[13] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, 569–586. DOI: https://doi.org/10.1145/3133956.3134002

[14] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe. 2016. Enabling network security through active DNS datasets. In *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions, and Defenses (RAID'06)*. 188–208. DOI: https://doi.org/10.1007/978-3-319-45719-2_9

[15] Lukas Krämer, Johannes Krupp, Daisuke Makita, Tomomi Nishizoe, Takashi Koide, Katsunari Yoshioka, and Christian Rossow. 2015. AmpPot: Monitoring and defending against amplification DDoS attacks. In *Research in Attacks, Intrusions, and Defenses*. Springer-Verlag, New York, NY, 615–636.

[16] Brian Krebs. 2016. FBI: $2.3 Billion Lost to CEO Email Scams. Retrieved February 1, 2020 from https://krebsonsecurity.com/2016/04/fbi-2-3-billion-lost-to-ceo-email-scams/.

[17] Jonghoon Kwon, Jehyun Lee, Heejo Lee, and Adrian Perrig. 2016. PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks* 97 (2016), 48–73. DOI: https://doi.org/10.1016/j.comnet.2015.12.008

[18] Lung Hao Lee, Kuei Ching Lee, Hsin Hsi Chen, and Yuen-Hsien Tseng. 2014. Proactive blacklist update for anti-phishing. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM, New York, NY, 1448–1450. DOI: https://doi.org/10.1145/2660267.2662362

[19] G. C. M. Moura, A. Sperotto, R. Sadre, and A. Pras. 2013. Evaluating third-party bad neighborhood blacklists for spam detection. In *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM'13)*. IEEE, Los Alamitos, CA, 252–259.

[20] Pierluigi Paganini. 2016. 150,000 IoT Devices Behind the 1Tbps DDoS Attack on OVH. Retrieved February 1, 2020 from http://securityaffairs.co/wordpress/51726/cyber-crime/ovh-hit-botnet-iot.html.

[21] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. 2013. On measuring the client-side DNS infrastructure. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC'13)*. ACM, New York, NY, 77–90. DOI: https://doi.org/10.1145/2504730.2504734

[22] O. van der Toorn, R. van Rijswijk-Deij, B. Geesink, and A. Sperotto. 2018. Melting the snow: Using active DNS measurements to detect snowshoe spam domains. In *Proceedings of the 2018 IEEE/IFIP Network Operations and Management Symposium (NOMS'18)*. IEEE, Los Alamitos, CA, 1–9. DOI: https://doi.org/10.1109/NOMS.2018.8406222

[23] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras. 2016. A high-performance, scalable infrastructure for large-scale active DNS measurements. *IEEE Journal on Selected Areas in Communications* 34, 6 (2016), 1887–1888.

[24] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2014. DNSSEC and its potential for DDoS attacks. In *Proceedings of the 2014 Internet Measurement Conference (IMC'14)*. ACM, New York, NY. DOI: https://doi.org/10.1145/2663716.2663731

[25] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2015. Making the case for elliptic curves in DNSSEC. *SIGCOMM Computer Communication Review* 45, 5 (Sept. 2015), 13–19. DOI: https://doi.org/10.1145/2831347.2831350

[26] Ricardo Villamarín-Salomón and José Carlos Brustoloni. 2008. Identifying botnets using anomaly detection techniques applied to DNS traffic. In *Proceedings of the 2008 5th IEEE Consumer Communications and Networking Conference (CCNC'08)*.476–481. DOI: https://doi.org/10.1109/ccnc08.2007.112

[27] Florian Weimer. 2005. Passive DNS replication. In *Proceedings of of FIRST 2005*.

[28] Bojan Zdrnja, Nevil Brownlee, and Duane Wessels. 2007. Passive monitoring of DNS anomalies. In *Proceedings of the 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'07)*. ACM, New York, NY, 129–139.

Received June 2019; revised November 2019; accepted November 2019