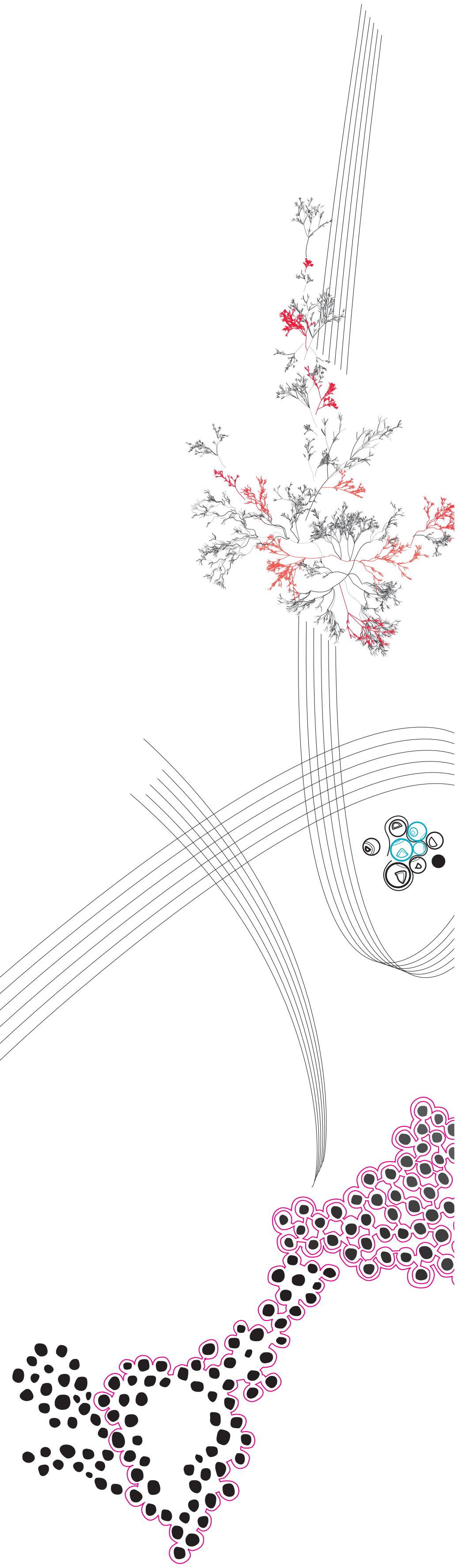
MSc Computer Science
Final Project



Early Warning System for Newly Registered Malicious Domains: A Machine Learning and Certificate Transparency Approach

Luuk Berenschot

Committee:
Roland van Rijswijk-Deij
Thijs van Ede
Antonia Affinito
Raffaele Sommese

August, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Chapter 1

Introduction

Upon agreement with my graduation committee, we have decided to write this paper for a conference. The intention is to submit the paper to the IEEE Euro S&P conference. The exact procedure will be discussed later.

The rest of this document discusses how the requirements of the Examination Board are met, even though the Master Thesis is written in paper form. The paper aimed to be handed in for a conference is included in Appendix A.

Chapter 2

Requirements

All the requirements for a master thesis are listed below, along with explanations of how each requirement is satisfied.

2.1 Scientific quality

2.1.1 Interpret a possibly general project proposal and translate it to more concrete research questions

The original project proposal aimed to create a methodology for early detection of malicious domain names after registration using OpenINTEL for newly registered domain detection. In the exploration phase, I designed an approach to measuring newly registered domain names on a regular interval. The data collected helps reach this goal, which is formulated in the following research questions.

1. To what extent can we distinguish between benign and malicious domains based on information collected during the early stages of a domain?
2. How does newly registered domain name detection based on certificate transparency compare to detection based on other sources?

2.1.2 Find and study relevant literature, software and hardware tools, and critically assess their merits

During the research topics phase, an extensive literature review was conducted to gather relevant and meritorious literature. The methodology, including the software needed for the research, was also outlined. Throughout the research process, additional literature was included, and a refined selection was made to determine which sources were most relevant to the paper I wrote.

2.1.3 Work in a systematic way and document your findings as you progress

During the research, a logbook is maintained, and all findings are recorded. The logbook is used for further investigation of interesting discoveries within the scope of the research.

2.1.4 Work in correspondence with the level of the elective courses you have followed

Elective courses relevant to this research include data science and cyber data analytics. These modules explain how to apply data science principles and how to collect and prepare data for machine learning. The research combined information learned from many courses but also required additional learning.

2.1.5 Perform original work that has sufficient depth to be relevant to the research in the chair

My work is original and provides substantial depth. It is relevant to the research in the chair, and the aim of presenting it at an external conference demonstrates its relevance.

2.2 Organisation, planning, collaboration

2.2.1 Work independently and goal-oriented under the guidance of a supervisor

My research has been performed independently. The measurement setup is self-designed and developed, and all results are processed independently. During the research, the guidance of the supervisors has been used on a weekly basis. For every meeting, preparations are made on topics to discuss and notes are made during the meetings containing new insights.

2.2.2 Seek assistance within the research group or elsewhere, if required and beneficial for the project

Within the DACS research group, assistance is sought to get access to my own measurement server and already existing measurement data. The research group also provides the computing cluster used for processing my data. Issues regarding these topics are resolved within the research group. In addition, contact is sought in the SCS research group regarding expertise in some machine learning approaches.

2.2.3 Benefit from the guidance of your supervisor by scheduling regular meetings, provide the supervisor with progress reports and initiate topics that will be discussed

Throughout the research process, weekly meetings were held. These meetings were prepared with notes for discussion, and detailed notes were taken during each meeting to ensure a smooth progression of the research and to address any issues that arose along the way. In addition, more elaborate meetings were held on an irregular basis with the entire graduation committee, during which presentations were held to showcase the current progress.

2.2.4 Organize your work by making a project plan, executing it, adjusting it when necessary, handling unexpected developments and finishing within the allotted number of credits

At the beginning of my research, I created a detailed project plan for my research topic, which included the methodology I intended to use. If adjustments would positively impact the research, they were made along the way.

2.3 communication

2.3.1 Write a Master thesis that motivates your work for a general audience, and communicates the work and its results in a clear, well-structured way to your peers

The final thesis, presented in paper form, explains the value of the research and communicates my results clearly and logically to my peers.

2.3.2 Give a presentation with similar qualities to fellow students and members of the chair

My work will be presented to the graduation committee at the university, just like any other thesis. I will deliver a presentation of similar quality to my fellow students and the committee members.

Appendix A

Conference Paper

Early Warning System for Newly Registered Malicious Domains: A Machine Learning and Certificate Transparency Approach

1^{st} Luuk Berenschot
University of Twente
Enschede, Netherlands
L.berenschot@student.utwente.nl

***Abstract*—Cybercrime is a significant and growing threat, resulting in substantial financial losses annually. The Domain Name System (DNS) is often exploited for malicious activities, such as command and control servers, malware hosting, and phishing campaigns. This research investigates the feasibility of using machine learning in conjunction with Certificate Transparency (CT) logs to detect newly registered malicious domain names. By actively monitoring newly registered domains, we label domains as malicious or benign using blocklists and train a classifier to distinguish between them. Our classifier detects 44% of newly registered malicious domains with a false positive rate of 0.47%. Additionally, our classifier offers customizable precision and recall, allowing for an increase in the detection rate up to 79% at the cost of the false positive rate. The classifier can support registries and registrars in identifying potentially harmful domains.**

1. Introduction

The Domain Name System (DNS) is a crucial part of the Internet that translates IP addresses to human-readable domain names, making it easier for people to locate specific servers connected to the Internet. DNS plays a critical role in the Internet as daily-used services depend on it. For example, email and content delivery networks rely on DNS to function [54].

Unfortunately, DNS is also misused for malicious purposes. Therefore, we monitor DNS for attempts of abuse. Malicious domain names are domains used for criminal activities. Examples are domains used for scams, malware hosting, command and control servers, and phishing. Malicious domain names can persist for a long time. However, in many cases, the malicious domain names are taken down or included in blocklists shortly after registration. Previous research has shown that most malicious domains are misused shortly after registration [18], [22]. Therefore, the challenge is to build a methodology to detect and analyze malicious behavior for newly registered domain names. Current approaches [23], [33], [37], [38] rely on data obtained from registries, zone files, and passive DNS, which are not easily accessible to all researchers and do not always provide real-time data. Our research instead makes use of Certificate Transparency (CT). CT is a publicly verifiable append-only log that creates a public record of issued X.509 certificates. The X.509 certificate can contain one or more domain names. From these certificates, newly registered domains are inferred.

Our research aims to find a method to analyze the behavior of newly registered domain names in the early stages after registration. We measure the potential and feasibility of detecting malicious domains using machine learning. When a newly registered domain is detected, we start to actively measure this domain for a period of 48 hours. We use blocklists to label the collected data as malicious or benign. Using the labeled data, we extract features after which a classifier is trained. The classifier that results from this is able to classify which newly registered domains are likely to be malicious.

We successfully designed an approach that can give an early indication of malicious domain names using a machine-learning classifier. The classifier can detect close to half of the newly registered malicious domains with a precision of 81%.

2. Background

2.1. DNS

The Internet connects servers and clients using unique identifiers called IP addresses. DNS maps IP addresses to human-readable texts, such as www.example.com.

DNS is managed by the Internet Corporation for Assigned Names and Numbers (ICANN) [28]. ICANN is responsible for overseeing the domain name system, accrediting registries and registrars and managing the root zone. DNS can be visualized as a tree structure, with each tree layer having authority over the connected levels below it. This structure is illustrated in Figure 1. ICANN manages the root node and has the power to delegate generic Top Level Domains (gTLDs), such as the .com domain, to registries. Registries are responsible for managing authoritative name servers for their Top Level Domains (TLDs). The .com TLD is managed by the registry Verisign [58]. Registries have the authority to delegate domain authority of the Second Level Domains (SLDs) of their TLDs to registrants. In the case of Verisign, this could be example.com. A registrar plays a role as a middleman between the registry and the registrant and registers domains on behalf of registrants. The registrants themselves can use their domain to create Fully Qualified Domain Names (FQDNs) shown as the bottom layer in the figure. An FQDN can be extended going further down the tree.

2.2. Certificate Transparency

Internet services can make use of X.509 certificates [6]. These digital certificates can bind the public key

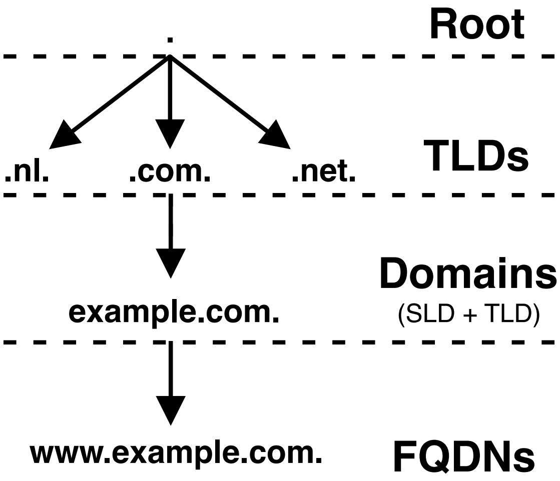


Figure 1. DNS tree representation of www.example.com.

of an entity (such as a web server) to information about the entity, such as its domain name. X.509 certificates help establish trust and security for various internet services, such as secure channels for web browsing [4].

Certificate Authorities (CAs) issue certificates and allow secure connections between clients and entities. CAs are organized in a tree-like hierarchy, similar to DNS, but with multiple root CAs, resulting in multiple chains of trust.

If a malicious actor gains control over a CA, they have the ability to issue fraudulent certificates for any domain in the tree below the CA. A well-known case of a hacked CA is that of DigiNotar [57]. Hackers used DigiNotar to issue fraudulent certificates for major websites such as Google and Microsoft, allowing them to perform man-in-the-middle attacks and steal user information. To prevent this from happening again, Google researchers proposed using Certificate Transparency (CT).

CT is a security standard to increase the security of the Public Key Infrastructure (PKI) used in internet services, including internet browsing [34]. CT has been standardized by the Internet Engineering Task Force (IETF). The current CT 2.0 standard is defined in `RFC 9162` [35]. CT helps investigate the issuance of fraudulent certificates by creating a public append-only log. CT has three components [36].

The first component is the certificate log, which is publicly auditable and contains a record of all issued certificates. Certificate authorities must be transparent and submit all issued certificates to these logs. Domain owners can monitor this log and detect unauthorized certificates issued for their domain.

The second component is a certificate auditor, who monitors the certificate logs and checks for suspicious and fraudulent behavior by certificate authorities. If a CA behaves suspiciously, certificate auditors can report it, which can result in action before significant damage is done.

The final component is the CT log policy, which contains rules and standards for CAs about how they must implement CT to participate in the CT program. A policy can, for example, state to which certificate logs participating CAs must publish their issued certificates [21].

2.3. Blocklists

A blocklist contains domain names or IP addresses that are deemed harmful or unwanted and thus should be blocked. Blocklists can, for example, be used for content filtering, ad-blocking, and network security.

Blocklists can be designed for specific categories, such as phishing or malware. Whenever a domain or IP is added to a blocklist, users can check if the party they are communicating with is present in the blocklist. If the party is deemed malicious, the connection is blocked to prevent damage.

Blocklists are managed by blocklist providers. The providers can be well-known groups or projects as well as individuals. Although blocklists are broadly used, these may not be entirely reliable due to the possibility of containing false positives and false negatives [16]. Blocklists can be created using different approaches.

The first approach is based on user reporting. When users discover a domain that is used for malicious activity, they can report it. If many users report a specific domain in a short period of time and a certain threshold is reached, this domain is added to the blocklist.

The second commonly used approach is based on automated monitoring for malicious behavior. This can be, for example, a network of honeypots or by using middleboxes. For such an approach, machine learning methods are often used to detect malicious traffic. Because of the methodology used to construct these blocklists, false positives can be present.

The third approach is based on fingerprints. Fingerprints are created by collecting identifying characteristics of, for example, malware, such as binary code or network packets. These characteristics are hashed and stored as a reference for future comparison. To detect malware, a new sample is hashed and compared to the stored hash. If the hashes match, malware is present and if there is no match, malware is not present [62]. When a fingerprint is detected in the network traffic, the connected domain or IP address can be added to the blocklist. This approach accurately identifies malicious traffic, but the disadvantage is that the fingerprint must be known beforehand. Another disadvantage is that if the used fingerprints are known, attackers can change their attack method to avoid detection.

3. Related Work

In this section we discuss related work. Our starting point is the work by Khormali et al., which compares over 170 peer-reviewed papers across the threat landscape, research methods, and entities scope [31]. The threat landscape section of this research provides valuable insights into malicious DNS activities.

3.1. Problem Scope

Cybercrime continues to rise yearly. Understanding how domain names are exploited for malicious purposes and to what extent is crucial. The study by the Interisle Consulting Group provides insights into the scale of malicious domain name registrations for phishing [1]. The study identified a total of 1,850,392 phishing attacks from May 2022 to April 2023. This is compared to 1,122,579 attacks during the same period the previous year, an increase of about 65%. This demonstrates that phishing campaigns continue to be a growing concern. Additionally, the report includes details about registrars with the highest ratio of domains registered for phishing. According to the report, the registrars with the highest phishing ratio are “NICENIC INTERNATIONAL” in first place, “TLD Registrar Solutions” in second place, and “REG.RU” in third place. The registrars with the highest count of phishing registrations are “NameSilo” in first place, “PublicDomainRegistry” in second place, and “NameCheap” in third place. We compare these results to our own registrar analysis to determine if the results are comparable. Another research paper written by Hao et al. examines the registration behavior of spammers [24]. It analyses the behavior of domain name registrations and identifies registration spikes and other patterns that can be associated with spammers. As the phishing landscape paper, it provides information on the most commonly used registrars for spam activities. Our research focuses on identifying patterns linked to malicious behavior. We leverage previous literature as an entry point. The European Commission has conducted a very extensive study regarding DNS abuse, showing its magnitude, impact and good practices to mitigate DNS abuse [10]. The study by Korczyński et al. performs a statistical analysis of the new gTLDs [33]. It examines the abuse rate for new gTLDs and compares it to legacy gTLDs. Information about the abuse in TLDs can be used to add a trust score to every TLD. The differences in abuse are likely correlated with the registration price of a domain. New gTLDs can also be misused to impersonate companies by buying the same domain with a different TLD. Another approach to impersonating companies is done by making use of Internationalised Domain Names (IDNs). IDNs are designed to create domain names using characters that are not included in the American Standard Code for Information Interchange (ASCII) characters (a-z, A-Z, 0-9, - and .). This feature is often misused for impersonation of domain names by using special characters that look very similar to ASCII characters [37]. In our research, we explore the possibility of using TLDs and IDNs to give a trust score to domain names. The research paper by Foremski et al. investigates the reasons for domain deletions and the speed at which they occur [18]. According to the findings of this study, on average 9.3% of newly registered domains are deleted within the first seven days, with 6.7% being due to blocklisting. The most common reason for a domain deletion varies depending on the type of TLD. Deletions of legacy gTLDs, such as .com and .net, are most commonly initiated by the domain names registrar. On the other hand, new gTLDs are more likely to be deleted due to blocklists.

3.2. Newly Registered Domain Detection

This section discusses existing approaches for detecting newly registered domains and their advantages and disadvantages.

The first detection approach makes use of registration data directly received from the registry. Using this method the registration is known as soon as the registry receives the request. Getting access to data directly from the registry is hard to achieve for researchers. A registry only has information about the zones they manage, which makes it harder for researchers to scale across multiple registries. To cover a larger part of the DNS researchers have to cooperate with multiple registries. Hao et al. conducted two studies that used registry data of the .com and .net TLD [22], [23]. Spooren et al. also conducted research that had access to registry data [12]. In this study, registry data of the .eu TLD is used.

The second detection approach is passive DNS. This approach makes use of a network of DNS sensors often connected to a database, as is the case for Farsight DNSDB [38], [53]. The advantage of this approach is that it can detect a wider range of domains compared to registry data. A disadvantage of passive DNS is mentioned by Sperotto et al. which is that passive DNS can only detect malicious activity when the domain name is already in use [51]. This means that it may have already been misused before it is detected. The amount of DNS traffic analyzed, and the speed at which malicious domains are detected depends on the amount and location of the DNS sensors [61]. Because the coverage of the sensors is important for the data collection, researchers often opt to use a passive DNS tool such as Farsight DNSDB [14] instead of deploying their own sensors.

The third detection approach makes use of zone files [17], [33], [37]. When a domain name appears in a zone file it will be considered as a newly registered domain. The research by Barron et al. discusses that domain names in a zone file can disappear for a short period even though the domains still exists [3]. Zone files are published and updated by the registries themselves and at different intervals depending on the registry. To access the zone files researchers must request access from each registry. With the introduction of the new gTLDs, the number of registry operators has increased significantly. Because of this, ICANN introduced the Centralized Zone Data System (CZDS) [27]. Researchers can request zone files from multiple registries in a single portal. The zone files published in the CZDS portal are updated every 24 hours, meaning that the zone files can be a day old. The CZDS system is designed to enhance transparency and simplify access to zone files. However, the research by Park et al. shows that over 10% of the requests made through this system do not receive a response within six months. Moreover, a significant number of requests are unjustly denied [45]. Although CZDS simplifies access to zone files, it does not guarantee complete transparency. To enhance the actuality of the zone files, the DNS Incremental Zone Transfer Protocol (IXFR) can be used [42]. IXFR is a protocol for obtaining intermediate zone file updates. For example, by using IXFR, the registry can update the zone information every five minutes instead of every 24 hours, resulting in up-to-date zone data.

The final detection approach makes use of OpenINTEL [43]. The detection of OpenINTEL is based on certificate transparency. The working principles of this detection are described by Sommese et al. [49]. OpenINTEL is able to detect 50% of the newly registered domains within a period of 45 minutes and 98% within a day.

3.3. Malicious Domain Detection

In our research, we use OpenINTEL as the detection method for newly registered domains. Research conducted by Sperotto et al. provides us with three case studies using OpenINTEL and identifies its potential to be used for fast detection of malicious domain names [51]. Other research often makes use of registration data provided by registries [12], [33], zone files [24] or passive DNS [5], [18], [41], [47], [48], [64]. Instead of only using one data source, researchers can use multiple data sources. For example, the research by Korczyński et al. uses both data provided by registries and zone files to enlarge the dataset [33]. During our research, we investigated where the approach using OpenINTEL fits in.

To train our classifier we make use of features introduced by previous research [5], [12], [23], [47]. These features are often based on a single measurement. In our research we make use of multiple measurements to see if changes occur and if they are a meaningful addition to distinguishing between malicious and benign domains. In addition, we introduce other features, such as features related to the FQDNs of a domain.

It is not easy to compare our results with the results of other research. Research that makes use of zone files and registry data often has access to a limited set of TLDs. As a result, their classifiers will be optimized for this specific set. Research that makes use of passive DNS has access to all TLDs. However, these domains are only observed when they are in active use and detected by a sensor. This approach will miss newly registered domains that are not in active use. The research PREDATOR makes use of zone files and registry data from Verisign. Verisign manages the .com and .net TLD [23]. These TLDs make up 59% of the newly registered domains that we observe. Because this is a large part of our measurements and the research also makes use of precision-recall metrics, we compare our results with this work.

Ideally, we want to prevent damage before it has occurred. By using active measurements with OpenINTEL, we aim to detect malicious domains before they are active. A comprehensive study on this topic is conducted by Zhauniarovich et al. [65]. The study compared various research papers introducing detection methodologies and categorized them based on their positive and negative effects. They categorized the research papers into three main categories. These categories are machine learning [2], [5], [23], knowledge-based [9], [20] and hybrid [63]. The research was conducted in 2018 and provides us with interesting methodologies we can use.

4. Data Collection

Since our approach uses machine learning, it is important to collect data that can be used to engineer features that help distinguish between malicious and benign domains.

In this section, we describe the measurement setup for our data collection and the data we collect. A visual representation of our measurement setup is presented in Figure 2. The setup supports two different types of measurements. The first type utilizes the OpenINTEL Newly Registered Domain (NRD) stream, while the second type uses the newly observed Fully Qualified Domain Name (FQDN) stream. When a new domain is detected by OpenINTEL a message is produced in the NRD stream. This message contains the domain that has been registered and the corresponding registration time. Similarly, the FQDN stream produces a message when a new FQDN is observed together with the observation time. The received information is used to start our measurements towards the received target.

The first path is our web crawler path. This path makes use of the OpenINTEL FQDN or NRD stream and places each newly received target into a sorted Redis queue. The targets are sorted based on the time received. For every target, we perform a total of ten measurements. We perform multiple measurements because we want to incorporate changes over time in our features. For every target received from the stream, ten values are placed in the queue at 0 seconds, 10 minutes, 30 minutes, 1 hour, 3 hours, 6 hours, 12 hours, 24 hours, 36 hours, and 48 hours after the target has been detected. These intervals are chosen as we expect the most changes to occur soon after registration. To prevent overflowing the targets with requests we set a limit of 10 measurements over the period of 48 hours. Once the values have been placed in the queue, a constant worker monitors and compares the current time against the values in the queue. All values below the current timestamp are moved to asynchronous HTTP requests, and the responses are collected. From the responses, data such as the HTML page and the response headers are stored. This provides insight into the data hosted by the web server and the architecture used to host the webpage such as the server type. The collected data is temporarily stored in local storage for a period of one hour. At the end of the hour, the data is transferred from local storage to permanent object storage.

The second path utilizes the Registration Data Access Protocol (RDAP) [25], [26]. We request registration data for each newly registered domain immediately upon receipt. From this data, we can retrieve information about the registration date, registrar, registrant, and the use of DNSSEC. In the case of FQDNs, we add the RDAP data for the corresponding domain name. This is done because RDAP is only available for domain names and not for FQDNs.

The third path involves DNS measurements. We employ a DNS crawler to gather DNS data for each NRD and FQDN every ten minutes over a 48-hour period. We collect information such as the IP addresses and nameservers observed during the 48-hour period. Using the collected IP addresses, we also utilize the CAIDA Prefix2AS to perform the mapping of Autonomous System (AS) numbers to our dataset [7]. The AS numbers provide information about the entity or organization that controls the network of the servers used by the domain or FQDN. The prefix-to-AS mapping is updated daily to ensure the numbers are up to date when added to the DNS measurement data.

In addition to adding AS numbers to the dataset, we include location and ISP data. We enhance our dataset using the dataset provided by IP2Location [29]. From the IP addresses we detect with our DNS measurements, we can map these IP addresses to a location and ISP information. This data is relevant as it allows us to train our classifiers to build a trust score for the origin of the

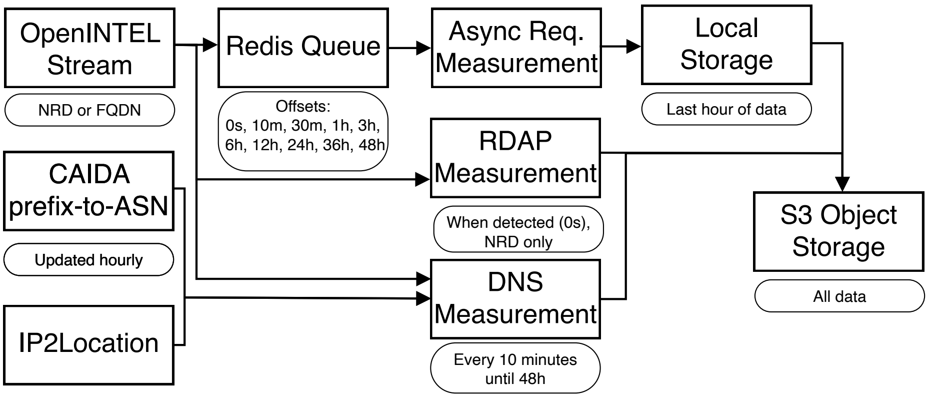


Figure 2. Visual representation of the active measurement

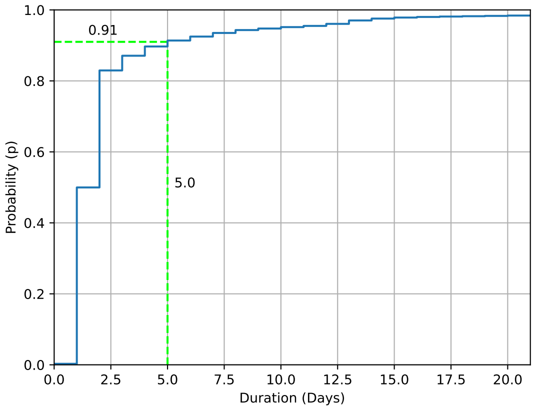


Figure 3. ECDF plot of time between registration and being blocklisted for blocklisted domains

IP addresses.

5. Ground Truth

To be able to train a classifier we need to have a labeled dataset. To label the data we make use of blocklists as they are considered to be our ground truth. First, we define the blocklists to be used. During our research, we crawled multiple blocklists. These blocklists are SpamHaus DBL [50], PhishTank [46], OpenPhish [44], Toulouse (DDoS, malware, crypto, and phishing) [8], CyberCrime Tracker [55], DigitalSide [13], URLhaus [56] and phishing army [15]. We have chosen this set of blocklists as they represent together a diverse set of domains that are blocked because of malicious activities related to DDoS attacks, malware, and phishing. It is also important to know when to consider a domain to be part of a blocklist. Blocklists contain different types of entries. We often see IP addresses, FQDNs and full URL paths. As a consequence, it is often not possible to map a domain one-to-one with a blocklist. To tackle this issue we use the Public Suffix List [19], [32] to extract the TLD and SLD from the blocklist entries. When both the TLD and SLD from our domain match with the TLD and SLD from the extracted entries, we consider our domain to be part of a blocklist.

The next important step is to determine for how long we need to look into the blocklist after a malicious domain has been registered. To determine this we collected newly registered domains for a period of two weeks and looked up the registration date using RDAP. This data is then compared against the appearance in blocklists. We unfortunately do not have a precise timing of when domains are added to the blocklist as our blocklist crawler runs on a daily basis. We only have information on which day the domain has been added to the blocklist. For the blocklist, we initially start by looking at the same period as the registrations but add two weeks to ensure that there is time for domains to appear in the blocklists. To provide a more precise indication of the number of days that we need to look ahead, we present Figure 3. In this figure, an Empirical Cumulative Distribution Function (ECDF) plot shows how many of the blocklisted domains are present in the blocklist after a given amount of time after registration.

A sharp increase of domains being detected can be observed for the first few days. Because of this, we decided to look for the threshold where the amount of blocklisted domains surpasses 90% for the first time. We find that this point occurs at the five-day mark indicated as a green line in the plot. At five days, 91% of the domains are blocklisted. Because of this, we use a look ahead of five days for the blocklist.

In addition to measuring the time it takes for malicious domains to appear in the blocklist, we also investigate how long it takes before a newly registered domain is detected using OpenINTEL. For this, we make use of the findings by Sommese et al. [49]. In this research the difference in time between the RDAP registration date and the appearance in CT logs is investigated. Sommese et al. find that 50% of the newly registered domains are detected within the first 45 minutes of their existence and at the one-day interval, more than 98% of the newly registered domains have been detected. A small increase after one day can still be observed. Because we work in daily intervals we choose to look back for two days in our research. This means that two days are added before our measurement on top of the already added five days after the measurement has taken place. We visually represented this in Figure 4.

Using blocklists as ground truth has its disadvantages. The ground truth is not perfect and may contain false positives and false negatives [16]. This negatively affects the

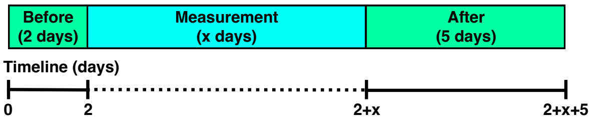


Figure 4. Timeline indicating how many days before and after the measurement we look at the blocklists

TABLE 1. PREFIXED DOMAINS

Blocklist	Domains Flagged
SpamHaus DBL	58,354
Phishing Army	1,354
PhishTank	740
OpenPhish	240
Tolouse Malware	173
Tolouse Phishing	173
URLHaus	22
DigitalSide	7
Tolouse Crypto	4
CyberCrime Tracker	2
Tolouse DDoS	0
Total Unique	59,329

training of the classifier. When the ground truth includes false positives and false negatives, it becomes harder for the classifier to differentiate the classes based on the labeled dataset. Despite these drawbacks, we currently rely on blocklists as we have no better alternative available to label our newly registered domains.

6. Domain Registration Analysis

It is important to identify the domain registration behavior of the data we collect. The behavior found in this section is used as a reference point for what we can expect in our measurements and helps identify possible differences between malicious and benign domains.

For our analysis, we used two weeks of measurement data. Over this period, we detected 1.3 million unique newly registered domains, resulting in, on average, 1.1 newly registered domains being detected per second. Of these newly registered domain names, 59,329 appeared in a blocklist, which corresponds to 4.5%.

To gain a better understanding of the contribution of individual blocklists to this number we show a breakdown of the count of domain names present in the blocklists in Table 1. Multiple blocklists may flag the same domain as malicious resulting in this domain name being counted for multiple blocklists. Because of this, we added a count at the bottom of the table indicating the unique number of domains blocklisted by the blocklists combined. Upon reviewing the table, it is evident that SpamHaus DBL is the blocklist that flags the most domain names and contributes to 98.4% of flagged domain names.

In Figure 5 the quantity of newly registered domains over time is presented. In the figure, we separated the domains that ended up in a blocklist from those that did not. The counts are normalized between zero and one using the minimum and maximum of each group. Normalizing the values allows us to compare the shapes of the two groups despite the significant difference in counts between them.

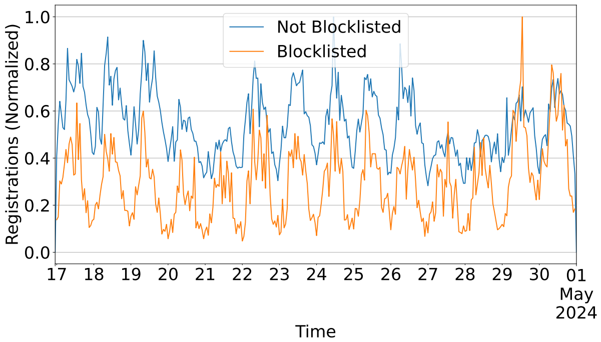


Figure 5. Hourly Registrations (Time = UTC)

We observe a daily cycle in the number of domains registered. Another interesting observation is that registrations during the weekend periods are lower compared to other days (20th, 21st, 27th, and 28th of April). This is especially notable for non-blocklisted domains. When looking at the blocklisted domains, we also see a small decrease, but the difference is less significant.

Because the registration patterns of both blocklisted and non-blocklisted domains show a similar pattern, we assume that both malicious and benign domains are registered with a human in the loop. If the process were fully automated, the registration pattern would look more like a flat line and be a constant stream of registrations.

We also observe spikes where the number of registrations is significantly higher than usual. These spikes may indicate bulk registrations. Research has shown that the ability to register many domain names in bulk aids malicious actors in their malicious practices [33].

We examined the high spike of malicious domains between April 29th and 30th. Upon closer analysis of the spike, we found that some bulk registrations occurred between 12:00 and 12:30 on April 29th. By manually examining the blocklisted domain names within this period, we identified three types of bulk registration. The first type uses a prefix followed by a random set of letters and numbers or numbers only. The second group consists of domain names with a fixed length of seemingly randomly generated numbers. The third type is a specifically chosen domain name often targeting widely used services such as postal services. Similar-looking domain names are often registered at different TLDs or with minor name changes. The groups that we derived from the given time interval are available in Appendix C. The registrations are often performed within a second or a few seconds of each other.

In the same time period, we observed that domains with similar patterns are also present in the non-blocklisted set. Both purely numeric and prefixed domains are present. We also found targeted domains, such as “ceskaposta-cze.top”. The occurrence of purely numeric domains is significantly less compared to the blocklisted set. Based on these results, we cannot directly conclude that all numeric domains are malicious, or that bulk registration is a direct indication of malicious intent. However, the presence of domains such as “ceskaposta-cze.top” in the non-blocklisted set suggests that it is likely that the non-blocklisted domain set still contains domains that should be in the blocklist. Therefore, blocklists are not

TABLE 2. DOMAIN FEATURES

Feature
TLD
SLD length
SLD digit count
SLD vowel count
SLD dash count
SLD entropy
Uses IDN

TABLE 3. RESPONSE FEATURES

Feature
Redirects (min, max, avg, change count)
Final destination (SLD, domain)
Errors (timeout, failures)
Server type
Content type
Content extra
First available offset
Last available offset

perfect for capturing all the malicious domains.

7. Feature extraction

To learn from the data we collect, we need to extract features that help machine-learning classifiers differentiate between malicious and benign domains. We categorize the features into seven different categories. The features used consist of self-engineered features and features used in previous research.

7.1. Domain Name Features

The first category is domain name features. This category contains features that can be obtained exclusively from the domain name. Previous research has shown that lexical features based on the domain name are valuable in distinguishing between malicious and benign domains [23]. The domain features that we extract are listed in Table 2. The first feature, TLD, contains the top-level domain of the newly registered domain. The SLD features capture details about the characters present in the SLD and its length. The digit count records the percentage of digits in the SLD, the vowel count reflects the percentage of vowel characters in the SLD, and the dash count contains the percentage of hyphen characters in the SLD. We also calculate the entropy of the SLD as a measure of how random the SLD is.

These SLD features are already used in existing literature [2], [5], [40], [47]. The features are applied to the SLD Since this is the part of the domain name that the registrant can completely choose themselves. Additionally, we have added a feature to check if the SLD makes use of an internationalized domain name (IDN). IDNs are domain names that use characters other than the Letter-Digit-Hyphen (LDH) subset (a-z), (0-9) or -. We can detect these domains by checking if the SLD starts with the ASCII Compatible Encoding (ACE) prefix `xn--`. We included this feature because research has shown that IDNs can be misused to create domains that look indistinguishable from already existing domains. These domains are often referred to as semantic homographs and make use of special characters that closely match with characters in the LDH subset [60].

7.2. Response Features

The second category describes response features. These features are created from the responses collected by our crawler using asynchronous HTTP requests. This category is introduced as we expected to see similar behavior for multiple domains such as hosting the same HTML and using similar configurations. The features that are extracted are presented in Table 3. The first set of features in this category are redirect features. During our measurement, we follow all redirects and store the chain of redirects. We calculate the minimum, maximum and average amount of redirects for each domain over all measurements. We also count how often the chain of redirects changes. Other features related to redirects are the final destination features. We have two of these features the first feature checks if the SLD is in the final destination when following the redirects. An example of a “True” value here would be from “example.nl” redirected to “example.com”. The second feature checks if the entire domain is in the final destination. Thus “example.nl” should be in the final destination. To identify the architecture of the server used, we collect the server type and content type from the response header. We store extra information about the content type, such as the charset, as a separate feature. We also store at which measurement the domain became first available and if it happens when the domain became unavailable. The last available offset is useful to detect early deletions of domains. If a domain becomes unavailable in between the available offsets a classifier can still learn from this as the measurement failure count is increased.

7.3. DNS Features

The third category is DNS features. These features are created from the collected DNS responses. The DNS response features are listed in Table 4. The first features are extracted from the A records. The A records contain the IPv4 addresses of the domain name. We store the minimum maximum and average amount of IP adresses in each query. Additionally, we count how often the set of IPv4 addresses per query changes and we store the time to live values for the DNS records. Exactly the same is done for the AAAA records. The AAAA records store the IPv6 addresses. The NS records contain information about how often the IPv4 and IPv6 addresses of the nameservers change. The last DNS feature we check concerns the MX record. The existence of the MX record indicates that the domain is set up to use Email.

7.4. RDAP Features

The fourth category contains RDAP features. These features are extracted from the RDAP responses. We send

TABLE 4. DNS FEATURES

Feature
A (min, max, avg, time to live, change count)
AAAA (min, max, avg, time to live, change count)
NS (ipv4 changes, ipv6 changes)
MX

TABLE 5. RDAP FEATURES

Feature
Registrar
Uses DNSSEC

an RDAP request for each domain directly after it is detected by OpenINTEL. The RDAP features are listed in Table 5. There are two features in this category. The first feature is the registrar. The registrar is the party that aided the registrant in registering the domain name. The second feature is “Uses DNSSEC”. This feature indicates if the domain makes use of the DNSSEC extension. DNSSEC provides a cryptographic security layer helping DNS to prevent DNS spoofing. This feature provides information regarding security measures set up for domains.

7.5. IP Features

In the fifth category, we focus on IP-based features. We gather additional information based on the IPs obtained in our DNS measurements. We utilize the IP2Location DB23 dataset [30] to include location, ISP, and usage type data. Additionally, we leverage CAIDA Prefix2AS [7] to map IPs to the corresponding Autonomous System Numbers (ASNs). The features belonging to this category are presented in Table 6. Because a domain can have multiple DNS records storing different IPs, the features in this category are stored as a set of unique values. This allows us to detect if a domain is hosted in multiple countries or makes use of multiple ISPs.

7.6. Registration Time Features

The sixth category focuses on features related to the time of registration. For these features, we examine the domain registration activity in a specific time window before a domain is registered, comparing it to other domains that have already been registered. In Section 9.1 we will determine the optimal window size. The features are presented in Table 7. Within the window, we analyze similarities between registrations. We record the number of domains registered within the window. Using this count, we calculate the normalized values for domains using the same registrar, registrant and server and domains hosting the same HTML. Additionally, we calculate the normalized Levenshtein distance between the SLD of the newly registered domain and the domains registered within the window [52]. From this distance, we calculate the minimum and average distance. Furthermore, we categorize all distance results into ten equally sized bins, indicating the percentage of domains falling within specific distance ranges. The features in this category are designed to detect bulk registrations and identify registrations that are similar, potentially indicating they are initiated by the same entity. Identifying bulk registrations can be useful as previous research identified that bulk registration lowers the bar for abuse and can be useful for malicious actors [33].

TABLE 6. IP FEATURES

Feature
Countries
Regions
Cities
ISPs
Usage types
ASN numbers

TABLE 7. REGISTRATION TIME

Feature
Domain count
Matching HTML (percentage)
Matching registrar (percentage)
Matching registrant (percentage)
Matching server (percentage)
Domain distance (min, avg, bins)

TABLE 8. FQDN FEATURES

Feature
FQDN count
FQDN length (min, max, avg)
FQDN vowel count (min, max, avg)
FQDN digit count (min, max, avg)
FQDN dash count (min, max, avg)
FQDN dot count (min, max, avg)

7.7. FQDN Features

The last category concerns FQDN features. This is a novel category for determining malicious domain names and, as of our knowledge, has not been used in previous research. The novelty here is that in comparison to other research, we also have access to information about FQDNs registered for a domain name. Because we make use of an approach based on certificate transparency, we do not only know information about issued certificates for domain names, but we also see information about fully qualified domain names. The first feature ”FQDN Count“ stores the number of FQDNs found for the newly registered domain name. We also extracted similar features as the domain features which we now calculate over all FQDNs belonging to a domain. Because this can result in multiple values per domain we calculate the minimum maximum and average for each of these features. In addition, we added a dot (.) count to calculate how many levels the FQDNs consist of.

TABLE 9. CLASSIFIER RESULTS

	Metric			
Classifier	Precision	Recall	PRC	ROC
LR	0.76	0.34	0.54	0.89
RF	N/A*	0.0	0.32	0.82
GBT	0.76	0.35	0.55	0.87
SVM	0.80	0.32	0.51	0.84

* N/A indicates that we could not calculate the value. This is because a division by zero occurs in the formula.

8. Classifier Selection

In this section, we determine the best classifier by training multiple classifiers and analyzing their performance. All classifiers are trained using a two-week training set and a validation set of the following week.

We test four different classifiers. These classifiers are Logistic Regression (LR), Random Forest (RF), Gradient-Boosted Trees (GBT) and Support Vector Machine (SVM). In this section, we analyze the results of these classifiers to determine the best classifier to use. The results of these classifiers are presented in Table 9.

In our research, we make our decisions based on the area under the Precision-Recall Curve (PRC). We use this scoring metric because it is less sensitive to class imbalance. This is because the area under the PRC does not take into account the number of true negatives in its calculation (how well the classifier can predict a domain is benign). As a result, it gives a better representation of how well the classifier is able to predict if a domain is malicious. In other research [5], [23], they make use of the ROC as a scoring metric. In the case of the area under the ROC, the number of true negatives is also used in the calculation. Because the majority of the values in our set are labeled as benign, predicting only the benign class results in many true negatives resulting in a high area under the ROC while the PRC would be low. The research by Cook et al. describes in depth why the area under the PRC is preferred over the area under the ROC in case of a class imbalance [11]. To be able to compare with related research, we also added the area under the ROC score in our tables and plots. Keeping this into account, we conclude that the GBT classifier is best suited for our goal as it has the highest PRC.

In the baseline analysis we have already identified a class imbalance. A class imbalance can have a negative impact on the performance of a classifier because when one class is more present, the classifier is more likely to predict the majority class. We explore the effects of two approaches to counter the imbalance. The first approach makes use of weighted training. In this approach, we ensure that each class has an equal weight of 50%. The second approach involves undersampling, where a random subset of the majority class (benign) is sampled to match the number of samples of the minority class (malicious). While undersampling may result in loss of information during training, it can help prevent overfitting on the majority class. Given that undersampling results in some information loss, there is a possibility of varied performance when different sets are sampled. To account for this, we conduct five measurements to obtain the undersampled results and calculate the average and standard deviation. This approach provides an understanding of the performance impact when a different set is sampled. The results of our balancing approaches for each classifier are presented in Table 10.

TABLE 10. BALANCED CLASSIFIER RESULTS
(W = WEIGHTED & U = UNDERSAMPLED)

	Metric*			
Classifier	Precision	Recall	PRC	ROC
LR (w)	0.25	0.79	0.48	0.89
LR (u)	0.25/0.007	0.78/0.010	0.40/0.02	0.89/0.005
RF (w)	0.19	0.65	0.33	0.84
RF (u)	0.20/0.026	0.62/0.025	0.29/0.05	0.84/0.010
GBT (w)	0.21	0.78	0.45	0.90
GBT (u)	0.21/0.004	0.78/0.002	0.46/0.01	0.90/0.002
SVM (w)	0.26	0.79	0.41	0.89
SVM (u)	0.27/0.002	0.79/0.003	0.40/0.01	0.89/0.002

* The results of undersampling are based on five training rounds and are represented as (average/standard deviation)

Based on the results for countering the imbalanced dataset, we observed that the PRC decreased for all classifiers except the RF classifier. Notably, the imbalanced GBT classifier still has the highest score. However, we noticed a significant difference in precision and recall. In the case of the imbalanced dataset, precision was high, but recall was low, while in the balanced datasets, recall increased significantly, but precision decreased. This trade-off between precision and recall is known as the precision-recall trade-off and will be discussed more elaborately in Section 10.3. For now, we continue with the classifier that has the highest PRC score, which is the imbalanced GBT classifier with an area under the PRC of 0.55 as shown in Table 9.

9. Classifier Tuning

9.1. Correlation Window

For our current scores, a correlation window of five minutes is used to detect bulk registrations. This five minutes is taken from the related work PREDATOR [23]. In PREDATOR a window of five minutes is used as they receive the newly registered domains in batches of five minutes. We believe that the performance could be improved by selecting a more appropriate window size that fits our data. Since the data we collect is continuous, we calculate a unique correlation window for each domain registration.

To determine the optimal correlation window size, we use one week of training data and one week of validation data. The correlation features are calculated for window sizes of 1, 2, 5, 10, 20, and 30 seconds and 1, 2, 4, 5, and 6 minutes. We selected these values because we anticipate that the most effective window size falls within the seconds range. This expectation is based on the observation that similar domain names are often registered within a few seconds. The minute measurements are added to make a comparison with the current five-minute window size.

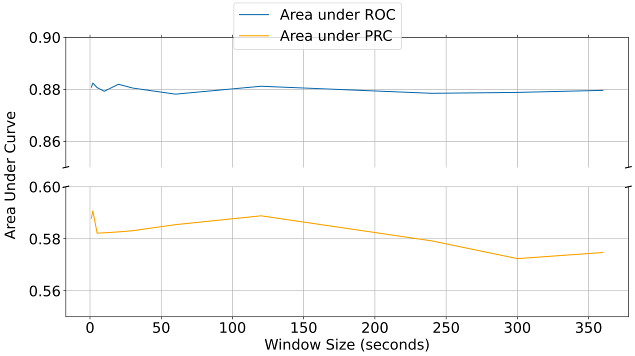


Figure 6. Window size performance scores

TABLE 11. Tuned classifier results

Tuning	Metric Precision	Recall	PRC	ROC
maxIter=40 & maxDepth=10 & stepSize=0.05	0.82	0.46	0.66	0.92

For each window size, we trained a classifier and calculated the performance scores of the area under the PRC and the area under the ROC. The results are presented in Figure 6. In the figure, we observe a slight increase in performance for the Area under the PRC and the Area under the ROC curve at a window size of two seconds, after which the performance decreases again and shows a more constant trend for both scores. Since a window size of 2 seconds gives the best performance, we adapt our window to this size.

9.2. Hyperparameter Tuning

We expect to improve the performance of the classifier by tuning its hyperparameters. The available hyperparameters can vary depending on the classifier used. For the GBT classifier, we use the depth, iterations, and step size. To find suitable hyperparameter values, we use cross-validation to validate whether the changes improve the classifier.

The untuned hyperparameters have a default value of five for maximum depth, 20 for maximum iterations, and 0.1 for the step size. The classifier is evaluated based on the area under the PRC. The results of the cross-validation indicate that the best parameters are a maximum depth of ten, maximum iterations of 40, and a step size of 0.05. In Table 11 the results are presented. We observe that the recall, precision, PRC and ROC scores all have increased by tuning the hyperparameters. All tuning parameters are at the maximum or minimum value of the cross-validation settings leaving room for improvement. Cross-validation is time-consuming and finding the optimal value can be a lengthy process. Increasing the depth or maximum number of iterations significantly increases the training time. Because of the already long training time, we settled with the current best-found hyper-parameters. This results in an increase of 0.11 in the area under the PRC.

TABLE 12. Impact of the measurement count on the classifier performance

Measurements	Metric PRC	ROC	Domains detected	Block-listed
1 measurement (0s)	0.67	0.92	478,540	26,480
5 measurements (3h)	0.66	0.92	547,477	27,949
10 measurements (48h)	0.66	0.92	621,208	31,675

9.3. Measurement Count

Since we are performing ten measurements over a 48-hour period, we want to understand how the number of measurements affects the performance of our classifier. To test this, we configured our classifier according to the best tuning we previously identified. Since forming the dataset and training the classifiers takes a significant amount of time, we use an approach where we try to find an optimum between the number of measurements and the best performances of the classifier. In this approach, we compare three measurement groups. The first measurement only, the first five measurements, and all ten measurements. The results are depicted in Table 12.

Upon reviewing the scores, we notice that the scores are very similar for each group of measurements. This suggests that the number of measurements does not significantly affect the results. However, this is not entirely accurate. In our approach, domains that failed measurement are excluded from the results. Taking this into consideration, we observe that the number of measurable domains after ten measurements is greater than at the first measurement. This is expected, as some domains may be inaccessible when they are just registered, for example, the DNS server might not resolve the domain yet. After 48 hours, we observe a total of 621,208 measurable domains. Right after detection, at the 0-second interval, we observe 478,540 domains, which amounts to 77% of the domains directly after registration. When considering the block-listed domains within this subset, we observe 31,675 after 48 hours and 26,480 directly after registration, which is 83.6% of blocklisted domains. From this, we can infer that even though we are able to identify more domains over a 48-hour period, we can already observe close to 80% of the blocklisted domains after the initial measurements and make predictions about them with a comparable area under the PRC and area under the ROC curve. Looking at the number of domains and blocklisted domains at the three-hour measurement mark, we see that the amount of domains detected has already significantly increased, showing that training classifiers at different timesteps allow for an increase in predictions over time. To take into account as many domains as possible we make use of all ten measurements.

9.4. Train Time

Another aspect we explore is the impact of varying sizes of the training data on performance. We test different training set sizes to ensure a fair comparison using the same validation set. We use data from one to 14 days before the validation set as a train set. The results are

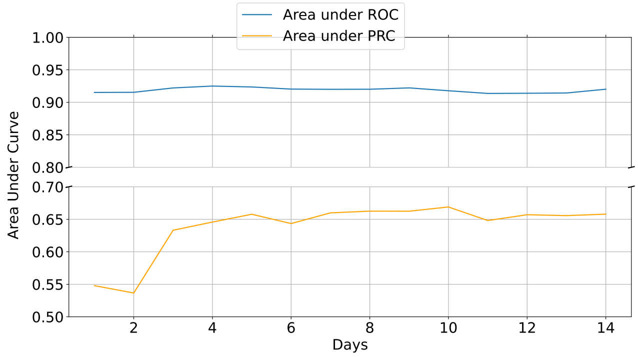


Figure 7. Scores on validation set based on x days of training

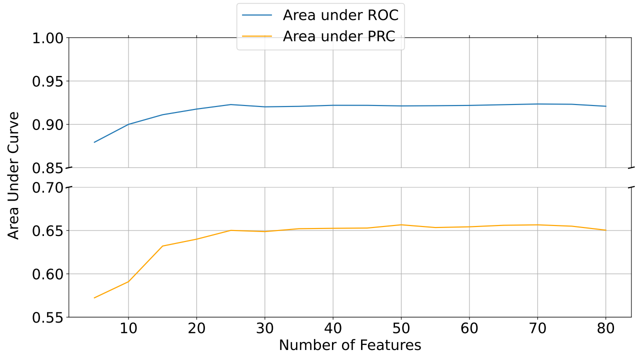


Figure 8. Scores of the classifiers trained on a subset of the features

outlined in Figure 7. In the figure, we can see that the area under the PRC increases until five days of training. For the area under the ROC curve we also observe a slight increase until five days. After the five days both scores stay consistent. This indicates that the data added after these five days is similar to the data already present in the training set and does not increase the performance of the classifier when added. Because of this finding, we use five days of training in the next steps.

9.5. Feature Analysis

Currently, many features are used, some of which may have no impact on the prediction. The GBT classifier we use has the ability to provide feature importances.

To determine which features are important we order the features based on their feature importance and train several classifiers using subsets of features. We begin with the top five significant features and then add the next five important features in each iteration until all features are included. In Figure 8 the results of the area under the ROC curve and the area under the PRC using the validation set are presented. We observe that the score increases up to the first 25 features. We present the top 25 features in Table 13

We analyze these top 25 features further. The first two features, “registrar” and “isps” have significant importance compared to other features in the list.

The first feature group we examine is the registrar. This feature consists of 965 unique registrars. The three registrars with the highest feature importance are presented in Table 14. We observe that NICENIC holds the most significant feature importance, followed by NameSilo and PublicDomainRegistry (PDR). It is noteworthy that these registrars are also found in the PhishingLandscape paper [1]. Where NiceNic is presented as the registrar with the highest ratio of phishing domains, and NameSilo and PublicDomainRegistry as the first and second registrars with the highest number of phishing domains. High feature importance indicates that the feature contributes significantly to distinguishing malicious and benign domains. Furthermore, a high feature importance does not imply that it can single-handedly distinguish the classes well. However, in combination with other features, it can play a major role in separating the classes into subgroups. Table 14 also provides the counts of how often the registrar occurs as blocklisted and as non-blocklisted along with the percentage of domains being blocked for this registrar. Notably, 91.8% of the newly registered domains of NiceNic are present in a blocklist, confirming it is a good predictor and showing that the feature is likely to distinguish between malicious and benign well on its own with reasonable accuracy. Looking at NameSilo and PDR, we observe a significantly lower percentage of blocklisted domains, thus requiring additional features to correctly classify domains using these registrars. The results imply that certain registrars are more likely to host malicious domain names compared to other registrars. However, even when the likeliness of hosting malicious domain names is low for a registrar the registrar can still be a valuable feature to classify a domain as malicious.

TABLE 13. Top 25 features

Feature	Importance
registrar	0.165
isps	0.131
tld	0.054
server_type	0.052
content_type	0.042
country_codes	0.037
NS_A_changes	0.036
avg_A_TTL	0.036
max_A_TTL	0.022
NS_AAAA_changes	0.021
regions	0.020
max_normalized_fqdn_dash_count	0.020
sld_len	0.017
min_fqdn_len	0.016
cities	0.016
max_normalized_fqdn_dot_count	0.016
A_changes	0.015
max_normalized_fqdn_digit_count	0.015
html_changes	0.014
unreachable_offset	0.014
min_normalized_fqdn_digit_count	0.013
timeouts	0.013
connection_failures	0.012
avg_normalized_fqdn_dot_count	0.011
normalized_sld_digit_count	0.009

The next feature we analyze is the ISP feature which consists of 3,639 unique ISPs. The top three ISP features are presented in Table 15. We observe that Unified Layer (UL) has the highest feature importance, followed by LIMENET and IP Khnykin Vitaliy Yakovlevich (IPK), which have very similar feature importance. In the table,

TABLE 14. TOP THREE REGISTRARS

Registrar	Importance	Blocklisted	Not blocklisted	Percentage blocked
NICENIC	0.021	993	89	91.8
NameSilo	0.014	1489	10068	12.9
PDR Ltd	0.013	989	8221	10.7

TABLE 15. TOP THREE ISPS

ISP	Importance	Blocklisted	Not blocklisted	Percentage blocked
UL	0.0113	467	3923	10.6
LIMENET	0.0096	275	17	94.5
IPK	0.0094	70	401	14.9

it is visible that both UL and IPK do not have a high percentage of being blocklisted, making them features unable to effectively classify malicious domains based solely on the ISP. When looking at the LIMENET ISP, we can see that 94.5% of the domains in the training set connected to this ISP end up in a blocklist.

It is noteworthy that UL has a higher feature importance compared to LIMENET. This can be explained by the fact that UL is a more common ISP and, in combination with other features, is able to extract more malicious domains compared to using LIMENET, making the UL feature more important for decision-making. These results imply that some ISPs are more likely to host malicious domain names compared to other ISPs. However, even if the likeliness of being malicious for an ISP is low it can still be valuable to aid the classification in predicting a domain to be malicious.

For both the "registrar" and "isps" feature we have a more elaborate plot showing the top 20 importances in Appendix D. From these plots, we derived that we were most interested in the top 3 for both features.

Analyzing the other features present in the top 25, we notice that the domain and FQDN features appear multiple times. These features are based on the lexical aspects of the domain name or FQDN. The frequent appearance of these features indicates that the lexical aspects play a significant role in determining whether a domain is malicious. These features have in common that they are known at the time of registration. Although there is no previous research about the impact of lexical features of newly observed FQDNs, our observation that lexical features of domain names impact the decision-making of the classifier is confirmed by previous research [23].

Other impactful features include the server type and content type obtained from the response of the target domain. Moreover, location features such as country codes and cities collectively have a significant impact. In terms of DNS features, we observe that the number of name server changes and the time to live of the A record also show a significant impact.

Since we found the 25 features presented in Table 13 contribute the most and adding the other features does not improve the classifier scores, upcoming steps only use these 25 features.

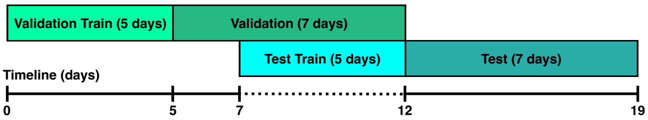


Figure 9. Timeline showing the difference in time between validation and test

TABLE 16. VALIDATION AND TEST SCORE COMPARISON

	Metric			
Tuning	Precision	Recall	PRC	ROC
Validation	0.78	0.47	0.65	0.92
Test	0.81	0.44	0.63	0.91

10. Evaluation

10.1. Test Set Performance

Our decisions so far have been based on a validation set, which raises the concern that the choices made to tune the classifier might be biased towards this particular set. To verify the reliability of our classifier, we use a test set consisting of unseen data. We train the classifier on the last five days of the validation set and test it on the following unseen week, as shown in Figure 9. This results in a train-test split of about 42% train data and 58% test data. The scores for both the validation and test set are listed in Table 16. We observe that the performance on the validation and test set is very similar. This suggests that the performance of our classifier is robust and can make predictions with similar accuracy on unseen data. In the test set, there are a total of 597,061 domains. It takes 69 seconds to make predictions for all these domains. This means that, on average, our classifier can classify a domain in less than 0.12 milliseconds.

10.2. False Positives

Our assumption is that blocklists are not complete and miss malicious domains. This would mean that some of our false positives might actually be true positives. Since the number of false positives is not too large (2,702), we further analyze them. We came up with four approaches to determine if our false positives are likely to be true positives.

The first approach involves checking the blocklist for a longer period. Currently labeling makes use of looking in the blocklist two days before the measurement to five days after. While this covers most of the domains in the blocklist, there is still a percentage that we miss. To address this, we decided to examine the blocklist data over the entire period we collected the blocklist data for our research. This measurement starts three weeks before the first value in the test set and extends for over a month after. By using this approach, we found 38 domains that were blocklisted outside our window.

The second approach involves using the VirusTotal API [59]. We waited for over a month after measuring the

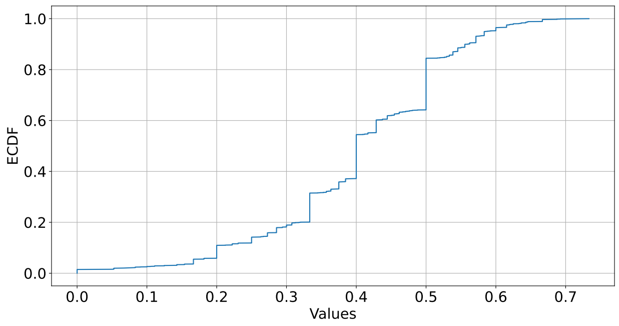


Figure 10. False positive blocklisted distance ECDF plot

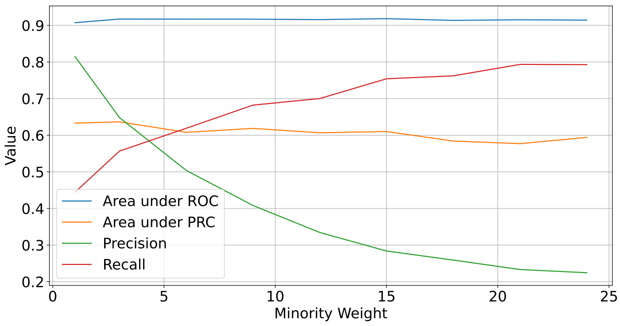


Figure 11. Plot showing the precision-recall trade-off

domains and then analyzed the false positives. VirusTotal provides us with information on how frequently domains are flagged as malicious or suspicious by other threat intelligence services and blocklists. We use this data to determine if the domains in our false positives are considered malicious by other parties that we previously did not take into account and when more time has passed. Through this approach, we discovered that 366 out of the 2,702 false positives have been identified as malicious, while 53 are listed as suspicious only. In our analysis, we treat both malicious and suspicious as malicious. Using VirusTotal, we were able to identify 419 domains in the false positives that should likely be classified as true positives.

The third approach involves checking the false positive domains against the set of newly registered domain names labeled as blocklisted within the same test set. We cross-joined the false positives with the blocklisted domains and calculated the normalized Levenshtein distance between the SLDs of the domain names [52]. For each false positive, we stored the blocklisted domain with the shortest distance to the false positive. When the distance is very small, it is likely that the domain has been missed by the blocklist. For example, we observed multiple domains with a distance of zero, indicating that they had exactly the same SLD as a newly registered domain that is blocklisted. We also often saw registrations with the same prefix but a few digits incremented.

To visualize the amount of false positives present below a certain threshold, we created an ECDF plot, shown in Figure 10. The figure shows small increases in domains and a significant increase at 0.2. Therefore, we set our threshold to consider all domains before this significant increase at 0.2 to be considered malicious, resulting in a total of 158 domains below this threshold.

Our final approach is a manual search for well-known companies and flagging domain names that could potentially be used in phishing campaigns. We look for domains that closely resemble legitimate companies or services, such as those related to post offices. Our search focused on common words and known company names such as “post”, “mail”, and “amazon”. Although these domains are not guaranteed to be registered for malicious purposes. For example, they can be registered as preventive measures by the company itself. The registrations of these domains do pose a risk to the companies and are likely undesired. In total, we found 108 of these domains that could be used for malicious activities such as scams or phishing.

After comparing the results from all four methods, we found evidence supporting our initial assumption that malicious domains are likely present in our false positives. When we combine all four approaches, we identify a total of 620 domains. It is important to note that this number is lower than the sum of all four approaches because there is an overlap in the results. This means that some domains are flagged as malicious by multiple approaches. Out of the 2,702 false positives, we have evidence that 620 of them are likely malicious. This implies that the precision of our classifier is expected to be higher than the score we obtained from our previous tests.

10.3. Precision Recall Trade-Off

We have identified that our classifier introduces some false positives. Therefore, it is recommended to manually inspect the positive predictions made by the classifier. In this section, we discuss how the amount of false positives introduced can be adjusted. This could be helpful for registries or registrars who wish to use the system and want to adjust the number of false positives according to their available analysis capacity in their work processes. For example, it could be used to automatically request additional verification from registrants. Increasing the amount of true positives predicted (recall) also increases the number of introduced false positives. As a result, registry or registrar operators may need to manually verify more domains, increasing their workload. If the workload becomes too high, it could become unmanageable.

In this section, we demonstrate how the precision-recall trade-off can be used to adjust the number of false positives introduced. We apply a balancing factor to the GBT classifier, starting with a one-to-one weight, which is fully imbalanced, and then increasing the weight of blocklisted domains from one to 24. We chose 24 because it aligns with 4% of malicious domain names present in the dataset which is close to the expected 4.6% blocklisted domains.

In Figure 11 we observe that the area under the PRC and the area under the ROC remain relatively unchanged. When the weight of blocklisted values is increased during training, more true positives are detected, but also more false positives are introduced. This adjustment can be used to tailor the number of acceptable false positives. For our test set, we observe that the precision starts at 81% with a recall (detection rate) of 44%. By increasing the recall to 79% the precision decreases to 22%.

11. Limitations and Future Work

In this section, we discuss the limitations of our system and how it can be improved in the future. Our system uses OpenINTEL to detect domain names. Research by Sommese et al. [49] shows that rapid DNS updates can improve detection speed. OpenINTEL data is publicly available, we expect that utilizing rapid zone updates would result in faster detection. However, gaining access to these files can be time-consuming and difficult. Therefore, we advocate for an open environment with rapid zone file updates accessible to the public at a frequent time interval, such as every five or ten minutes. This information is valuable for early classification of malicious domain names, preventing harm, and making the Internet a safer place.

One of the major limitations in our research is the lack of a reliable ground truth. Using blocklists as a ground truth is not ideal for multiple reasons. It is often unknown how the blocklists are created, and the reasons why a domain shows up in a blocklist are missing. Furthermore, blocklists, as ground truth, also miss a significant number of malicious domains that never end up in a blocklist, even though they are malicious. This has a negative impact on both the training and evaluation of the classifiers. It negatively impacts training because malicious domains are present in the benign class, which affects the distinction between the two classes. Also, when evaluating the results, they are not fully reliable. A low precision score might mean that the classifier is actually performing very well when the false positives turn out to be true positives. For future work, it would be a good idea to optimize the ground truth. Approaches that try to optimize the ground truth could have a positive impact on this research and may improve scores. An example of this is removing parked domain names from the training data, as proposed by Lloyd et al. [39].

We have not conducted an extensive system comparison with existing literature. Two related works, PREDATOR and Premadoma [12], [23], are very similar, but unfortunately, we do not have access to their source code or the ability to run our data through their implementation. Therefore, our comparison is limited to the recall and false positive rate results presented in their papers. PREDATOR claims a recall of 70% for the .com TLD and 61% for the .net TLD with a false positive rate of 0.35%. Premadoma achieves a recall of 66.23% with a false positive rate of 0.30%. In comparison, our classifier achieves a recall of 44% with a false positive rate of 0.47% for all TLDs. There could be several reasons for this difference in performance. The results of the related works show that the recall can vary significantly per TLD. Our research includes all TLDs, unlike the related works that focus on specific TLDs. This makes our classifier more versatile, but it may come at the expense of recall. Our research is conducted years after the work we compare to. In the meantime, malicious actors may have changed their approach to avoid detection. Without comparing the exact same data across these implementations, we cannot make a proper performance comparison. Further research is needed to understand how our implementation compares to other existing works.

In our research, we conducted hyperparameter tuning using cross-validation. We discovered that these parameters greatly impact the performance by increasing both precision and recall. Due to the long time it takes to perform cross-validation, we decided not to add extra parameters that would increase the training time of the classifiers even more. However, we have not yet determined the point at which the results start to decrease, or the improvements become less significant. We anticipate that further tuning of the hyperparameters will likely lead to improvements in the scores.

Our methodology describes a measurement setup that performs ten measurements for each input value over a period of 48 hours. In the results, we have seen that the number of measurements does not really make a difference in classifier scores. Also, analyzing the features resulted in the most important features not being dependent on multiple measurements. Additionally, we have seen that performing measurements at a later time frame does contain more domains with a successful measurement. We expect that some domains are not fully active yet at the first few measurements and become active later in time. Because of this, we expect an approach where a measurement is performed once, and if this measurement fails, a new measurement will be performed at a later time is better. This methodology of measurement will decrease the amount of traffic introduced by our crawlers while we expect the results to stay the same.

In our research, the classifier performance is determined by predicting a one-week period. In future research, we aim to examine how the performance of our classifier is affected when predicting for shorter or longer periods. This investigation will provide insight into the duration for which the same classifier can be utilized before requiring retraining.

12. Conclusion

In this research, we developed a system to detect malicious domains through active measurement of newly registered domain names. Our approach demonstrates the potential for identifying malicious domain names close to their registration, thereby mitigating risks posed by malicious domain registrations.

Our findings show that our system is able to classify domains based on registration data, DNS responses, and characteristics collected by our web crawler. However, we also identified areas for improvement. One of these improvements is our detection timeliness. Although our system can be quicker compared to passive DNS, the current reliance on OpenINTEL does show a delay in domain detection compared to rapid zone updates. We aim to address this by advocating for more accessible rapid zone updates.

Furthermore, there is a trade-off between precision and recall. When striving for high precision, a significant number of malicious domains may slip through undetected. On the contrary, prioritizing high recall to catch more malicious domain names can negatively impact precision, resulting in more false positives.

In any scenario, false positives can be present, which makes the system less usable as a fully automated system for flagging malicious domains. We recommend using the system as an indicator for malicious domains, for instance, registrars could use it to request additional verification of the registrant. In addition, involving a human in the evaluation process is a good solution. The amount of work required can vary significantly depending on the registrar. Therefore, it is beneficial that our classifier is adjustable to find the right balance between recall and precision that suits the registrar.

References

[1] Greg Aaron, Lyman Chapin, David Piscitello, and Colin Strutt. Phishing landscape 2023.

[2] P. Mohan Anand, T. Gireesh Kumar, and P.V. Sai Charan. An ensemble approach for algorithmically generated domain name detection using statistical and lexical analysis. *Procedia Computer Science*, 171:1129–1136, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[3] Timothy Barron, Najmeh Miramirkhani, and Nick Nikiforakis. Now you see it, now you Don't: A large-scale analysis of early domain deletions. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, pages 383–397, Chaoyang District, Beijing, September 2019. USENIX Association.

[4] Diana Gratiela Berbecaru and Antonio Lioy. An evaluation of x.509 certificate revocation and related privacy issues in the web pki ecosystem. *IEEE Access*, 11:79156–79175, 2023.

[5] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. Exposure: A passive dns analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.*, 16(4), apr 2014.

[6] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and David Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008.

[7] CAIDA. CAIDA pfx2as. https://www.caida.org/catalog/datasets/routeviews-prefix2as/. [Accessed 24-5-2024].

[8] Université Toulouse Capitole. Blacklists ut1. http://dsi.ut-capitole.fr/blacklists/index_en.php. [Accessed 27-5-2024].

[9] Hyunsang Choi, Hanwoo Lee, Heejo Lee, and Hyogon Kim. Botnet detection by monitoring group activities in dns traffic. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 715–720, 2007.

[10] European Commission, Content Directorate-General for Communications Networks, Technology, I Paulovics, A Duda, and M Korczynski. *Study on Domain Name System (DNS) abuse*. Publications Office of the European Union, 2022.

[11] Jonathan Cook and Vikram Ramadas. When to consult precision-recall curves. *The Stata Journal*, 20(1):131–148, 2020.

[12] Lieven Desmet, Jan Spooren, Thomas Vissers, Peter Janssen, and Wouter Joosen. Premadoma: An operational solution to prevent malicious domain name registrations in the .eu tld. *Digital Threats*, 2(1), jan 2021.

[13] DigitalSide. DigitalSide Threat-Intel. http://osint.digitalside.it. [Accessed 27-5-2024].

[14] DomainTools. Farsight DNSDB 2.0. https://www.domaintools.com/products/farsight-dnsdb/. [Accessed 4-2-2024].

[15] Andrea Draghetti. PhishingArmy — The Blocklist to filter phishing! https://phishing.army/. [Accessed 27-5-2024].

[16] Álvaro Feal, Pelayo Vallina, Julien Gamba, Sergio Pastrana, Antonio Nappa, Oliver Hohlfeld, Narseo Vallina-Rodriguez, and Juan Tapiador. Blocklist babel: On the transparency and dynamics of open source blocklisting. *IEEE Transactions on Network and Service Management*, 18(2):1334–1349, 2021.

[17] Simon Fernandez, Maciej Korczyński, and Andrzej Duda. Early detection of spam domains with passive dns and spf. In Oliver Hohlfeld, Giovane Moura, and Cristel Pelsser, editors, *Passive and Active Measurement*, pages 30–49, Cham, 2022. Springer International Publishing.

[18] Pawel Foremski and Paul Vixie. The modality of mortality in domain names. *Virus Bulletin*, 2018.

[19] Mozilla foundation. Public Suffix List. https://publicsuffix.org/. [Accessed 27-5-2024].

[20] Hachem Guerid, Karel Mittig, and Ahmed Serhrouchni. Privacy-preserving domain-flux botnet detection in a large scale network. pages 1–9, 01 2013.

[21] Josef Gustafsson, Gustaf Overier, Martin Arlitt, and Niklas Carlsson. A first look at the ct landscape: Certificate transparency logs in practice. In Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann, editors, *Passive and Active Measurement*, pages 87–99, Cham, 2017. Springer International Publishing.

[22] Shuang Hao, Nick Feamster, and Ramakant Pandrangi. Monitoring the initial dns behavior of malicious domains. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, page 269–278, New York, NY, USA, 2011. Association for Computing Machinery.

[23] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 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, page 1568–1579, New York, NY, USA, 2016. Association for Computing Machinery.

[24] Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. Understanding the domain registration behavior of spammers. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, page 63–76, New York, NY, USA, 2013. Association for Computing Machinery.

[25] Scott Hollenbeck and Andy Newton. JSON Responses for the Registration Data Access Protocol (RDAP). RFC 9083, June 2021.

[26] Scott Hollenbeck and Andy Newton. Registration Data Access Protocol (RDAP) Query Format. RFC 9082, June 2021.

[27] ICANN. Centralized Zone Data Service. https://czds.icann.org/help. [Accessed 4-2-2024].

[28] ICANN. Internet Corporation for Assigned Names and Numbers. https://www.icann.org/. [Accessed 4-2-2024].

[29] IP2Location. IP2Location. https://www.ip2location.com/. [Accessed 24-5-2024].

[30] IP2Location. IP2Location DB23. https://www.ip2location.com/database/ip2location. [Accessed 9-6-2024].

[31] Aminollah Khormali, Jeman Park, Hisham Alasmary, Afsah Anwar, Muhammad Saad, and David Mohaisen. Domain name system security and privacy: A contemporary survey. *Computer Networks*, 185:107699, 2021.

[32] ko zu. publicsuffixlist python library. https://pypi.org/project/publicsuffixlist/. [Accessed 27-5-2024].

[33] Maciej Korczynski, Maarten Wullink, Samaneh Tajalizadehkhoob, Giovane C. M. Moura, Arman Noroozian, Drew Bagley, and Cristian Hesselman. Cybercrime after the sunrise: A statistical analysis of dns abuse in new gtlds. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, page 609–623, New York, NY, USA, 2018. Association for Computing Machinery.

[34] Ben Laurie. Certificate transparency. *Commun. ACM*, 57(10):40–46, sep 2014.

[35] Ben Laurie, Adam Langley, Emilia Kasper, Eran Messeri, and Rob Stradling. Certificate Transparency Version 2.0. RFC 9162, December 2021.

[36] Bingyu Li, Fengjun Li, Ziqiang Ma, and Qianhong Wu. Exploring the security of certificate transparency in the wild. In Jianying Zhou, Mauro Conti, Chuadhry Mujeeb Ahmed, Man Ho Au, Lejla Batina, Zhou Li, Jingqiang Lin, Eleonora Losiouk, Bo Luo, Suryadipta Majumdar, Weizhi Meng, Martín Ochoa, Stjepan Picek, Georgios Portokalidis, Cong Wang, and Kehuan Zhang, editors, *Applied Cryptography and Network Security Workshops*, pages 453–470, Cham, 2020. Springer International Publishing.

[37] Baojun Liu, Chaoyi Lu, Zhou Li, Ying Liu, Haixin Duan, Shuang Hao, and Zaifeng Zhang. A reexamination of internationalized domain names: The good, the bad and the ugly. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 654–665, 2018.

[38] Cricket Liu. Actively boosting network security with passive dns. *Network Security*, 2016(5):18–20, 2016.

[39] Siôn Lloyd, Carlos Hernandez-Gañan, and Samaneh Tajalizadehkhoob. Towards more rigorous domain-based metrics: quantifying the prevalence and implications of "active" domains. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 539–545. IEEE, 2023.

[40] Sourena Maroofi, Maciej Korczyński, Cristian Hesselman, Benoît Ampeau, and Andrzej Duda. Comar: Classification of compromised versus maliciously registered domains. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 607–623, 2020.

[41] Mohamed Nabeel, Issa M. Khalil, Bei Guan, and Ting Yu. Following passive dns traces to detect stealthy malicious domains via graph inference. *ACM Trans. Priv. Secur.*, 23(4), jul 2020.

[42] Dr. Masataka Ohta. Incremental Zone Transfer in DNS. RFC 1995, August 1996.

[43] OpenINTEL. OpenINTEL Website. https://openintel.nl/. [Accessed 4-2-2024].

[44] OpenPhish. OpenPhish - Pishing Intelligence. https://openphish.com/. [Accessed 27-5-2024].

[45] Jeman Park, Jinchun Choi, Daehun Nyang, and Aziz Mohaisen. Transparency in the new gtld era: Evaluating the dns centralized zone data service. *IEEE Transactions on Network and Service Management*, 16(4):1782–1796, 2019.

[46] PhishTank. PhishTank.com - Join the fight against phishing. https://www.phishtank.com. [Accessed 27-5-2024].

[47] Yong Shi, Gong Chen, and Juntao Li. Malicious domain name detection based on extreme machine learning. *Neural Process. Lett.*, 48(3):1347–1357, dec 2018.

[48] Marcos Rogério Silveira, Leandro Marcos da Silva, Adriano Mauro Cansian, and Hugo Koji Kobayashi. Detection of newly registered malicious domains through passive dns. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3360–3369, 2021.

[49] Raffaele Sommese, Gautam Akiwate, Antonia Affinito, Moritz Muller, Mattijs Jonker, and KC Claffy. Darkdns: Revisiting the value of rapid zone update. *arXiv preprint arXiv:2405.12010*, 2024.

[50] Spamhaus. The Spamhaus Project. https://www.spamhaus.org/. [Accessed 4-2-2024].

[51] Anna Sperotto, Olivier van der Toorn, and Roland van Rijswijk-Deij. Tide: Threat identification using active dns measurements. In *Proceedings of the SIGCOMM Posters and Demos*, SIGCOMM Posters and Demos '17, page 65–67, New York, NY, USA, 2017. Association for Computing Machinery.

[52] Keiichiro Tashima, Hirohisa Aman, Sousuke Amasaki, Tomoyuki Yokogawa, and Minoru Kawahara. Fault-prone java method analysis focusing on pair of local variables with confusing names. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 154–158, 2018.

[53] Domain Tools. DNSDB API Technical Datasheet. https://www.domaintools.com/resources/user-guides/dnsdb-api-technical-datasheet/. [Accessed 14-01-2024].

[54] Sadegh Torabi, Amine Boukhtouta, Chadi Assi, and Mourad Debbabi. Detecting internet abuse by analyzing passive dns traffic: A survey of implemented systems. *IEEE Communications Surveys & Tutorials*, 20(4):3389–3415, 2018.

[55] CyberCrime tracker. CyberCrime-Tracker. https://cybercrime-tracker.net/. [Accessed 27-5-2024].

[56] URLhaus. URLhaus — Malware URL exchange. https://urlhaus.abuse.ch/. [Accessed 27-5-2024].

[57] Nicole van der Meulen. Diginotar: Dissecting the first dutch digital disaster. *Journal of Strategic Security*, 6(2):46–58, 2013.

[58] Verisign. Verisign Website. https://www.verisign.com/. [Accessed 4-2-2024].

[59] VirusTotal. VirusTotal API. https://docs.virustotal.com/reference/overview. [Accessed 29-6-2024].

[60] Maoli Wang, Xiaodong Zang, Jianbo Cao, Bowen Zhang, and Shengbao Li. Phishhunter: Detecting camouflaged idn-based phishing attacks via siamese neural network. *Computers & Security*, 138:103668, 2024.

[61] Florian Weimer. Passive dns replication. In *FIRST conference on computer security incident*, volume 98, pages 1–14, 2005.

[62] Ban Xiaofang, Chen Li, Hu Weihua, and Wu Qu. Malware variant detection using similarity search over content fingerprint. In *The 26th Chinese Control and Decision Conference (2014 CCDC)*, pages 5334–5339, 2014.

[63] Bin Yu, Les Smith, and Mark Threefoot. Semi-supervised time series modeling for real-time flux domain detection on passive dns traffic. pages 258–271, 07 2014.

[64] G. Zhao, K. Xu, L. Xu, and B. Wu. Detecting apt malware infections based on malicious dns and traffic analysis. *IEEE Access*, 3:1132–1142, 2015.

[65] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. A survey on malicious domains detection through dns data analysis. *ACM Comput. Surv.*, 51(4), jul 2018.

A. Ethics

Our research makes use of active measurements. This means that by performing this research, extra internet traffic is created. This may require extra resources from the servers that are actively measured. During our research we kept this into account. We limit the amount of requests to ten measurements per domain or FQDN and only measure within a period of 48 hours after registration. Depending on the amount of FQDNs and domains linked to the same server this can result in our system sending more requests to the same server. To investigate how the domains and FQDNs change over time it is required that we perform these measurements. The measurements consist of a get request. We do not spider the webpages but only visit the main page resulting in limited traffic created by our crawler. The server conducting the measurements hosts a webpage explaining the purpose of the research and how to request removal from the measurements if there are objections. If removal is requested, we will ensure that these targets are excluded from any future measurements. The system makes use of machine learning to flag domains as malicious. We have seen that our system can introduce false positives. If used in practice, some domains may be flagged as malicious even though they are not. In case of implementation, we suggest having an appeal system where requests can be sent for flagged domains to be removed if misclassified. Data made public by this research can contain personal information such as names of registrants. This information is already publicly available. Thus, our research does not pose an extra risk by making this information publicly available.

B. Data Availability

The newly registered domain stream used for this research has already been made publicly available in previous research [49]. Similarly, the newly observed FQDN stream is also available from the same source. These streams are needed to reproduce our research. We will not

TABLE 17. Prefixed domains

Measure Date	Domain
2024-04-29 12:08:47.011	bxbx9.vip
2024-04-29 12:08:47.023	bxbx1.vip
2024-04-29 12:08:47.027	bxbx10.vip
2024-04-29 12:08:47.042	bxbx5.vip
2024-04-29 12:08:47.050	bxbx4.vip
2024-04-29 12:08:47.091	bxbx7.vip
2024-04-29 12:08:47.091	bxbx8.vip
2024-04-29 12:08:47.121	bxbx3.vip
2024-04-29 12:08:47.307	bxbx6.vip
2024-04-29 12:08:47.426	bxbx2.vip
2024-04-29 12:27:11.802	htmdc.vip
2024-04-29 12:27:11.836	htxfo.vip
2024-04-29 12:27:12.361	ht2o3.vip
2024-04-29 12:27:14.503	hto4v.vip
2024-04-29 12:27:22.880	ht6rg.vip
2024-04-29 12:27:45.272	htg0v.vip
2024-04-29 12:27:45.811	htw4m.vip

make available the code of the crawlers we used as these are directly built up on our measurement environment, including spark and S3 storage configurations. We will make available the final model of our best-performing setup so our scores can be verified.

C. Identified Bulk Registration Patterns

In this appendix, we display domain names that are registered in close proximity to each other, indicating bulk registrations. We have identified three patterns. The first pattern consists of prefixed domains, which are presented in Table 17. The second pattern involves fully numeric domains, as presented in Table 18. It is important to note that the numeric domains often appear to have high randomness, but upon examining the registration times, we often find that they are registered in groups within a short period, often within seconds. The last group we identified is targeted domains. These domains target a specific company or service and often closely resemble existing company names. The results are presented in Table 19.

D. Extra Feature Importance Figures

In this appendix, the top 20 most important registrars and ISPs are listed. The top 20 registrars can be found in Figure 12, representing 80.6% of the total registrar importance. Similarly, the top 20 ISPs are shown in Figure 13, accounting for 64.9% of the total ISP importance.

TABLE 18. Numeric domains

Measure Date	Domain
2024-04-29 12:04:39.831	62404.ooo
2024-04-29 12:04:40.614	42530.ooo
2024-04-29 12:04:50.643	49883.ooo
2024-04-29 12:04:51.560	88618.ooo
2024-04-29 12:04:52.623	26833.ooo
2024-04-29 12:05:07.601	74193.link
2024-04-29 12:05:16.930	57041.link
2024-04-29 12:05:17.258	47164.link
2024-04-29 12:05:17.261	48142.link
2024-04-29 12:05:32.817	88862.link
2024-04-29 12:06:37.005	965574.com
2024-04-29 12:06:38.214	845731.com
2024-04-29 12:06:55.895	884020.com
2024-04-29 12:06:55.896	263292.com
2024-04-29 12:07:06.894	509648.com
2024-04-29 12:07:50.896	719228.com
2024-04-29 12:08:01.897	991691.com
2024-04-29 12:18:06.627	131961.net
2024-04-29 12:18:38.061	309497.net
2024-04-29 12:18:50.791	67123.ooo
2024-04-29 12:19:01.170	517813.net
2024-04-29 12:19:11.241	389298.net
2024-04-29 12:19:54.948	555739.net
2024-04-29 12:20:06.086	496621.com
2024-04-29 12:20:16.902	343396.net
2024-04-29 12:20:16.994	857107.net
2024-04-29 12:20:17.041	268146.net
2024-04-29 12:20:27.988	839214.net
2024-04-29 12:20:27.989	541591.net
2024-04-29 12:20:39.004	957774.net
2024-04-29 12:21:01.186	989519.net
2024-04-29 12:21:12.955	376217.net

TABLE 19. Targeted domains

Measure Date	Domain
2024-04-29 12:13:25.392	uspsooe.top
2024-04-29 12:14:09.779	uspsooe.shop
2024-04-29 12:17:43.026	post-royal.shop
2024-04-29 12:17:43.050	post-royal.site
2024-04-29 12:17:43.058	post-royal.icu
2024-04-29 12:17:43.063	post-royal.cloud
2024-04-29 12:17:43.068	post-royal.store
2024-04-29 12:26:36.899	office-poste.top
2024-04-29 12:26:59.838	uspostoinb.top
2024-04-29 12:27:00.469	uspostoinbq.top
2024-04-29 12:27:22.174	royalmaireceivingmilacountsurl.top
2024-04-29 12:28:05.208	ptt-post.mom
2024-04-29 12:28:39.091	royalmaireceivingmilacountsusps.top
2024-04-29 12:28:50.890	ceskaposta-cze.top

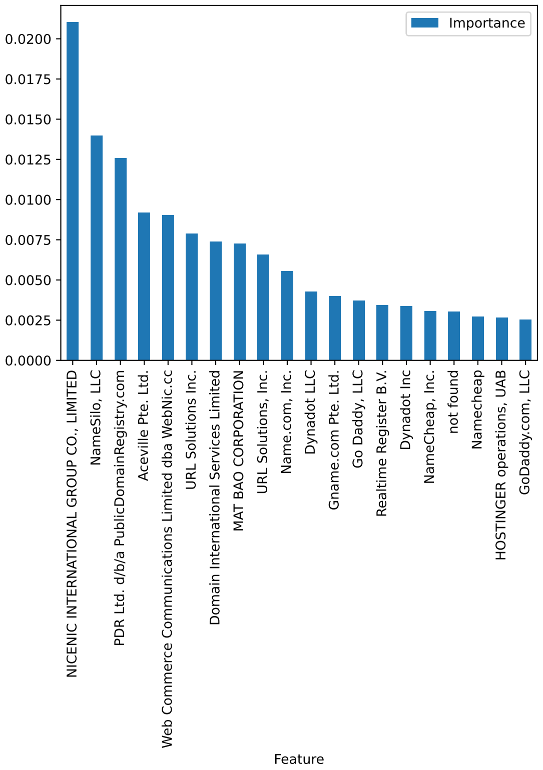


Figure 12. Top 20 most significant registrars

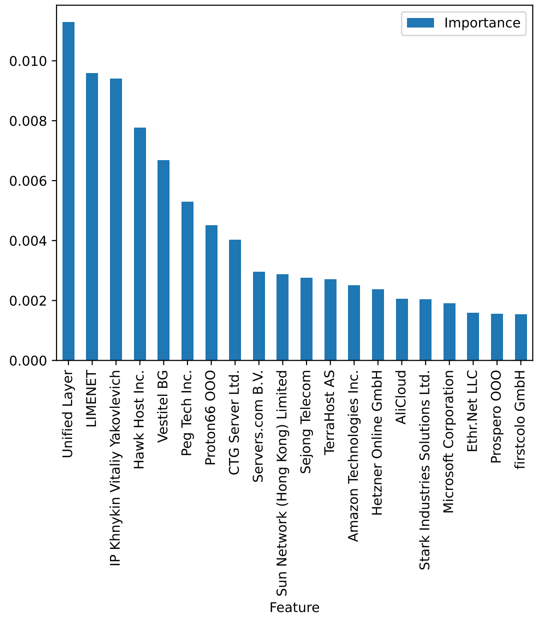


Figure 13. Top 20 most significant ISPs