COMAR: Classification of Compromised versus Maliciously Registered Domains

Sourena Maroofi*, Maciej Korczyński*, Cristian Hesselman‡, Benoît Ampeau§, Andrzej Duda*
*Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG ‡SIDN Labs §AFNIC Labs

***Abstract*—Miscreants abuse thousands of domain names every day by launching large-scale attacks such as phishing or malware campaigns. While some domains are solely registered for malicious purposes, others are benign but get compromised and misused to serve malicious content. Existing methods for their detection can either predict malicious domains at the time of registration or identify indicators of an ongoing malicious activity conflating maliciously registered and compromised domains into common blacklists. Since the mitigation actions for these two types domains are different, we propose COMAR, an approach to differentiate between compromised and maliciously registered domains, complementary to previously proposed domain reputation systems. We start the paper with a thorough analysis of the domain life cycle to determine the relationship between each step and define its associated features. COMAR uses a set of 38 features costly to evade. We evaluate COMAR using phishing and malware blacklists and show that it can achieve high accuracy (97% accuracy with a 2.5% false-positive rate) *without* using any privileged or non-publicly available data, which makes it suitable for the use by any organization. We plan to deploy COMAR at two domain registry operators of the European country-code TLDs and set up an early notification system to facilitate the remediation of blacklisted domains.**

***Index Terms*—DNS, domain name abuse, phishing, malware, malicious domain registration, compromised domain names**

1. Introduction

Domain names play an important role in almost all types of cybercriminal activities. Miscreants tent to use domains in various attack scenarios such as phishing (e.g., to collect sensitive information) or spam campaigns, or as part of command and control (C&C) services with algorithmically generated domain names (AGDs). In all these cases, the involved domains are either solely registered for malicious purposes (which we refer to as *malicious* for simplicity) or registered for legitimate reasons but have been compromised at some time to serve malicious content (we refer to these domains as *compromised*).

One common way to fight malicious activities is to build domain blacklists so a security system can check whether a domain exists on the blacklist and decide on how to treat the incoming traffic related to that domain [1]. However, this method is effective when the blacklist only contains the malicious domains because if it includes the compromised ones, the legitimate services associated with the domains may be interrupted and cause financial loss.

At the time of registration, each domain has two possible states: either it is registered for a malicious purpose or a legitimate one. Then, when the domain is active, there are three possible states: (1) *Benign*: the incoming traffic from the domain is benign and can be passed to users safely, (2) *Malicious*: the traffic related to the domain should be considered as malicious and treated differently (e.g., blocked), and (3) *Compromised*: an attacker leverages an arbitrary vulnerability to upload malicious content, e.g., a phishing page. In this way, while the legitimate website is likely to continue serving benign content to its customers, the attacker benefits from the good reputation of the website to conduct her phishing attack. Therefore, the traffic related to the domain can be either malicious or benign.

The existing domain name reputation systems only consider the first two states. They can detect malicious domains either at registration (e.g., PREDATOR [2]) or after they exhibit malicious behavior (e.g., EXPOSURE [3]). However, none of them can detect compromised domains due to two major problems: (1) there is no such state as *compromised* at the registration time, and (2) compromised domains may exhibit the same behavior as malicious domains while they are benign and abused to serve malicious content. In this regard, domain reputation systems may detect a compromised domain as malicious and blacklist it [1]. While this method successfully prohibits malicious traffic, it also blocks the traffic to the legitimate part of the compromised domain. If such a system identifies a compromised domain as benign, it helps attackers achieve their goals. Therefore, in both cases, the decision on the state of the domain may cause collateral damage. For this reason, a complementary system is required to work along with domain reputation systems to differentiate the compromised domains from the malicious ones.

Apart from creating effective domain blacklists, distinguishing compromised from malicious domains is also important for intermediaries involved in the domain name registration and deployment process. When confronted with a malicious URL, it is critical to assess the registrant's intention for registering the underlying domain since the mitigation action could be different if the registration is for malicious purposes or not. Regarding Top-Level Domain (TLD) registries, one appropriate action for malicious domains is domain delisting, i.e., removing the name from the zone file and changing its status to *hold* to completely deactivate it [4]. Another appropriate action is to block access to the domain (*domain blocking*) or redirect the traffic of the domain to another server under the control of authorized entities (also known as domain *sinkholing*), which can be done by registrars. The latter is a popular and widely used technique to identify the victims infected by malware and to reduce its spread [5].

Taking appropriate action against blacklisted domains

is also important for hosting providers since hosting malicious content can adversely affect their reputation [6]–[9]. Canali et al. studied the reaction of hosting providers when confronted with compromised websites [10]. They showed that in more than 50% of the cases, the reaction of the hosting providers was to suspend (or terminate) users' accounts. For large providers, which may receive hundreds of abuse notifications every day, it is not feasible to manually investigate each case. Therefore, there is a need for a system that can help hosting providers identify the compromised domains and differentiate them from the malicious ones for taking appropriate actions.

Distinguishing between malicious and compromised domains may also lead to revealing the profit-maximizing behavior of attackers. For example, there has been anecdotal evidence indicating that miscreants choose to abuse registrars that offer low domain registration prices [11]–[14]. However, no study has systematically proved this conjecture mainly because the existing URL blacklists conflate compromised and malicious domains. One attacker may indeed prefer lower registration prices but, others may choose to abuse a registrar that offers specific payment methods or a free API allowing for domain registration in bulk. On the other hand, registrars might offer cheap domains but, to prevent domain abuse, perform additional checks to confirm the identity of registrants.

In this paper, we propose **COMAR** (Classification of COmpromised versus MAliciously Registered Domains), a system capable of differentiating compromised (and misused) domains from the malicious ones to 1) create more effective domain blacklists, 2) help registries, registrars, and hosting providers to take appropriate mitigation actions depending on the abuse type, and 3) gain better insights into the attackers' behavior for choosing candidate domains to hack and intermediaries to abuse.

We thoroughly study the domain life cycle to understand the intentions of both miscreants and benign users and determine the relationship between each step and its associated features. We use OpenPhish [15], PhishTank [16], Anti-Phishing Working Group (APWG) [17], and URLhaus [18] as our initial URL blacklist resources, but the system is not only limited to phishing or malware feeds. Our results illustrate that COMAR achieves high classification accuracy by leveraging only *publicly available* data without relying on any privileged resources like historical WHOIS or passive DNS traffic. We also show how it is possible to compensate for the lack of domain creation time if there is no access to WHOIS information.

In summary, we make the following contributions:

- We develop a system to classify domains exhibiting malicious behavior as either compromised or maliciously registered by *only* using publicly available and readily accessible resources, and achieve 97% accuracy with 2.5% of false positives.
- We leverage 38 features to identify the state of a domain, 14 of which are new and have not been used in previous work.
- We introduce a new method to estimate the domain creation time in cases there is no access to WHOIS information, which outperforms standard statistical methods in filling missing values.
- We show that content-based features are the most important ones in representing the domain status.

2. Domain Life Cycle

To understand better the intentions of both malicious actors and benign users for registering and maintaining a domain name, we need to thoroughly inspect the domain life cycle and determine the relationship between each step and its associated features. In this way, we can capture the characteristics of the benign but compromised and malicious domain registration. We divide the domain life cycle into four phases as follows:

L1. Choosing the domain name. In this phase, both miscreants and benign users try to register an appropriate domain name based on their needs. Benign users tend to choose easier to remember, meaningful domain names related to the service provided by the domain. Malicious actors with the purpose of a phishing attack in mind, may try to choose a deceptive name to lure benign users and steal their personal information (e.g., facebook-account.support). In the case of malware C&C panels, miscreants may choose the names that can be generated by the malware family as part of a domain generation algorithm (DGA). These domain names are likely to be long and meaningless (to increase the chance of availability). We expect spammers to use domains that contain keywords of the targeted service to effectively persuade users to click on the link to increase the click rates and search engine ranks (e.g., earn-bitcoin.biz). The knowledge we gain from this phase can help us to build appropriate lexical features related to the characteristics of the domain name.

L2. Registration of the domain name. A user (registrant) registers a domain either through a registrar or a reseller by paying the registration fee. The registration period can be between one to ten years depending on the registrars and registries (although shorter registration periods also exist [19]). In this phase, malicious actors tend to choose less expensive (or free) TLDs to maximize their profit [11]–[14]. The name of the registrar, domain creation, and expiration dates are stored by registrars and registries as part of WHOIS information. The registrant's information, i.e., the registrant name, address, phone number, are often obscured and not publicly available due to the European General Data Protection Regulation (GDPR) [20]. The COMAR system uses the public part of the WHOIS data as well as TLD-related registration features such as retail domain pricing to discriminate between malicious and compromised domains.

L3. DNS record configuration. After the domain name registration phase, DNS records should be set up to allow the discovery of the services associated with the domain. Each resource record provides information about the service behind the domain name. For example, the DNS 'A' record gives the IP address of the server providing the content for that domain (sometimes, the 'A' record points to a reverse proxy server, responsible for fetching the content from the backend server and delivering it to end-users). The 'MX' records point to mail servers whereas 'DMARC' and SPF 'TXT' records are for giving the email domain owners the ability to protect their domain from unauthorized use. Passive DNS datasets (e.g., Farsight Security [21]) come from monitoring DNS responses and extracting DNS information. For legitimate domains, we expect more stability and *availability* of DNS records

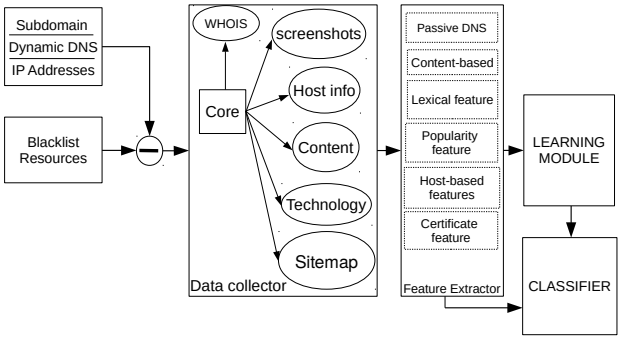


Figure 1: COMAR system structure.

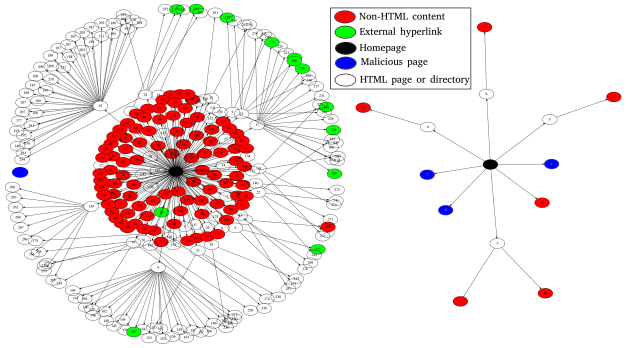


Figure 2: Website structure of a compromised (left) and a malicious domain (right) with the depth level of 3.

while for malicious domains, we expect frequent changes or unavailability of some records (e.g., 'TXT', 'MX', or 'DMARC'). COMAR uses the monitoring approach of passive DNS to construct a feature set, but it *does not* rely on passive DNS datasets since they are not always publicly available. We also use active DNS features by querying blacklisted domains.

L4. Service deployment process and its activity period. This step consists of all the activities to set up the necessary infrastructure for the (legitimate or malicious) service offered by the domain. The activities may include setting up a web server, deploying the application to manage the web content, or ordering a Transport Layer Security (TLS) [22] certificate for the domain name to build trust of the service visitors. We expect that legitimate domain owners put the effort in creating content to increase user interest and therefore, the website popularity, i.e., the amount of web traffic the site receives. Miscreants may or may not take the effort of setting up real websites depending on the type of abuse. We also expect to observe more (vulnerable) libraries and technologies to build a legitimate website, which is not required for the correct operation of malicious domains. In this phase, COMAR collects data mainly through a crawl of blacklisted domains and extracts host-based, popularity, and its most important content-based features.

3. Methodology

Our system comprises three main modules: 1) data collector, 2) feature extractor, and 3) learning and classification modules. Figure 1 presents its structure.

The data collector module gathers data related to the domains derived from URL blacklists. The feature extractor module derives features from the collected data. It can be further used to support efforts of manual labeling domains as maliciously registered or compromised. The learning module takes the labeled data on an input to build a classifier using an appropriate supervised learning technique. Finally, the classification module uses the extracted features and the generated model to classify unlabelled domains derived from URL blacklists in real-time.

3.1. Data Collector Module

We use OpenPhish, PhishTank, APWG, and URLhaus as our initial blacklist resources, but the system is not limited to these URL feeds and can use other types of blacklists on input.

The system downloads URL blacklists every 5 minutes to one hour (depending on the blacklist) to get the newly blacklisted URLs. Some URLs are already not operational by the time they are downloaded (domains are taken down or websites are suspended). Some URLs do not contain domain names and use IPv4 addresses instead, whereas some of them use free subdomain services or dynamic DNS services. We use the private part of the public suffix list [23] to exclude dynamic DNS and free subdomain services from further analysis. For each remaining newly appeared URL in the blacklist, we collect the following information:

Technology information. We define technology information as frameworks and libraries used to build websites (both client-side libraries like JQuery and WordPress, and server-side technologies like PHP or ASP programming languages). To extract such data, we use the Wappalyzer [24] signature list. For each signature in the list, we search in (a) the URL, (b) HTTP headers, and (c) page content to extract all the libraries and tools used to build the website.

Page content. For each domain name, we download the corresponding homepage for further analysis and extracting features. To catch the real content of the domains, which are behind the reverse proxy service (e.g., Cloudflare) with the anti-DDoS feature enabled, we emulate the behavior of a real browser to solve the JavaScript anti-DDoS challenge [25] by using a headless version of the Firefox and Selenium browsers.

Sitemap structure. We further extract all the hyperlinks on the homepage and generate the tree structure of the domain name. For professionally designed websites, the sitemap is often stored in the root directory of the website. However, most of the compromised websites are not well designed, whereas malicious domains rarely have a sitemap file (even if they do, they are not trustworthy). Therefore, we develop our crawler to generate a sitemap for domains. For example, Figure 2 shows the website structure of two sample domains with 3 levels of depth for compromised (left) and malicious (right) domains. The black node in the center of the image is the homepage of the domain name. The green nodes are the links to external domain names, the red nodes are the links to non-HTML data types such as PDF or ZIP files, whereas the white nodes are either HTML pages (leaf) or directories (non-leaf). The blue node shows the malicious page. Having this graph, we can extract various information about the website. For example, the number of internal

TABLE 1: Features and their characteristics. Feature types are binary (B) or continuous (C). The availability column shows the availability of features as highly available (high), medium, or low. The source column shows the features defined by us (*new*) or appeared in previous work.

Feature#	Type	Availability	Source	Feature#	Type	Availability	Source	Feature#	Type	Availability	Source
(1) - (3)	B	High	[26], [27], [11]	(12),(13)	C	High	new	(28)	C	Medium	[32]
(4)	B	High	new	(14)-(16)	B	Medium	new	(29)-(31)	B	High	[33]
(5)	C	High	[28]	(17)	B	Medium	[29]	(32)	B	High	[33]
(6)	C	High	[3]	(18)	C	Medium	new	(33)	C	High	[34]
(7)	C	High	[29]	(19)	C	Medium	new	(34)	B	High	new
(8)	B	High	[30]	(20)-(24)	C	High	[31]	(35)	C	Low	new
(9)	B	High	new	(25)	B	High	new	(36)	B	Low	new
(10)	C	Medium	[7]	(26)	B	Medium	[2]	(37)	B	High	[29]
(11)	B	High	new	(27)	C	Medium	[26]	(38)	B	Medium	[35]

links to pages with different HTML content is higher in the compromised domain compared to the maliciously registered domain because compromised domains have legitimate parts for their users. More importantly, most of the time, there is no connection between the phishing page and the homepage in compromised domains since malicious actors do not tend to change the homepage of the compromised domain. Malicious domains have often a connection between the homepage (if there is one) and the malicious page.

DNS resource records. For each domain, we actively collect the 'A', 'AAAA', 'NS', 'TXT', 'SOA', 'DMARC', and 'MX' resource records. Then, using the Maxmind database [36], we convert the 'A' record to the country code and the autonomous system number (ASN) for further use. We also extract the sender policy framework (SPF) [37] rule from the 'TXT' record if available.

Host information. The host information module is responsible for collecting all the host information related to the input domain (and a possible subdomain) at the time of blacklisting. This information includes the TLS certificates of the domain, the HTTP headers of the web server, the AS number, and its related organization name.

WHOIS data. We collect and parse the WHOIS data, however, we only use the domain creation date in our features. Since this field is not available as part of the WHOIS data for all TLDs, we estimate the missing value based other features (see Section 3.4 for more details).

Screenshots. The lifespan of the blacklisted URLs is short [38]. Therefore, for each domain, we save the screenshots of the homepage as well as the malicious URL (and a subdomain if there exists any) for further manual analysis and labeling of domains in case the website has been taken down by registrars, hosting providers, or miscreants.

3.2. Features

The feature extractor module extracts features from the collected data. It operates along with the data collector in a real-time manner to convert plain data into features. In total, we extract 38 features divided into seven main categories (feature set F_1 through F_7) as presented below:

1) Lexical features (F_1)
2) Ranking system and popularity features (F_2)
3) Passive DNS features (F_3)
4) Content-based features (F_4)
5) WHOIS and TLD-based features (F_5)
6) TLS certificate features (F_6)
7) Active DNS features (F_7)

TABLE 2: Lexical features used in maliciously registered domain names.

Domain name	Attack type	Lexical features
paypala.com	Phishing	(1) (2) (3)
suportaccount-services.com	Phishing	(4) (5)
3lf4vlxegj1luy6kbs.com	AGD (Rovnix)	(6)
erdoypf-inr.net	AGD (Redyms)	(5)
applid.appsgr.girtrusgirs.com	Phishing	(7)

Table 1 shows the characteristics of each feature along with their availability, types (B: binary or C: continuous), and if they appeared in previous work or are defined by us.

3.2.1. Lexical features. They are the features extracted from the registered domain (e.g., *example.com*), the subdomain (e.g., *sub.example.com*) as well as the path part of the URL.

Famous brand name in the domain name ($\mathbf{f_1}$). We have identified 231 brand names mostly targeted by attackers in phishing attacks (e.g., PayPal, Amazon, Yahoo, or Gmail). We have created a list of keywords by manually inspecting phishing pages and the corresponding domain names. If the domain name consists of one of these words, it is an indication of maliciousness.

Misspelled target brand name in the domain name ($\mathbf{f_2}$). We use dnstwister [39] to generate possible similar domain names for each of 231 brand names and compare them with the domain name to check the existence of these words. We also consider internationalized domain names and convert the unicode characters to their look alike ASCII equivalent to cover homograph attacks.

Levenshtein distance of the domain name and targets ($\mathbf{f_3}$). We calculate the Levenshtein distance (LD) between the domain name and every 231 targets on our list. We choose $LD = 1$ as the threshold as proposed by Korczyński et al. [11].

Special words but not brand names in the domain name ($\mathbf{f_4}$). Some specific words (e.g., verification, account, support) are not brand names but, based on our word frequency analysis, miscreants tend to use them as part of the domain name to lure victims to enter their credentials. We split the domain name into a word list using the hyphen character. For each word in the domain name, we look for a complete or partial match of that word and our predefined list of 28 keywords. For example, for the domain name 'supportacc-paypal.com', we have one brand name match (i.e., 'paypal') and one special word match (i.e., 'support').

Number of hyphens in the domain name ($\mathbf{f_5}$). The only special character that can be used in a domain name is hyphen ('-'). Both phishing [28] and algorithmically generated domain names (e.g., Redyms malware [40]) tend to use hyphens as part of the domain name.
Digit ratio ($\mathbf{f_6}$). AGDs tend to have more digits than legitimate domain names [3]. This feature is more suitable for domains generated by malware families.
Level of subdomains ($\mathbf{f_7}$). As miscreants control the DNS records of the malicious domains[1], they can create as many subdomains as necessary for a successful attack [29].
Presence of a brand name in the path part of URL ($\mathbf{f_8}$). For compromised domains for which attackers generally do not have access to the domain zone file to create new subdomains, the only way for the malicious actors to use the target brand name is to include it as a part of URL.
Presence of the dot character in the path part of URL ($\mathbf{f_9}$). By manual analysis of blacklisted URLs, we have observed that some malicious actors tend to use the dot character ('.') before file or directory names, for example: *https://masseffect.co.za/.lilman/login.php?cmd=submit*, which may allow the attacker to deceive an unskilled administrator who may not notice the hidden malicious content on the compromised system.

For features $\mathbf{f_1}$-$\mathbf{f_6}$, we only consider the domain name part of the blacklisted URL. Feature $\mathbf{f_7}$ considers the subdomain, whereas $\mathbf{f_8}$ and $\mathbf{f_9}$ are only related to the path part of the URL. Table 2 shows the use of selected features in various types of maliciously registered domains.

3.2.2. Content-based features. The ultimate goal of domains is to identify a website or a web service that serve content to their customers in various forms. While it is not trivial to examine the content validity, yet it is feasible to extract informative content-based features.
Content length ($\mathbf{f_{10}}$). Malicious domains tend to have less content [7]. For this feature, we only consider the content length of the homepage for each domain part of the blacklisted URL. If there is no index page for that domain (i.e., default directory listing page of the web server), or the web server returns any HTTP code other than the success code (e.g., 404 not found or 403 not authorized), we consider the length to be zero.
Number of used technologies ($\mathbf{f_{11}}$). Using different frameworks and libraries in building a website needs time and effort. The more different technologies used in creating a website, the more time spent on the development. Therefore, we consider the number of used technologies as an indication of the domain being benign. We crawl websites to fingerprint software using unique words and patterns found in the source code. We derive the fingerprints and signatures used in Wappalyzer [43].
Vulnerable technology ($\mathbf{f_{12}}$). It is a binary feature that indicates whether the website uses a technology with at least one known vulnerability. For example, 271 known vulnerabilities have been found in the WordPress content management system (CMS), including themes and plugins that enable the attackers to upload an arbitrary file to the server [44]. Other familiar technologies with known vulnerabilities are *Joomla* or *Drupal* CMSes, and *Magenta*, *PrestaShop*, or *DotNetNuke* frameworks. The intuition is that if a website uses one of these CMSes, frameworks, modules, or libraries, then there is more chance to get compromised. To obtain the list of technologies with at least one reported vulnerability, we use the exploit [45] and vulnerability databases [46].
Number of internal working hyperlinks ($\mathbf{f_{13}}$). The website with some content is not always benign since miscreants may create fake content on the website so that it looks legitimate. The easiest way for malicious actors is to clone the content of a legitimate website. For each internal hyperlink in the homepage, i.e., a page belonging to the same domain, we fetch the content (only HTML content not files) and calculate the fuzzy hash as proposed by Kornblum [47] to make sure all the pages are not the same and then count the number of unique hashes as the number of working internal hyperlinks.
Content-related domain name ($\mathbf{f_{14}}$). This feature defines the relationship between the content of the homepage and the domain name itself. We extract the meaningful words (based on a dictionary) of the domain name and search for those words in the visible text of the homepage. It is a binary feature with the values 1 (at least one match between a word from the domain name and a related word in the textual content) or 0 otherwise. Another approach would be to use the 'Google trends' service but it is paid and difficult to use on a larger scale.
Presence of the index page ($\mathbf{f_{15}}$). It is common for attackers to upload their files to the web server and just use them without appropriate configuration. In this case, if they forget to upload an appropriate index page (e.g., index.html, index.php, or index.asp depending on the server and server-side programming language), the default behavior of the most web servers (e.g., NGINX or Apache) is to list the directory content. One possibility is that the attacker could remove the index page from the compromised domain but it leads to immediate reaction of the webmaster.
Presence of the default index page ($\mathbf{f_{16}}$). The index pages (homepages) of some domains are the default sites deployed by the registrars, hosting providers, or resellers after the domain name registration process is complete. Resellers often offer free software installation plans like WordPress or Joomla CMSes along with hosting plans. The whole process of installing a CMS on the server takes a few minutes. Sometimes, attackers leverage these free plans to make the domain looks more legitimate. For each domain name, we compare the content of the index page with our pre-defined list of default pages from familiar CMSes and default control panels to check whether the home page is a default page or not.
Using page redirection ($\mathbf{f_{17}}$). Homepage redirection and web cloaking [48] are two common methods among attackers to conceal their malicious intention by displaying benign contents to web crawlers and bots. Regarding homepage redirection, when users try to visit the homepage of the malicious domain, they will be redirected to a benign website. In case of phishing, the redirection is mostly to the real website of the phishing target (e.g., the real Bank of America website). In case of malware domains, it can be a random website like google.com. To

1. In some types of attacks like zone poisoning [41] or domain shadowing [42], it is still possible for miscreants to change almost any DNS record of a benign domain and generate arbitrary subdomains.

distinguish between page redirection and web cloaking, we crawl each malicious URL with the Selenium browser and with the Python *requests* library. We set this feature to true if the destination URL of the homepage requested using both the browser emulation and the *requests* library shows the same domain name but it is different from the domain name of the blacklisted URL.

Number of external hyperlinks ($\mathbf{f_{18}}$). This feature works the same way as *internal working hyperlinks* but counts the number of hyperlinks that refer to external domains. Sometimes miscreants, especially in phishing attacks, tend to clone the target website to perform a more successful attack. In these cases, the cloned website often has hyperlinks to the real target. Using this feature, we can capture such fake content.

3.2.3. Passive DNS features. Passive DNS has become an industry-standard tool for more than a decade. It can give us insights into how the behavior of a domain changes over time (e.g., changing the IP address) and how popular the domain name was in the past. Although the features based on passive DNS data proved to be significant, they can be compensated by other features without lowering accuracy. Therefore, the absence of this feature set does not affect the classification results.

First passive DNS query before the blacklist time ($\mathbf{f_{19}}$). The number of days between the first occurrence of a passive DNS query (for 'A' or 'NS' records) and the blacklist time. This feature provides the estimation of the age of the domain in terms of usage and not only with respect to the registration.

Passive DNS queries ($\mathbf{f_{20}}$-$\mathbf{f_{24}}$). The number of queries for each resource record before appearing in the blacklist resources (i.e., 'A', 'AAAA', 'NS', 'MX', 'TXT' records). For example, the higher the number of observed 'MX' queries, the higher the chance that the domain has an active mail service.

3.2.4. Active DNS features. We extract the following features from DNS data queried shortly after the blacklisting time.

Presence of the sender policy framework (SPF) ($\mathbf{f_{25}}$). 'TXT' records are used (among others) for setting SPF rules [37], domain message authentication reporting and conformance (DMARC) rules [49], and in some cases for domain ownership verification by third-party services (like Google App verification). The presence of SPF for a specific domain can be considered as an indication of legitimacy. For example, a domain owner for whom protection against email spoofing is important would set an appropriate SPF rule in the 'TXT' record [50]. Nevertheless, the malicious actors may also set up SPF rules to increase domain reputation.

Self-resolving name server ($\mathbf{f_{26}}$). Miscreants may use self-resolving name servers i.e., name servers responsible for resolving their own domain names (e.g., *ns1.domain.com* for resolving *domain.com*) [2], whereas legitimate users tend to use the default DNS resolvers of their DNS service providers.

3.2.5. WHOIS features. Due to the introduction of GDPR and our requirement that the proposed method should only depend on publicly available data sources, we only derive the domain creation date from WHOIS and propose the following feature:

Domain age ($\mathbf{f_{27}}$). The older the domain name, the higher the chance to be legitimate. However, according to the 2016 APWG report [26], some miscreants age registered domains waiting weeks or sometimes months before using them. In this way, they can gain reputation for the domain and bypass the detection methods that work based on the registration date. However, according to the report, the number of such domains is low because maintaining a domain name for a long time needs extra effort and money, not always possible for attackers. We use the time lapse between the domain registration and the blacklist dates.

3.2.6. Ranking system and popularity features. This feature set consists of 8 features related to search engine results, the Internet Archive [51], and domain name popularity in different ranking systems.

Search engine results ($\mathbf{f_{28}}$). The number of results returned by the Bing search engine for 'site:example.com' queries. The higher the number of results, more popular the domain is. We do not consider Yahoo and Google search engine results because although they are free, with publicly available APIs, the number of requests per day is limited. For example, at the time of writing, the Google custom search engine only allows 100 queries per day. While the Bing search engine is not free (we used the trial version), the price ($3/1000 requests [52]) is low compared to its equivalent alternatives.

Top ranking websites ($\mathbf{f_{29}}$-$\mathbf{f_{32}}$). The presence of the domain name in the Alexa [53], Majestic [54], Quantcast [55], and Umbrella [56] top 1 Million website and domain ranking lists. While we could merge features $\mathbf{f_{29}}$-$\mathbf{f_{32}}$ into a single one based on the Tranco list [33], each of these ranking systems uses its own metrics to calculate domain popularity and therefore, captures different characteristics. We only consider the presence of a domain in such lists as a sign of its popularity.

Wayback Machine ($\mathbf{f_{33}}$). The Internet Archive project started in 1996 by archiving the Internet itself. The sources of the captures come from different plans of the project, e.g., capturing Alexa top domains, domains that have at least one link from different domains that the Wayback Machine already captured at least one time, and several more plans covering the most part of the Internet [51]. We consider the high number of captures as a sign of benignness for domains.

3.2.7. TLD-related features. Chosen TLD is not random among miscreants [11]. They tend to use TLDs based on some factors like the TLD price. We extract two features related to TLD.

TLD maliciousness index ($\mathbf{f_{34}}$). It is a number greater than or equal to zero corresponding to the proportion of abused to all registered domains for each TLD introduced by Spamhaus [57].

TLD price ($\mathbf{f_{35}}$). The price of domains is very important among miscreants since they want to maximize their profit by minimizing the costs. For example, free TLDs (i.e., those provided by Freenom) are among the most common TLDs used in phishing attacks [12], [26].

3.2.8. TLS certificate features. COMAR uses three features related to TLS certificates.

TLS certificate price ($\mathbf{f_{36}}$). The purpose of making a TLS certificate available free of cost was to make access to HTTPS available for all websites [58], which means that miscreants can also benefit from it. By using a TLS certificate, attackers can make their attacks look more legitimate (e.g., by showing the green lock in the address bar of the browsers). Free TLS certificates do not either require their owners to provide any personal information. Therefore, the conjecture is that miscreants would prefer to choose free TLS certificates rather than the paid ones.
Presence of TLS certificate ($\mathbf{f_{37}}$). Although the report published by Phishlab [59] shows that almost half of the phishing websites were hosted on domains with an active TLS certificate, we can still leverage this feature since our analysis is not limited to only phishing attacks.
Valid TLS certificate ($\mathbf{f_{38}}$). Trusted but expired TLS certificates or those issued by untrusted certificate authorities (CAs) trigger an error (or a warning) in most standard browsers (e.g., Chrome or Firefox). This behavior may alert victims about an attack. Therefore, most of the phishing URLs are either HTTP or HTTPS with valid certificates. However, for websites that are used in malware spread or domains for C&C panels, the victims are not humans but infected machines. Having a TLS certificate let the infected machines to communicate with their hosts (e.g., bot masters) securely regardless of the validity of the certificates [35]. For each domain, with a TLS certificate, we define a binary feature indicating whether the certificate is valid or not.

3.3. Further Notes on Features

So far, we have introduced 38 features in 7 categories. There are some aspects to consider regarding these features.

- Not all features are available for domain names but some features rely on the presence of other ones. For example, all the content-related features are available if there is a homepage available for the domain name. As another example, a TLS certificate price solely relies on the existence of a TLS certificate (i.e., the domain should be HTTPS-enabled). Such dependency enables suitable handling of missing values discussed below in Section 3.4.
- For each type of domain abuse, only a specific set of features may be related to that type. For example, lexical features (more specifically, URL-based features) are mostly used in phishing attacks and are not relevant to algorithmically generated domain names. However, we apply all the features in the classifier and let the classifier decide the relevance of each feature. Then, by interpreting the results, we can choose the appropriate feature set for each type of domain abuse.
- Another important aspect of feature engineering is feature evasion, i.e., how robust each feature is against manipulation. In Appendix C, we discuss potential evasion strategies and how difficult they are for attackers to deploy.

3.4. Handling Missing Values

Ideally, the classifier operates on a complete ground-truth dataset without missing values. However, in practice, it is not always possible to collect all the features due to several reasons. Two important considerations regarding missing values are their types as explained by Little and Rubin [60], and the reasons for the absence of data. For example, one important feature in our set is the domain age, which depends on the availability of the registration date. However, it is not always feasible to parse the WHOIS data [61]. Some registries do not provide the registration date as part of WHOIS information or WHOIS data at all (e.g., Freenom registries for .ml, .tk, .ga, or the German registry for .de). Therefore, the lack of the registration date means losing important information, which may result in misclassification. The common strategy to fill missing values is to use statistical methods such as the mean (or median) of the feature. However, the mean and median values may lead to biased results since each sample in the dataset (i.e., each registered domain name) is independent of other samples [62]. Another way to fill missing values is to estimate the best value based on the available evidence. In the case of the registration date, although we cannot find the exact date, we can use the earliest day we observed the domain name in the wild. We use the following formula:

$$\min_{date}\{wayback\_machine, SSL\_certificate, first\_pDNS\} \quad (1)$$

with respect to the following constraints:

- Regarding passive DNS, we consider the first seen 'A' (or 'NS') record that matches the 'A' (or 'NS') record of the domain name before the time it was submitted to one of the blacklist resources. The justification comes from the possibility that a domain name was registered by someone before, then re-registered by another user (miscreant) and misused.
- Regarding TLS certificates, we use Certificate Transparency logs to retrieve all the previous certificates of the domain (and subdomains, if any) and extract the issuance date of the oldest one that matches the certificate of the domain name before the blacklisting time.

In this way, we obtain the earliest date the domain appeared in the 1) Wayback Machine [63], 2) Certificate Transparency [64], [65], and 3) the passive DNS database [21], which ensures that the real domain registration date is earlier than (or equal to) our estimated value.

Figure 3 shows the proportion of the domains for which the difference between the real registration dates and the estimated ones using the proposed method, mean, and median approaches is less than 1 year, between 1 and 2 years, and so on. As the ground-truth data, we use 10,000 domains with different TLDs with known registration dates. For approximately 67% of the domains, the difference is less than one year, while for the mean and median, the result is less than 30%. Furthermore, filling the registration date with the mean for a specific TLD requires to have at least partially the data for that TLD, while for some TLDs (e.g., .ml, .tk), the responsible registries do not provide registration dates at all.

Apart from the registration date, there may be some more missing values in our feature set. For the *'TLD maliciousness index'*, whenever we do not have the data, we fill the value with zero. For *content-related* features, we send requests to domains using the headless version

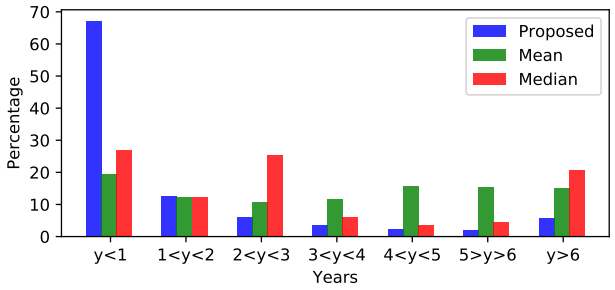


Figure 3: Proportion of the domains vs. the difference between real registration dates and the estimated ones using the proposed method, mean, and median approaches.

of Selinium and Firefox browsers to mimic user-oriented actions. If we do not get any response from the server, we can assume that the domain has no content to offer to visitors. Therefore, we consider *'content length'*, the *'internal hyperlink'*, and *'external hyperlink'* features as zero. However, in some rare cases, it is possible that the attack type is location-based that either serves the content to specific IP addresses or serves different contents to different IP addresses [66]. In this case, due to our limited resources, we may not be able to fetch the real content. Concerning the TLS certificate price, our approach is to create a binary feature, paid vs. free. However, for some certificate authorities, there is no clear cut boundary between these two options. For example, Comodo CA [67] (also known as Sectigo) offers both free and paid TLS certificates for domains. For these CAs, we consider the validity period of the certificate. If the validity period is less than three months, then we consider the certificate as free.

In Section 6, we compare handling of missing values, data availability, and usage limitations of previously proposed methods with COMAR.

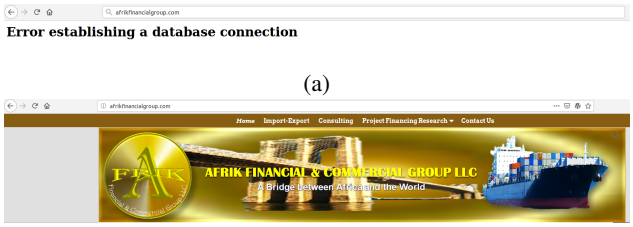
4. Experimental Results

In this section, we provide the details of the phishing and malware ground-truth datasets and describe our method to classify compromised and maliciously registered domains.

4.1. Ground-Truth Datasets

We have collected 41,002 URLs from four blacklists. Figure 8 in Appendix B shows the number of collected URLs for each blacklist and the overlap between them—it is only the number of working (live) URLs at the time of crawling (March to July, 2019), after removing URL shorteners, free subdomain services, and inactive URLs. Then, we have created two ground-truth datasets from the subset of collected URLs with: 1) URLs from phishing blacklists (APWG, PhishTank, and OpenPhish) and 2) URLs from malware distribution blacklist (URLhaus).

We start with labeling URLs by manually visiting the homepage of the domain and investigating its content and functionality. It is not always trivial even for a human to decide if a domain name is compromised or a malicious one. For instance, it is easy to label 'pyplcompte.fr' (without any homepage and one URL to a fake PayPal login



(a)

(b)

Figure 4: (a) The homepage of the 'afrikfinancialgroup.com' captured in the first scan showing a database connection error. (b) The homepage of the same domain name re-visited after 10 days.

page) as malicious while for 'afrikfinancialgroup.com', the domain name does not contain any suspicious word and the registration time is 2017 but looking at the homepage of the domain, there is only a database connection error description (Figure 4a). The error can be the result of an attack or it can be just a simple message to fill the homepage of the maliciously registered domain. To be certain that the chosen label is correct, we re-visit each domain manually after a period of 10 days and check the homepage and the presence of the malicious URL again (the hypothesis is that a 10 day period is long enough for a webmaster to notice that the website is defaced). If the homepage is fixed after 10 days (see Figure 4b), we consider the domain as compromised.

We have manually labeled domains as either 1) maliciously registered, 2) compromised, 3) subdomain/free service, or 4) false positive. Although the data collector module automatically excludes free subdomain services, still some of them, which were not in our predefined subdomains list, appeared in the labeling process. After removing subdomain services and false positives (i.e., URLs mistakenly blacklisted) the final datasets consist of 1,321 domains from phishing blacklists and 1,008 malware domains from URLhaus. The proportion of the phishing dataset is 58% malicious - 42% compromised and for the malware dataset 57% compromised - 43% malicious.

4.2. Classifier

We use two classification methods: 1) Logistic Regression and 2) Random Forest. We apply each method separately on the malware and phishing datasets. We choose the methods because of their characteristics. Logistic regression is a machine learning algorithm that works on linearly separable data and uses the combination of the weighted input features to predict the output class. It is a parametric method known for its efficiency, low computational resources, and interpretability. However, feature engineering plays an important role with respect to its performance. On the other hand, the random forest is a non-parametric machine learning algorithm capable of training a non-linear model based on the input samples. Generally, it does not need any feature transformation or any assumption about the underlying mapping function. With a sufficient number of training samples, it may result in a better performance model compared to logistic regression [68]. As for evaluation metrics, we use accuracy, precision, recall, F1-score, and Matthews

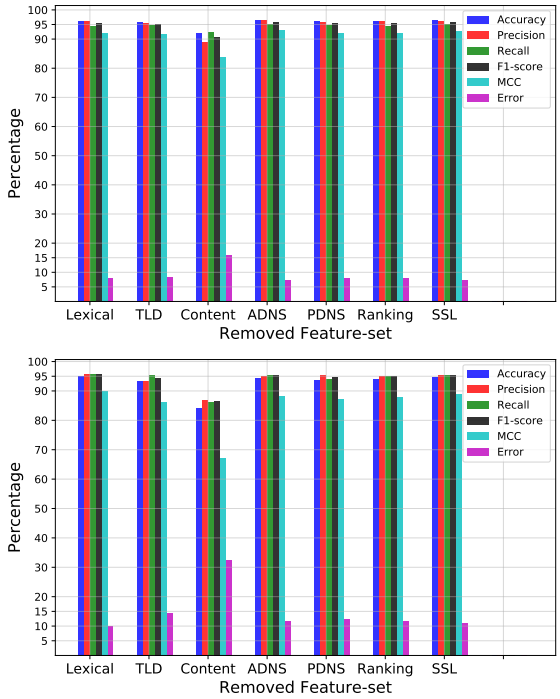


Figure 5: Evaluation metrics of phishing (top) and malware (bottom) datasets using logistic regression.

correlation coefficient (MCC) defined in Appendix A. We use the MCC metric since our datasets are not completely balanced and we also need to consider false positives and false negatives in the final results of the classifier.

Table 3 shows the results of the random forest (RF) and logistic regression (LR) classifiers for phishing and malware datasets. We can notice that the classification results of the random forest are slightly better than logistic regression for both datasets. However, we use logistic regression to describe the data and explain the relationship between input features and output classes since it can produce interpretable coefficients.

Figure 5 shows the classification results by applying logistic regression on the phishing and malware datasets, and eliminating one feature set at a time. We set the maximum number of iterations to 10,000, using 10-fold cross-validation to evaluate the algorithm and ridge regularization to create a less complex model and avoid overfitting. The classification error is the sum of false positives and false negatives. A false positive refers to the malicious domains misclassified as compromised and a false negative refers to the compromised domains misclassified as malicious ones. We can observe that removing content-based features can severely affect the results of the classifier both in phishing and malware datasets, and increase the classification error up to 16% and 30%, respectively. On the other hand, removing the *passive DNS* feature set has almost no effect on the final results (Acc: 96.14%, Precision: 95.78%, Recall: 94.91%, F1: 95.34%, MCC: 0.92 for phishing datasets). Moreover, content-based features are more important for malware samples than phishing. The reason is that most of the maliciously registered domains related to malware spreading or C&C panels have no content in their homepages.

TABLE 3: Evaluation of the Random Forest (RF), Logistic Regression (LR), and the APWG method on phishing and malware datasets.

Method	DB	Acc	Precision	Recall	F1	MCC
RF	Phish	97%	95%	97%	96%	0.93
LR	Phish	96.5%	96.59%	95%	95.7%	0.92
APWG	Phish	85%	82%	93%	88%	0.69
RF	Mal	96%	97%	96%	97%	0.92
LR	Mal	94.5%	95.6%	95.2%	95.4%	0.89

5. Evaluation of the Results

In this section, we first compare our results with the simple approach used in the 2016 APWG phishing survey [26] to distinguish between malicious and compromised domains. Then, we study the features extracted and used in the classification process. We analyze the 'strength' of each feature (i.e., how it is related to each output class) and select those with the highest impact on the classification results. This section provide a better insight into how we can select the features to create a more effective classifier. We also present three case studies that may influence the classification results.

5.1. Comparing COMAR with APWG Method

In the 2016 global APWG phishing survey [26], Aaron and Rasmussen used a simple set of heuristics to distinguish maliciously registered from compromised domains in phishing attacks. They considered a domain to be malicious if it was reported within a very short time after registration and/or contained a brand name and/or was registered in a batch or there existed a pattern indicating common ownership or intent. Since we do not have access to the registrant's information in the WHOIS data to detect batch registration or any pattern of common intent, we use only the first two conditions to evaluate the APWG method on our ground-truth data. The report did not specify the exact meaning of the 'very short time of being registered', so we chose three months as it is used in the previous study [11]. If the domain has appeared in a blacklist in less than three months of its registration time, or if it has a famous brand name/string in its name, we consider it as a malicious one otherwise it is categorized as compromised.

Table 3 shows the classification results of the APWG method. Although the accuracy of the result is relatively high (85%), the false-positive rate is also very high (27%), which results in low MCC (69%). The reason for the high false-positive rate is that the method is unable to detect malicious domains that were registered more than three months before blacklisting and that have no famous brand name or a misleading string as part of the domain name.

In general, there are three limitations of this and other methods that use the registration date as the main feature for classification. As discussed in Section 3.4, the registration date is not always available for all TLDs. Therefore, the evaluation is limited to TLDs with the registration date available as part of the WHOIS data. The second drawback is that identifying patterns or evidence of bulk registrations need registration information such as the registrant's name and the address no longer publicly

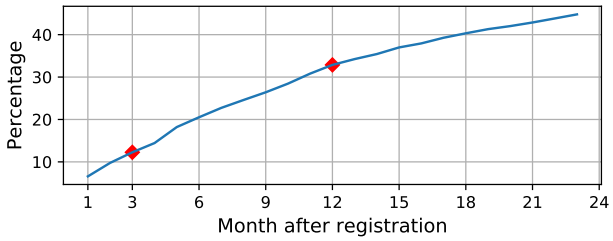


Figure 6: Partial cumulative distribution of the compromised domains after registration.

available [20]. Finally, the third caveat of using this heuristic is the fact that it does not consider legitimate domains compromised in the first few months or even days after registration.

Figure 6 shows the partial cumulative distribution of the compromised domains after their registration date. We collect the data of hacked websites for 18,810 domains from various resources like accounts of the hacker groups on Facebook, Twitter, and hacking forums for 2 months. The results show that 12% of the domains get compromised in the first three months of their registration, and approximately 32% get hacked in the first year after registration probably because of the lack of appropriate configurations or because the website is still under active development. These types of domains may lead to false-negative results (classifying newly registered benign but compromised domains as maliciously registered). However, COMAR does not suffer from these limitations since it does not heavily rely on the registration dates (only one feature out of the other 38 proposed ones), and we estimate the missing values of the domain registration dates for TLDs that do not provide the WHOIS data.

5.2. Feature Analysis

By using logistic regression, we can measure how important individual features are to the overall performance. Table 4 and 5 show the logistic regression weights for 24 most significant features. We use the L2 norm regularization to keep the weights small to avoid overfitting and reduce model complexity. Moreover, small weights help us making sure that one feature with a large value cannot heavily affect the final classifier result. We also use log transformation for some features (e.g., *'number of Bing result'*) to increase the linearity between the input features and the output class. The sign of each coefficient shows the relationship between the feature and the *compromised* output class. For example, the *'TLD_maliciousness_index'* feature has a negative relationship with the *compromised* (and a positive relationship with the *malicious*) output class. Therefore, a higher maliciousness index of TLD indicates a higher probability of a domain being maliciously registered rather than compromised.

We can observe that the *'number of internal hyperlinks'*, *'number of Bing search results'*, *'number of technologies'*, and *'content length'* features are in the top five strongest features indicating compromised domains for both malware and phishing datasets. The '*number of internal hyperlinks*', '*number of technologies*', and '*content length*' features are content-based and capture the effort the owner (legitimate or malicious) put into creating a fully-featured website. The results support the conjecture that attackers spend less time to deploy a fully-functional website with rich content since it is time consuming. Content-based features play an important role in the classification: 5 out of 10 most significant features are content-based. The '*number of Bing search results*' is related to domain popularity, which reflects the conjecture that malicious domains are less popular than compromised domains since they have legitimate traffic generated by benign users.

Another interesting feature is *'number of external hyperlinks'* with different signs for phishing and malware datasets probably because phishers tend to copy the entire HTML code of the target website to create the exact look and feel experience, and most of the time, the cloned

TABLE 4: Logistic regression coefficients of the significant features for the phishing dataset.

#	Feature	Category	Weights
1	$f_{\text{number of internal hyperlink}}$	Content-based	1.88
2	$f_{\text{number of technology used}}$	Content-based	1.28
3	$f_{\text{Bing search result}}$	Ranking	1.26
4	$f_{\text{content length}}$	Content-based	0.98
5	$f_{\text{first PDNS before blacklist}}$	Passive DNS	0.78
6	$f_{\text{number of PDNS MX}}$	Passive DNS	0.56
7	$f_{\text{TLD maliciousness index}}$	TLD-based	-0.56
8	$f_{\text{domain aging}}$	WHOIS-based	0.49
9	$f_{\text{using redirection}}$	Content-based	-0.46
10	$f_{\text{has vulnerable tech}}$	Content-based	0.41
11	$f_{\text{presence of index page}}$	Content-based	0.39
12	$f_{\text{wayback machine captured}}$	Ranking	0.30
13	$f_{\text{URL has famous brand name}}$	Lexical	0.28
14	$f_{\text{is content related}}$	Content-based	0.21
15	$f_{\text{special word in domain name}}$	Lexical	-0.18
16	$f_{\text{number of external hyperlink}}$	Content-based	-0.17
17	$f_{\text{using HTTPS}}$	SSL-based	0.15
18	$f_{\text{using brand name in domain name}}$	Lexical	-0.12
19	$f_{\text{presence of default homepage}}$	Content-based	-0.10
20	$f_{\text{has SPF}}$	Active DNS	-0.07
21	$f_{\text{self-resolve NS}}$	Active DNS	-0.05
22	$f_{\text{presence in quantcast}}$	Ranking	0.03
23	$f_{\text{using misspelled brand name}}$	Lexical	0.03
24	$f_{\text{presence in umbrella}}$	Ranking	0.02

TABLE 5: Logistic regression coefficients of the significant features for the malware dataset.

#	Feature	Category	Weights
1	$f_{\text{number of technology used}}$	Content-based	0.87
2	$f_{\text{number of internal hyperlink}}$	Content-based	0.84
3	$f_{\text{content length}}$	Content-based	0.82
4	$f_{\text{Bing search result}}$	Ranking	0.74
5	$f_{\text{TLD maliciousness index}}$	TLD-based	-0.72
6	$f_{\text{number of PDNS MX}}$	Passive DNS	0.50
7	$f_{\text{wayback machine captured}}$	Ranking	0.50
8	$f_{\text{presence of index page}}$	Content-based	0.19
9	$f_{\text{number of external hyperlink}}$	Content-based	0.18
10	$f_{\text{domain aging}}$	WHOIS-based	0.16
11	$f_{\text{has vulnerable tech}}$	Content-based	0.14
12	$f_{\text{presence of default homepage}}$	Content-based	-0.14
13	$f_{\text{self-resolve NS}}$	Active DNS	-0.13
14	$f_{\text{is content related}}$	Content-based	0.11
15	$f_{\text{presence in umbrella}}$	Ranking	0.05
16	$f_{\text{using HTTPS}}$	SSL-based	0.05
17	$f_{\text{first PDNS before blacklist}}$	Passive DNS	0.04
18	$f_{\text{using redirection}}$	Content-based	0.04
19	$f_{\text{URL has famous brand name}}$	Lexical	0.02
20	$f_{\text{using brand name in domain name}}$	Lexical	0.01
21	$f_{\text{presence in quantcast}}$	Ranking	0.01
22	$f_{\text{using misspelled brand name}}$	Lexical	0.01
23	$f_{\text{special word in domain name}}$	Lexical	0.0
24	$f_{\text{has SPF}}$	Active DNS	0.0

HTML code contains hyperlinks related to different pages of the target website. On the other hand, malware domains (e.g., algorithmically generated) often have less (or no) content, which leads to less (or no) external hyperlinks. Therefore, in this case, having an external hyperlink is the indication of a compromised domain.

For URL-based features (e.g., *'URL has famous brand name'*), we observe a significant decrease from phishing dataset to malware dataset because URL-based features are mostly related to phishing attacks. For example, the weight of *'URL has famous brand name'* is 0.28 for phishing while it is 0.02 for the malware dataset.

Considering ranking and popularity features, the presence of the domain name in four ranking websites (i.e., Alexa, Quantcast, Majestic, and Umbrella) has a weak association with the output class in favor of compromised domains in both datasets. Although these features are less significant and it is not difficult to manipulate these ranking lists [33], still using these features combined with others can provide more accurate results.

The *'presence of HTTPS'* feature has a small weight in the phishing dataset (0.15) and near zero (0.05) for the malware dataset, which is not surprising since more than 58% of the phishing attacks used TLS certificates in the first quarter of 2019 according to the phishing activity report [69]. Therefore, the presence of a TLS certificate cannot be considered as a strong feature to distinguish malicious and compromised domains due to the popularity of using TLS certificates among both attackers and legitimate users.

Figure 7 shows the distribution of six selected features for each output class for the phishing dataset. For better representation of the distribution, we use logarithmic scales for the *'Bing search result'*, *'number of passive DNS for MX'*, *'content length'*, and *'number of internal hyperlink'* features. For example, in Figure 7 (d) the average length of the homepage content for compromised domains is greater than for maliciously registered domains. Looking at Table 4, the weight of the *'content length'* feature is 0.98 in favor of compromised domains, which means that more content on the homepage is an important characteristic of the benign but compromised domains.

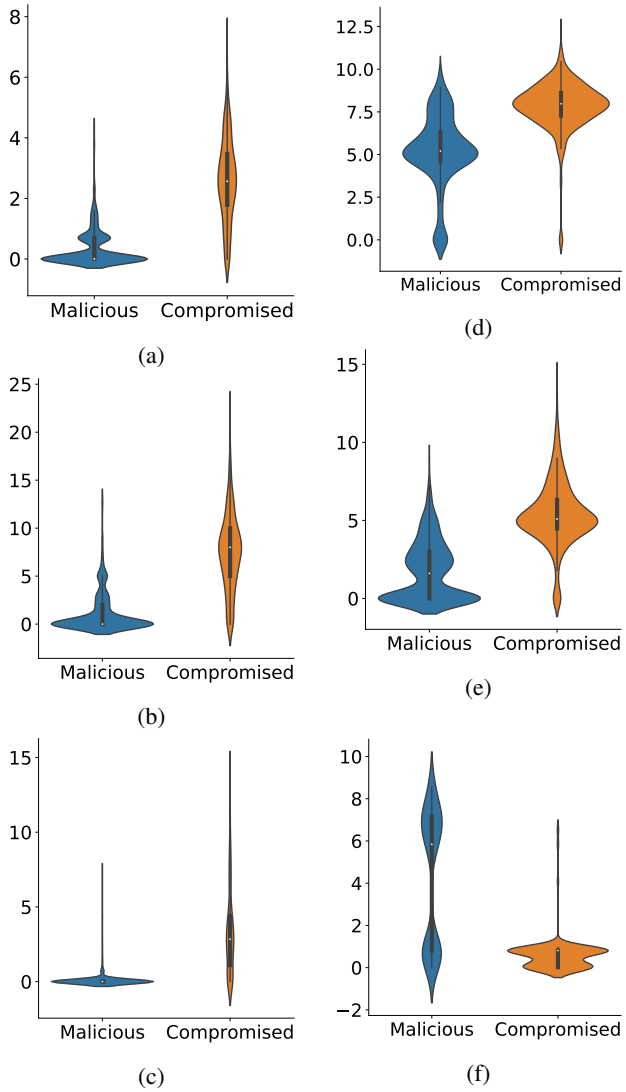


Figure 7: Distribution of the (a) *'internal hyperlink'*, (b) *'number of technologies'*, (c) *'Bing search results'*, (d) *'content length'*, (e) *'number of passive DNS MX'*, and (f) *'TLD maliciousness index'* features in phishing datasets. Y-axis is log-transformed for (a), (c), (d), and (e).

5.3. Case Studies

In this section, we present three case studies that may influence the classification results. The first one is related to website defacement when an attacker changes the visual appearance of a website by replacing the index page of the domain. To the best of our knowledge, the second case presents a new technique observed in phishing attacks for the first time. Finally, the third case is related to domain dropcatching in which attackers register expired benign domains to take advantage of their residual trust.

5.3.1. Case 1: Homepage defacement. It concerns a compromised domain name registered back in 2017 but detected by COMAR as malicious. We manually investigated the results, visited the homepage of the domain, and compared it with the data and screenshots from the data collection process. When we found the domain name in the OpenPhish blacklist, the homepage of the domain name was defaced and the content replaced by following HTML code:

```
<html><head></head>
    <body>ddddddddd</body>
</html>
```

The COMAR classifier uses 9 content-based features (as explained in Section 3.2.2). With the replaced homepage, COMAR was not able to extract features effectively, therefore, misclassified the domain as malicious (with the probability of 67.1% in favor of the malicious class).

As a matter of fact, this result is one of the drawbacks of the content-based features. If we do not fetch the real content of the domain for any reason, the classification results are uncertain. However, homepage defacement is very rare since attackers tend to keep the homepage of the compromised domains as intact as possible to avoid early detection by the website owners.

5.3.2. Case 2: New anti-phishing evasion technique. Phishers always look for new techniques to extend the lifetime of the phishing pages by evading anti-phishing bots and detection systems. One of the best ways to do so is to use defense techniques like page redirection, web-cloaking or server-side techniques like filtering famous user agents like *googlebots* or known scanners IP addresses [70]. As mentioned in Section 4.1, we scan each URL and domain twice, within ten days in between, to make sure that the state of both URL and the domain in the labeling process is correct. During our scan, we noticed an URL labeled as safe by Google safe browsing in both scans. Since the URL was in the Phishtank blacklist, which is a community based URL blacklist based on user reports, we had to manually check it to avoid a false positive. By visiting the URL, we noticed that the attacker used Google CAPTCHA to hide the real content of the malicious page. Therefore, even the browser emulation technique was not able to fetch the real phishing content unless a human solves the CAPTCHA manually. Figure 9 in Appendix D shows the website homepage, the phishing page protected by Google CAPTCHA, and a fake PayPal login page for phishing the user's credentials. Although COMAR classified correctly the domain as compromised (since we do not use any feature related to the content of the phishing URL), any phishing (fraud) detection system based on the content of the phishing URL cannot probably automatically fetch the page content. This is the first time we observe an evasion strategy using one of the strongest counter-attack techniques (CAPTCHA). Using techniques like CAPTCHA by phishing attackers may raise a serious challenge to security vendors in detection of malicious pages.

5.3.3. Case 3: Domain dropcatching. Domain dropcatching is the practice of registering a domain name once it is expired and released for new registration [71]. In this process, it is possible for miscreants to register an already expired benign domain name and inherit its residual trust. Miramirkhani et al. [71] showed that approximately 10% of the dropped domains are picked and registered by attackers for malicious purposes. The problem with these domains is that while they should be treated as newly registered domains (as they are), some of the features will match the original registration leading to misclassification of the domains as compromised. The feature sets concerned by drop-caught domains are TLS certificate, passive DNS, and ranking and popularity features. To show the effect of domain dropcatching, we compared the domain registration date with the date related to the first observed DNS query in DNSDB and the first captured page in the Wayback machine only for domains manually labeled as malicious. Whenever the date of the first captured page in the Wayback machine or first seen DNS query is older than the real registration date, we consider the domain as a drop-caught one. In this way, we found 7 samples in our dataset, 6 of them correctly classified as malicious.

Then, we applied the classifier two times on the samples: first, by removing passive DNS features (since they are affected by dropcaching and COMAR does not heavily rely on them) and then, by removing content-based features (since they can be relatively easily evaded and may affect classification). COMAR misclassified 1 and 2 samples (out of 7 samples) in the first and second experiment, respectively. While the number of samples is not enough to evaluate the generalizability of the method in the context of domain dropcaching, we assume that the benign history of the domain may mislead the classifier. We believe that this situation can be worse when attackers clone the content of the original website using the Wayback machine (we have not observed such a case in our dataset).

To reduce the negative impact of the drop-caught domains on the classifier, we could improve the Bing search engine result feature ($\mathbf{f_{28}}$) by only retrieving the results for a specific time slots i.e., after the registration date. Regarding the Wayback machine ($\mathbf{f_{33}}$), we already count only the number of captured pages after the domain registration date. However, passive DNS features and the TLS feature set are still heavily affected by the benign history of the domain and in the worst case scenario, attackers could also consider bypassing content-based features by cloning the content of the original website.

6. Related Work

Detecting malicious activity from URLs. Several authors proposed techniques in this category, which makes it one of the most prevalent research topic in the field. The main purpose of these methods is to detect phishing pages and malware C&C panels using machine learning techniques. In case of phishing attacks, Jain and Gupta, for example, proposed a machine learning approach that uses a set of 20 features to identify the input URL as malicious or legitimate [72]. Tian et al. proposed a combination of visual and content-based features to detect phishing attacks [73]. Their assumption is that even if attackers can evade content-based features by using obfuscation techniques, the final appearance of the phishing page should be the same as the target to persuade users to enter their credentials. Tan et al. proposed a phishing detection technique using lexical, URL-based, and content-based features combined with the Google search engine results to detect phishing URLs [74]. However, in a large scale detection system, it is not feasible to use the Google search engine due to the limitation of the number of requests [75]. COMAR uses some of the features from the above mentioned systems but the primary goal of COMAR is not to detect the malicious content of the URL since we create the domain classification system on top of already blacklisted URLs.

Detecting maliciously registered domains. Several effective methods have been proposed in this category. Although the ultimate goal of these methods is not the same as in COMAR, it might still be possible to apply some of the techniques on each domain in the URL blacklists and potentially identify the malicious ones. NOTOS [1] is a reputation system based on passive DNS queries to rank input domains. It extracts 41 features in three categories: 1) network-based, 2) zone-based, and 3) evidence-based features. Except for two features related to the lexical characteristics of the domain name itself, all other ones are derived from the IP address associated with the domain. NOTOS calculates the reputation of IP addresses, networks, and autonomous systems. Therefore,

if the domain is behind a reverse proxy system (e.g., CloudFlare [76]). NOTOS is unable to capture the true IP address and instead, it calculates the reputation of the network related to the reverse proxy rather than the reputation of the true network that hosts the domain. Another limitation is that it needs a large passive DNS dataset to perform well. COMAR does not rely on passive DNS queries and even by excluding passive DNS features, it can still obtain high accuracy with low false positive rate. PREDATOR [2] is a proactive recognition method to detect maliciously registered domains at the time of registration. It uses lexical features, IP-based features, and batch registration patterns to identify malicious domains. PREDATOR suffers from the same limitation as NOTOS in confronting reverse proxies. It also heavily uses WHOIS information and historical WHOIS data, which makes it only practical at registries that have access to such data. Le Pochat et al. proposed an automated method for classifying maliciously registered, algorithmically generated domain names and benign ones that accidentally collide with AGDs, within the constraints of the real-world takedown context of the Avalanche botnets [77]. MENTOR [78] is a system designed to remove benign domains from a blacklist of C&C domains. Both COMAR and MENTOR look for features related to the benign parts of the domains. While the goal of COMAR is to use these features to identify a domain as compromised, the goal of MENTOR is to distinguish benign domains (that are not abused) from malicious ones. One important caveat of MENTOR is the training and testing datasets. The authors used top 500 domains in the Alexa ranking list as the benign dataset. To form the malicious dataset, they used domains from various blacklists double-checked with the Google safe browsing (GSB) system. However, top 500 domains in the Alexa list are professionally designed, well-structured websites, which make them inappropriate to be used as fair samples of the benign domains in the wild. Moreover, for the malicious training set, if a domain is labeled by GSB as *'not safe'*, it does not necessarily mean that the domain name is completely malicious, since the goal of the GSB system is to detect malicious content (also hosted on benign but compromised domains) rather than malicious domains.

Detecting malicious activity on compromised domains. The main purpose of these methods is to detect malicious activity on compromised domains. Rao and Pais proposed a technique based on Google search engine queries to detect phishing activity [79]. Apart from the limitation of the number of queries, during the manual labelling of the domains in our dataset, we observed that most of the compromised domains are low ranked websites and many of them had been compromised in the first month of their registration and never got indexed by search engines. Corona et al. [80] proposed 11 content-based features along with image similarity combined using a fusion classifier to detect phishing URLs on compromised websites. We leverage some of their features in our work. However, our ultimate goal is not to detect phishing URLs but to classify domains as maliciously registered or compromised ones. Le Page et al. [34] proposed a method to classify maliciously registered domains and compromised ones. They used 15 features in three categories of lexical (5 features), domain name popularity (3 features), and 7 features related to the Internet Archive. Their results show that features derived from the Internet Archive perform the best among all features. However, relying heavily on the Internet Archive may lead to generate feature vectors with a considerable number of missing values since there is, high likely, no archive history for newly registered domains compromised in a short period after their registration.

7. Conclusion and Future Work

In this paper, we present COMAR, a system capable of distinguishing maliciously registered from compromised domains. COMAR leverages publicly available data and makes classification decisions based on the extracted features. Registries, registrars, and hosting providers can use it to decide on appropriate mitigation actions for each domain with malicious content. It can also serve as an effective tool for creating domain blacklists from the existing URL ones.

We show that the content-based features are the most effective in capturing the 'amount of benignness' of domains during their life cycles. We examine features regarding their robustness and the possible ways attackers can bypass them. High cost and effort for attackers complicates the evasion from COMAR and may therefore discourage malicious actors.

We introduce a new technique to compensate missing values in the 'domain registration date' field of the WHOIS data that outperforms the existing methods. We also show that approximately 12% of the domains get compromised in the first three months of their registration, which suggests that domain reputation systems based on the domain age cannot distinguish maliciously registered from compromised domains with high accuracy.

We plan to deploy COMAR at two European registry operators: SIDN (.nl domains) and AFNIC (.fr domains) and set up an early notification system to contact the owners of compromised domains and domain registrars for maliciously registered domains.

We also plan to correlate the concentrations of maliciously registered domains with a specific registration policy (prices, available payment methods, etc.) at the time of the domain creation. We intend to systematically distill a set of registration features preferred by attackers and analyze individual campaigns as well as long-term trends.

Acknowledgments

We thank: the anonymous reviewers and Thymen Wabeke (SIDN Labs), Pierre-Aymeric Masse (AFNIC), Paul Vixie (Farsight Security) for their valuable feedback; Anti-Phishing Working Group, OpenPhish, PhishTank, URLhaus for providing access to their URL blacklists; Farsight Security for sharing DNSDB, and the DNSDB data contributors. This work has been carried out in the framework of the COMAR project funded by SIDN, the .NL Registry and AFNIC, the .FR Registry. It was partially supported by the ANR projects: the Grenoble Alpes Cybersecurity Institute CYBER@ALPS under contract ANR-15-IDEX-02, PERSYVAL-Lab under contract ANR-11-LABX-0025-01, and DiNS under contract ANR-19-CE25-0009-01.

References

[1] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a Dynamic Reputation System for DNS." in *Proc. USENIX Security Symposium*, 2010, pp. 273–290.

[2] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-of-Registration," in *Proc. ACM CCS*, 2016, pp. 1568–1579.

[3] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in *Proc. NDSS*, 2011, pp. 1–17.

[4] (2019) EPP Status Codes – What Do They Mean, and Why Should I Know? [Online]. Available: https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en

[5] (2016) Avalanche Network Dismantled in International Cyber Operation. [Online]. Available: https://www.europol.europa.eu/newsroom/news/'avalanche'-network-dismantled-in-international-cyber-operation

[6] A. Noroozian, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Developing Security Reputation Metrics for Hosting Providers," in *Proc. Workshop on Cyber Security Experimentation and Test (CSET)*, 2015.

[7] M. Kührer, C. Rossow, and T. Holz, "Paint It Black: Evaluating the Effectiveness of Malware Blacklists," in *Proc. RAID*. Springer, 2014, pp. 1–21.

[8] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda, "Fire: Finding Rogue Networks," in *Proc. ACSAC*. IEEE, 2009, pp. 231–240.

[9] A. Noroozian, M. Ciere, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Inferring Security Performance of Providers from Noisy and Heterogenous Abuse Datasets," in *Workshop on the Economics of Information Security (WEIS)*, 2017.

[10] D. Canali, D. Balzarotti, and A. Francillon, "The Role of Web Hosting Providers in Detecting Compromised Websites," in *Proc. WWW Conference*. ACM, 2013, pp. 177–188.

[11] M. Korczyński, M. Wullink, S. Tajalizadehkhoob, G. Moura, A. Noroozian, D. Bagley, and C. Hesselman, "Cybercrime after the Sunrise: A Statistical Analysis of DNS Abuse in New gTLDs," in *Proc. Asia CCS*. ACM, 2018, pp. 609–623.

[12] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. v. Eeten, "Reputation Metrics Design to Improve Intermediary Incentives for Security of TLDs," in *IEEE EuroS&P*, 2017, pp. 579–594.

[13] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse," in *Proc. NDSS*, 2015.

[14] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage, "On the Effects of Registrar-level Intervention," in *Proc. 4th USENIX Conference on Large-scale Exploits and Emergent Threats*, ser. LEET'11, 2011, pp. 5–5.

[15] OpenPhish. [Online]. Available: https://openphish.com/

[16] PishTank: Join the Fight Against Phishing. [Online]. Available: https://www.phishtank.com/

[17] APWG: Anti-Phishing Working Group. [Online]. Available: https://apwg.org

[18] URLhaus: Sharing Malicious URLs That Are Being Used for Malware Distribution. [Online]. Available: https://urlhaus.abuse.ch/

[19] Freenom. (2017) Free and Paid Domains. [Online]. Available: https://www.freenom.com/en/freeandpaiddomains.html

[20] ICANN. (2018) Temporary Specification for gTLD Registration Data. [Online]. Available: https://www.icann.org/resources/pages/gtld-registration-data-specs-en

[21] Farsight Security. Passive DNS Historical Internet Database: Farsight DNSDB. [Online]. Available: https://www.farsightsecurity.com/solutions/dnsdb

[22] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8446.txt

[23] Public Suffix List. [Online]. Available: https://publicsuffix.org

[24] Wappalyzer: Identify Technology on Websites. [Online]. Available: https://www.wappalyzer.com/

[25] T. Miu, A. Hui, W. Lee, D. Luo, A. Chung, and J. Wong, "Universal ddos mitigation bypass," *Black Hat USA*, 2013.

[26] (2016) Global Phishing Survey: Trends and Domain Name Use in 2016. [Online]. Available: https://docs.apwg.org/reports/APWG_Global_Phishing_Report_2015-2016.pdf

[27] V. L. Pochat, T. van Goethem, and W. Joosen, "A Smörgåsbord of Typos: Exploring International Keyboard Layout Typosquatting," in *Proc. IEEE WTMC Security and Privacy Workshop*, 2019.

[28] W. Wang and K. Shirley, "Breaking Bad: Detecting Malicious Domains Using Word Segmentation," in *Proc. 9th Workshop on Web 2.0 Security and Privacy*, 2015.

[29] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, and N. Asokan, "Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application," *IEEE Transactions on Computers*, vol. 66, no. 10, pp. 1717–1733, 2017.

[30] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in *Proc. NDSS*. The Internet Society, 2010.

[31] P. Lison and V. Mavroeidis, "Neural Reputation Models Learned from Passive DNS Data," in *IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3662–3671.

[32] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A Content-Based Approach to Detecting Phishing Web Sites," in *Proc. WWW Conference*. ACM, 2007, pp. 639–648.

[33] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *Proc. NDSS*, 2019.

[34] S. Le Page, G.-V. Jourdan, G. V. Bochmann, I.-V. Onut, and J. Flood, "Domain Classifier: Compromised Machines Versus Malicious Registrations," in *International Conference on Web Engineering*. Springer, 2019, pp. 265–279.

[35] B. Anderson, S. Paul, and D. McGrew, "Deciphering Malware's Use of TLS (Without Decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, 2018.

[36] MAXMIND: Detect Online Fraud and Locate Online Visitors. [Online]. Available: https://www.maxmind.com

[37] S. Kitterman, "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1," Internet Requests for Comments, RFC 7208, April 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7208.txt

[38] L.-H. Lee, K.-C. Lee, H.-H. Chen, and Y.-H. Tseng, "Poster: Proactive Blacklist Update for Anti-Phishing," in *Proc. ACM CCS*, 2014, pp. 1448–1450.

[39] DnsTwister: The Simple and Fast Domain Name Permutation Engine. [Online]. Available: https://dnstwister.report/

[40] D. Plohmann. (2018) DGArchive. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de

[41] M. Korczyński, M. Król, and M. van Eeten, "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates," in *Proc. Internet Measurement Conference*. ACM, 2016, pp. 271–278.

[42] NormShield Blog. (Retrieved: August 2019) Domain shadowing. [Online]. Available: https://www.normshield.com/domain-shadowing/

[43] Wappalyzer Signature List. [Online]. Available: https://github.com/AliasIO/Wappalyzer

[44] (2019, May) WordPress Vulnerability Statistics. [Online]. Available: https://wpvulndb.com/statistics

[45] Exploit Database. [Online]. Available: https://www.exploit-db.com/

[46] VULDB: The Community-Driven Vulnerability Database. [Online]. Available: https://vuldb.com/

[47] J. Kornblum, "Identifying Almost Identical Files Using Context Triggered Piecewise Hashing," *Digital investigation*, vol. 3, pp. 91–97, 2006.

[48] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein, "Cloak of Visibility: Detecting When Machines Browse a Different Web," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 743–758.

[49] M. Kucherawy and E. Zwicky, "Domain-based Message Authentication, Reporting, and Conformance (DMARC)," Internet Requests for Comments, RFC 7489, March 2015. [Online]. Available: https://tools.ietf.org/html/rfc7489

[50] S. Maroofi, M. Korczyński, and A. Duda, "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2020.

[51] (2019) Wayback Machine General Information. [Online]. Available: https://help.archive.org/hc/en-us/articles/360004716091-Wayback-Machine-General-Information

[52] (Retrieved: March 2020) Cognitive services pricing - bing search api. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/search-api/

[53] Alexa: SEO and Competitive Analysis Software. [Online]. Available: https://www.alexa.com/

[54] MAJESTIC: Find Out Who Links to Your Website. [Online]. Available: https://majestic.com/

[55] Quantcast Ranking. [Online]. Available: https://quantcast.com

[56] Umbrella: Top 1 Million Websites. [Online]. Available: http://umbrella-static.s3-us-west-1.amazonaws.com/

[57] The Spamhaus Project. [Online]. Available: https://www.spamhaus.org/

[58] M. Aertsen, M. Korczyński, G. C. M. Moura, S. Tajalizadehkhoob, and J. van den Berg, "No Domain Left Behind: Is Let's Encrypt Democratizing Encryption?" in *Proc. ANRW*, 2017, pp. 48–54.

[59] (2019) Phishing Trends and Intelligence Report: The Growing Social Engineering Threat. [Online]. Available: https://info.phishlabs.com/2019-pti-report-evolving-threat

[60] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. John Wiley & Sons, 2019, vol. 793.

[61] S. Liu, I. Foster, S. Savage, G. M. Voelker, and L. K. Saul, "Who Is .com?: Learning to Parse WHOIS Records," in *Proc. Internet Measurement Conference*. ACM, 2015, pp. 369–380.

[62] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A Gentle Introduction to Imputation of Missing Values," *Journal of Clinical Epidemiology*, vol. 59, no. 10, 2006.

[63] (2020) Internet Archive: Wayback Machine. [Online]. Available: https://archive.org/web/

[64] (2018) Google Transparency Report. [Online]. Available: https://transparencyreport.google.com/https/certificates

[65] O. Gasser, B. Hof, M. Helm, M. Korczynski, R. Holz, and G. Carle, "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements," in *Proc. PAM*, 2018, pp. 173–185.

[66] (2016) Location-Based Threats: How Cybercriminals Target You Based on Where You Live. [Online]. Available: https://news.sophos.com/en-us/2016/05/03/location-based-ransomware-threat-research/

[67] COMODO Certification Authority. [Online]. Available: https://ssl.comodo.com/

[68] S. J. Russell and P. Norvig, *Artificial Intelligence: a Modern Approach, 3rd Edition*. Prentice Hall, 2009.

[69] (2019) Phishing Activity Trends Report, 1stQuarter 2019. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf

[70] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a Phisher's Mind: Understanding the Anti-Phishing Ecosystem through Phishing Kit Analysis," in *IEEE eCrime*, 2018, pp. 1–12.

[71] N. Miramirkhani, T. Barron, M. Ferdman, and N. Nikiforakis, "Panning for gold.com: Understanding the Dynamics of Domain Dropcatching," in *Proc. WWW Conference*, 2018, pp. 257–266.

[72] A. K. Jain and B. B. Gupta, "Towards Detection of Phishing Websites on Client-Side Using Machine Learning Based Approach," *Telecommunication Systems*, vol. 68, no. 4, pp. 687–700, 2018.

[73] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild," in *Proc. ACM IMC*, 2018, pp. 429–442.

[74] C. L. Tan *et al.*, "PhishWHO: Phishing Webpage Detection via Identity Keywords Extraction and Target Domain Name Finder," *Decision Support Systems*, vol. 88, pp. 18–27, 2016.

[75] (2019) Google Custom Search API. [Online]. Available: https://developers.google.com/custom-search/v1/overview

[76] (2019, August) How We Made Our DNS Stack 3x Faster. [Online]. Available: https://blog.cloudflare.com/how-we-made-our-dns-stack-3x-faster/

[77] V. L. Pochat, T. van Hamme, S. Maroofi, T. V. Goethem, D. Preuveneers, A. Duda, W. Joosen, and M. Korczyński, "A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints," in *Proc. NDSS*, 2020.

[78] N. Kheir, F. Tran, P. Caron, and N. Deschamps, "Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists," in *IFIP SEC*. Springer, 2014, pp. 1–14.

[79] R. S. Rao and A. R. Pais, "Jail-Phish: An Improved Search Engine Based Phishing Detection System," *Computers & Security*, vol. 83, pp. 246–267, 2019.

[80] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "Deltaphish: Detecting Phishing Webpages in Compromised Websites," in *Proc. ESORICS*, 2017.

[81] B. W. Matthews, "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, 1975.

[82] D. Y. Wang, S. Savage, and G. M. Voelker, "Juice: A Longitudinal Study of an SEO Botnet," in *Proc. NDSS*, 2013.

[83] P. Thomas. (2010) Web Application Fingerprinting and Vulnerability Inferencing. [Online]. Available: https://media.blackhat.com/bh-us-10/presentations/Thomas/BlackHat-USA-2010-Thomas-BlindElephant-WebApp-Fingerprinting-slides.pdf

[84] (2018) Save Pages in the Wayback Machine. [Online]. Available: https://help.archive.org/hc/en-us/articles/360001513491-Save-Pages-in-the-Wayback-Machine

[85] T. Van Goethem, N. Miramirkhani, W. Joosen, and N. Nikiforakis, "Purchased Fame: Exploring the Ecosystem of Private Blog Networks," in *Proc. Asia CCS*, 2019, pp. 366–378.

Appendix A. Evaluation Metrics

We use the following metrics to evaluate our machine learning algorithms.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precission = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 - score = \frac{2TP}{2TP + FP + FN} \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{6}$$

where *TP*, *TN*, *FP*, *FN* are the number of true positives, true negatives, false positives and false negatives, respectively. Compromised domains are considered positive

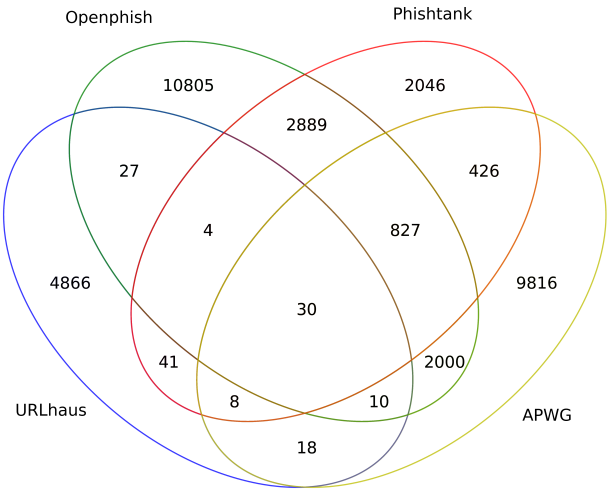


Figure 8: Venn diagram of the collected URLs from four blacklists.

and maliciously registered domains negative. Accuracy is the ratio of the number of correct predictions to the total number of input samples. Precision means the percentage of relevant results. Recall refers to the percentage of total relevant results correctly classified by the algorithm. The F1 score is the harmonic mean of precision and recall.

The Matthews correlation coefficient (MCC) [81] is a measure of the quality of binary classification. The return value of MCC is between -1 and +1 which +1 represents a perfect prediction, 0 means random prediction and -1 means total disagreement between the predictions and true labels. The advantages of MCC over accuracy and F1-score is that it considers the size as well as the imbalance of dataset. Most importantly, MCC takes into account true and false positives and negatives (all the entries of the confusion matrix not only true-positives and true-negatives).

Appendix B. Phishing and Malware Datasets

Figure 8 shows the Venn diagram of the collected URLs from a) URLhaus, b) APWG, c) OpenPhish, d) Phishtank, and the overlap between them.

Appendix C. Evasion Techniques

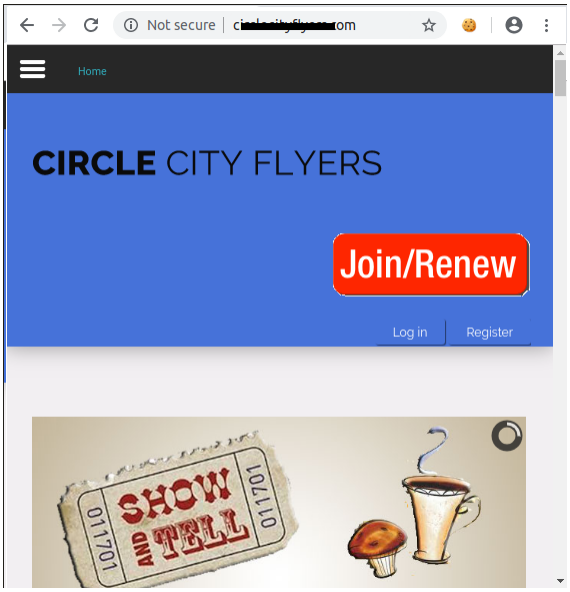
In Section 1, we discussed the appropriate mitigation actions for compromised and maliciously registered domains by different intermediaries. For malicious domains, one recommended action is to take down the domain or suspend the hosting service related to that domain. This action may generate extra costs for malicious actors (losing the domain name or the hosting service), which makes it a good reason to avoid their domain being classified as maliciously registered. However, manipulating COMAR features also requires extra effort. In this section, we examine the possibility of feature evasion and associated costs. We take into account i) the amount of money the attackers should pay to bypass a specific feature, ii) the amount of time the attacker should spend to evade each feature and, iii) the necessary skills the attacker should have to bypass a feature.

Generally, it is safe to consider external features as more difficult to evade compared to the features under the control of the attacker. For example, manipulating search engine results, the Wayback Machine as well as passive DNS data are more difficult compared to content-based or lexical features in case of maliciously registered domains. However, it does not necessarily make external features completely bulletproof against manipulation. Furthermore, any feature with a one-time cost (either in terms of time or money) for the attacker cannot be considered as robust.

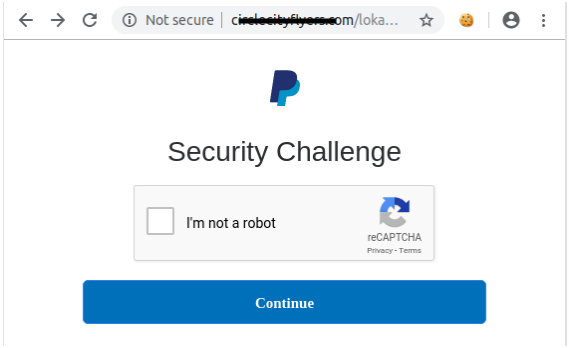
Content-based features. In Section 5.2, we show that content-based features are among the best ones, yet most available in our set. Through this feature set, we exploit the *benignness* of the domain by analyzing 1) the length of the generated content on the homepage of the domain, 2) the relationship between the homepage and other (possible) pages related to that domain (i.e., the number of internal and external hyperlinks), 3) the amount of effort required by the domain owner to design a professional websites (i.e., the number of technologies that are used to create the website), and 4) the number of technologies prone to attacks. We now consider possible evasion techniques the attacker can use to bypass content-related features.

1) **Content length and hyperlinks**. To bypass the content length feature ($\mathbf{f_{10}}$), the attacker needs to generate lengthy content either manually (which is not feasible in large-scale attacks) or automatically through third-party applications. The same methodology can be applied to features related to internal and external hyperlinks (i.e., $\mathbf{f_{13}}$ and $\mathbf{f_{18}}$). Wang et al. [82], studied the effectiveness of black hat search engine optimization (SEO) campaigns to evaluate the possibility of manipulating search engine results for specific keywords by generating fake contents and leveraging various linking strategies. This method can be used to evade features related to the content length and hyperlinks but it requires a fair amount of effort and costs not always available for the attacker.
2) **Technology-related features**. As mentioned in Section 3, we use Wappalyzer to enumerate the technologies used by the domain owner to design the website. Wappalyzer is a fast, free, easy to use, signature-based tool able to extract the used technologies by partial string and regular expression matching. Unfortunately, it is also easy to evade. For example, using PHP as a server-side programming language, the default name for the session ID stored as a cookie in the client machine is '*PHPSESSID*'. Wappalyzer uses this name to decide if the server-side code is PHP or not. Therefore, it is possible to mislead Wappalyzer and force it to make a wrong decision on the server-side language just by changing one keyword in cookies. However, decisions can be made using more advanced techniques e.g., hash-based fingerprinting [83].

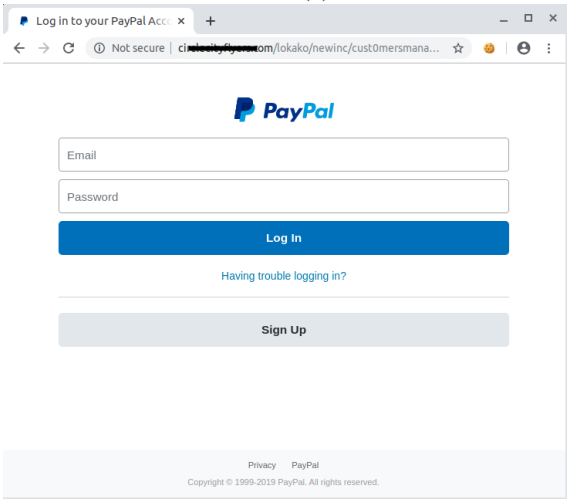
Overall, to evade content-based features, the miscreant must establish a fully-functional website with different content and hyperlinks related to the domain name itself either manually, which takes **time**, or automatically, which impose additional **costs** on them.



(a)



(b)



(c)

Figure 9: Home page of the compromised domain (a), Google reCAPTCHA with a fake PayPal logo (b), and a fake PayPal login page for phishing user's credentials.

Ranking and popularity features. Manipulating features in this category is not completely under control of the attacker as it represents an external feature. However, it is feasible for a sufficient amount of time and effort.

1) Regarding the Wayback Machine, the attacker can manually submit URLs related to their domains to the Internet Archive project [84].
2) Regarding the Bing search engine results, using SEO techniques (e.g., black hat SEO as explained earlier [85]), it is possible to increase the number of indexed pages for each domain name in search engines.
3) Regarding top ranking websites (e.g., Alexa ranking system), previous research shown that it feasible to manipulate them [33].

However, the cost of evading *'ranking and popularity'* features is related to the **expertise** and **amount of time** the attacker should spend to make her domains as popular as it is necessary to evade the COMAR classifier.

TLD and WHOIS features. COMAR uses one WHOIS-based feature (i.e., *'domain age'*) and two TLD-based features (i.e., *'TLD maliciousness index'* and *'TLD price'*).

1) To evade the *'domain age'* feature, the attacker should register domains long time before using them since we use the number of years before blacklisting, which imposes costs in terms of **money** on the attacker as she needs to register or re-new the domain for a period of a few years to evade this feature.
2) *'TLD maliciousness index'* is another strong feature of COMAR to decide on the state of a domain. One of the factors affecting the value of this feature is pricing. Cheap TLDs (or the free ones) have a higher 'maliciousness' value compared to the expensive TLDs [11], [57]. A higher value of the maliciousness index increases the chance that the domain name is maliciously registered. Therefore, to avoid being detected by the COMAR classifier, the attackers should register domains with TLD suffixes with low maliciousness values, which means they should pay more **money**.

Lexical, passive, and active DNS features.

1) Lexical features are relatively easy to evade. For malware distributors, the domain name is not important since the victims that download the malicious content from the website are not humans but infected machines. However, for phishers, the choice of the domain name is relatively important to conduct a successful attack. For example, insta-support.com is more appealing to lure Instagram users compared to the name that has no indication of Instagram.
2) Regarding active DNS features, it is feasible to setup a mail server and/or define, for example, SPF rules in 'TXT' records. However, for attacks performed at a larger scale, the process needs automation.
3) Passive DNS features are the most difficult to evade as the sensors are distributed all around the world. Attackers are not aware of their locations and even if they were, it is not trivial to inject a large number of DNS packets as the monitoring sensors are placed above the local recursive resolvers.

Appendix D.
Captcha Evasion Technique

Figure 9 shows a compromised website hosting a phishing page protected by Google CAPTCHA to prevent anti-phishing bots from accessing the malicious page content (details in Section 5.3.2).