MASTER

Improving Early Phishing Detection using SSL & WHOIS data an Application to PhishDetect for Civil Society Protection

Mulder, C.J.

Award date:
2020

Improving Early Phishing Detection using SSL & WHOIS data: an Application to PhishDetect for Civil Society Protection

Chris Mulder
MSc Thesis
Student no. 1038186
c.j.mulder@student.tue.nl
TU Eindhoven
June 1, 2020

***Abstract*—Early phishing detection has the purpose of identifying domains that are likely to be employed for phishing at some point in the future shortly after registration. One tool that focusses on this is PhishDetect, which purpose is to protect at-risk users by detecting and blocking phishing domains at an early stage. This paper analyzes the usability of SSL Certificates and WHOIS information to enhance the early detection capabilities of PhishDetect. In this paper we propose several logistic regression models, which rely on SSL and WHOIS data along with several lexical features of the domain as input, that can achieve up to 86% accuracy when classifying domains which at one point will be used for phishing. Our results can be used to improve other existing phishing detection solutions, especially those that focus on early detection, as our models do not make use of content deployed on the website for classification. We also show that features relying on SSL data to detect phishing websites proposed in previous research are less useful in modern-day situations because of an updated threat landscape.**

***Index Terms*—phishing,early detection,SSL certificate,WHOIS**

1. Introduction

Phishing is a set of techniques that aim at stealing information from users by deceiving them in interacting with attacker-controlled systems (e.g. a phishing webpage) that resemble other systems the user regularly interacts with (e.g., a home-banking login page). To exemplify the magnitude and importance of this attack vector, a recent report suggests phishing costed businesses $1.6 billion between October 2013 and December 2016 in the US alone [1]. Importantly, phishing campaigns have also been aimed at civil society organizations and media collectives [2]. The Anti-Phishing Working Group (APWG) reports that in the third quarter of 2019, a total of 266, 387 phishing websites were detected, more than two-thirds made use of some form of encryption deceiving users of the legitimacy of the website.

In the domain of civil rights protection, one important resource for phishing reporting and detection is `PhishDetect` [3]; `PhishDetect` started as a project with the support of Amnesty International[1] and Security Without Borders[2]. `PhishDetect` is a tool to help at-risk users block and report potential phishing websites. It makes use of SSL certificate log streams to identify suspicious newly registered domains, as well as additional lists of newly registered domains provided by registrars to detect phishing domains as early as possible. Currently the system relies on suspicious URL heuristics[3] as well as word blacklists as their detection mechanism. `PhishDetect` has been in production for several years, but no specific research has been conducted on the effectiveness of the system, and means to improve it. In this paper, we pose the following research question: "**How can we improve the PhishDetect early detection system with additional data sources?**". To address our research question, we rely on `PhishDetect` data spanning 20 months from the end of 2017 to mid 2019. This dataset consists of domains and SSL certificates belonging to these same domains which are flagged as suspicious by `PhishDetect`. We do not have access to a full dataset of legitimate domains, but only already predetermined suspicious domains. Because of this, we focus our research on extracting true and false positives from the PhishDetect dataset in order to test its successfulness, and analyze URLs and metadata of these domains to find any potential distinguishing properties present. The results of these experiments can in turn be used to improve the current product.

Earlier research has shown that the majority of phishing campaigns are short-lived, with up to 70% of phishing campaigns being concluded within 5 hours [4]. Stressing the importance of early phishing detection. The main goal of PhishDetect is early phishing detection, thus we focus our efforts on data that is available shortly after registration, such as the domain name, WHOIS information and SSL certificate data.

Previous research [5], [6] on detecting phishing domains using SSL certificates showed promising results, with claimed phishing precision classification of 95% and and a recall rate of above 93% [5]. However, the used top sites lists for ground truth derivation in these studies may contain biases [7] and could be prone to manipulation [8]. Also, an updated threat landscape requires a re-evaluation of the robustness of the proposed models in modern-day

[1] https://www.amnesty.org/

[2] https://www.securitywithoutborders.org/

[3] https://github.com/phishdetect/phishdetect/blob/master/urlchecks.go

situations.

This paper is set up as follows: we first introduce some background, discuss related works and present our methodology and data collection. After that we carry out an analysis on our data and then present the results of our models. Finally we benchmark similar papers on our dataset to test the effectiveness of these papers compared to ours.

2. Background

2.1. Phishing and domain names

In order to trick users into visiting phishing websites, criminals often try to make the phishing website resemble the original website as closely as possible. This includes methods where phished URLs closely resemble the URLs of the original website. One commonly used technique is *typosquatting* [9], where attackers register a domain name that resembles the targeted domain name very closely. Whereas the legitimate domain may be, say, **bankname.com**, a corresponding phishing domain targeting that bank could be **b*o*nkname.com** or **bank*k*name.com**. Similarly, phishers make use of subdomains to make domains look more legitimate. Often one or a combination of the following methods are used [10]:

- Putting the name of the targeted brand in the subdomain, e.g. **bankname.fakewebsite.com**.
- Using a fake TLD in the subdomains, e.g. **bankname.com.fakewebsite.com**.
- Using so called “function words” in domains and subdomains in hopes of throwing off focus on the actual domain name. Function words are often words related to the phishing page and attempt to give a false sense of security, examples of such function words are “verify”, “login” or “account”.
- URL padding, i.e. making the URL as long as possible to mask the actual domain name, e.g. **bankname.com.account.login.secure.php.id555 .fakewebsite.com**
- So called homograph attacks were introduced when ICANN made Internationalized Domain Names (IDNs) available for registration [11], allowing attackers to use non-latin characters that look similar to latin characters. E.g. ”*ạ*” (U+7841) instead of a regular ”a” (U+0061), tricking the user into visiting **b*ạ*nkname.com** instead of **bankname.com**.

2.2. WHOIS information

WHOIS is a query/response protocol that allows WHOIS servers to be queried which provide information regarding registered domain names in human readable format. WHOIS information typically contains data such as domain registration and renewal dates, as well as information about the registrant (the owner of the domain) and the registrar (the organization that sells domain names to registrants). As privacy regulations have become more strict over the past few years, WHOIS data has become more shielded. As a result of the GDPR, ICANN [12] has published a specification [13] to define (temporary) requirements to ensure ICANN and gTLD registry operators as well as registrars worldwide comply with the GDPR. This means analysis on registrants given names/email adresses is no longer possible.

2.3. SSL Certificates

SSL Certificates are small files which contain a cryptographic key along with several properties that are bound to an entity. SSL certificates contain basic information regarding a domain and the organization behind the domain. Analyzing these fields might show differences between legitimate and phishing websites.

2.3.1. Certificate transparency logs. Certificate transparency logs are logs that contain issued digital SSL certificates, described in RFC6962 [14]. In order for most applications, such as browsers, to accept an SSL certificate, they must be present in a certificate log. These logs are publicly available.

2.3.2. Types of SSL certificates. SSL certificates are often categorized by their validation method, which results in three distinct types of certificates. Domain Validated (DV), Organization Validated (OV) and Extended Validation (EV) certificates. DV certificates are of interest to phishers since they have low to no cost, and require no verification of identities except for the domain. In recent years, services such as LetsEncrypt [15] and SSL For Free [16] have started providing free DV certificates. This made it easy for any website to get an SSL certificate on their website within a few minutes. OV and EV certificates require additional verification steps and cost money to acquire.

2.4. Phishing feeds

Several popular services provide feeds containing blacklisted domains to the public, the most popular being PhishTank and Google Safe Browsing.

2.4.1. PhishTank. PhishTank[4] is a community driven site where people can submit, verify and share information about phishing sites. A user is able to report a phishing website to PhishTank, after which other users can verify if the URL is actually a phishing URL or not. PhishTank periodically checks if phishing websites are still online. Once a website is offline, PhishTank will remove it from their phishing feed.

2.4.2. Google Safe Browsing. Google Safe Browsing[5] is used by popular webbrowsers such as Chrome [17], Firefox [18] and Safari [19] as well as most Google products such as Google Search and Gmail [20]. In the last quarter of 2019, it identified over $30,000$ new phishing sites every week, and over $1,000$ sites hosting malware every week [21].

[4] https://www.phishtank.com/ Last visited on 05-2020

[5] https://safebrowsing.google.com/ Last visited on 05-2020

2.5. A note on detection accuracy

For anti-phishing campaigns, it is important that the number of false positives is extremely low, as blocking access to legitimate domains could severely hinder users in their daily operations. As most domains are legitimate, low false positive rates may still produce high volumes of inaccurately blocked domains. Incorrectly classifying a website as phishing eventhough it is legitimate could also cause monetary loss, depending on how widespread the use of the results of the system are. For example: if a legitimate webshop somehow got into the Google Safe Browsing list for a small amount of time, it could discourage anyone who visits the website from buying anything at that moment or even at some point in the future. Another side effect of a too high false positive rate is that if a user gets phishing warnings when visiting legitimate websites, it could make users lose trust in the system, causing them to ignore any further warnings for sites which are actually malicious. Furthermore, the true positive rate should be as high as possible, as a system with a low true positive rate is not very effective at detecting phishing websites.

3. Related work

We look at several different methods of detecting and classifying malicious websites and at the successfulness of these methods. We also discuss other studies that are focussed on the interaction with users, to determine what kind of methods phishers might use in phishing attacks to increase their effectiveness.

3.1. Early phishing detection

Early phishing detection has the purpose of identifying domains (as soon as they are registered) that are likely to be employed for phishing at some point in the future. Hence, these methods are employed to flag potentially dangerous domains before the phishing content is actually deployed at the corresponding URL.

3.1.1. URL-based early detection. In order to detect phishing websites early, several URL based detection methods have been proposed. Marchal et al. [22] leveraged natural language modelling techniques to build proactive blacklists, allowing discovery of malicious websites during or shortly after registration by querying generated potential phishing domain names periodically. When testing their generated proactive blacklists they were able to detect several phishing websites in advance (270 when generating a list of $200,000$ domains, 350 when generating $600,000$). Although this allows detection of phishing websites before they are even registered, its false positive rates are unknown. Furthermore, probing the generated lists of domains periodically to determine if they are in use can be resource intensive. This will only go up the more domains are generated. Properties used to generate these proactive blacklists could however be used in our phishing detection models.

Bo et al. [23] proposed a system that makes use of daily DNS query logs to detect phishing domains early. Although this allows for fast detection of phishing domains, their system processes a batch of DNS logs once a day, meaning that phishing domains could be in use for up to 24 hours before being detected by their system. Such a system might not be very effective for early phishing detection, since 75% of phishing campaigns are finished after 24 hours according to [4]. PhishStorm is an URL detection system proposed by Marchal et al. [24] relying only on lexical URL analysis using features derived from search engine query data. They achieved a classification accuracy of 94.91% with a false positive rate of 1.44%. Later, Bahnsen et al. [25] proposed a similar approach by extracting URL features and using a recurrent neural network with an accuracy of up to 98.7%.

In order to combat *homoglyph* attacks, where normal ASCII characters are replaced by similar looking counterparts from non-latin alphabets, solutions such as translation tables have been proposed [26] [27]. These translation tables allow domain names to be mapped to how they would be interpreted by humans. Mayank Dhiman et al. [26] show this is very effective in combatting attempts to bypass content based filtering used by spam filters. These mappings can also help find phishing domains by comparing registrar information of domains containing unicode characters with their mapped counterpart, as shown by Elsayed & Shosha [27], who propose an IDN monitoring solution which detects non-ASCII domains by replacing homoglyph characters with their ASCII lookalikes, detecting 225 homoglyph attacks in a list of $41,500$ IDNs. PhishDetect makes use of blacklists of (brand)names with homoglyph lookalikes to detect these attacks, but its effectiveness has yet to be measured. Eric Lin et al. [28] have looked at how users determine if a website URL is likely legitimate or not. Eric Lin et al. found that participants of their study quickly got confused by complex URLs. To counter this, the authors propose a method to highlight relevant parts of the domain, but this seemed to improve protection only marginally, as they found out their test subjects spent little to no time analyzing the URL but mostly looked at the content of the website, confirming earlier research which came to similar conclusions [29], [30]. Since early phishing domain detection relies heavily on URL features aimed at appearing legitimate, phishers might opt to use arbitrary domains unrelated to the targeted websites without too much of a decrease in success rate. Hence, in our early detection models we aim to also provide features that do not rely on domain name characteristics.

One of the most commonly used methods for blocking detected malicious URLs is using blacklists [31]. Sheng et al. [4] discovered that 63% of phishing campaigns in their dataset lasted less than two hours, emphasizing the importance of early phishing detection. When comparing their dataset with existing blacklists, only 20% of phishing websites were present in the blacklist at hour zero, $47\% - 83\%$ of phishing domains appeared in blacklists after 12 hours. Our research is aimed at improving early detection, potentially allowing phishing feeds to provide a list of phishing domains which are not yet actively used.

3.1.2. SSL based detection. Mishari et al. [6] proposed a machine learning approach to detect phishing domains using SSL certificates. The research focussed on analyzing

features such as the number of days the certificate is valid, its signer as well as the country of origin of the signer. The dataset used for this research consisted of 2410 unique SSL certificates belonging to malicious websites, and 19021 SSL certificates of legitimate websites. Their results showed that there exist differences between SSL certificates belonging to malicious domains and those belonging to popular domains. The proposed classifiers achieved an accuracy of up to 88%.
Using a similar approach, Z. Dong et al. [5] were able to get 95.5% precision and 93.7% recall on classifying phishing domains that used SSL certificates, and 94.7% precision and 96.3% recall on classifying non-phishing domains. This research paper was published in 2015. At the time less than 5% of phishing websites had an SSL certificate, according to the Anti-Phishing Working Group [32]. However, since then, more than half of phishing websites make use of SSL certificates [32], questioning the discriminative power of this feature to detect phishing domains, as discussed by Drury & Meyer [33] who concluded that it is generally impossible to differentiate between benign sites and phishing sites based on the content of their certificates alone. So although the results of detecting phishing domains based on SSL certificate information seem promising, and results of this research could potentially prove to be useful in our research, its modern day effectiveness is unknown. Therefore we will benchmark these two papers on our dataset and compare the results with the results of our models.

3.2. Detection of deployed phishing websites

3.2.1. Content based detection. For detecting phishing websites that are already in use, several content based detection approaches have been proposed. Many of these research efforts make use of scrapers or automated browsers [34]–[39] to look at the content and structure of a webpage to extract features and characteristics for use in their models. Additionally, research in this direction also look at the composition of the queries and operations encoded in URLs (e.g. GET requests). For example, [24], [25] leverage full URLs for their detection mechanisms, including path and query string present in the URL. The problem with this is that these properties are often not known until the phishing campaign is active. Our research is aimed at just domains alone thus we cannot make use of all proposed features in these papers. We will however still be able to make use of domain features used in this research.

”Know your Phish” [34] describes techniques to classify phishing websites based on the HTML content that is served on the website. The authors have implemented a scraper which visits a website using an automated web browser and stores its response, as well as a feature extractor which is able to extract 212 features from the data sources in the webpage. These features are then fed into a previously trained classification model. The proposed technique requires only a small dataset to train the model on, whilst still achieving relatively high precision of $90.5 - 97.3\%$. One upside of the proposed method is that the features used in their approach are language-independent.

As opposed to only using content on a webpage, a combination of both URL characteristics as well as page content can be used to achieve accurate phishing detection techniques [38]–[40]. CANTINA+ [38] is a neural net which apart from URL features, also extracts features from the content of the webpage such as log in forms and URLs on the page. The paper proposes a solution which includes using a sliding window of two weeks to ensure the model stays up to date. This solution nets them the same true positive rate as when using the full dataset of 92% but slightly increases their false positive rate (from 0.4% on their training data to 1.4% after two weeks). Although the false positive rate increased, this approach could be a good self learning method of staying up to date with how phishing attacks evolve. Although content based detection has been proven to be good at detecting malicious websites [41], its usefulness in early detection is questionable. Given the fact that many phishing campaigns are often short-lived [4], and time between the malicious website being deployed and it being used in a phishing campaign is short, content based detection systems only have a short window of time to detect potential phishing websites. In our early detection research we will not make use of content placed on websites, but are potentially able to use URL features used in these papers.

Several other more novel approaches to detect deployed phishing websites have been proposed, such as making use of favicons [42], CSS styling [43], host based information [44] and looking at URL shorteners [45], [46] achieving similar results to earlier mentioned content based detection systems, with a true positive rate of up to 99.5% in [42]. However, these detection mechanisms suffer from the same drawbacks as these content based systems, namely requiring the phishing website to already be deployed at time of analysis.

There also exists a field of research of detecting ongoing phishing campaigns by analyzing email content [47]–[50] and email metadata [51]. Although our research focusses on detection of phishing domains before content is placed on the webpage and the campaign has started, we can still leverage findings on URLs used in phishing emails. Furthermore our proposed models could be used in this research area to increase phishing detection by analyzing URLs present in emails.

4. Methodology

This section will detail the methodology used in our research. A full overview of our approach is shown in Figure 1.

4.1. Data collection

4.1.1. PhishDetect dataset. Our dataset of phishing domains is from PhishDetect [3]. PhishDetect checks incoming domains from the Certificate Transparency Log network stream [52] as well as lists of newly registered domains given by registrars. The entire dataset contains 374,972 domains that were deemed as “suspicious” by PhishDetect in the period between 14-11-2017 and 28-06-2019. The fields in the dataset are described in Table 1. Each entry has a value for “domain” and “score”, all non required fields can be left empty.

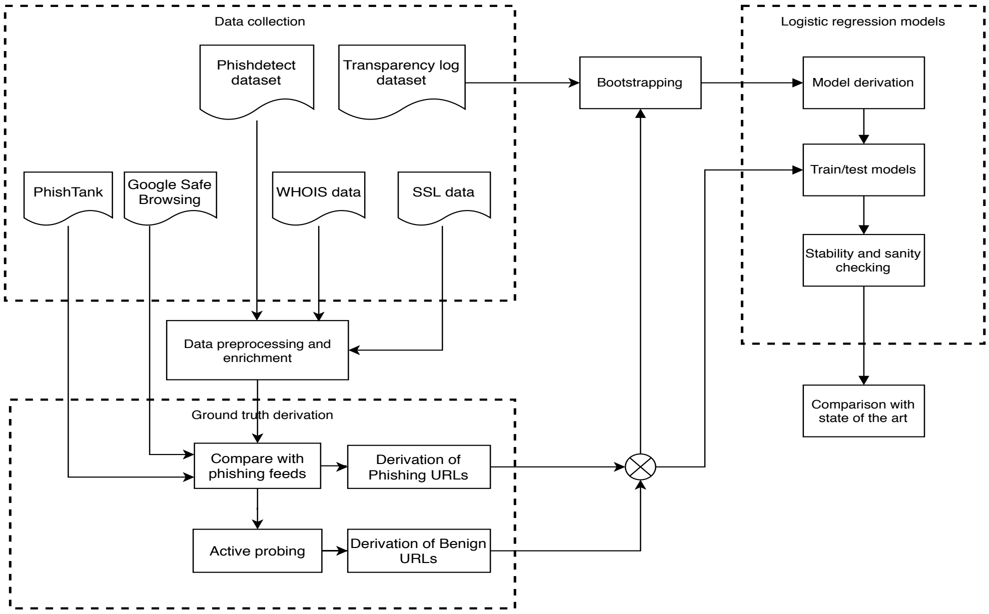

Data collection
Phishdetect dataset
Transparency log dataset
PhishTank
Google Safe Browsing
WHOIS data
SSL data
Data preprocessing and enrichment
Ground truth derivation
Compare with phishing feeds
Derivation of Phishing URLs
Active probing
Derivation of Benign URLs
Bootstrapping
Logistic regression models
Model derivation
Train/test models
Stability and sanity checking
Comparison with state of the art

Figure 1: General overview of construction of datasets and their usage.

TABLE 1: Fields in PhishDetect dataset

Column name	**Required**	**Value**
domain	yes	The full domain name including subdomain.
score	yes	The suspicousness score PhishDetect assigned to this domain.
datetime	no	The date and time the suspicious website was detected and stored in the database.
resolutions	no	Array of IPs belonging to the domain along with the date this IP was resolved.
certificates	no	Array of SSL certificates belonging to the domain, along with the date the certificate was found.
brand	no	The brand that is targeted by this domain.
warnings	no	Features of the domain PhishDetect deemed suspicious.

TABLE 2: Warnings produced by PhishDetect

Warning	Condition
suspiciouswords	The domain contains a word present in a brand name blacklist or two words have Levenshtein distance of less than two compared to the list of suspicious words.
dotsdashes	The combined number of dots and dashes in a domain is higher or equal to 4.
suspiciouspattern	The string ".com-" or ".org-" is present in the domain.
suspicioustld	The TLD of the website is in the list of predefined suspicious TLDs.
nohttps	The website does not make use of HTTPS.
mimicsbrand	The domain mimics a known brand using *homographs*.

The "score" field is a custom score PhishDetect assigns based on predefined metrics. It looks at certain risk factors of a domain such as suspicious words in the domain, word similarity to existing brands, as well as checks for suspicious TLDs. For every risk indicator, a predefined value gets added to the score, depending on the weight assigned to that risk indicator. A full overview of warnings produced by PhishDetect can be found in Table 2.

4.1.2. Additional data collection on PhishDetect dataset. Since the initial PhishDetect dataset does not contain all the information we need in our experiments, we need to extract further information from the domains which then could be used in features for our models.

SSL Data. The PhishDetect dataset contains certificates of a domain in DER [53] format. Using the Python package "pyopenssl" [54] we can parse these certificates and extract their properties.

WHOIS data. Since the original PhishDetect dataset does not contain WHOIS information about a domain, we built our own resolver using the pythonwhois[6] library. One of the limitations of the PhishDetect dataset is that since the collection started over two years ago, many of these domains will already have expired by the time we look up the WHOIS information. Unfortunately, we were not able to find and get access to a WHOIS database that contained domains which have already expired. This means that, for our WHOIS analysis, we are limited to websites which are still registered and whose registrars support WHOIS queries.

4.1.3. PhishTank and Google Safe Browsing. In order to verify the effectiveness of PhishDetect, as well as helping us create a clear list of true and false positives from

[6] http://cryto.net/pythonwhois/ Last visited on 04-2020

the PhishDetect dataset, we need a secondary dataset of known phishing domains to compare suspicious PhishDetect domains with. For this set, we make use of PhishTank and Google Safe Browsing.

PhishTank collection. PhishTank, as mentioned in 2.4, only stores currently active phishing websites. This means that any phishing website which is no longer active or has been taken down is removed from the PhishTank feeds. However, online repositories exist which store a daily list of phishing URLs present in PhishTank, such as [55]. We were able to create a dataset of all verified phishing URLs from PhishTank between 18-08-2017 to 26-10-2019. This dataset consists of $364,350$ unique phishing URLs along with the time and date these URLs were verified to be a phishing website by PhishTank users. Out of these phishing URLs, there are $126,858$ unique domains.

Google Safe Browsing. In order to query Google Safe Browsing we make use of its Update API [56] using the gglsbl Python package [57]. This allows us to store partial hashes of URLs deemed malicious by Google Safe Browsing, which can then be queried using gglsbl. We do not have exact numbers on number of domains present in this database, as it only stores partial hashes, and only when a hash is found, gglsbl will do a further lookup using the Google Safe Browsing Lookup API [56].

4.1.4. Transparency log dataset. Since our PhishDetect dataset limits our visibility to domains that are already deemed 'suspicious' by the platform, we consider a secondary data source to identify newly registered domains regardless of their 'suspiciousness'. In order to create our secondary list of legitimate domains, we opted to look at the certificate transparency logs. We can download a large set of SSL certificates quickly using this method. In total, we retrieved $219,121,968$ SSL certificates from the Google Rocketeer Log[7]. These were all the SSL certificates added to this transparency log from 02-02-2018 to 12-10-2018. Since we are dealing with such a big dataset, it is unfeasible to WHOIS every single entry in our dataset. For this reason, a random subset from this log was extracted. This was done by randomly selecting certificates from the log and discarding any entries for which no WHOIS data was found; to remove domains that have resulted in phishing, we discard domains which are found in PhishTank or Google Safe Browsing. We end up with a dataset of $31,315$ entries for our secondary legitimate domain dataset, that we call the `Transparency log dataset`. We decided to use this strategy as opposed to using lists of 'top' domains (e.g. most visited, such as Alexa's list) used in earlier work [5], [6] for numerous reasons. The primary reason being that using this technique, we already have SSL data for our legitimate domains and do not need to iterate over a list of domains to retrieve certificates. A secondary reason is that the origin of domains on top lists are often unknown, and can be manipulated as shown by Scheitle et al [7].

4.2. Data enrichment

4.2.1. SSL Data. From the parsed SSL certificates found in PhishDetect, we extract every field present in a certificate and store additional information such as the number of domains present in the certificate and the number of characters provided in user filled text fields.

4.2.2. WHOIS data. We parse the acquired WHOIS data belonging to a domain, and extract every WHOIS property present in the response. Additionally we accumulate extra information based on this response such as the registration length of a domain (by looking at creation and expiration date) and counting the number of nameservers and statuses present in the WHOIS response.

4.3. Ground truth derivation

4.3.1. Extracting phishing URLs from PhishDetect. To verify if a domain in our PhishDetect dataset has at some point been used for phishing or not, we do a lookup in our PhishTank dataset. If the queried domain is not found in our PhishTank set, a Google Safe Browsing lookup is done. This check is done for every domain in our PhishDetect dataset. Our dataset of phishing domains consists of all domains present in our PhishDetect dataset which were also marked as phishing by either PhishTank or Google Safe Browsing.

4.3.2. Extracting benign URLs from PhishDetect. To get information regarding false positives from the PhishDetect dataset, we resolve every domain *not* marked as phishing, and check if they are still online. From the websites that are still online we check the returned HTTP status code. If this status code is valid (i.e. the status code is in the range 200-299), we check if the webpage body contains at least 500 characters (including HTML and javascript code) to ensure there is at least some content on the webpage. We employ a set of simple regular expressions to remove domains pointing to default pages (e.g., Apache's), 404 pages with wrong status codes as well as domain parking messages. Any domain remaining after this filtering process is marked as being likely legitimate. We expect these domains to be likely to be legitimate as the newest entry to our PhishDetect dataset at time of resolving is 136 days old. Earlier research has shown that phishing websites are short-lived, with [4] reporting that only 27.7% of phishing domains in their dataset were still online after 48 hours and 90% of phishing domains were present in phishing feeds within 48 hours. Given the fact that the domain is not present in our PhishTank or Google Safe Browsing sets, the website has been online for an extended period of time ($\geq$ 136 days) and has content deployed, we expect these domains to have a high chance of being benign.

4.4. Bootstrapping

To find potentially useful features of domains reported in the collected SSL certificates and WHOIS data for our model, we make use of bootstrapping by randomly sampling, with replacement, from our `phishing`, `benign` and `transparency log` dataset. Apart from executing bootstrapping using the full datasets, we also run bootstrapping on subsets of our datasets with matching attributes. We, for example, create subsets with matching certificate issuer and domain registrars to see if there exist

[7] https://ct.googleapis.com/rocketeer Last visited on 01-2020

any distinguishing features for these issuers and registrars specifically. We only run bootstrapping on subsets derived from combinations of properties where at least 10 entries exist in each of the `phishing`, `benign` and `transparency log` dataset to ensure we have enough entries we can use for bootstrapping.

For our bootstrapping we sample, with replacement, $N = 10$ domains (the minimum number of entries in our subsets) from our two legitimate domain datasets and our phishing dataset as well as the created subsets for different combinations of registrars and certificate issuers. We repeat this process $M = 100000$ times for every property. This technique allows us to recreate the 'real' distribution of features without requiring to query the WHOIS database for all the 219 million domains captured by the transparency log dataset. To represent the distribution we report the mean of the measured feature across each sample.

4.5. Logistic regression models

4.5.1. Model derivation. In order to reach our goal of detecting phishing websites, we propose several logistic regression models whose parameters (i.e. explanatory variables) are derived from the results of our bootstrap analysis. When creating our models we check every attribute for its significance and confidence intervals. Logistic regression was chosen since we are dealing with a classic binary classification problem, allowing us to quantify the probability of a website being used for phishing or not based on the selected features.

4.5.2. Training and evaluation criteria. *Classification* In order to test the effectiveness of our proposed solution, we need to measure the accuracy of our system. The following terminology is used throughout the paper:

- **True Positive (TP)** A domain marked as phishing by the system is an actual phishing domain.
- **True Negative (TN)** A domain marked as legitimate by the system is an actual legitimate domain.
- **False Positive (FP)** A domain marked as phishing by the system is actually a legitimate domain.
- **False Negative (FN)** A domain marked as legitimate by the system is actually a phishing domain.

Using the above definitions, we can determine the following metrics, where TP, TN, FP, FN denotes the amount of true positives, true negatives, false positives and false negative respectively:

$$\begin{aligned}
TruePositiveRate(TPR) &= \frac{TP}{TP+FN}\\
TrueNegativeRate(TNR) &= \frac{TN}{TN+FP}\\
FalsePositiveRate(FPR) &= \frac{FP}{FP+TN}\\
FalseNegativeRate(FNR) &= \frac{FN}{FN+TP}\\
Accuracy(ACC) &= \frac{TP+TN}{TP+TN+FP+FN}\\
Precision &= \frac{TP}{TP+FP}\\
Specificity &= \frac{TN}{TN+FP}\\
Recall(Sensitivity) &= \frac{TP}{TP+FN}
\end{aligned}$$

Model stability. Further, we are interested in determining if the weights assigned to features of phishing and legitimate domains for the classification change over time. Therefore we train and evaluate our models based on various intervals to test the model's performance and stability. We test multiple parameters for different training intervals to see how this impacts the performance of the model. This also allows us to continuously update the cutoff value of the models used for making the final classification if they change over time.

4.6. Comparison with state of the art

In order to test the overal performance of our model we benchmark our model against [6] and [5]. These papers were published in 2009 and 2015 respectively and so the question arises if the approach taken in these papers still hold today. We are also interested if these approaches are effective when using our dataset, as we have taken a different approach to collecting legitimate domains as opposed to using top lists which were used in these papers (a sometimes problematic approach, as these top lists are prone to manipulation [8] and contain biases [7]). We have reimplemented the proposed solutions and models from these papers and ran it on our PhishDetect dataset to check whether the original results could be replicated in our setup.

5. Data analysis

5.1. Ground truth derivation

5.1.1. Extracting phishing URLs from PhishDetect. From our original PhishDetect dataset, $10,354$ entries were found in either PhishTank or Google Safe Browsing. After removing duplicate domains, we end up with $3,127$ unique known phishing domains from the PhishDetect dataset.[8]

[8] During our exploratory analysis, we ran into one particular phishing campaign in the PhishDetect dataset consisting of $1,264$ domains which all used the same characteristics. These domains consisted of exactly 10 seemingly random characters and used either the .top or .site TLD. In order to not skew our models these domains were removed from our dataset.

TABLE 3: Number of usable entries extracted from PhishDetect dataset after ground truth derivation. “WHOIS” dataset is subset of “Full” but only contains entries with available WHOIS information.

Dataset	Full	WHOIS
Benign	2597	1515
Phishing	3127	1207
Total	5724	2722

TABLE 4: Percentage of user submitted fields in SSL certificate not left empty in our datasets.

	Benign	Phishing
Organization	1.69%	0.83%
Organizational unit	6.05%	1.92%
Locality	1.69%	0.83%
State	1.54%	0.83%
Country	2.54%	0.86%
Email[9]	0%	0%

5.1.2. Extracting benign URLs from PhishDetect. Out of all requested websites in our PhishDetect dataset that were not known phishing domains, $21,260$ domains were still reachable and returned a valid status code in the 200-299 range. After removing all websites which contain a word in our blacklist, we are left with $5,104$ domains which we assume have a high chance of being legitimate. From this set of likely legitimate domains we remove entries with duplicate second and top level domain combinations. In the case of a duplicate, we keep the earliest entry, as this research is focussed on detecting phishing domains as early as possible. After removing duplicates, $2,597$ entries remain which we can use in our legitimate dataset. At time of our resolving, the newest entry in our PhishDetect dataset was 136 days old. Analysis on our known phishing domains shows that 80% of known phishing domains were detected by PhishTank or Google Safe Browsing within 9 days after being added to PhishDetect. This number increases to 90% within 56 days and 94.6% within 136 days. [4] states that 90% of phishing campaigns were present in phishing feeds within two days of inception. Given the fact that the websites we use for our benign domain dataset were online for a sufficient amount of time, and were not present in any phishing feeds, we assume these domains are very likely not used for phishing, we mark these as benign.

5.1.3. Analysis on WHOIS information. Table 3 shows the number of entries remaining from the original PhishDetect dataset after our ground truth derivation. Our dataset consists of a total of $5,724$ usable entries with SSL data, of which $2,722$ returned a valid WHOIS response. Our dataset is fairly balanced. In our full dataset we have more phishing domains than legitimate domains. However, we were able to WHOIS more legitimate websites, as many phishing websites in our dataset were already taken down or had expired at the time of the WHOIS request. Even though every domain in our legitimate set was online at the time of the WHOIS query, a valid WHOIS response was returned for only 58.3% of legitimate domains.

TABLE 5: Distribution of warnings produced by PhishDetect on the phishing and benign dataset. Percentage of total shown in brackets.

PhishDetect warning	Benign	Phishing
suspiciouswords	2276 (87.64%)	2506 (80.14%)
dotsdashes	487 (18.75%)	2351 (75.18%)
suspiciouspattern	298 (11.47%)	906 (28.97%)
suspicioustld	141 (5.43%)	588 (18.8%)
nohttps	2 (0.07%)	1 (0.03%)
mimicsbrand	2 (0.07%)	0 (0.00%)

5.1.4. Analysis on user submitted SSL information. We mentioned earlier that most WHOIS data is now shielded to protect the privacy of the registrant. However, SSL certificates still allow the user to enter information such as organization, organizational unit, locality and state which will show up in the SSL certificate. Features used in [6] and [5] rely on these fields as input for their models, therefore we conducted our own analysis on the importance of these fields.

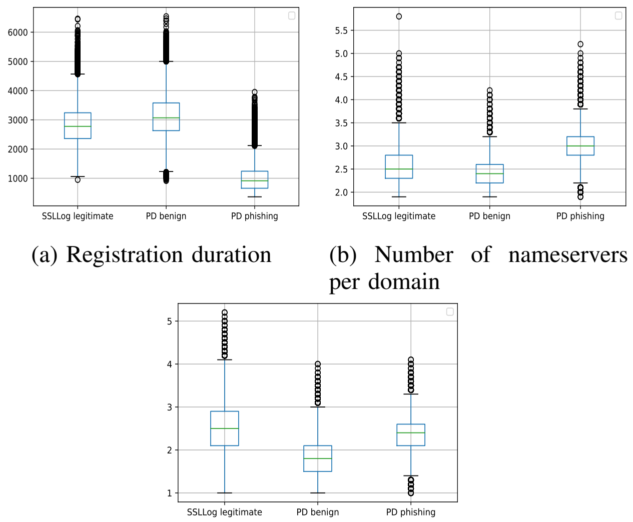
Table 4 contains an overview of user submitted fields not left empty in our `benign` and `phishing` dataset. It shows that user submitted fields in SSL certificates are often left empty. Eventhough legitimate websites are slightly more likely to not leave these fields empty, this difference is not significant enough to justify using a boolean feature whether this data is present or not in our models. Thus we conclude that the existence of values in these properties does not help identify potential malicious websites. Furthermore, the value of these properties can be freely chosen by the person registering the certificate in the case of Domain Validated certificates. Should any of these properties contain indicators increasing the likelihood of a website being phishing, phishing domain owners can easily circumvent any detection methods using these fields by filling out different information.

5.1.5. Analysis on PhishDetect warnings. Table 5 shows the distribution of PhishDetect warnings on the datasets produced by our ground truth derivation. It can be observed that the `nohttps` and `mimicsbrand` warnings are barely produced and will not be reliable features for our model. The `dotsdashes`, `suspiciouspattern` and `suspicioustld` warnings show a significant difference between phishing and benign domains, with 75.2% of phishing domains producing the `dotsdashes` warning as opposed to only 18.75% for benign domains. The `suspiciouswords` warning is the most common warning for both the phishing and benign domains. Indicating that this warning is most likely responsible for most false positives produced by PhishDetect.

5.2. Bootstrap analysis

In order to identify features for our models, we need to first analyze and detect any potential differences between metadata of legitimate and phishing websites. Using bootstrapping we can find any potential properties that indicate a higher likelihood of a website being a phishing domain. This paragraph is dedicated to findings during our bootstrapping experiments which could be of use for our phishing models. We also mention and describe

[9] Email field is deprecated but still supported by some certificate issuers.



(a) Registration duration

(b) Number of nameservers per domain

(c) Number of EPP statuses per domain

Figure 2: Bootstrapped WHOIS data of the Transparency log (SSLLog legitimate), benign (PD benign) and phishing (PD phishing) datasets.

several investigated properties which show no indication of maliciousness.

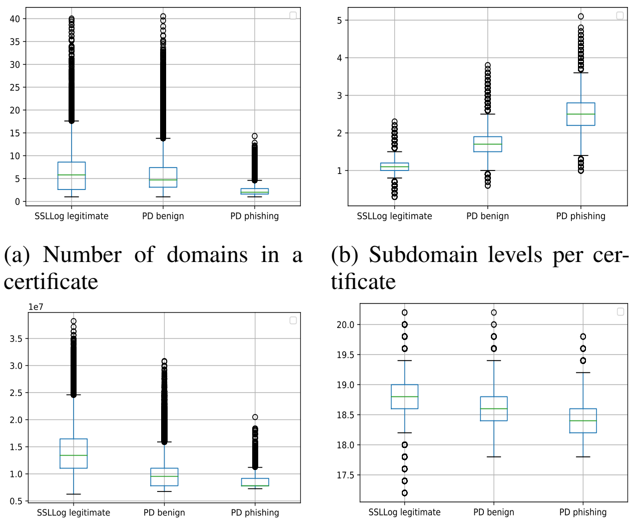
5.2.1. WHOIS data.

Registration duration We define the registration age of a domain as $expiration\_date - creation\_date$, where $expiration\_date$ and $creation\_date$ are fields returned by the WHOIS query. It can be observed that when comparing the bootstrapped results of the registration durations (Figure 2a), the average registration age of a phishing domain is significantly lower than those of legitimate websites. This is in line with our initial assumptions, as we assumed phishing websites to be short-lived, as opposed to legitimate websites which, on average, exist longer. Furthermore, legitimate websites are expected to be more likely to register a domain name for a longer period of time, for instance for 5 years at a time, whereas phishers might only register a domain for one year in order to save costs.

We also looked at other WHOIS data, such as number of EPP status codes[10] and number of nameservers assigned to domains, there were no clear differences between phishing and legitimate websites. Figure 2b and 2c show the bootstrapped number of nameservers and number of EPP statuses associated with a domain. Although the mean number of nameservers is 0.5 higher between the `phishing` dataset and the `Transparency log` dataset, this difference is not statistically significant and cannot therefore be reliably used as a feature for our classification models.

5.2.2. SSL certificate data.

Number of subdomains. The bootstrapped number of subdomains of the domain with the deepest subdomain per certificate is significantly higher in phishing domains compared to legitimate domains as can be seen in Figure 3b, which makes this



(a) Number of domains in a certificate

(b) Subdomain levels per certificate

(c) Certificates validity (duration)

(d) Number of extensions in SSL certificates

Figure 3: Bootstrapped SSL data of the Transparency log (SSLLog legitimate), benign (PD benign) and phishing (PD phishing) datasets.

property a good candidate for our models. The bootstrapping results show that the average number of subdomains for our samples from the `Transparency log` dataset is even lower than those from the `benign` dataset. This is expected as the initial PhishDetect dataset consists of domains which were deemed already slightly suspicious by PhishDetect as mentioned before. As a result, models using this property are assumed to be more effective on models deployed on real world situations as opposed to models tested against our test dataset. However, our bootstrapping has shown that for suspicious looking domains, as defined by PhishDetect, phishing domains still use more subdomains on average compared to benign domains.

Certificate validity length. The bootstrapping analysis has also shown the average validity length of a certificate belonging to a legitimate domain to be slightly higher on average (Figure 3c). This is highly influenced by the fact that most certificates for phishing websites were issued by free issuers (e.g. LetsEncrypt) which only allow for a certificate duration of 90 days, whereas certificates issued by other (paid) issuers can be valid for up to 825 days [11].

Other SSL certificate properties. We were unable to find any significant differences in other properties of SSL certificates. We looked at number of extensions present in SSL certificates, as [5] uses this as one of their features for their model and also makes use of 24 features related to the existence of common extensions. Figure 3d shows that the number of extensions present in SSL certificates does not heavily differentiate between benign and phishing domains. This is most likely because 96% of certificates issued to benign and phishing domains extracted from the PhishDetect dataset came from only three different issuers.

Properties specific to certain combinations of registrars and certificate issuers. We were unable to find any properties for specific combinations of certificate issuers

[10] https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en Last visited on 04-2020

[11] https://www.digicert.com/shortening-validity-periods-for-ov-dv-certificates/ Last visited on 04-2020

and domain registrars that showed an increase in phishing likelihood which are not already derived from the bootstrapping on the full datasets.

5.3. Logistic regression models

We identify a set of nine features for our model derived from the bootstrap analysis (feature no. 5 to no. 9) as well as warnings already produced by PhishDetect that were deemed significant by our analysis in 5.1.5 (feature no. 1 to no. 4).

5.3.1. Model features.

We here list the identified features.

Feature 1: ***dotsdashes*** Warning produced by PhishDetect when the combined number of dots and dashes in a domain is higher or equal to 4.

Feature 2: ***suspiciouswords*** Boolean property provided by PhishDetect. Set to true if the domain contains a suspicious word related to a brand defined in PhishDetect. This includes common misspellings as well as Internationalized Domain Name variations of brands. If a suspicious word is not found, it will calculate the Levenshtein distance between the domain and a list of known brand names as well as Levenshtein distance between the domain and a list of function words. If two matches are found where the Levenshtein distance is less than one, this boolean is true.

Feature 3: ***suspicioustld*** The TLD of the website is in the list of suspicious TLDs defined in PhishDetect. This list of TLDs is created based on certain factors. These are user familiarity (e.g. .com, .info), possibility to be misleading (e.g. .support, .tech, .bank) or TLDs which are free to register (e.g. .tk, .ml).

Feature 4: ***suspiciouspattern*** Set to true if the string ".com-" or ".org-" is present in the domain. This is used by phishers to confuse users into thinking (part of) the subdomain is the TLD. This feature could be extended with more TLDs, but were limited to the mentioned two as this was part of PhishDetects lexical analysis.

Feature 5: $\boldsymbol{num\_subdomains >= 3}$ $num\_subdomains$ is the number of subdomains in the domain with the deepest subdomain nesting. This boolean feature is set to true if the $num\_subdomains$ is higher than or equal to 3. This feature looks at any domain present in the "Common name" field as well as those in the "subjectAltName" extension. For example an SSL certificate containing the domains { login.to.account.example.com, account.example.com} has a value of 3 for $num\_subdomains$.

Feature 6: $\boldsymbol{num\_domains <= 2}$ Boolean feature indicating if the total amount of unique domains present in the SSL certificate is 2 or less. SSL certificates will always have at least one domain, which is required for the "Common name" field, but can also contain extra domains in the "subjectAltName" extension. Bootstrapping has shown that when comparing SSL certificates from phishing domains to those that are legitimate, certificates belonging to phishing domains often only have one domain present while legitimates often have more. This includes different subdomains as well as different second level domains. For example, for an SSL certificate where the domains are {twitter.example.com, facebook.example.com, google.com}, the number of domains is 3.

Feature 7: ***numdash*** The number of dashes (-) of the domain with the most dashes present in the SSL certificate. This feature was chosen as instead of subdomains as separators, phishers also make use of dashes as separators between words (e.g. log-in-to-your-account.com). This is the only non-boolean feature for our model.

Feature 8: $\boldsymbol{registration\_length < 368}$ The time between creation date and expiration date of WHOIS data as defined in 5.2.1 is less than 368 days.

Feature 9: $\boldsymbol{registration\_length < 1200}$ Using the same data as the previous property, but using a longer registration length could help identify more legitimate websites.

5.3.2. Model evaluation.

Table 6 shows the resulting logistic regression coefficients of the chosen features. Although most of our features rely on lexical properties of the domain, the SSL and WHOIS data are significant enough to make a more accurate assumption on the legitimacy of a domain. Low registration lengths have a significant impact on classification ($P < 0.001$). The number of domains in a certificate also helps classification ($P < 0.05$). When looking at lexical features in the more extensive models the `suspiciouspattern` warning produced by PhishDetect, number of subdomains and a high number of dots and dashes seem to be the most significant. The `suspiciouspattern` feature being significant is unsurprising, as a legitimate website is unlikely to deliberately use ".com-" or ".org-" in their domain, as it might result in making their domain look suspicious. For combinations of features, only the combination of the `dotsdashes` and `suspiciouswords` warnings produced by PhishDetect were deemed significant ($P < 0.001$ in Model 2, $P < 0.05$ in Model 3 and 4). The other impactful lexical features such as number of subdomains are also expected. Although there are legitimate use cases for benign domains to use a high number of dashes or subdomains in a URL, the number of phishing domains using these features greatly outweighs the legitimate domains thus making this a significant feature in our models.

In order to test the performance of our models we use repeated K-Fold cross-validation ($N_{splits} = 10$, $N_{repeats} = 3$) on our PhishDetect dataset. Since we do not have WHOIS information for all domains in our set, we evaluate the best performing models over both the dataset with and without WHOIS information. The results are shown in Table 7. When comparing Model 5, which is the model built using data available via only SSL certificates we get $79.2\% - 79.7\%$ accuracy on our datasets. Appending domain registration length information to this model (Model 7) we increase the performance to 86.3% accuracy with sensitivity and specificity values of 86.6% and 85.8% respectively.

5.3.3. Stability checking.

Sliding window approach. One question we have is if we could improve the performance of our model if we only use data from N previous months instead of using the entire dataset. Phishers might adapt their strategy in order to circumvent existing detection methods. By using a sliding window approach over our

TABLE 6: Logistic regression results of our models on PhishDetect dataset with WHOIS data. McFadden Pseudo-R^2 is reported with relation to baseline model.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
(Intercept)	-3.64^{***}	-3.14^{***}	-3.26^{***}	-3.61^{***}	-3.76^{***}	-4.09^{***}	-4.34^{***}
	(0.40)	(0.42)	(0.43)	(0.43)	(0.44)	(0.46)	(0.46)
dotsdashes	2.41^{***}	1.66^{***}	1.28^{***}	1.16^{***}	1.00^{***}	0.88^{**}	0.88^{**}
	(0.10)	(0.22)	(0.23)	(0.23)	(0.23)	(0.28)	(0.29)
suspiciouspattern	2.39^{***}	2.34^{***}	2.41^{***}	2.35^{***}	2.28^{***}	1.67^{***}	1.48^{***}
	(0.39)	(0.39)	(0.40)	(0.40)	(0.40)	(0.40)	(0.41)
suspicioustld	0.86^{***}	0.81^{***}	0.80^{***}	0.83^{***}	0.84^{***}	0.74^{***}	0.48^{*}
	(0.19)	(0.19)	(0.19)	(0.19)	(0.19)	(0.22)	(0.23)
suspiciouswords	2.13^{***}	1.56^{***}	1.57^{***}	1.40^{**}	1.44^{***}	1.24^{**}	1.11^{*}
	(0.40)	(0.42)	(0.43)	(0.43)	(0.43)	(0.45)	(0.46)
dotsdashes & suspiciouswords		0.90^{***}	0.69^{**}	0.79^{**}	0.65^{*}	0.56	0.54
		(0.24)	(0.25)	(0.25)	(0.26)	(0.30)	(0.31)
num_subdomains >= 3			1.23^{***}	1.23^{***}	1.40^{***}	1.49^{***}	1.53^{***}
			(0.12)	(0.12)	(0.13)	(0.15)	(0.16)
num_domains <= 2				0.69^{***}	0.70^{***}	0.50^{***}	0.34^{*}
				(0.11)	(0.12)	(0.13)	(0.14)
numdash					0.19^{***}	0.18^{**}	0.18^{**}
					(0.05)	(0.06)	(0.06)
registration_length $<$ 368						2.77^{***}	1.82^{***}
						(0.13)	(0.16)
registration_length $<$ 1200							1.49^{***}
							(0.14)
Pseudo R^2	0.26	0.26	0.29	0.30	0.31	0.46	0.48
AIC	2774.05	2763.13	2660.49	2624.69	2613.93	2056.54	1947.95
Deviance	2764.05	2751.13	2646.49	2608.69	2595.93	2036.54	1925.95
Num. obs.	2722	2722	2722	2722	2722	2722	2722

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

TABLE 7: Performance of two best performing models using repeated K-Fold cross-validation ($N_{splits} = 10$, $N_{repeats} = 3$). Model 5 uses features based domain name and SSL certificate. Model 7 contains features based on SSL certificate data and WHOIS properties. 95% confidence interval for accuracy is added in brackets.

	Sensitivity		Specificity		Accuracy	
Model	(Full)%	(WHOIS)%	(Full)%	(WHOIS)%	(Full)%	(WHOIS)%
Model 5	0.784	0.835	0.797	0.750	0.792 (0.7858, 0.798)	0.797 (0.7881, 0.8056)
Model 7	N/A	0.866	N/A	0.858	N/A	0.8625 (0.8548, 0.8699)

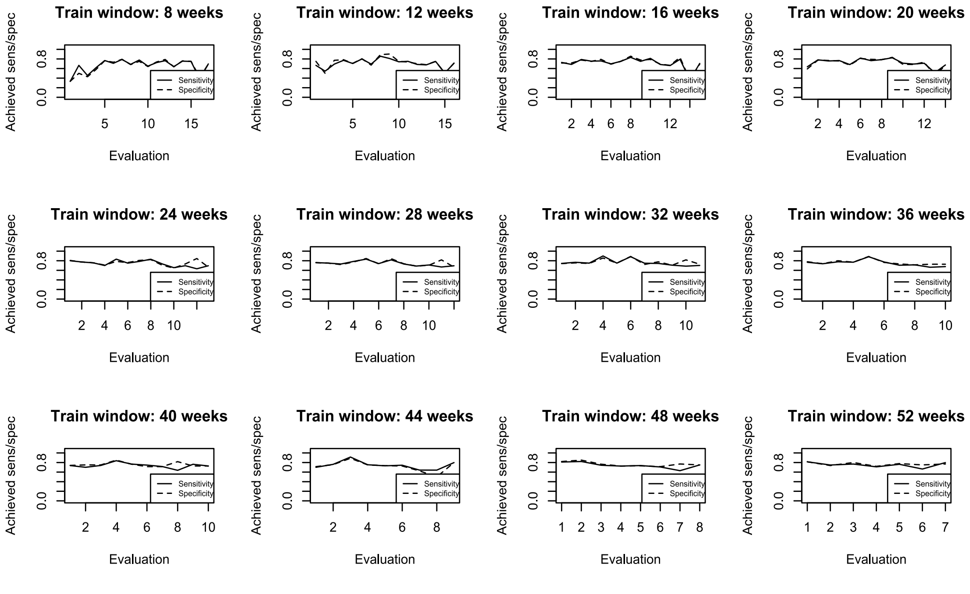


Figure 4: Achieved sensitivity and specificity using different length training windows using the best available model (Model 7) on the WHOIS dataset.

data, we may be able to make our model more resistant against such changes by recalculating the weights assigned to features. Since the original PhishDetect dataset stores timestamps for every entry that is added, we can simulate a real time detection system to test our model. When selecting our model we optimize for both sensitivity and specificity, using the intersection point of the two as the cutoff for our predicted logistic regression model. Thus for the ideal cutoff point of our logistic regression model result we pick the point where $abs(sensitivity - specificity)$ is closest to 0. Using the sliding window approach we can also adjust the cutoff point for our model in case this needs to be updated over time to achieve a higher model performance.

Sliding window performance. Figure 4 shows the achieved sensitivity and specificity using different training windows. This experiment was done by training the model for N weeks, and evaluating on the next 4 weeks after which the window was moved by 4 weeks. This was repeated until the evaluation start date exceeded the date of our last sample. The figure shows that results of the model are quite volatile when using short windows for our training data (from 8 to 20 weeks). This is most likely due to the number of samples being too low to create an accurate logistic regression model.

The performance of the model seems to stabilize when using a sliding window of size $24 - 36$ weeks. When training on 8 weeks of data using our PhishDetect dataset using the best performant model with WHOIS features (Model 7), the number of usable training samples varied greatly (between $14-810$). When using a sliding window of 20 weeks the number of training samples per window is much higher on average $(97-1448)$, increasing to $741 - 1565$ when using a window of 32 weeks. When using this window of 32 weeks, we achieve an accuracy of $69.7\% - 88.6\%$ depending on the evaluation month. Because of this, we believe our models and the data we have perform best when using the earlier mentioned window size of $24-36$ weeks, but this may vary depending on the number of samples available in a given time window.

Determining logistic regression cutoff value. One of the problems that arose during the research into using the sliding window approach is that the value of the result of our logistic regression model varied when picking the ideal cutoff point to maximize sensitivity and specificity. We optimize for both sensitivity and specificity, but the ideal threshold to mark a domain as phishing may vary depending on the training data the model is fed. When using the full dataset one can pick the ideal cutoff based on predictions after the regression is done on validation data. If we wish to use our model in real-time using our sliding window we must pick a threshold to use during real-time classification, based on the previous months of data. By dividing the training window into 10 folds of equal size, we train our model on 90% of the data, and test it on the remaining 10%, repeated for every fold, selecting the mean cutoff from these regressions as our threshold to use in real-time detection. Using this method on our sliding window of 32 weeks on the full dataset without WHOIS information, we achieve an average sensitivity of 75.2%, an average specificity of 76.7% and an average accuracy of 75.6%. These results seem promising, but are subject to variance, with sensitivity ranging from $67\% - 85.6\%$ and specificity ranging from $68.2\% - 81.3\%$.

6. Comparison with state of the art

Since the performance of our models is lower than earlier claimed studies which also looked at domain names and their SSL certificates, we wanted to find out the reasoning behind it. There are several differences in the way our research was conducted as opposed to [6] and [5]. These papers used top site lists for their legitimate datasets, whereas we decided to use the PhishDetect dataset. For our implementations, we used the scikit-learn[12] library and implemented every feature used in their models still applicable with current SSL certificates. Validation is done using stratified K-Fold cross-validation with $N_{splits} = 10$.

In our experiments, we were not able to reproduce the claimed results using our dataset. Table 8 shows the results claimed by “Harvesting SSL Certificate Data” [6] compared to the results of their proposed models on our PhishDetect dataset. It can be noticed that the negative recall results we obtained on our dataset are significantly lower than claimed (0.78 in the original paper as opposed to 0.16 on our dataset). These results can be explained by looking at the features used to feed their models. Many of their chosen features rely on user provided SSL fields such as the organization and country fields, which depending on the field is only filled in $0.83\% - 1.92\%$ of the time for phishing websites, and $1.54\% - 6.05\%$ of the time for benign websites according to our analysis (Table 4). Furthermore, one feature relies on marking fields in the subject such as the `organization name` and `issuer country` as “bogus” but do not provide a list of values deemed to be bogus apart from three examples. After manually checking our dataset, we could not find any values in subject fields which we would deem “bogus” thus this feature was removed from the model. The feature indicating whether a certificate was self-signed or not was also removed, as none of the certificates in our dataset were self-signed. The only features remaining which possibly have an impact on classification are validity duration of the certificate (deemed not significant by our research) and features related to the “common name” field. Although the positive recall on our dataset is high for most algorithms used (0.98), the negative recall is very low (0.16). All results combined, we expect this model to no longer be viable in modern day situations.

“Beyond the Lock Icon” [5] makes use of 42 features to feed into the model. A full comparison of claimed results compared to their proposed models used on our dataset can be found in Table 9. The majority of the features used in their proposed models rely on extensions present in SSL certificates as well as used algorithms by the certificate and the certificate version. The extensions and chosen algorithms are often the same for every certificate issued by the same issuer, which makes these features less useful, as most of the certificates in our dataset were issued by the same three entities (Let’s Encrypt $4,104$, cPanel $1,250$ and COMODO CA 143, 96% of all certificates issued). Just like [6], it also makes use of user provided fields such as organization and country which

[12] https://scikit-learn.org/stable/ Last visited on 04-2020

TABLE 8: Claimed results (Original) of “Harvesting SSL Certificate” [6] compared with results of the same models tested on our PhishDetect (PD) dataset. All results are done using Stratisfied K-Fold with $N_{splits} = 10$.

	Positive recall		Positive precision		Negative recall		Negative precision	
Classifier	(Original)%	(PD)%	(Original)%	(PD)%	(Original)%	(PD)%	(Original)%	(PD)%
Random Forest	0.94	0.98	0.88	0.66	0.778	0.16	0.881	0.85
Decision Tree	0.939	0.98	0.881	0.66	0.779	0.16	0.88	0.85
Bagging - Decision Tree	0.935	0.98	0.877	0.66	0.773	0.16	0.874	0.85
Boosting - Decision Tree	0.94	0.98	0.881	0.66	0.78	0.16	0.882	0.85
Nearest Neighbour	0.94	0.02	0.879	0.76	0.774	0.99	0.882	0.38

TABLE 9: Claimed results (Original) of “Beyond the Lock Icon” [5] compared with results of the same models tested on our PhishDetect (PD) dataset. All results are done using Stratisfied K-Fold with $N_{splits} = 10$.

	Phishing recall		Phishing precision		Non-Phishing recall		Non-Phishing precision	
Classifier	(Original)%	(PD)%	(Original)%	(PD)%	(Original)%	(PD)%	(Original)%	(PD)%
Random Forest	0.937	0.97	0.955	0.66	0.963	0.19	0.947	0.82
K-Nearest Neighors	0.936	0.07	0.953	0.56	0.961	0.9	0.947	0.37
Decision Table	0.85	0.97	0.926	0.66	0.942	0.19	0.882	0.82
Naive Bayes Tree	0.905	0.94	0.909	0.66	0.942	0.20	0.92	0.68
Simple logistic	0.873	0.97	0.936	0.66	0.738	0.18	0.738	0.77

do not seem very relevant in detecting phishing anymore. As shown in Table 9, similarly to [6], the non-phishing recall is very low on our dataset ($0.18 - 0.20$), making this model not very useful in current day situations.

After evaluating our logistic regression models results against these papers, it seems that the data available in SSL certificates is less usable than previous years since the landscape of SSL has changed rapidly over the years. SSL certificates are freely available and the field is dominated by a few issuers issuing the majority of certificates. A lot of the SSL certificate registration is automated by services such as CertBot [13] making features used in previous research less valuable when it comes to detecting phishing domains.

7. Conclusion

In this work we examined the provided PhishDetect dataset and extracted its true and false positives. We have shown that using data available in SSL certificates and WHOIS data help increase detection of malicious websites. Eventhough many SSL certificates are issued by only few distinct entities, causing many properties to be the same among certificates, some useful properties can still be extracted beyond just the original domain name of a website.

Although our proposed models cannot be used as a standalone solution, as an accuracy of $79.2\% - 86\%$ still presents many false positives when spanning the entire internet, they can still be used to strengthen existing detection methods. Since our proposed models do not require the content of a website as input, classification is fast, can be done shortly after registration and can be used on a large scale. The results of our research could be used as a first step in filtering suspicious domains from SSL certificate logs, or as a browser extension to warn users in realtime. We have also shown that earlier work is less relevant in current-day situations, because of an updated threat landscape.

7.1. Future work

There are several ways we believe our work could be improved. Eventhough we take certain lexical features of domains into account, a more thorough look into the lexical properties of domain names could potentially increase the detection rate of phishing domains. Since phishing websites are short-lived, one could also make use of entire Certificate Transparency logs. A feature indicating whether a domain has requested an SSL certificate in the past could also be useful.

Acknowledgement

I would like to thank Luca Allodi and Pavlo Burda for supervising and providing valuable feedback during the execution of the project. I would also like to express my gratitude to Claudio Guarnieri for providing the initial PhishDetect dataset used in the thesis.

References

[1] “Internet crime complaint center (ic3) — business e-mail compromise e-mail account compromise the 5 billion dollar scam.” https://www.ic3.gov/media/2017/170504.aspx, 12 2016.

[2] “Phishing attacks using third-party applications against egyptian civil society organizations.” https://www.amnesty.org/en/latest/research/2019/03/phishing-attacks-using-third-party-applications-against-egyptian-civil-society-organizations/, 03 2019.

[3] C. Guarnieri, “Phishdetect homepage.” https://phishdetect.io/.

[4] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” 01 2009.

[5] Z. Dong, A. Kapadia, J. Blythe, and L. J. Camp, “Beyond the lock icon: real-time detection of phishing websites using public key certificates,” in *2015 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–12, May 2015.

[13] https://certbot.eff.org/ Last visited on 04-2020

[6] M. A. Mishari, E. D. Cristofaro, K. M. E. Defrawy, and G. Tsudik, "Harvesting SSL certificate data to mitigate web-fraud," *CoRR*, vol. abs/0909.3688, 2009.

[7] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A long way to the top," *Proceedings of the Internet Measurement Conference 2018 on - IMC '18*, 2018.

[8] V. L. Pochat, T. van Goethem, and W. Joosen, "Rigging research results by manipulating top websites rankings," *CoRR*, vol. abs/1806.01156, 2018.

[9] T. Dam, L. D. Klausner, D. Buhov, and S. Schrittwieser, "Large-scale analysis of pop-up scam on typosquatting urls," *CoRR*, vol. abs/1906.10762, 2019.

[10] E. S. Aung, C. T. Zan, and H. Yamana, "A survey of url-based phishing detection," 2019.

[11] ICANN, "Icann bringing the languages of the world to the global internet — fast track process for internationalized domain names launches nov 16." Blog, Oct. 2009.

[12] "Icann website." https://www.icann.org/.

[13] I. Board, "Temporary specification for gtld registration data." https://www.icann.org/en/system/files/files/gtld-registration-data-temp-spec-17may18-en.pdf, May 2018.

[14] B. Laurie, A. Langley, and E. Kasper, "Certificate transparency," RFC 6962, June 2013.

[15] "Let's encrypt - free ssl/tls certificates." https://letsencrypt.org/.

[16] "Ssl for free - free ssl certificates in minutes." https://www.sslforfree.com/.

[17] "Manage warnings about unsafe sites — google chrome help." https://support.google.com/chrome/answer/99020?hl=en.

[18] "How does built-in phishing and malware protection work? — firefox help." https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work.

[19] "Google safe browsing faq." https://www.google.com/safebrowsing/static/faq.html.

[20] A. M. Stephan Somogyi, "Safe browsing: Protecting more than 3 billion devices worldwide, automatically." https://www.blog.google/technology/safety-security/safe-browsing-protecting-more-3-billion-devices-worldwide-automatically/, September 2017.

[21] "Google safe browsing - google transparency report." https://transparencyreport.google.com/safe-browsing/overview?hl=en&unsafe=dataset:0;series:malwareDetected,phishingDetected;start:1546214400000;end:1587279600000&lu=unsafe.

[22] S. Marchal, J. François, R. State, and T. Engel, "Proactive discovery of phishing related domain names," in *Research in Attacks, Intrusions, and Defenses* (D. Balzarotti, S. J. Stolfo, and M. Cova, eds.), (Berlin, Heidelberg), pp. 190–209, Springer Berlin Heidelberg, 2012.

[23] H. Bo, W. Wei, W. Liming, G. Guanggang, X. Yali, L. Xiaodong, and M. Wei, "A hybrid system to find fight phishing attacks actively," in *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 506–509, 2011.

[24] S. Marchal, J. François, R. State, and T. Engel, "Phishstorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014.

[25] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing urls using recurrent neural networks," in *2017 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–8, April 2017.

[26] M. Dhiman, M. Jakobsson, and T.-F. Yen, "Breaking and fixing content-based filtering," pp. 52–56, 04 2017.

[27] Y. Elsayed and A. Shosha, "Large scale detection of idn domain name masquerading," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–11, May 2018.

[28] E. Lin, S. Greenberg, E. Trotter, D. Ma, and J. Aycock, "Does domain highlighting help people identify phishing sites?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), p. 2075–2084, Association for Computing Machinery, 2011.

[29] M. Jakobsson, A. Tsow, A. Shah, E. Blevis, and Y.-k. Lim, "What instills trust? a qualitative study of phishing," vol. 4886, pp. 356–361, 02 2007.

[30] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Commun. ACM*, vol. 50, pp. 94–100, 10 2007.

[31] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *CoRR*, vol. abs/1701.07179, 2017.

[32] A.-P. W. Group, "Phishing activity trends report." Blog, 2019.

[33] U. Meyer and V. Drury, "Certified phishing: Taking a look at public key certificates of phishing websites," in *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, (Santa Clara, CA), USENIX Association, Aug. 2019.

[34] S. Marchal, K. Saari, N. Singh, and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," *CoRR*, vol. abs/1510.06501, 2015.

[35] 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," *CoRR*, vol. abs/1707.00317, 2017.

[36] S. Afroz and R. Greenstadt, "Phishzoo: Detecting phishing websites by looking at them," in *2011 IEEE Fifth International Conference on Semantic Computing*, pp. 368–375, 2011.

[37] G. Xiang and J. I. Hong, "A hybrid phish detection approach by identity discovery and keywords retrieval," in *WWW '09*, 2009.

[38] G. Xiang, J. Hong, C. Penstein Rosé, and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing web sites," *ACM Trans. Inf. Syst. Secur.*, vol. 14, p. 21, 09 2011.

[39] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages.," in -, 01 2010.

[40] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, (USA), p. 447–462, IEEE Computer Society, 2011.

[41] S. Marchal and N. Asokan, "On designing and evaluating phishing webpage detection techniques for the real world," in *11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18)*, (Baltimore, MD), USENIX Association, Aug. 2018.

[42] Guang-Gang Geng, Xiao-Dong Lee, Wei Wang, and Shian-Shyong Tseng, "Favicon - a clue to phishing sites detection," in *2013 APWG eCrime Researchers Summit*, pp. 1–10, 2013.

[43] J. Mao, P. Li, K. Li, T. Wei, and Z. Liang, "Baitalarm: Detecting phishing sites using similarity in fundamental visual features," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pp. 790–795, 2013.

[44] J. Ma, L. Saul, S. Savage, and G. Voelker, "Identifying suspicious urls: An application of large-scale online learning," p. 86, 01 2009.

[45] S. Le Page, G. Jourdan, G. V. Bochmann, J. Flood, and I. Onut, "Using url shorteners to compare phishing and malware attacks," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–13, May 2018.

[46] N. Gupta, A. Aggarwal, and P. Kumaraguru, "bit.ly/malicious: Deep dive into short url based e-crime detection," *eCrime Researchers Summit, eCrime*, vol. 2014, 06 2014.

[47] M. Khonji, Y. Iraqi, and A. Jones, "Mitigation of spear phishing attacks: A content-based authorship identification framework," in *2011 International Conference for Internet Technology and Secured Transactions*, pp. 416–421, Dec 2011.

[48] R. Amin, J. Ryan, and J. Dorp, "Detecting targeted malicious email using persistent threat and recipient oriented features," *Security & Privacy, IEEE*, pp. 1 – 1, 01 2012.

[49] S. Hardy, M. Crete-Nishihata, K. Kleemola, A. Senft, B. Sonne, G. Wiseman, P. Gill, and R. J. Deibert, "Targeted threat index: Characterizing and quantifying politically-motivated targeted malware," in *23rd USENIX Security Symposium (USENIX Security 14)*, (San Diego, CA), pp. 527–541, USENIX Association, Aug. 2014.

[50] G. Ho, A. Cidon, L. Gavish, M. Schweighauser, V. Paxson, S. Savage, G. M. Voelker, and D. Wagner, "Detecting and characterizing lateral phishing at scale," in *28th USENIX Security Symposium (USENIX Security 19)*, (Santa Clara, CA), pp. 1273–1290, USENIX Association, Aug. 2019.

[51] G. Ho, A. Sharma, M. Javed, V. Paxson, and D. Wagner, "Detecting credential spearphishing in enterprise settings," in *26th USENIX Security Symposium (USENIX Security 17)*, (Vancouver, BC), pp. 469–485, USENIX Association, Aug. 2017.

[52] C. D. Security, "Certstream." https://certstream.calidog.io/.

[53] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," RFC 5280, May 2008.

[54] "pyopenssl." https://www.pyopenssl.org/en/stable/introduction.html.

[55] astaruch, "master-thesis-phishtank-data." https://github.com/astaruch/master-thesis-phishtank-data, 2019.

[56] "Overview — safe browsing apis (v4)." https://developers.google.com/safe-browsing/v4.

[57] afilipovich, "afilipovich/gglsbl: Python client library for google safe browsing api." https://github.com/afilipovich/gglsbl, 2019.