Computers & Security 88 (2020) 101613

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Heuristic-based strategy for Phishing prediction: A survey of URL-based approach

Carlo Marcelo Revoredo da Silva[a,b,*], Eduardo Luzeiro Feitosa[c], Vinicius Cardoso Garcia[a]

[a] Federal University of Pernambuco, Center of Informatics (CIn/UFPE) P.O. Box 7851, Recife-PE, Brazil
[b] University of Pernambuco (UPE), Garanhuns-PE, ZIP 55294-902, Brazil
[c] Federal University of Amazonas, Computing Institute (IComp/UFAM) ZIP 69077-000, Manaus-PE, Brazil

ARTICLE INFO

Article history:
Received 8 August 2019
Accepted 12 September 2019
Available online 14 September 2019

Keywords:
Phishing
Social Engineering attacks
Cybercrimes
Frauds
Heuristic prediction

ABSTRACT

Context: In the fight against phishing attacks, phishing prediction heuristics are important in developing solutions. However, phishing attacks continue to grow today, reflecting on the need for higher precision solutions.

Objective: This article focuses on phishing prediction based on a set of features. The purpose of this proposal is to evaluate the static features used and observe their occurrence in the current phishing. Static aspects refer to elements such as keywords and patterns over the phishing URL.

Method: The study methodology makes use of a survey with a set of 12 features, raised both in this study and from third-party studies, submitted to three distinct samples of phishing and legitimate sites during the year 2018.

Results: Although research on phishing prediction has developed considerably, it is possible to note that certain features are of low relevant and others have not accompanied the changes in the scenario and may need to be discarded. Some features are found more regularly in phishing and could be more efficiently exploited, indicating that further investigations need to be carried out.

Conclusion: In addition to the quantitative data, the study also performed a qualitative analysis of behaviors, managing to identify aspects such as relevance, relationships, and similarities among the features. It is expected that these results obtained can help in developing new heuristic approaches or improve the robustness of the existing ones.

1. Introduction

With regard to the detection of web phishing, browsers generally act as the first layer of defense, giving browser developers the arduous responsibility of providing protection capable of identifying a fraudulent page at the moment it is accessed (Schneier, 2013).

Like strategy, the developers build denunciation platforms, which act as an external service that periodically provides blacklists to be used by browser protection mechanisms. In general, these lists are feed by voluntary community denunciations (OpenDNS, 2019; Whittaker et al., 2010) but is possible to use, for instance, automated security software like IDS or IPS for this (Gowtham and Krishnamurthi, 2014). Actually, *SafeBrowsing* (Google, 2019), *PhishTank* (OpenDNS, 2019) and *SmartScreen* (Windows, 2019) are the principal denunciation platforms. SafeBrowsing is maintained by Google and operates in the Chrome, Firefox, and Safari browsers. PhishTank is maintained by OpenDNS[1] and operates in the Opera. Finally, SmartScreen is Microsoft's solution for Internet Explorer and Edge.

Although efficient and simple, denunciation platforms solutions (and their blacklists) face off some issues. The main is the newly-created phishing site, commonly referred to as **zero-day (0-day) phishing** (AlEroud and Zhou, 2017; Srinivasa et al., 2019), because it will not be flagged as phishing until the time that it is registered in the blacklist. This time period can be considered a **window of vulnerability**. As a result, this phishing kind can take hours or even months to be registered into blacklist, resulting in enough time to apply scams to multiple users.

* Corresponding author.
E-mail addresses: cmrs@cin.ufpe.br (C.M.R.d. Silva), vcg@cin.ufpe.br (V.C. Garcia).

[1] https://www.opendns.com/.

https://doi.org/10.1016/j.cose.2019.101613
0167-4048/

Another issue is the expended effort to properly maintain blacklists since the lifetime of a phishing site is short, especially those running on fast-flux networks (Almomani, 2018; Moore and Clayton, 2007). Maintainers of blacklists end up storing an incredible amount of phishing sites in their lists, many of them that not exist more.

An extra issue is related to the semantics of the URL, where the modification of even a single character in the phishing URL will make it unknown, and consequently, the blacklist will store redundant phishing URLs. There is also the case of more targeted attacks that are not widely propagated by the Web, such as *spear phishing*, with very reduced dissemination, consequently decreasing the chances that these sites will end up in blacklists (Vayansky and Kumar, 2018). Finally, it is not uncommon for genuine sites to be erroneously blacklisted, intentionally or not, resulting in a **false positive**. Considering that a particular site can be banned across several browsers, such a situation can bring significant trouble to the parties as damaging as a successful fraud.

Faced with these obstacles, **heuristics-based solutions** have been developed as the intent to identify phishing through **prediction**, where a set of features existing either in the URL or in the page content itself makes up the knowledge necessary to classify a page as malicious. Classification models with machine learning are commonly used (Alkhozae and Batarfi, 2011; Chelliah and Aruna, 2014).

Based on a number of studies (AlEroud and Zhou, 2017; Chaudhry et al., 2016; Chelliah and Aruna, 2014; Kirda and Krugel, 2005; Naresh et al., 2013), heuristics-based solutions need to demonstrate **responsiveness** and **response time**. Responsiveness is the accuracy prediction of the mechanism on a threat, considering false positives or negatives. The response time refers to the speed of response to the incident, that is, the window of vulnerability often provided in 0-day phishing incidents

Comparing blacklist and heuristic-based solutions, it is noted that each approach has its pros and cons. While it is impossible for a blacklist to catch 0-day phishing, its responsiveness is less complex. In the heuristic-based model, responsiveness is conditioned on the adequacy of the **concept drift** (Elwell and Polikar, 2011) concepts, common in the dynamic phishing environment. In other words, the number of features must be balanced so as not to compromise response time.

Another challenge is ensuring the preservation of user privacy. For example, a URL may contain sensitive data in its querystring, meaning that a blacklist-based solution will intercept this information from the URL address accessed by the user. Similarly, in addition to the URL, heuristic models that oversee content can intercept sensitive data from the page itself (Chaudhry et al., 2016). Such solutions must state that the intrusive nature of their actions does not violate privacy, and such a statement must be convincing. Therefore, heuristics-based solutions have to have a solid justification for the choice of features they employ.

This study presents a **survey** that investigates **static behaviors** based on a self-structuring defined as **URL-based taxonomy**. For this, we propose a taxonomy to classify the action scenario of phishing in 3 distinct categories that define behaviors, namely: URL blacklist bypass, URL morphology and User susceptibility. Each category has a group of features peculiar to its behavior, totaling 12.

The study analyzed some features in confirmed phishing and legitimate sites as a comparative means to test hypotheses that justify the behavior. The PhishTank was adopted as a reporting platform and database for data extraction and sample definition. Lastly, collection and interpretation is presented, which describes **relevance, relation**, and **similarities** between the features, as well as limitations, conclusions and further works. The research methodology used in this survey is illustrated in Fig. 1.

The present study is structured as follows: Section 1 is an introductory section about the proposed study. Section 2 presents a current overview of the phishing research scenario, briefly describing the timeline and strategies of the proposed studies. Section 3 presents the methodology of the study, as a method of investigation and extraction of data based on real phishing. In Section 4, the results obtained by the extraction methodology are presented, making use of another methodology to present the data extracted. Section 5 presents the threats and limitations of the study in its current state. Section 6 presents related studies and the differences between them and this study. Finally, Section 7 presents the conclusions and future perspectives based on an evolution of the current study.

2. *Phishing* prediction background

According to Kaspersky (2014), phishing is an attack characterized by fraudulent attempts against Internet users. The attacker develops a fake page that presents itself as if it is a trusted environment, inducing victims to submit sensitive data, for example, forms with credentials for access to a genuine service (Mohammad et al., 2015). Based on the quarterly reports of the Anti-Phishing Working Group (APWG), it is possible to consider the problem to be chronic and constantly on the rise.[2] A brief investigation in the ScienceDirect Digital Library[3] made it possible to draw a timeline showing the emergence of and concerns regarding the problem, namely:

- **1998**: The term phishing is first used to refer to an online fraud scheme.
- **1999 to 2003**: Criminal organizations use this practice to target banks.
- **2004**: The practice becomes more common in the global scenario. Studies in the literature are motivated to explain the phenomenon.
- **2005**: Anti-phishing practices gain ground in the literature, especially with regard to authentication levels.
- **2006**: The literature begins to describe combined phishing and malware attacks.
- **2007**: Reports of bank fraud in many countries. Forensic computer techniques are motivated to identify fraud.
- **2008**: Approaches that use computational intelligence appears in the literature as a justification for identifying real-time fraud.
- **2009 to 2018**: Proposals based on behavioral patterns consolidate and become a trend in the literature.

Given the scenario presented, approaches in the literature emerge with the proposal of observing standard behaviors as support for phishing prediction, a technique similar to anti-spam filters. Considering the context, there are elements of URL (protocol, path, querystring, domain and subdomains) that be relevant to observing behavioral patterns. An analysis of these behaviors can be defined as URL-based.

The exploitation of a phishing attack is centered in the context of the exploitation of human aspects, analyzing means of access and events to abuse trust, or more precisely, susceptibility (Whittaker et al., 2010). However, computational resources, when exploited, can facilitate the carrying out of the attack, either through the context of the content, such as the HTTP protocol, or through the context of third parties involved, such as browsers and maintainers, for example, browser behavior that can elevate privileges or appear to be trustworthy. In addition, through social engineering, the attacker analyzes the existence of a 6- or 8-digit

[2] https://apwg.org/resources/apwg-reports/.
[3] ScienceDirect: https://bit.ly/2MzFfwx.

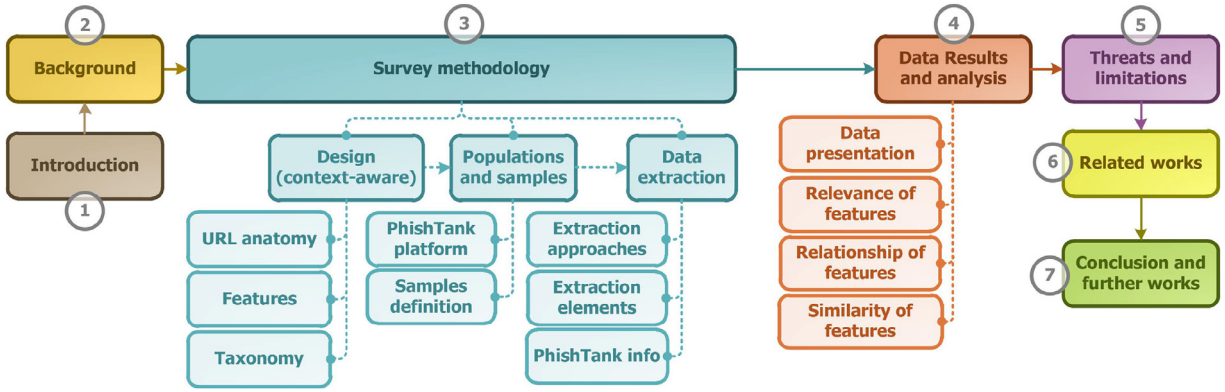

Fig. 1. Study structure.

access password requested by a maintaining service, resulting in a set of profiles (Khonji et al., 2013).

The attacker can also investigate aspects of the URL morphology, which, along with the profile, can inspire reliability in order to exploit susceptibility. The higher the quality of information, the greater the reliability, reducing suspicions of fraud. Therefore, the attacker acts as a "Man in the Middle" (MiTM) obtaining privileges, either by proxy, browser extension, router redirects, or other malicious applications (Whittaker et al., 2010). In addition, the attacker can use the infected computer to spread the malicious site among the victim's contacts or obtain sensitive data, which the criminal commonly sells on the black market. In the following subsections, terms, nomenclatures, and strategies adopted will be presented in the context of commonly-adopted anti-phishing solutions that are present in the literature.

2.1. Heuristic-based approaches

The academic literature reports several techniques for constructing prediction heuristics. One of the great challenges is to have enough sensitivity to not only identify pre-defined concepts, but also perceive concept changes, which are common in very dynamic scenarios, such as the phishing environment. Such conceptual changes are known as concept drift (Elwell and Polikar, 2011). In general, there are four types of concept drift, namely:

- **Abrupt change**: when changes occur suddenly, causing heuristics to drop significantly in precision.
- **Gradual change**: when the changes oscillate regularly over a period of time, showing slight random noise before making a definitive change.
- **Incremental change**: similar to gradual change, but with the transition occurring more slowly, until it stabilizes for a new concept change.
- **Recurring context**: occurs when certain behaviors that had previously ceased to exist will occasionally reappear, consequently oscillating between a new or existing concept change.

There are several anti-phishing heuristic strategies that are susceptible to the problems of concept drift and that have been debated in the literature. However, the approaches are commonly divided into 3 types (Afroz and Greenstadt, 2011; Amiri et al., 2014): the Non-Content-based Approach, the Content-based Approach, and the Visual similarity-based approach.

2.1.1. Non-content-based approach

This approach does not need to analyze the content of the suspected page in order to classify it as legitimate or fraudulent. In general, such solutions employ either filters based on blacklists or whitelists, analysis of lexical patterns in the URL, or both. Faced with the limitations already presented regarding the use of blacklists, lexical pattern strategies have gained strength.

Examples of such patterns would be a check on the size of the URL, the number of characters used as separators, the reputation or geographical location of the page host, among others (Ma et al., 2009). This option is attractive due to two aspects: (i) the processing response speed, because the only element analyzed is the URL; and (ii) the absence of intrusion into the page content, offering less risk to end-user privacy. However, because it does not evaluate the page content, this approach is less sensitive to aspects of context (Elwell and Polikar, 2011).

2.1.2. Content-based approach

This approach bases its predictions on elements or information contained within the suspicious page itself. Elements of this nature include password fields, misspellings, CSS formatting, as well as HTML or JavaScript code. Despite the possibility of discarding the use of blacklists, because of the dynamic nature of the phishing environment, heuristics of this type need to perform constant revisions in order handle the issue of concept drift.

Aburrous et al. (2008) highlight the use of hashing to identify malicious sites duplicated on the Web. This approach can be easily broken, however, if any modification is made to the malicious site, because its resulting hash would then also be modified. Solutions using this approach, compared to those based on blacklists, need to have an even greater concern about potential false positives, because the process of undoing an undeserved block involves a reassessment of the precision metric, a task that is considerably more expensive and involved than simply removing from a list.

2.1.3. Visual similarity-based approach

Another commonly-used approach is to perform prediction based on a screen capture of suspicious pages. It is possible to perform prediction by similarity using image contrast and clustering of key points with a k-mean algorithm. The calculation of Euclidean distance can also be used to identify similarities (Jain and Gupta, 2017). Visual similarity can also be determined through optical character recognition by converting the images into text and comparing the results. Yet another strategy is that of identifying similarity through the CSS layout.

As with the page content-based approach, this methodology better addresses the issue of 0-day phishing. However, the similarity classifier must be constantly trained. Another factor is that malicious pages do not always faithfully reproduce the look of the genuine page, either through failure to load images or CSS style sheets, resulting in a page that is aesthetically different from the genuine one, producing false negatives.

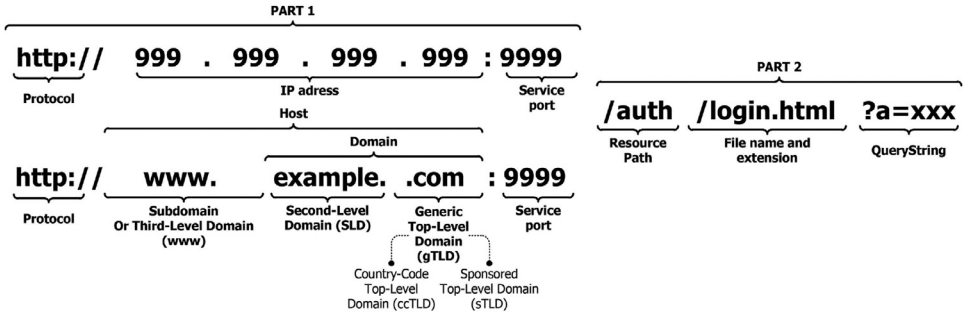

Fig. 2. URL anatomy.

3. Survey methodology

The objective of this study is to present a survey that investigates patterns of URL-based phishing behaviors in real environments. According to Wohlin et al. (2000), a survey is understood as a primary study, that is, an experimental software engineering approach that observes behaviors in a specific context. Such behaviors can be translated as the features proposed by the heuristic that can classify a page as malicious. These features will be observed in pages confirmed as fraudulent and pages considered legitimate, in order to test the hypotheses and investigate whether a certain behavior occurs only on fraudulent sites or may be found anywhere.

Because the samples and dependent and independent variables cannot be controlled, and a real environment is being observed, a survey methodology was adopted for the study, as discussed in the works of Molleri et al. (2016), Robson (2002), and R. Babbie (2019). Generally, a survey is presented as a retrospective investigation, analyzing behaviors through data collected from a representative sample of the context population. From the results of this collection, conclusions can be drawn that are generalized to the population from which the sample was derived.

3.1. Design

This section describes the design of the methodology as a way of delimiting the scope and objectives. Data, categories, and features are defined. Additionally, the composition of the relationships and the similarity between the features allowed for the definition of a taxonomy.

3.1.1. Data definition

During the study, some terminology used during the steps of the study will be described in this section. For example, this study proposes the analysis of a URL according to the strategy illustrated in Fig. 2, where the URL is divided into two parts in order to observe its anatomy with distinct and necessary strategies to investigate certain features.

Part 1 consists of the protocol, IP address or host, and the Web service port. The host is composed of a subdomain, also called the third level, and a domain. The domain is the combination of a top-level (gTLD) and second level (SLD) domain. GTLD domains are intended to specify the country or general segment of the website. The SLD domain is the name of the site owner in the DNS server registry. Part 2 of the URL consists of the path, file name, file extension, and the GET method parameters.

Elements such as the protocol, IP address, and port can be defined by the attacker, however, they will have slightly more predictable manipulations. For example, in part 1, it is possible to predict that the protocol will have values followed by the characters "://", and the port will be represented by a colon ":", followed by a series of integers. On the other hand, elements such as the IP address or host offer more flexibility for the attacker to exploit certain tactics, considering that he is free to define the values as he wishes. In part 2, the parameter elements will have their values preceded by the character "?" or "&" for multiple values. However, the attacker is free to assign the respective values. The remaining elements of part 2 can have arbitrary values.

3.1.2. Features

Each feature represents a means of measuring a particular aspect of URL-based phishing behavior. However, the presence of a certain feature may not always indicate that the page is malicious. Because of this, phishing prediction models adopt strategies in which a specific set of existing feature represents a certain class in the classification model. This is a result of the need for balance between the sensitivity and specificity of the heuristic.

The intent of this study is to evaluate the individual occurrences of each feature in actual phishing and on legitimate pages. In order to do so, 12 features were evaluated, categorized by different approaches. Some these features were extracted from studies published in the literature, related and presented in Section 6. These features often make up a classification model heuristic, and have become available as a dataset on the Kaggle[4] and UCI[5] sites. For questions of convenience, details on each feature are described in Section 4, where the data and interpretations are presented.

3.1.3. The phishing URL-based taxonomy

A taxonomy is an artifact responsible for presenting a categorized structure of concepts. Such an artifact has the potential to provide a greater understanding of the systematics and shared similarities between behaviors to be debated, as their classification can provide a clinical view of the behaviors and their relationships. In this case, the behaviors in question are the features of the heuristic prediction model.

The proposed taxonomy, as shown in Fig. 3, is considered URL-based because considers URL aspects of the phishing, such as the domain, subdomain, protocol, path, querystring and other elements as describes in Fig. 2. Each approach operates to 3 distinct categories such as the behaviors of the phishing approach based on blacklist approaches, morphology, and the user susceptibility exploited by malicious users, representing 12 features. The details on each categories are described in Section 4, where the data and interpretations are presented.

3.2. Populations and samples

There are two population types to be considered, one containing all existing phishing sites in the world and one containing all

[4] https://www.kaggle.com/akashkr/phishing-website-dataset.
[5] https://archive.ics.uci.edu/ml/datasets/phishing+websites.

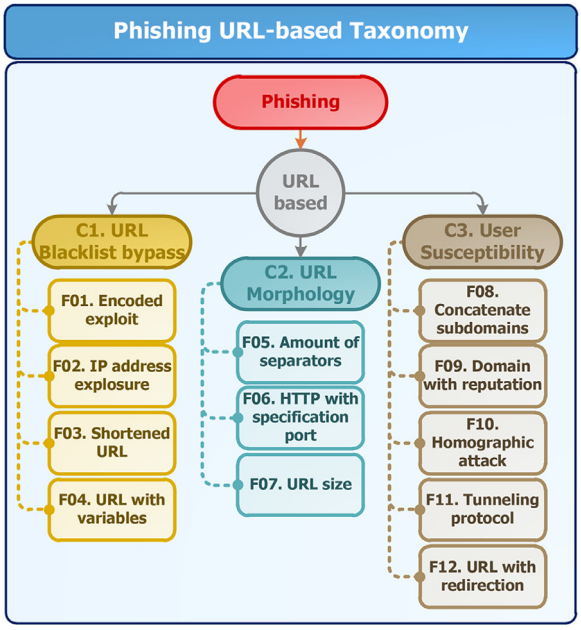

Fig. 3. Phishing URL-based taxonomy.

legitimate pages in the world. For the sample process, the study adopted the PhishTank platform as a database containing both actual (valid) phishing samples and legitimate (invalid phishing) sites. Despite the possibility of reduced precision, samples are a viable means of achieving the objectives of such a survey. Although other services are available, such as OpenPhish[6] and SafeBrowsing,[7] PhishTank was chosen because of its open base and higher volume, as well as having its features available for free.

3.2.1. The PhishTank platform

PhishTank is designed to be a free community where anyone can send, verify, track, and share phishing data.[8] In addition, it also provides an open API to share anti-phishing data with third parties for free. It is important to emphasize that the PhishTank team does not consider its platform as a protection measure,[9] however, the information provided by it serves to support incident response mechanisms at a number of organizations,[10] including Yahoo!, McAfee, APWG, Mozilla, Kaspersky, Opera, and Avira.

PhishTank is described as a **community** because it involves a large number of users who share phishing data on the Web among themselves. Its **collaborative** nature refers to the fact that all registered users have the possibility of feeding information into the phishing database. Each phishing record is built from complaints consisting of information regarding **confirmation** and **availability**.

With regard to confirmation, PhishTank enables a user to submit a suspicious URL and other users to vote in order to determine a verdict on the complaint, i.e. to consider it to be **valid** or **invalid** phishing. As for availability, the platform looks at whether the phishing is **online** or **offline**. It is important to point out that an unavailable phishing means that the request returned 400 or 500 HTTP status code, meaning that the site is inaccessible, with a status of "offline". The life cycle between the phishing site, the platform, and its users can be divided into five steps, as shown in Fig. 4.

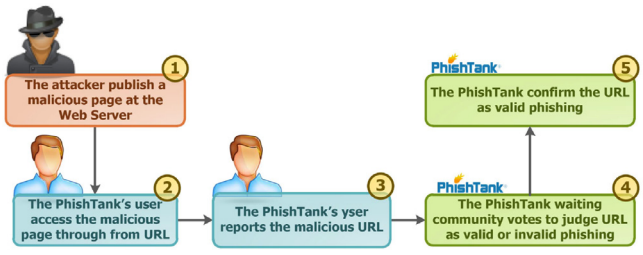

Fig. 4. Community life-cycle in PhishTank platform.

In step 1, the attacker publishes his/her malicious page on a web server, making it available to be propagated on the Web. Step 2 refers to the discovery of the malicious URL by a user. Subsequently, the user accesses PhishTank and reports the URL in step 3. Step 4 is when the platform waits for community votes on the newly URL. Finally, step 5 occurs when the voting system receives enough votes to consider the URL malicious or not. It is worth noting that the "sufficient" number of votes is not explicit. The platform states that it may vary according to the history of denunciations.[11]

3.2.2. Samples definition strategies

To define the samples, it was necessary to obtain a significant amount of "valid" phishing sites, either online or offline. As an alternative, the platform makes available a web service that provides a **JSON file**.[12] This file is updated hourly and contains an average of 15,000 records. In addition to the URL, status and **publication date**, it provides the **confirmation date** and the **target brand** of the fraud. The confirmation date is the moment that voting was closed and the URL was confirmed to be a phishing.

Considering that a number of organizations execute simultaneous requests in the API, to avoid overloading the platform's servers, requests to the file are carried out with a key that identifies the user in the HTTP header along with the limits and intervals of the requests to be carried out periodically.[13] Therefore, an application for the periodic consumption of the JSON file was developed. As the platform does not establish a deadline for voting, the study adopted a **collection interval** margin of one month adjacent to the current month, that is, the January collection was closed on the last day of February, and this pattern was followed successively, concluding the collection for the month of December 2018 on 31 January 2019.

However, there were some obstacles to the process. About 90% of the URLs were kept in subsequent files. Considering that each compressed JSON had 9 mb of data, the platform eventually overloaded, giving *509 bandwidth limit exceeded* errors, indicating that the request key had been banned. In order to circumvent this problem, registries of several keys were performed in advance to use as substitutes whenever a specific key was banned, according to the flowchart on the left in Fig. 5.

Because it is so informative, JSON is an excellent sample for analyzing behaviors, however, it has some limitations. Because it registers only "valid" and "online" phishing, temporal features will have skewed results. An example of this was observed in a file downloaded on 15 January 2019, in which the months of January and February of 2018 had, respectively, 358 and 617 records. In the same file, the more recent months of November and December included, respectively, 1524 and 1791 records.

As an alternative, the platform provides the **phish archive**[14] functionality in the **phish search** menu, which contains a history

[6] https://openphish.com/.
[7] https://safebrowsing.google.com.
[8] https://www.phishtank.com/faq.php#whatisphishtank.
[9] https://www.phishtank.com/faq.php#doesphishtankprotect.
[10] http://www.phishtank.com/friends.php.
[11] https://www.phishtank.com/faq.php#howmanypeoplehavetov.
[12] http://data.phishtank.com/data/online-valid.json.bz2.
[13] https://www.phishtank.com/developer_info.php.
[14] https://www.phishtank.com/phish_archive.php.

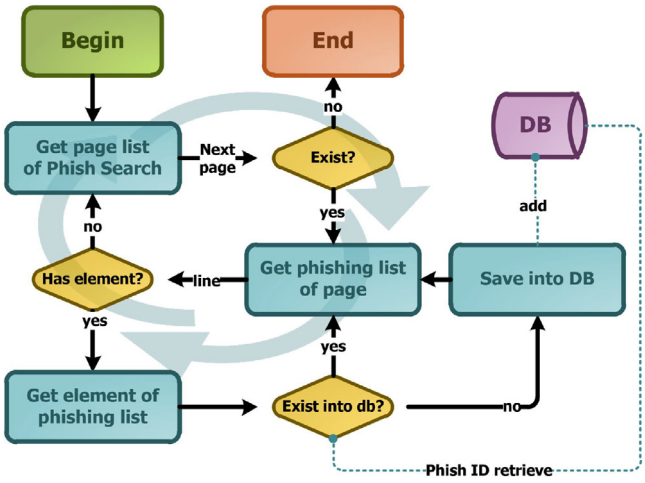

Fig. 5. Workflow for "Phish Search" extractions.

of all submitted URLs, making it possible to filter the records by "valid" or "invalid" and "online" or "offline". Unfortunately, PhishTank does not offer a download of all records in this report, and there have been several attempts to contact the PhishTank team in order to obtain this information. However, as the requests were not responded to, a **Web Crawler** was developed to collect and store the information from the records in an automated way. This process is illustrated in the flowchart on the right in Fig. 5.

Another difficulty is that the listing page only displays the first 70 characters of each URL. When a URL is longer than this, the rest of its characters are replaced by "...". That is, in cases where the number of characters in the URL exceeds this limit, the Web crawler had to access the detail page to get the full URL. For some records, however, the details page was inaccessible. Because there were very few cases where this occurred, they were discarded. In addition, in an attempt to find phishing from over 10 years ago, *HTTP 504 Gateway time-out* errors were often introduced, indicating a server limitation, which limited access older than 10 years.

The report provides information such as the phishing ID on the platform, the URL, the submission date, its confirmation (whether "valid" or "invalid"), and its availability (whether "online" or "offline"). The submission date records the time when a particular user reported the potentially malicious URL. The sample definition flowchart is shown in Fig. 6.

The Web Crawler was able to extract 2,156,759 phishing records from a 10-year period from 2009 to 2018. With this data, it was possible to define **Sample #1**, which considered valid phishing site from the year 2018, resulting in 189,892 records. Simultaneously, **Sample #2** was defined, following the same rule except for "invalid" phishing, resulting in 1384 legitimate site registrations. From comparing these samples, the objective was to determine any differences in behavior between valid and invalid phishing sites.

Even with the reduced population size in Sample #1, the extraction process was still not feasible for features which could not be gathered automatically. Therefore, the **Sample #1.1** was defined according to Eqs. (1) and (2) [39, 40]. The sample size defined was 384 for valid phishing. As the purpose of these samples is also to be temporal, the number of entries was divided proportionally by the months of the year, resulting in 32 entry for each month of sample #1.1.

$$\frac{z^2 \times p(1-p)}{e^2} \tag{1}$$

$$1 + \left(\frac{z^2 \times p(1-p)}{e^2 N} \right) \tag{2}$$

The calculation has two steps. In (1), z represents the degree of confidence in standard deviations (95% or 1.96 on the Z-scale), e represents the margin of error (5% or 0.05), and N the total population of URLs (189,892 for valid and 1384 for invalid). Finally, p is the true probability of the event, that is, the individual probabilistic chance of a URL being chosen, represented by a constant 0.50 (50% probability).

3.3. Data extraction

Because the extraction process needed to be constant for the year 2018, it was necessary to perform an extraction routine with a well-defined beginning and end. Therefore, data extraction was performed according to two distinct schedules, namely: the **timeless samples** were extracted every hour daily during the year 2018; while the **temporal samples** were all extracted at a single moment, on 1 Feb 2019.

As shown in Fig. 7, each sample defined by the study required a distinct set of data. For example, Sample #1 contains data to the HTTP scope, such as the URL and source code, but does not store the page header and body. The **PhishTank Info** column is made up of data provided by the PhishTank, such as submission date, verification date, status, confirmation, and target mark as described in Section 3.2.2. There were two types of extraction methods used: manually, which could be applied subjectively; and automated, which was handled objectively through an algorithm.

4. Data results and analysis

This section describes the data obtained from the extraction methods and their interpretation through analysis of their behavior. Fig. 8 describes the features with their respective extraction contexts and the samples used to obtain the data. The order of display and analysis of the data will respect the taxonomy from Section 3.1.3.

The Goal Question Metric (GQM) methodology proposed by Basili et al. (1994) was adopted, in order to provide formalism and planning regarding the measurement of the results to be extracted and interpreted. Each of the six categories defined by the taxonomy represents a type of **objective** to be evaluated against the samples, which act as **objects of measurement**. Each feature represents a **question**, and each behavior represents a **metric**, totaling 49.

In addition, the study seeks to present a verdict on the relevance of the feature, based on the data obtained, on a scale of *WEAK, MODERATE* e *STRONG*. The criterion used to determine relevance considers both quantitative and qualitative aspects. For example, a quantitative result, being discreet and expressive, can influence the relevance considerably; however, temporal behaviors or aspects of context, when applicable, can balance this verdict.

4.1. URL blacklist bypass

This category describes features designed to **bypass blacklist based mechanisms**. With this data, patterns can be highlighted that apply to the composition of the URL.

4.1.1. F01. Encoded exploit

This feature evaluates the URLs that have their hostname or IP address encoded in hexadecimal or base64, both supported by the browser. Unlike the *punycode* exploit described in feature F01, this exploit does not directly target the end user, but instead focuses on the blacklist mechanism. By generating the hash of a URL from the encoded hostname or IP, the result will be different from the hash of an unencrypted URL. The data are shown in Fig. 9 and the GQM analysis is in Table 1.

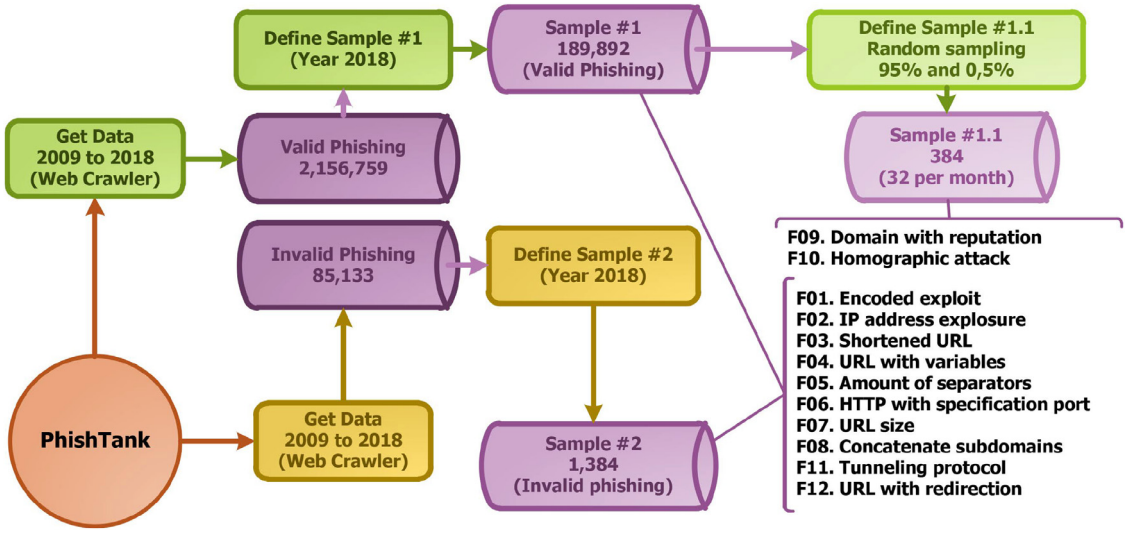

Fig. 6. Population and Sample definition strategies.

Samples	Extraction type		HTTP				PhishTank Info				
							Submission time	Status		Confirmation	
	Auto	Manual	URL	Code	Header	Body		Online	Offline	Valid	Invalid
Sample #1	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗
Sample #1.1	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Sample #2	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓

Fig. 7. Details of *phishing* info.

C1. URL blacklist bypass	Extract Element	Sample #1	Sample #1.1	Sample #2
F01. Encoded exploit	URL part 1	✓	✗	✓
F02. IP address explosure	URL part 1	✓	✗	✓
F03. Shortened URL	URL part 1 & 2	✓	✗	✓
F04. URL with variables	URL part 2	✓	✗	✓
C2. URL morphology	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**
F05. Amount of separators	URL part 1 & 2	✓	✗	✓
F06. HTTP with specification port	URL part 1	✓	✗	✓
F07. URL size	URL part 1 & 2	✓	✗	✓
C3. User susceptibility	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**
F08. Concatenate subdomains	URL part 1	✓	✗	✓
F09. Domain with reputation	URL part 1	✗	✓	✗
F10. Homographic attack	URL part 1 & 2	✗	✓	✗
F11. Tunneling protocol	URL part 1	✓	✗	✓
F12. URL with redirection	URL part 2	✓	✗	✓

Fig. 8. Relation between categories and samples.

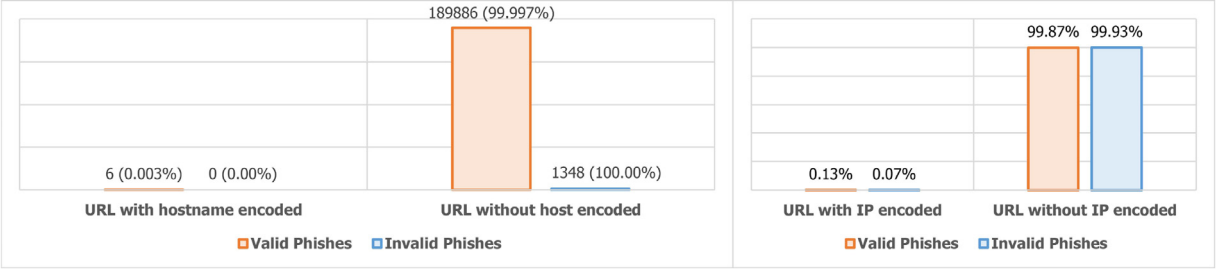

Fig. 9. Occurrences of F01 for encoded hostname and IP address.

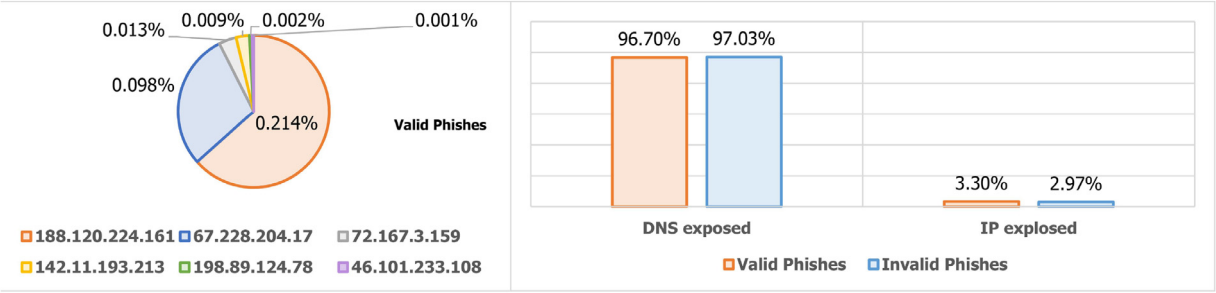

Fig. 10. Occurrences of F02 IP address explosure.

Table 1
GQM of F01. Encoded exploit.

Goal 1	Analyze attack techniques used to bypass blacklist mechanisms.
Question	Q17. What records are propagated by using encode on hostname or IP address?
Metrics	[M01] Count of records that have hostname encoded for valid phishing. [M02] Count of records that have the hostname encoded for invalid phishing. [M03] Count of records that have the IP encoded for valid phishing. [M04] Count of records that have the IP encoded for invalid phishing.
Hypotesis	Malicious users have an interest in propagating fraudulent URLs by encoding the hostname or IP address, to possibly bypass in cases where the filtering is restricted to part 1 of the URL.
Sample	1 and 2 **Relevance** STRONG **Relations** F02 and F07
Extraction	Get parts 1 and 2 of the URL and analyze the presence of an encoded hostname or IP address.
Limitations	-
Observations	-
Analysis	The encoded display of the URL hostname is reasonably present in malicious URLs and does not exist on legitimate sites. Similarly, encoded IP addresses are also reasonably present in malicious URLs. They do have, however, a discreet presence on legitimate sites.

Table 2
GQM of F02. IP address explosure.

Goal 1	Analyze attack techniques used to bypass blacklist mechanisms.
Question	Q18. What records are propagated using a public IP address?
Metrics	[M05] Count of records that use an explicit public IP in valid phishing. [M06] Count of records that use an explicit public IP in invalid phishing.
Hypotesis	There is an interest on the part of malicious users to propagate malicious URLs in two ways, with or without their DNS, in order to perform a possible by-pass if the blacklist in question does not provide for hash modification through this maneuver.
Sample	1 and 2 **Relevance** WEAK **Relations** F01 and F07
Extraction	Get the URL and parse for the presence of a public IP. Also check for the incidence of pages with DNS that refer to the same IP, providing evidence for propagation using this strategy.
Limitations	-
Observations	There may be cases where the approach deprecates the presence of the default port and generates the hash without it, thus avoiding a possible by-pass through this technique. This feature only analyzes the presence and not the effectiveness of protection mechanisms.
Analysis	It was possible to observe that 3.30% of reported phishing sites used such a technique, a number that deserves attention since it represents more than 6000 records. In addition, it was possible to note that 406 URLs with DNS referenced the same IP number.

According to the left bar graph in Fig. 9, only 0.003% of valid phishing has an encoded hostname. However, an interesting fact is that none of the invalid phishing sites in the sample used hostname encoding, making its mere existence in the URL a very suspicious sign. In the same figure, the right graph shows that the occurrences of codification in the IP address were more frequent, with a presence of 0.13% in valid phishing sites and 0.07% in invalid phishing sites. Although little applied, the behavior can be considered suspicious, theoretically within a low margin of error. Given this, the feature was considered to have STRONG relevance.

4.1.2. F02. IP address explosure

This feature evaluates URLs that use their IP address instead of the DNS. This appears to be set up as an attempt to bypass blacklist engines that generate the hash based on the URL using DNS. These cases raise so many suspicions that some applications, such as WhatsApp, do not allow URL hyperlinks in messages if they use an IP address, in order to mitigate phishing attacks. The extracted data are shown in Fig. 10 and the GQM analysis is in Table 2.

The pie chart in Fig. 10 describes the IP addresses that were most duplicated in valid phishing records, that is, the same malicious page on the same server, but with a different URL. For example, 406 records, 0.21% of all phishing samples in Sample #1, refer to the same IP address. In the bar graph, it was possible to observe that the IP address use cases were present in both valid and invalid phishing. Given this, the feature was considered to be of MODERATE relevance.

4.1.3. F03. Shortened URL

This feature evaluates URLs that use shortening services to encapsulate the original URL. The hypothesis in question refers to the interest on the part of malicious users to propagate the shortened URL so that a different hash is generated. The occurrence of exploitation by depth was identified, that is, several shortenings of

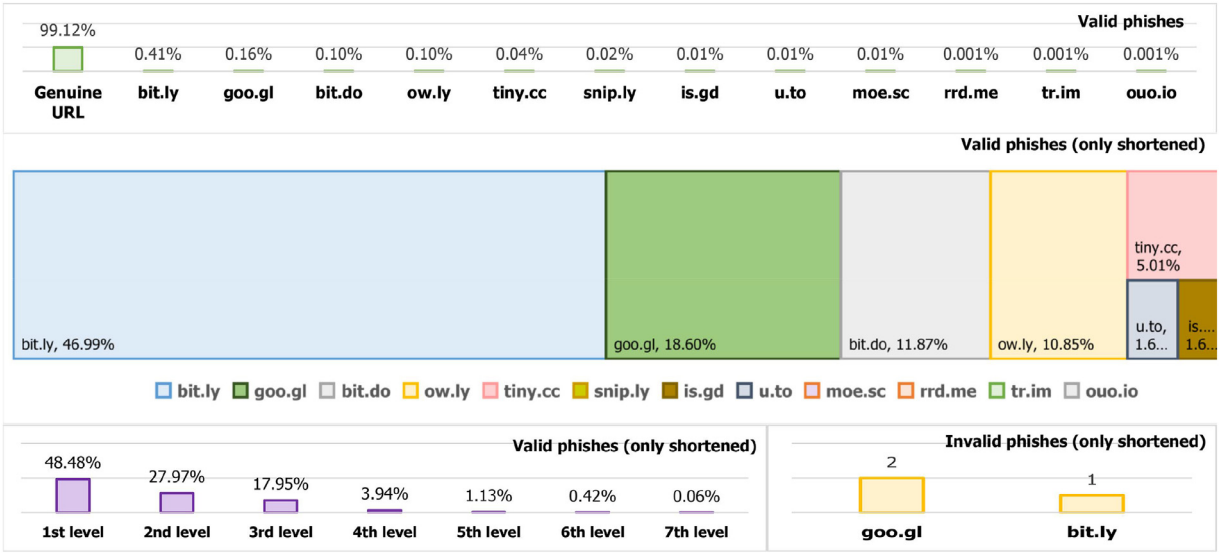

Fig. 11. Occurrences of F03. Shortened URL.

Table 3
GQM of F03. Shortened URL.

Goal 1	Analyze attack techniques used to bypass blacklist mechanisms.
Question	Q19. What entries are propagated using URL shortening services?
Metrics	[M07] Count of valid phishing records that use URL shortening. [M08] Recognize shortening levels on valid phishing sites. [M09] Count of invalid phishing records that use URL shortening. [M10] Recognize shortening levels on invalid phishing sites.
Hypotesis	There is an interest on the part of malicious users to propagate malicious URLs by making use of URL shorteners in order to perform a possible by-pass if the blacklist in question does not provide for hash modification through this maneuver.
Sample	1 and 2 **Relevance** STRONG **Relations** F07
Extraction	Get the shortened URL and parse the URL that results from the deshortening.
Limitations	-
Observations	This analysis also noted that certain URLs made use of 2 or more shortenings from different services for the same URL, that is, a particular URL had encapsulation levels in its final URL.
Analysis	It was observed that services like bit.ly and goo.gl were the most used in 2018. However, many of these services, like goo.gl, have been discontinued. Others, such as ow.ly, begin to adopt a stricter rules for the creation of shortened URLs, such as forcing the user to have a registration. A curious fact is that the bit.ly service allows shortened URLs that point directly to files, such as executables or PDFs. Other services, such as goo.gl, did not allow this kind of situation. It was also possible to observe URLs with up to 7 levels of encapsulation, that is, the same URL that was shortened 7 times.

the same shortened URL. The extracted data are shown in Fig. 11 and the GQM analysis is described in Table 3.

As shown in Fig. 11, the X-axis of the bar graph at the top segments the records by the shortening service used, or "Genuine URL" for cases of unshortened. The Y-axis presents the quantity as a percent of the sample. According to the data, 99.04% of valid phishing did not make use of shortening services, that is, only 0.96% had this feature. Although little used, the tactic still even as the years go by.

Some services, such as goo.gl, have been discontinued[15] and others, ow.ly, have become more critical, requiring prior registration and applying restriction policies, such as the prohibition of shortened URLs for full addresses that access files. However, the bit.ly service still allows such behavior and does not adopt many usage policies, explaining its popularity in scams. It seems that the practice of shortening as a third-party URL has been discouraged. Site owners have been motivated to shorten the URLs of their own resource, such as YouTube shortening its links through youtu.be.

In the same figure, the treemap graphic groups the shortened URL records by their respective shortening services, the difference being that the Y-axis considers only the records that have been shortened. The graph makes it clear that bit.ly is the most exploited shortening service. Finally, the bar graph at the bottom describes valid phishing sites that used the depth technique. This technique refers to when the attacker shortens an already shortened URL several times, deepening the encapsulation of the genuine URL in order to bypass blacklists, and making multiple links to the same page. It was observed that 44.82%, that is, almost half of the shortened URL used a deep technique. Given this, the feature was considered to have STRONG relevance.

4.1.4. F04. URL with variables

This feature evaluates URLs where the attacker has manipulated the path or querystring values so that the resulting hash will be different. The extracted data are shown in Fig. 12 and the GQM in Table 4.

As shown in Fig. 12, the line graphs make a comparison of the exploitation of path or querystring variables between invalid and valid phishing sites, demonstrating clearly that the behavior is heavily exploited in valid phishing, with thousands of records, while in invalid phishing, the total barely reaches dozens. The bar column graphs make a comparison between valid and invalid

[15] The service https://goo.gl/ was discontinued on March 30, 2018.

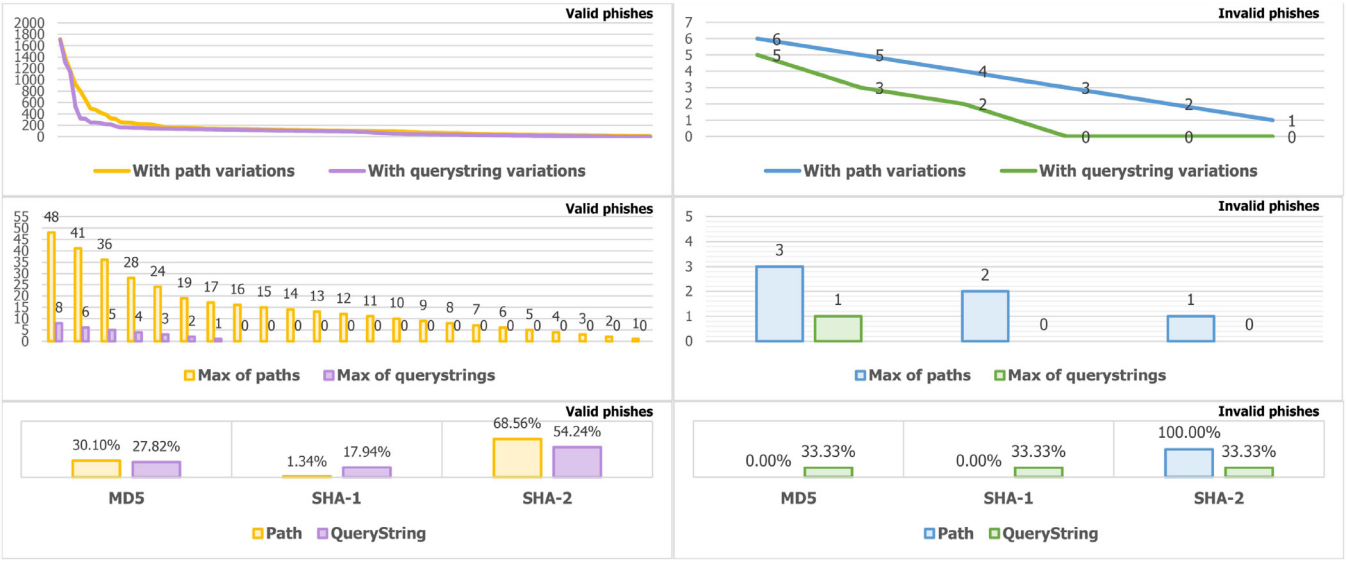

Fig. 12. Occurrences of F04. URL with variables.

Table 4
GQM of F04. URL with variables.

Goal 1	Analyze attack techniques used to bypass blacklist mechanisms.
Question	Q20. Which records are propagated with variables in their URLs?
Metrics	[M11] Count the types of values in the URL querystring for valid phishing sites. [M12] Count the types of values in the URL path for valid phishing sites. [M13] Count records that use values in the URL querystring for valid phishing sites. [M14] Count records that use values in the URL path for valid phishing sites. [M15] Count the types of values in the URL querystring for invalid phishing sites. [M16] Count the types of values in the URL path for invalid phishing sites. [M17] Count records that use values in the URL querystring for invalid phishing sites. [M18] Count records that use values in the URL path for invalid phishing sites.
Hypotesis	There is an interest on the part of malicious users to propagate fraudulent URLs by making use of values in the querystring or path, in order to perform a possible by-pass, if the blacklist in question does not provide for hash modification through this maneuver.
Sample	1 and 2 **Relevance** MODERATE **Relations** F05, F07 and F10
Extraction	Obtain the URL and analyze its second part.
Limitations	-
Observations	-
Analysis	Certain blacklist approaches do not create the hash on only the first part of the URL, but rather on both parts. In these cases, dynamic values provide considerable variations in URLs for the same fraudulent page. It was possible to observe a considerable number of variations using this technique. A notable fact is that approximately 50 malicious URLs had 48 paths in their composition, different from legitimate URLs that presented a maximum of 3 paths in their composition.

phishing, quantifying the exploitation by path or querystring in the same URL, making it possible to identify that a single URL had 48 paths and 8 querystrings, a situation considered quite unusual. In cases of invalid phishing, the occurrences are much more modest, proving that the presence of this exploit raises valid suspicions about the URL.

In the same figure, the penultimate bar graph shows the URLs that perform hashes on the path or querystring values, thereby modifying the URL and bypassing the blacklist mechanisms. From the size of the hash, it was possible to identify the algorithm used, whether MD5, SHA-1, or SHA-2. Given this, the feature was considered to be of MODERATE relevance.

4.2. URL morphology

This category describes features that are not necessarily intended to carry out any type of attack, but rather to analyze *morphological aspects of a malicious URL*, such as character patterns and URL size, that result from attacks.

4.2.1. F05. Amount of separators

This feature evaluates the URLs in which the attacker uses a significant amount of character separators, resulting in a URL with a distinctive pattern. Unlike exploitations intended to provide a simulation of veracity, which are addressed in the "User susceptibility" category, the aspects covered in this section refers to the patterns that result from exploitation and which become apparent in the URL morphology. The extracted data are shown in Fig. 13 and the GQM analysis is described in Table 5.

It is important to note that the amount of slash characters "/" was not considered in this feature. In part 1 of the URL, the character is used only in the specification of the protocol (with a double slash), while in part 2. it is used to specify a path, a situation already analyzed previously in feature F04.

As shown in Fig. 13, the identification of the separators was divided by the parts of the URL. In part 1, three commonly used separators, "-", "_", and "@" were identified. It was notable that, in valid phishing records, the incidence is considerably higher. For example, the hyphen had an average of 8.33 occurrences per URL, a value greater than the maximum number of occurrences found

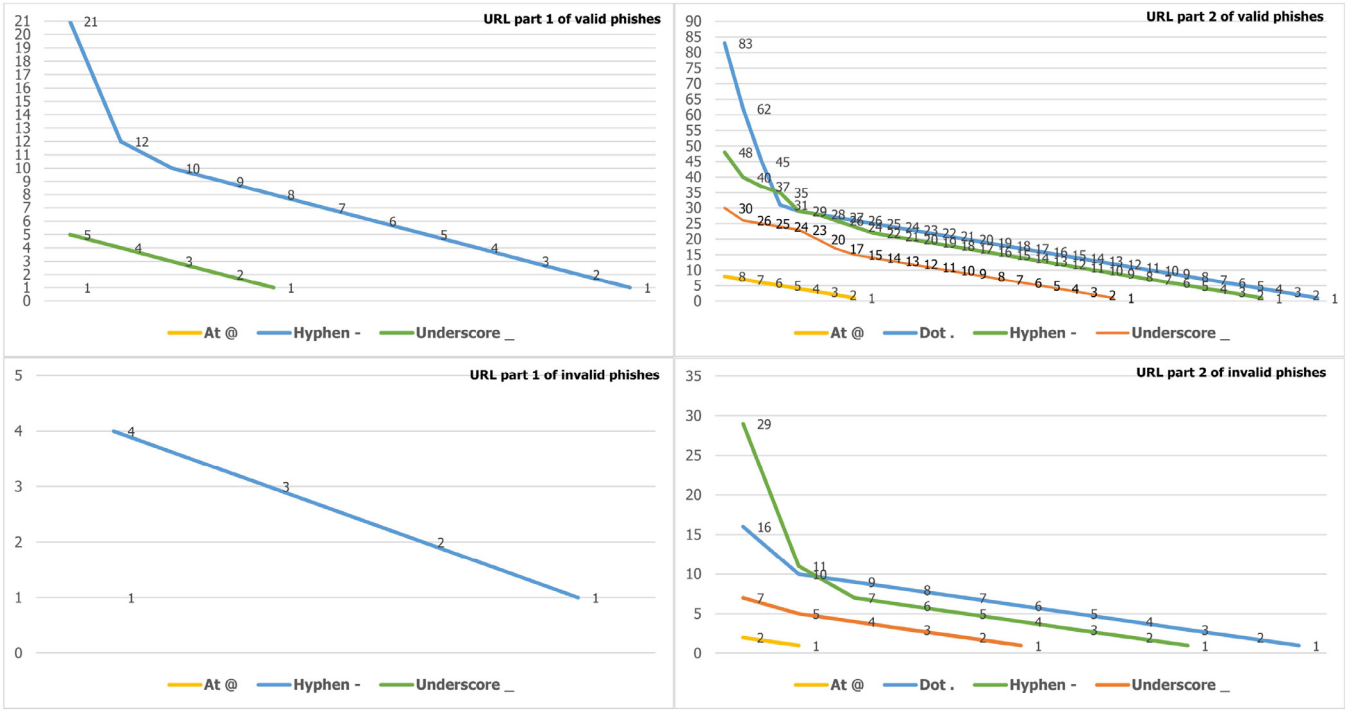

Fig. 13. Occurrences of F05. Amount of separators.

Table 5
GQM of F05. Amount of separators.

Goal 2	Analyze the morphological patterns of a malicious URL.
Question	Q21. Which records have the greatest number of separate characters in the URL's composition?
Metrics	[M19]: Count records that use the "-" character in the first part of the URL of valid phishing sites. [M20]: Count records that use the "_" character in the first part of the URL of valid phishing sites. [M21]: Count records that use the "@" character in the first part of the URL of valid phishing sites. [M22]: Count records that use the "-" character in the first part of the URL of invalid phishing sites. [M23]: Count records that use the "_" character in the first part of the URL of invalid phishing sites. [M24]: Count records that use the "@" character in the first part of the URL of invalid phishing sites. [M25]: Count records that use the "-" character in the second part of the URL of valid phishing sites. [M26]: Count records that use the "." character in the second part of the URL of valid phishing sites. [M27]: Count records that use the "_" character in the second part of the URL of valid phishing sites. [M28]: Count records that use the "@" character in the second part of the URL of valid phishing sites. [M29]: Count records that use the "-" character in the second part of the URL of invalid phishing sites. [M30]: Count records that use the "." character in the second part of the URL of invalid phishing sites. [M31]: Count records that use the "_" character in the second part of the URL of invalid phishing sites. [M32]: Count records that use the "@" character in the second part of the URL of invalid phishing sites.
Hypotesis	Typically, a malicious URL will adopt strategies that use separators in its URL.
Sample	1 and 2 **Relevance** MODERATE **Relations** F04, F07 and F08
Extraction	Get the URL and analyze parts 1 and 2.
Limitations	-
Observations	The URL was divided into two parts, the first consisting of its default protocol, subdomains, domain, and port. The second part contains the paths, querystrings, and filenames. In part 1 the dot character was discarded, as it represents the separation between subdomains, analyzed in F08.
Analysis	It was possible to observe a large number of separators being used in fraudulent sites. Apparently, domains without many characters can raise the suspicion of the end user. This feature reflects on F07, resulting in URLs of considerable size.

in non-malicious URLs. In part 2, in addition to the characters presented, the use of the "." was also observed. The fact that it was not addressed in the first part of the URL is due to its function as a delimiter of subdomains, a situation that will be analyzed with another view in feature F08.

In the second part of the URL, separators are used considerably more than in the first. For example, the hyphen makes up an average of 15.37% of the characters used in a malicious URL, compared to 7.60% in non-malicious. Given this, the feature was considered MODERATE relevance.

4.2.2. F06. HTTP with specification port

This feature evaluates URLs where the attacker uses a port other than the default port. One of the reasons for using a specific port is to bypass the blacklist, since the attacker could periodically modify it to allow the URL to "bypass", however, the hypothesis in question examines whether specifying port usage is common in phishing incidents, thereby being a recurring pattern of URL morphology. In short, the feature in question aims to see if the fact that the site uses a port other than the default strengthens sus-

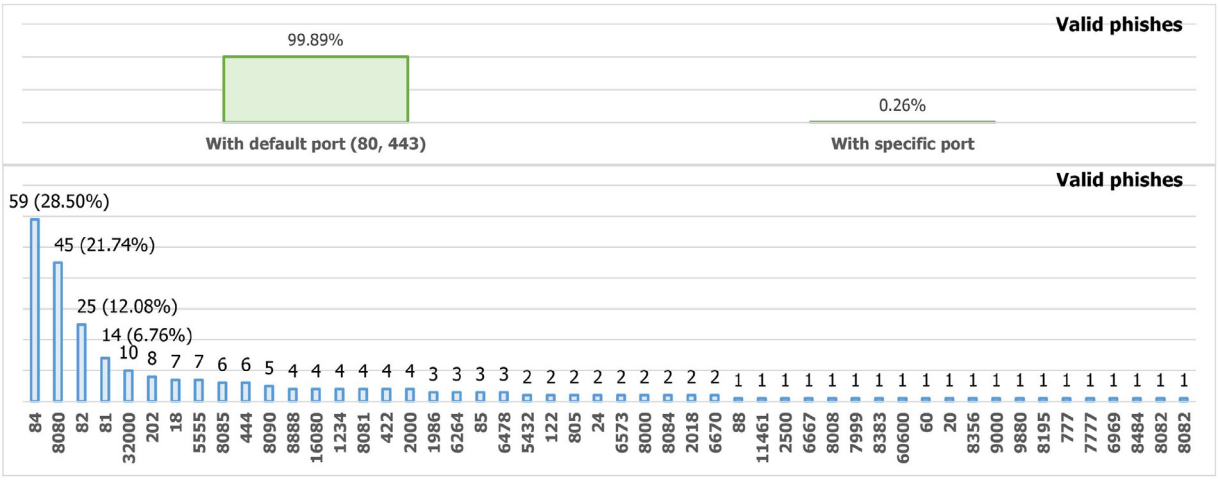

Fig. 14. Occurrences of F06. HTTP with specification port.

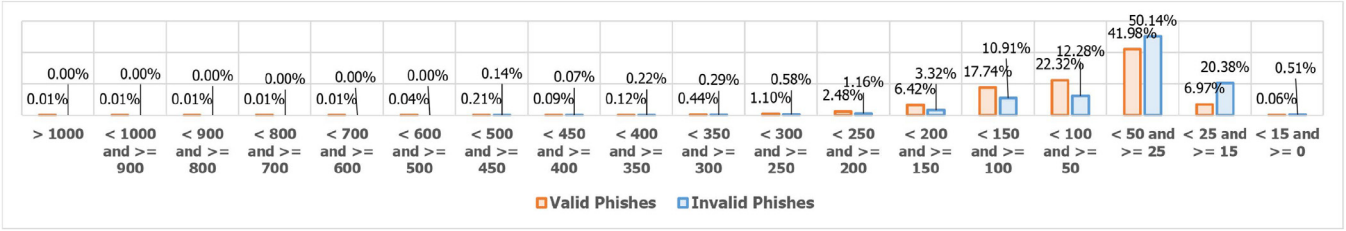

Fig. 15. Occurrences of F07. URL size.

Table 6
GQM of F06. HTTP with specification port.

Goal 2	Analyze the morphological patterns of a malicious URL.
Question	Q22. Which records make use of a specified port to run the HTTP service?
Metrics	[M33]: Count valid phishing records that use ports other than 80 or 443. [M34]: Count invalid phishing records that use ports other than 80 or 443.
Hypotesis	Typically, a malicious URL adopts strategies for using specified ports in its URL.
Sample	1 and 2 **Relevance** WEAK **Relations** F07
Extraction	Get the URL and parse its second part to see if there are mentions of ports other than 80 and 443.
Limitations	There were no entries of invalid phishing sites with specified port usage.
Observations	-
Analysis	This feature is present in malicious URLs, however, not at significant levels.

picions about it possibly being fraudulent. The extracted data are shown in Fig. 14 and the GQM analysis is described in Table 6.

As shown in Fig. 14, it was possible to observe a small number of ports being used in valid phishing records, only 0.26%, as described in the first bar graph. However, what stands out is that there were no specific port usage cases in valid phishing records, making the feature an indicator of a suspicious URL. In addition, port usage is considerably diversified when it occurs, as shown in the second bar graph. Given this, the feature was considered WEAK relevance.

There is no precise way to find out what technology is being used by malicious users to develop a fraudulent site. In contrast, the port could be a possible indicator to help answer this question. For example, many application servers adopt a certain port as the standard one. As specified at the IANA[16] site, port 8080 is commonly used to run services such as tomcat, in other words, suggesting the possibility that fraudulent sites are mostly developed using Java technology.

[16] https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

4.2.3. F07. URL size

This feature evaluates the size of the URL and aims to investigate whether the URL length pattern may raise suspicions that the URL is malicious, because other strategies, such as features F01, F02, F03 and others can result in an increased number of characters in the URL. The extracted data are shown in Fig. 15 and the GQM analysis is depicted in Table 7. Additionally, Fig. 16 shows that the feature also evaluated the size of the domains and subdomains within the respective URLs.

The X-axis of the bar graph in Fig. 15 shows the number of characters, grouped by margins as a scale. The same graph displays the results of this feature as categorical data, that is, it proposes a scale of intervals for the number of characters, as presented on the Y-axis. In this way, it was possible to observe that most URLs have between 25 and 50 characters. One behavior observed was that most valid phishing sites were concentrated in the intervals with the most characters, while invalids were more prevalent as the number of characters was reduced, as shown in the bar graph.

One notable fact is that all URLs over 500 characters was found to be malicious, making this a possible indicator to raise suspicion about a particular site. Given this, the feature was considered to be of MODERATE relevance. In Fig. 16, the X-axis shows the number of

Table 7
GQM of F07. URL size.

Goal 2	Analyze the morphological patterns of a malicious URL.
Question	Q23. Which URL records have a large number of characters?
Metrics	[M35]: Count of the characters in the URL of a valid phishing site. [M36]: Count of the characters in the URL of an invalid phishing site. [M37]: Count of the characters in the domain of the URL of a valid phishing site. [M38]: Count of the characters in the domain of the URL of an invalid phishing site. [M39]: Count of the characters in the subdomain of the URL of a valid phishing site. [M40]: Count of the characters in the subdomain of the URL of an invalid phishing site.
Hypotesis	Typically, many malicious URLs will have a large number of characters.
Sample	1 and 2 **Relevance** MODERATE **Relations** F01, F02, F03, F04, F05, F06, F08 and F12
Extraction	Get parts 1 and 2 of the URL and count the number of characters.
Limitations	-
Observations	-
Analysis	Malicious URLs have a large number of characters when no shortening has been applied due to variable techniques, separators, and subdomains. It has been observed that more than 1% of URLs are composed of over 250 characters. Meanwhile, more than 50% of legitimate URLs have less than 50 characters.

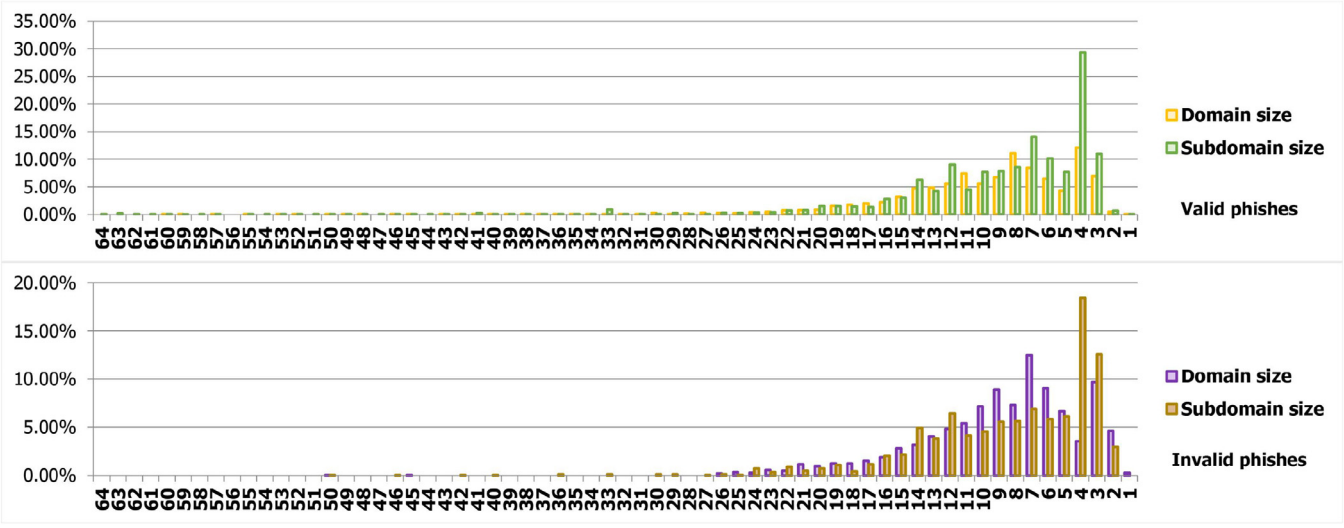

Fig. 16. F07 Domain and subdomain size analysis.

characters used in the domain and subdomain, ranging from 1 to 64 characters, according to the minimum and maximum defined in the DNS protocol. Because a URL can have 1 or more subdomains, each subdomain (delimited by a point) was considered individually. It was possible to observe that both valid and invalid domains have around 7 characters and subdomains have around 4.

4.3. User susceptibility

This category contains features directed towards attacks that seek to provide **greater truth** in content in order to **persuade the end user**. These data will be used to highlight patterns that apply to content page and the composition of the URL.

4.3.1. F08. Concatenate subdomains

This feature evaluates URL patterns that have multiple concatenations of subdomains, in order to lead users to believe, through careless observation, that the URL displayed in the browser is a legitimate domain. For example, *facebook.edit.youraccount.com*, when in fact the domain in question is a record named youraccount.com that was created by malicious intent, and the subdomains edit and facebook are visual effects manipulated by the fraudster. The extracted data are shown in Fig. 17 and the GQM in Table 8.

As illustrated in Fig. 17, it was possible to observe that the use of many subdomains is quite common in phishing attacks. In addition, when comparing between valid and invalid phishing, the incidence is much greater in valid phishing cases, justifying the feature to have STRONG relevance. As shown in Fig. 17, it was possible to detect cases of 25 subdomains in a single URL. The most common keywords in this concatenation referred to consolidated services, such as Facebook or Dropbox, in combination with terms such as security, login, or authentication.

4.3.2. F09. Domain with reputation

This feature evaluates cases where the fraudster has been able to gain control over a particular trusted site and uses its reputation to persuade the end user, i.e. a hijacked domain. Hijacking cases commonly occur due to application sanitization failure, for example, upload sessions that allow for the injection of malicious pages. Cases where the fraudulent site has a domain with a more restricted access registry, such as ".gov" and ".org" domains, also fall under this feature. It is not uncommon to find government and organizational domains being used for crime, either hijacked or officially obtained. The extracted data are shown in Fig. 18 and the GQM is described in Table 9.

As shown in Fig. 18, this type of incident occurs quite frequently and in places all around the world. However, what attracts attention is the high number of hijacked Brazilian domains (.com.br), which are the most exploited and have twice as many occurrences in comparison to the domain in second place. The.org domains are

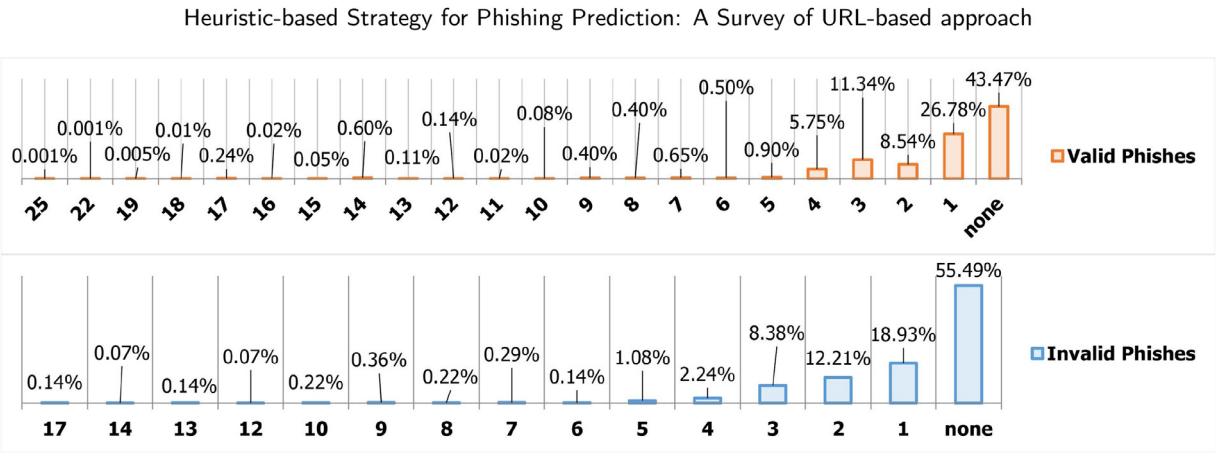

Fig. 17. Occurrences of F08. Concatenate subdomains.

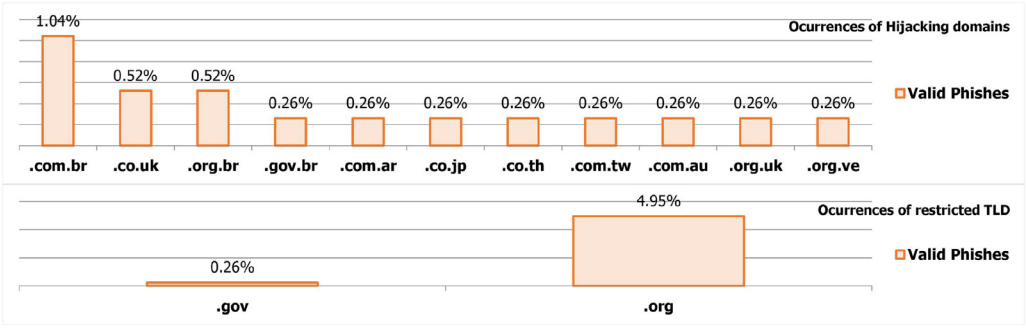

Fig. 18. Occurrences of F09. Domain with reputation.

Table 8
GQM of F08. Concatenate subdomains.

Goal 3	Analyze patterns adopted by attackers to minimize end user suspicion.
Question	Q24. Which records exploit the use of subdomains?
Metrics	[M41]: Count subdomains in valid phishing records. [M42]: Count subdomains in invalid phishing records.
Hypotesis	Malicious URLs will uses numerous subdomains to hide the real domain from the end user.
Sample	1 and 2 **Relevance** MODERATE **Relations** F05, F07 and F10
Extraction	Get part 1 of the URL and count the subdomains used.
Limitations	-
Observations	-
Analysis	To simulate greater trustworthiness, a malicious URL can persuade a user to think that they are browsing a legitimate domain, but which is actually a play on words created through subdomains. This technique can be combined with F10 for further elaboration of fidelity simulation.

Table 9
GQM of F09. Domain with reputation.

Goal 3	Analyze patterns adopted by attackers to minimize end user suspicion.
Question	Q25. What phishing sites have appropriated reputable domains?
Metrics	[M43]: Count the amount of valid phishing sites on domains that have been hijacked. [M44]: Count the amount of valid phishing sites on domains that have restricted registration for organizations.
Hypotesis	In order to simulate greater trustworthiness, a malicious URL may make use of a reputable domain appropriated by the attacker, either through hijacking it or through lack of criteria when registering it.
Sample	1.1 **Relevance** STRONG **Relations** F10, F11 and F12
Extraction	Obtain part 1 of the URL and manually count phishing sites that belonged to a hijacked domain and those registered in organizational domains.
Limitations	It was not necessary to perform comparative analysis between valid and invalid phishing sites. To obtain the data, a manual process was performed. It was initially planned to use the Domage API to obtain the data through WHOIS, however, it was not possible to confirm whether a domain was hijacked or not. In addition, it was necessary to analyze cases where a particular domain had been hijacked that belonged to a restricted domain. In these cases, records were only counted in the hijacked domain metric.
Observations	Domain theft can occur in a number of ways, whether it is exploitation of the application, the hosting server, or the DNS maintainer. On the other hand, this feature is presenting the data in the context of user persuasion.
Analysis	There are a considerable number of cases of site hijacking, especially in Brazilian domains.

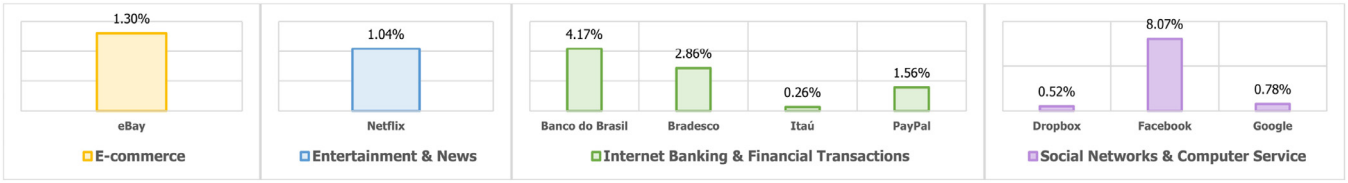

Fig. 19. Occurrences of F10. Homographic attack.

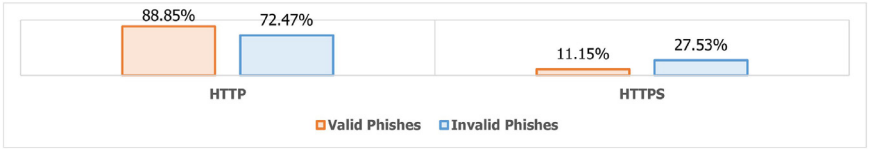

Fig. 20. Occurrences of F11. Tunneling protocol.

Table 10
GQM of F10. Homographic attack.

Goal 3	Analyze patterns adopted by attackers to minimize end user suspicion.				
Question	F10. Which records use word games to appear like a more reliable service?				
Metrics	[M45]: Count of valid phishing records that use service spoofs.				
Hypotesis	To simulate greater trustworthiness, a malicious URL can make use of combined words so that it is trusted by an inattentive user.				
Sample	1.1	**Relevance**	STRONG	**Relations**	F04, F08 and F09
Extraction	Obtain parts 1 and 2 of the URL and analyze them.				
Limitations	It was not necessary to perform a comparative analysis between valid and invalid phishing sites.				
Observations	Words that change the letter "O" to "0" (faceb00k) or other characters that are similar.				
Analysis	Some records were found that used these word games. Social networking services are the most exploited by this trick.				

another point that deserves to be highlighted, which, because they are more restricted, inspire even more confidence on the part of users, making them very attractive to fraudsters and justifying the classification of this feature as having STRONG relevance.

4.3.3. F10. Homografic attack

This feature evaluates cases in which the fraudster makes use of substitutions, which can be words having spelling errors, paronyms, homographs, and homonyms, among others, that are often associated with word games that may pass by an inattentive end user. The extracted data are shown in Fig. 19 and the GQM analysis is in Table 10.

Figure 48 shows brand name exploited with spoofing, that is, occurrences of URLs using words such as "faceb00k", "Netfliiix", or "dr0pbox", a practice referred to as *typosquatting* (Stout and McDowell, 2012). In the case of Facebook, six substitutions were detected, totaling 8.07% of Sample #1.1. Brazilian banks, in particular Banco do Brasil, also stand out with 4.17% of the total. Given this, the feature was considered to have HIGH relevance. Other interesting behavior is the considerable number of domains that are registered to intentionally appear similar to a famous brand, but created by third parties, a practice known as *cybersquatting* (Stout and McDowell, 2012).

The action is not always done with criminal intent, since it is a long time practice to register domains that refer to famous brands and later resell them to the respective representative groups, however, it is still an opportunity for exploitation by malicious users. An interesting stance adopted by some companies, such as Facebook and Netflix, was the appropriation of domains with typosquatting in order to redirect them to the correct page, such as "facbook.com", "fcebook.com", "netfiix.com", "netflix.com", or "netfliix.com" in order to protect their users.

4.3.4. F11. Tunneling protocol

This feature evaluates cases in which the fraudster does not measure efforts to ensure trustworthiness in his fraud, by hiring domains with tunneling resources and sometimes even registering them in digital certifiers. The extracted data are shown in Fig. 20 and the GQM in Table 11.

As shown in Fig. 20, 88.85% of valid phishing sites do not use the HTTPS protocol, whereas with invalid sites, only 27.53% lack the padlock. In view of this, it was possible to consider the feature to have high relevance, however, in 2017 the number of pages with padlocks was only 4.92%, less than half of the occurrences in 2018, where this had increased to 11.15%. This behavior shows that the feature is losing force over time. It is true that free resources, such as *Let's Encrypt*,[17] increase the occurrence of padlocked pages, making it questionable whether it should be considered as a criterion for increased or decreased trustworthiness. Because of this, the relevance of the feature was defined as MODERATE.

4.3.5. F12. URL with redirection

This feature evaluates cases where the fraudster exploits URLs from legitimate sites that enable redirection through *path* or *querystring* manipulation. In practice, the HTTP protocol allows the values of these parameters to be modified arbitrarily during GET requests. In many cases, the application that handles the URL does not handle such entries, enabling a malicious user to place a malicious URL within the legitimate URL. By appearing legitimate, the user may end up trusting the site. However, it will redirect the user to a page that the fraudster inserted into the GET parameters, representing a danger for the end user. Data extracted about this type of attack are shown in Fig. 21 and the GQM analysis is described in Table 12.

As shown in Fig. 21, for both valid and invalid phishing sites, a small number make use of redirection, showing that legitimate pages commonly make use of the practice of redirecting through a URL parameter in almost the same proportion as fraudulent pages,

[17] https://letsencrypt.org/.

Table 11
GQM of F11. Tunneling protocol.

Goal 3	Analyze patterns adopted by attackers to minimize end user suspicion.
Question	F11. Which records make use of HTTPS?
Metrics	[M46]: Count of valid phishing records that use HTTPS. [M47]: Count of invalid phishing records that use HTTPS.
Hypotesis	To simulate greater trustworthiness, malicious URLs will make use of HTTPS.
Sample	1.1 **Relevance** MODERATE **Relations** F09
Extraction	Obtain part 1 of the URL and parse the protocol used.
Limitations	-
Observations	-
Analysis	It was observed that few URLs used HTTPS. However, there are some attackers that do not hesitate to invest so that their fraud appears more reliable to the end user.

Table 12
GQM of F12. URL with redirection.

Goal 3	Analyze patterns adopted by attackers to minimize end user suspicion.
Question	F12. Which records are propagated by making use of URL with redirection?
Metrics	[M48]: Count of valid phishing records that use redirects in their URL. [M49]: Count of invalid phishing records that use redirects in their URL.
Hypotesis	There is an interest on the part of malicious users to have their malicious URLs appear in the parameters of a legitimate URL, making it appear that the malicious URL is trusted because the main URL provides that trust.
Sample	1 and 2 **Relevance** WEAK **Relations** F07 and F09
Extraction	Obtain URLs that have another URL in their path or querystring parameters.
Limitations	-
Observations	In certain cases, the application does not handle the main URL, so it will redirect the end-user to the malicious URL reported for redirection, either in the querystring or path.
Analysis	A considerable number of URLs of this nature have been identified, nearly 5%.

Fig. 21. Occurrences of F12. URL with redirection.

C1. URL blacklist bypass	Scale			C2. URL morphology	Scale			C3. User susceptibility	Scale		
	weak	moderate	strong		weak	moderate	strong		weak	moderate	strong
F01. Encoded exploit	✗	✗	✓					F08. Concatenate subdomains	✗	✗	✓
F02. IP address explosure	✓	✗	✗	F05. Amount of separators	✓	✗	✗	F09. Domain with reputation	✗	✗	✓
F03. Shortened URL	✗	✗	✓	F06. HTTP with specification port	✗	✓	✗	F10. Homographic attack	✗	✗	✓
F04. URL with variables	✗	✓	✗	F07. URL size	✗	✓	✗	F11. Tunneling protocol	✗	✓	✗
								F12. URL with redirection	✓	✗	✗

Fig. 22. Relevance of features.

storing it in the *path* or *querystring*. Given this, the feature was evaluated to have WEAK relevance.

4.4. Relevance analysis

Considering the results obtained, the study pondered the level of relevance to assign to each feature, either *WEAK, MODERATE* or *STRONG*, as shown in Fig. 22. This analysis is not limited only to quantitative aspects, because, in certain situations, subjective aspects, such as content and context, were determinant, resulting in an objective and subjective analysis. Through the data presented, it was possible to observe that a large part of the most relevant features is concentrated in "URL morphology" and "User susceptibility".

The category "User susceptibility" emphasizes that abuse of browser features, subdomain concatenation, domain hijacking, malicious code, and typosquatting in the URL are all factors that can be decisive in reaching a verdict. In the same line, the category "URL blacklist bypass" was only considered to have weak relevance because the behavior in question was little exploited in the sample records, such IP exposure. The occurrences of path and querystring manipulation are situations whose main result is the duplication of the URL, which can be remedied with policies that identify the fraud by the URL domain. Since the features in the "URL morphology" category have their own legitimate importance, however, their particularities do not directly represent a malicious action, but may be consequences of other actions, thus justifying the low relevance.

4.4.1. Relationship and similarities analysis

This section describes the relationships observed between features. Such aspects can directly or indirectly influence the result of each feature, as well as impact on one or more distinct features. Nevertheless, the relationship can be crossed with distinct categories, providing greater sensitivity to the taxonomy with regard to the aspects that are similar between the categories.

Certain combinations describe the use of a language's keywords to attract the attention of victims (F08, F10), whether using arbitrary values in URL parameters (F04), port number values (F06) or practicing typosquatting in the page content and URL (F10), allud-

ing to popular services (F09, F12). Moreover, due to the low acuity of the domain maintainers, malicious users can register domains to gain advantages. One would be to reduce the size of the URL (F01, F02, F03, F07) and consequently increase the SEO score (F09, F11). The practice of registering domains also offers greater freedom to exploit the composition of words in a URL domain (F09) with cybersquatting. However, certain features may increase suspicion, such as the exhaustive use of separators (F05) and subdomains (F08), impacting the size of the URL (F07), and attackers appear to be concerned.

Other combinations of features show that many confirmed phishing sites do not last long enough to receive a final verdict confirming their complaint on the platform. Specifically, in many cases, the complaint submitted on the platform ends up receiving a verdict when the URL reported has already been taken offline. In addition, it would be interesting to adopt better strategies for the complaints platform in order to avoid unnecessary voting, such as in the cases of duplicity as features F02 and F04. For example, when generating a hash for occurrences of F04, the PhishTank platform considers the two parts of URL. Any variation in second part of the URL becomes susceptible to bypass and, in a way, also represents a duplicity, generating unnecessary effort to vote on a URL already confirmed.

Other combinations are aimed at exploring aspects of service reputation, with social engineering regarding as technique that increase trustworthiness (F11). This practice strengthens attacks that are directed at a specific organization (F09), known as Spear Phishing. However, instead of registering, attackers may use a legitimate URL that triggers a redirect to a URL that is found as the value of a GET parameter, thereby taking advantage of the reputation of the legitimate URL domain in question (F12).

5. Threats and study limitations

This section describes the threats and limitations to be considered by the study, which were grouped by objectives and methodology phases.

5.1. Threats from features

Some features may be found in the literature that are not be present in the proposed study. Firstly, through definition of the scope, the study limited the number of features to those that fit the planned extraction time and adopted as criteria the features that were most influenced by trends over time. For example, the Google page rank feature is present in many studies in the literature, however, it has become obsolete since the maintainer decided on April 18, 2016 to no longer make this information available (Dunlop et al., 2010).

5.2. Threats from URL-based phishing taxonomy

In Fig. 3, the categories are classified for the purpose of separately analyzing the context and characters of the URL. Although dismembered, it is possible to observe an intersection between features where one feature will influence others. These behaviors were identified in the "Relations" field of the tables in Section 4. For example, features F01, F03, and F08, if present, can considerably influence feature F07. It is important to stress that, although segmented, they are not both superimposed on the taxonomy structure, thus meeting the prerequisites presented in Section 3.1.3.

5.3. Threats from sampling and data extraction

The process of generating new JSON files considered events from which the PhishTank platform periodically **removed** or **added** entries. It is not possible to accurately answer the reasons for all of the record additions or removals, however, it is possible to explain some of the behaviors.

Regarding removals, the voting system aims to minimize the occurrence of false positives, however, it is still not exempt from this possibility. The platform therefore provides a section where a user can warn about an inappropriate judgment. The platform itself declares that this type of complaint is taken very seriously,[18] and if the error is confirmed, the URL in question is changed from valid phishing to invalid, meaning the record is eventually removed in the next generation of the JSON file. A false positive of a genuine and very popular site would affect the credibility of the platform.

With regard to additions, besides the natural process of the emergence of new phishing sites on the Web, the JSON file can receive phishing sites that were left over from previously pending polls. For example, a URL may have been submitted to the platform, but it will only be considered in the JSON file when the platform has rendered a verdict through voting, which can take hours or even days. In other words, the transition time from "invalid" to "valid" status can vary, causing new records to appear gradually in future files.

These behaviors mean that, in comparison to the most recent months, older ones become more susceptible to removals, as explained in Section 3.2.2. As an example, in a file downloaded on 01/15/2019, the months of January and February of 2018 had, respectively, 358 and 617 records. In the months of November and December of the same year, there were 1524 and 1791 records, respectively. As a form of verification, a search on "phish search", as presented in Section 3.3, showed that the months of January and February had, respectively, 11,503 and 18,953 records. Therefore, this study performed monitoring throughout 2018, observing each update to the JSON file, taking note of the addition of new phishing sites and analyzing the removal of old ones.

5.4. Threats from data results

Because it contains subjective aspects, the scale proposed by the study (*WEAK, MODERATE* amd *STRONG*) results in verdicts defined by interpretations derived from analyzed observations. The study covered contexts that belong to its scope, allowing for different interpretations if another type of context were considered. These limitations will be presented in segments following the structure proposed by the taxonomy.

5.4.1. Threats from URL blacklist bypass results

Some URLs use a shortening service that is no longer active. For those cases, it was not possible to shorten the URL to visit the genuine page or analyze the depth level applied. Therefore, these cases were computed only as occurrences of shortened URLs. In the case of "goo.gle", although it no longer has the option to create shortenings, the service still performs unshortening of already shortened URLs, enabling a depth analysis to be performed.

5.4.2. Threats from URL morphology results

Regarding separator characters, it would be interesting to investigate more possibilities, less common but recurrent, to be able to observe unanalyzed patterns. Regarding the size of the URL, establishing a quantity of characters to define as long or short can be subjective. Therefore, the study was based on an average obtained through the literature, transforming this feature from continuous to categorical.

[18] https://www.phishtank.com/developer_info.php.

Studies	Year	Proposed Taxonomy	C1. URL blacklist bypass				C2. URL morphology			C3. User susceptibility				
			F01	F02	F03	F04	F05	F06	F07	F08	F09	F10	F11	F12
Khonji et al.	2013	✓	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓
AlEroud and Zhou	2017	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
Sharma et al.	2017	✗	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Goel and Jain	2018	✓	✗	✓	✗	✓	✓	✓	✓	✓	✗	✗	✓	✗
Chiew et al.	2018	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓
Qabajeh et al.	2018	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
This Study	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fig. 23. Related works.

5.4.3. Threats of User susceptibility results

An interesting feature for analysis would be to measure the malware detection policy adopted by the browser, looking at page elements and user actions, such as clicks for browsing and downloading. Nevertheless, another important policy to observe would be the installation and use of plug-ins and extensions. Many services, especially banking, adopt hardening strategies on the client side through plug-ins or extensions that are owned or maintained by third parties, such as Brazilian banks using the Warsaw solution[19]. In this context, it would be interesting to analyze the efficiency of these solutions.

With regard to tunneling phishing records, it was not considered whether the digital certificate in question was revoked or not. The study extraction process did a lexical analysis of the URL that considered sample # 1. To analyze the validity of certificates, we would need to extract from sample # 1.1 due to the need to observe the browser response on each certificate.

With regard to malicious redirects, another exploited medium that was not part of the scope of this study were cases of the attacker filling out false profiles in an environment that shares personal profiles, such as social networks, forums, etc. It is not uncommon for this type of environment to allow the user to enter a URL into their profile, indicating a possible personal page. In these cases, the application does not usually check the origin of the URL, allowing it to propagate a malicious URL through the shared environment.

6. Related works

In this section, studies found in the literature that have solutions related to the proposal of this study will be described. One of the motivations for the development of this *survey* was to demonstrate the static behavior of heuristics that use the 12 proposed features. Fig. 23 illustrates the related studies and the relationship between their contributions and the objectives of this study. Like the present study, almost all related studies also proposed a taxonomy.

In Khonji et al. (2013), techniques for phishing detection are presented that consider the human factor, and an overview is provided of the detection technique categories, resulting in a taxonomy. The techniques consider the approaches of offensive defense, correction, and attack prevention. In the study by AlEroud and Zhou (2017), the main focus is to present a taxonomy of phishing attacks and anti-phishing techniques, which results from a literature review. This study differs from others because the proposed model considers emerging techniques, specific environments, and countermeasures that mitigate new types of phishing. The purpose of the taxonomy is to provide guidance to incident combat teams on detection techniques.

[19] https://www.dieboldnixdorf.com.br/gas-antifraude.

Sharma et al. (2017) propose the comparison of eight tools for phishing detection, subjecting them to a sample that evaluates the efficiency of each one. The sample was extracted from bases such as APWG and PhishTank. The analysis also compares the phishing sample against another sample of 500 legitimate sites. The study by Goel and Jain (2018) evaluates some anti-phishing mechanisms for mobile devices, dividing them into four steps, namely: (i) detailing the context of the attack for mobile devices; (ii) analyzing the types of attacks involved, resulting in a taxonomy of attacks; (iii) defining a taxonomy of countermeasures for the attacks in the previous taxonomy; and finally (iv), discussing the challenges of combating phishing.

In Leng Chiew et al. (2018), a discussion of phishing attack execution approaches is presented, in order to improve the understanding of the features used. The study also intended to point out existing gaps in the heuristic proposal segment. The organization and arguments presented resulted in a taxonomy. Finally, the study by Qabajeh et al. (2018) analyzes the heuristics of anti-phishing solutions, taking into consideration legal aspects, training, awareness, and intelligent approaches. In addition, it also highlights positive and negative aspects of the performance of these mechanisms. The paper also states that the study's outcome serves as a support for the development of such solutions.

7. Conclusion and further works

This study presented a survey as a methodology to obtain evidence regarding certain behaviors and analyze them using in samples extracted from the real phishing environment. This evidence was described using graphs and argued based on GQM metrics. Considering that there are more than a few phishing prediction proposals, the problem remains chronic today, justifying the need for and applicability of these solutions. Because many of the solutions are guided by a set of features, the present study, as a reflection, analyzes the relevance of certain features commonly used in the prediction process. The study divided the features into types that considered the lexical structure of the URL, the context, the content, and similarities.

7.1. Challenges of the URL blacklist bypass

Regarding the URL-based blacklist bypass features, a very modest number were present for cases where the valid phishing exposed the default port or the IP address, but these were also repeated in some cases of invalid phishing, lowering the relevance to WEAK for these features. On the other hand, although encoding exploitation seldom occurred and was somewhat out of the ordinary, it was considered STRONG. Similarly, with a reasonable number of occurrences, shortened URLs and abuse of *path* or *querystring* variables, were considered to have MODERATE relevance, because, although they also appear in some invalid phishing cases, at a certain level or depth they will not be found at conventional sites.

7.2. Challenges of the URL morphology

For features based on URL morphology, behaviors such as the number of separators and the URL size also raise suspicions about certain URLs when used extensively. In contrast, the number of URLs that used a port other than the default was almost non-existent, and thus a low relevance.

7.3. Challenges of the User susceptibility

Finally, with regard to the features based on "User susceptibility", the behaviors that most attracted attention were the concatenation of subdomains and *cybersquatting* or *typosquatting* in domains, aspects that are increasingly being exploited in recent years. In contrast, deciding whether a page is malicious or not based on the absence or presence of the padlock eventually loses its strength over time, as does relying on the SEO score of certain available services. As already mentioned, site hijacking cases have raised Brazil to be a prominent region exploited by phishing attacks.

7.4. The Road Ahead for the Heuristic URL-based

Because it extracted a considerable number of real phishing sites and data, the analysis carried out by this study considered mostly quantitative aspects, as shown in the graphs. Nevertheless, the study also offered qualitative results through the consideration of content and context, as well as the determination of relevance and similarities. With this data, it was possible to conclude that temporal aspects, in the perspective of this study, influenced the relevance of the commonly-adopted heuristic for prediction.

The study can provide support for the development of a model evaluator that uses as evaluation metrics those already presented, such as sensitivity, specificity, and efficiency, as well as other metrics such as prediction value and coefficient of variation, in order to judge the maturity of the precision of the proposed new model. The evaluator would have the role of assigning weights to the features of the heuristic, representing the factors of relevance, proposing something like a maturity model for new prediction models. The definition of a classification model has, among its challenges, some problems that can be minimized through the results of this study, such as (i) **categorization**, (ii) **relevance analysis**, and (iii) **grouping of features**.

In light of (i), a feature can be considered to be a variable with either continuous or categorical value. An example of a categorical value would be whether or not a URL has tunneling (F11). A variable with a continuous value would be, for example, the result for URL size (F07), that is, it can present a different value for each URL. As the classification model needs to deal with categorical values, it would be necessary to transform the continuous variables into categorical ones. Through descriptive statistics, the results of this study can bring a new perspective to the process of converting these variables, for example the scale of size intervals presented in Fig. 15. In addition, the URL-based taxonomy proposed considers aspects of the context that are biased in static behaviors. That is, the features proposed are not restricted to the lexical or content aspects of the malicious page.

Based on (ii), and considering the anti-phishing context, it is important to assign a weight, as discussed in Section 4.4, so that a given set of variables can define a class for pages accessed. As the phishing environment is susceptible to concept drift, it is important to consider the content and context in the definition of relevance, in addition to static aspects such as lexical source code and URL patterns. Content refers to trends and patterns of behavior, while in terms of context, it consists of anti-phishing strategies, activity, time, computational resources, and seasonality. Another important criterion to consider is the occurrence of certain combinations of features.

Finally, in (iii) the problem of grouping the features served by the classification model is addressed. In this context, (iii) serves as a base of support for (i) and (ii), that is, when in possession of the data from this study, it is possible to perform a cluster analysis for greater sensitivity to similarities in the context between the features, as discussed in Section 4.4.1. Along the same lines, a well-defined grouping avoids overlapping attributes that work with a relevance evaluation.

Closed-scope phishing attacks, such as spear phishing, because of their non-disseminated nature, suggest a predictive approach that is more directed to the context of the attack. Despite this, it is possible that, in a way, the results of this study can support prediction models focused on an organization's **brand protection**. Such a solution aims to monitor aspects of **nominal identity**, i.e. **cybersquatting** and **typosquatting** in domains and subdomains, as well as false publications on social networks and use of keywords in search engines. In addition, it also aims to protect **visual identity**, that is, the abuse of elements that represent the organization visually, such as templates and logos.

However, the study did not have a way to present the **intrinsic features of the target brand**, a gap which could be analyzed in future studies. In short, pillars (i), (ii), and (iii) are intended to provide support to the **response capacity** and **response time** of the proposed heuristic model, as mentioned in Section 1.

Acknowledgments

This research was partially funded by INES 2.0, FACEPE grants PRONEX APQ 0388-1.03/14 and APQ-0399-1.03/17, CAPES grant 88887.136410/2017-00, and CNPq grant 465614/2014-0.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cose.2019.101613.

References

Aburrous, M., Hossain, M., Thabtah, F., Dahal, K., 2008. Intelligent phishing website detection system using fuzzy techniques. In: Proceedings of 3rd International Conference on Information and Communication Technologies: From Theory to Applications.

Afroz, S., Greenstadt, R., 2011. Phishzoo: detecting phishing websites by looking at them.. ICSC. IEEE Computer Society.

AlEroud, A., Zhou, L., 2017. Phishing environments, techniques, and countermeasures: a survey. Comput. Secur.

Alkhozae, M.G., Batarfi, O.A., 2011. Phishing websites detection based on phishing characteristics in the webpage source code. Int. J. Inf. Commun. Technol.

Almomani, A., 2018. Fast-flux hunter: a system for filtering online fast-flux botnet. Neural Comput. Appl. 29 (7), 483–493. doi:10.1007/s00521-016-2531-1.

Amiri, I.S., Akanbi, O.A., Fazeldehkordi, E., 2014. A Machine-Learning Approach to Phishing Detection and Defense. Syngress.

Basili, V.R., Caldiera, G., Rombach, H.D., 1994. The goal question metric approach. Encyclopedia of Software Engineering. Wiley.

Chaudhry, J.A., Chaudry, S.A., Rittenhouse, R.G., 2016. Phishing attacks and defences. Int. J. Secur. Appl.

Chelliah, G.A., Aruna, S., 2014. Preventing phishing attacks using anti-phishng prevention technique. Int. J. Eng. Dev. Res.

Dunlop, M., Groat, S., Shelly, D., 2010. Goldphish: using images for content-based phishing analysis. In: 2010 Fifth International Conference on Internet Monitoring and Protection, pp. 123–128. doi:10.1109/ICIMP.2010.24.

Elwell, R., Polikar, R., 2011. Incremental learning of concept drift in nonstationary environments. IEEE Trans. Neural Netw 22 (10), 1517–1531. doi:10.1109/TNN.2011.2160459.

Goel, D., Jain, A.K., 2018. Mobile phishing attacks and defence mechanisms: state of art and open research challenges. Comput. Secur. 73, 519–544. doi:10.1016/j.cose.2017.12.006.

Google, 2019. Safe browsing. Available at: https://safebrowsing.google.com/.

Gowtham, R., Krishnamurthi, I., 2014. A comprehensive and efficacious architecture for detecting phishing webpages. Comput. Secur. 40. 23 to 37.

Jain, A.K., Gupta, B.B., 2017. Phishing detection: analysis of visual similarity based approaches. Secur. Commun. Netw.

Kaspersky, 2014. What is a phishing attack? Available in: https://goo.gl/4EEtxk.
Khonji, M., Iraqi, Y., Jones, A., 2013. Phishing detection: a literature survey. IEEE Commun. Surv. Tutorials 15 (4).
Kirda, E., Krugel, C., 2005. Protecting users against phishing attacks. Comput. J.
Leng Chiew, K., Yong, K., Tan, C.L., 2018. A survey of phishing attacks: their types, vectors and technical approaches. Expert Syst. Appl 106. doi:10.1016/j.eswa.2018.03.050.
Ma, J., Saul, L.K., Savage, S., Voelker, G.M., 2009. Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York, NY, USA, pp. 681–688.
Mohammad, R.M., Thabtah, F., McCluskey, L., 2015. Tutorial and critical analysis of phishing websites methods. Comput. Sci. Rev. 17 (C).
Molleri, J.S., Petersen, K., Mendes, E., 2016. Survey guidelines in software engineering: an annotated review. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, New York, NY, USA, pp. 58:1–58:6. doi:10.1145/2961111.2962619.
Moore, T., Clayton, R., 2007. Examining the impact of website take-down on phishing. In: Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit. ACM, pp. 1–13.
Naresh, U., Sagar, U.V., Reddy, C.V.M., 2013. Intelligent phishing website detection and prevention system by using link guard algorithm. IOSR J. Comput. Eng.
OpenDNS, 2019. Phishtank. Available at: https://www.phishtank.com/.
Qabajeh, I., Thabtah, F., Chiclana, F., 2018. A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Comput. Sci. Rev. 29, 44–55. doi:10.1016/j.cosrev.2018.05.003.
Babbie, R.E., 2019. Survey research methods/earl r. babbie. SERBIULA (sistema Librum 2.0).
Robson, C., 2002. Real World Research - A Resource for Social Scientists and Practitioner-Researchers, 2nd Blackwell.
Schneier, B., 2013. Phishing has gotten very good. Available at: https://bit.ly/2O2wnR0.
Sharma, H., Meenakshi, E., Bhatia, S.K., 2017. A comparative analysis and awareness survey of phishing detection tools. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp. 1437–1442.
Srinivasa, R., Alwyn, R., Pais, R., 2019. Jail-phish: an improved search engine based phishing detection. Comput. Secur.
Stout, B., McDowell, K., 2012. United States Patent. Technical Report. Citizenhawk, Inc., CA (US).
Vayansky, I., Kumar, S., 2018. Phishing challenges and solutions. Comput. Fraud Secur. 2018, 15–20. doi:10.1016/S1361-3723(18)30007-1.
Whittaker, C., Ryner, B., Nazif, M., 2010. Large-scale automatic classification of phishing pages. NDSS '10.
Windows, 2019. Windows smartscreen. Available at: https://bit.ly/2ER8yow.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell, MA, USA.

Carlo Marcelo Revoredo da Silva has a degree in Systems Analysis and Development from the Union of Brazilian Institutes of Technology - Unibratec (2009), Specialization in Information Security in Software Engineering from the Center for Advanced Studies and Systems of Recife - Cesar.edu (2012) and MD in Computer Science from Federal University of Pernambuco - UFPE (2014). He is currently assistant professor from the University of Pernambuco (UPE) at Campus Garanhuns (2018), Ph.D student in Computer Science from Federal University of Pernambuco (2018) and chapter leader of OWASP Recife. Has experience in Computer Science, focusing on Software Engineering and Information Security.

Eduardo Luzeiro Feitosa has a Ph.D. in Computer Science from the Federal University of Pernambuco (2010), a MD in Computer Science from the Federal University of Rio Grande do Sul (2001) and a degree in Data Processing from UFAM (1998). He is Associate Professor at the Institute of Computing (IComp) and Permanent Professor of the Graduate Program in Informatics (PPGI) of the Federal University of Amazonas (UFAM). He is one of the leaders of the Emerging Technologies and Systems Security (ETSS) research group. It works in partnership with several research groups at home and abroad and has coordinated projects with national and international institutions. He is a reviewer of various journals and member of committees of various conferences. He is coordinator of the Graduate Program in Informatics (PPGI) of UFAM. Member of the Special Committee on Information Security and Computer Systems (CESeg) of SBC.

Vinicius Cardoso Garcia has a degree in Computer Science from Salvador University (2000), a MD in Computer Science from Federal University of Sao Carlos (2005) and a PhD from Federal University of Pernambuco (2010). He is an Associate Professor at Centro de Informática of UFPE since 2010. Besides, he is an associate researcher at INES (National Institute of Science and Technology in Software Engineering) and he worked as a software and systems engineer and software reuse consultant at CESAR (Center for Advanced Studies and Systems of Recife) from 2005 to 2010, where he carried out several industrial projects focused on various aspects of software engineering. Since 2010 he has been working as a researcher in agreements and partnerships with startups, small and medium sized companies of Porto Digital (an initiative in Recife, Brazil to foster technological innovation in the northern region of Brazil) in the areas of Cloud Computing, Software-Defined Storages, Site Reliability Engineering and Continuous Software Engineering with a primary focus on improvements for the Brazilian industry and state improvement. practice of these areas.