Heuristic-based Strategy for Phishing Prediction: An Empirical Research of Context-Aware Approach

ARTICLE INFO

Keywords:
Phishing
Social Engineering attacks
Cybercrimes
User Susceptibility
Heuristic prediction

ABSTRACT

Context: It is not uncommon for many studies in the literature to report URL-based features for phishing prediction. By adopting a static URL-only analysis, this approach has its performance benefits, however, it also has limitations on accuracy. features of a dynamic nature, that is, that consider behaviors in phishing content, time and context of action, can contribute to the accuracy in anti-phishing solutions. **Objective**: The present study reports the relevance, relationship, and similarity of dynamic features to contribute to the direction of phishing prediction proposals that need to improve accuracy and thus minimize false positive and negative problems. **Method**: The study is conducted by an empirical research that considers actual phishing samples extracted from the PhishTank repository. The methodology is driven by hypothesis testing that analyzes behaviors and compares them in confirmed phishing samples and legitimate sites to investigate whether or not the features involved are present in both samples. **Results**: It was observed that certain features are more or less relevant to be adopted in prediction strategies. Besides, some features have strong relationships or similarities with each other. **Conclusion**: The study developed an empirical research of quantitative and qualitative nature, in which they propose to contribute their results to new proposals in the literature that perform phishing prediction considering dynamic aspects as defined by the study.

1. Introduction

Commonly-applied strategies for web phishing detection are based on two types of approaches: blacklists (i) (Vayansky & Kumar, 2018; Whittaker et al., 2010) and prediction-based (ii) (Alkhozae & Batarfi, 2011; Chelliah & Aruna, 2014). In general, the tactic employed in (i) is to make use of whistleblowing platforms that act as third-party services, periodically providing blacklists to computational resources on the client side of protection, for example, web browsers, IDS, and IPS (Gowtham & Krishnamurthi, 2014). In (ii), the strategy is to make use of phishing prediction heuristics based on features patterns, such as anti-spam filters.

Solutions based on (i) have challenges in detecting newly-created phishing sites, known as **zero-day phishing** (Srinivasa et al., 2019; AlEroud & Zhou, 2017). Because phishing naturally has a volatile life cycle and high propagation, it is not uncommon for false negatives to occur (Srinivasa et al., 2019; AlEroud & Zhou, 2017), representing a **window of vulnerability** in the incident response for this type of solution. As a result, proposals based on (ii) have gained momentum and, in the current state of the art, have become a trend as an alternative to mitigate the gaps caused by zero-day phishing.

In contrast, solutions based on (ii) have problems related to **privacy** and **prediction rate**. Although the approach may consider only the URL, in some cases, heuristics may rely on features extracted from the page content or visual patterns in order to address false negatives, increasing the chances of violating end-user privacy (Chaudhry et al., 2016). Even a URL-only-based approach can still violate privacy, depending on its use of analytics.

Regarding the prediction rate, the problem is skewed in **responsiveness** and **response time** (Chelliah & Aruna, 2014; Kirda & Krugel, 2005; Naresh et al., 2013; AlEroud & Zhou, 2017; Abdelhamid et al., 2014; Moghimi & Varjani, 2016; Adebowale et al., 2019). Responsiveness can be assessed through metrics such as **sensitivity**, **specificity**, **efficiency**, **accuracy**, and **precision**. Sensitivity and specificity are generally opposed to each other. High sensitivity to positives may result in a high number of false positives, and vice versa; therefore, a balance needs to be established. The equilibrium between these is efficiency and these metrics are all quantitative. Accuracy is the ability of the model to provide, after a series of measurements, an average value that is close to reality, without considering false positives or negatives. Precision is the ability of the model to have low dispersion between measurements when under similar conditions. Therefore, accuracy and precision are qualitative-type metrics that work together.

Prediction response time describes how fast the heuristic responds to the incident by increasing or decreasing the 0-day phishing vulnerability window. However, speed is one aspect that can generate varied interpretations, whether with regard to the minimum time to respond to the incident or the performance of the prediction processing. The minimum response time refers to the window of vulnerability, while performance is the computational cost of the operation (Li et al., 2019). Due to the wide variety of browsers available, the performance becomes a relevant criterion with regard to browser choice, that is, heuristic solutions need to balance the number of features adopted.

To deal with the challenges mentioned, it is not uncommon to employ anti-phishing mechanisms that adopt a hybrid solution between (i) and (ii), exploiting the positive aspects of each approach. However, a much-debated phenomenon in the literature, the so-called concept drift (Elwell & Polikar, 2011), which describes concept changes in an active dynamic scenario, may compromise the assertiveness of the prediction. Considering the volatility and widespread occurrence of phishing (Almomani, 2018; Moore & Clayton,

ORCID(s):

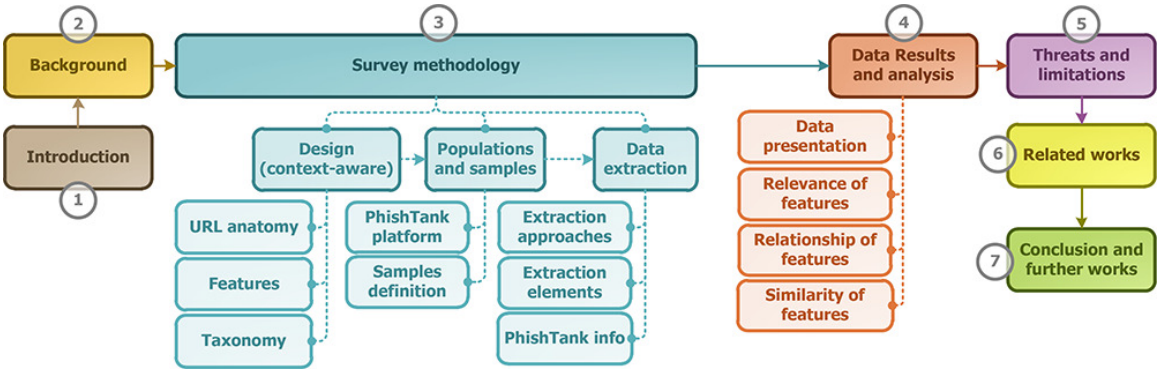

Figure 1: Study structure

2007), these aspects provide a very dynamic scenario, that is, a scenario highly susceptible to change. Such behavior makes it necessary to rethink the importance of each feature employed in the heuristic model.

Beyond this, **dynamic features**, i.e., temporal and behavioral features, observed in approach (i) can be employed as features in approach (ii) to strengthen the hybrid model. It is interesting to investigate the use of such features as a way to enhance a heuristic model, taking context-sensitivity into account and identifying the relevance, relationship, and similarity between them.

The study investigates the relevance, relationship and similarity of **dynamic features** commonly present in phishing sites. It is also proposed a taxonomy that structures and relates identified behaviors, based on 16 features and grouping them into 4 distinct categories identified in the context of phishing performance, namely: web browser, the intervention of the community platforms, life cycle and trend profiles.

The study looked at actual (valid) phishing samples and legitimate (invalid) sites and found features that consolidate into valid and lack invalid ones as a way to justify that the features in question are commonly found in phishing attacks. The samples were obtained from the PhishTank footnote repository https://www.phishtank.com/. Thus, it was possible to present a report that describes **relevance**, **relationship** and **similarity** between the observed features. The flow adopted by the methodology can be observed in Figure 1 and described in more detail on Section 3.

The study is structured as follows: Section 2 describes a macro view of the problem, as well as the main strategies to mitigate the problems addressed. Section 3 presents the methodology adopted by the study to conduct the empirical research on the dynamic features. In Section 4 we present the quantitative result in graphs and the qualitative analysis on the relevance, relationship, and similarity between the features. In Section 5 the study presents the threats that should be considered in the empirical research. Section 6 describes the studies related to the present study. Finally, Section 7 presents the final considerations and perspectives towards new prevention approaches that may benefit from the results of the present study.

2. Background

According to the Association of Certified Fraud Examiners (AFCE) (ACFE, 2018), fraud is an action that represents any kind of intentional or deliberate act of depriving property or money through cunning, deception or other unjust acts. According to *Kaspersky* (Kaspersky, 2014), *phishing* is an attack characterized by fraudulent attempts against Internet users. At this point, an attacker develops a malicious page that proves itself to be legitimate in order to exploit end-user susceptibility to intercept sensitive data. This type of attack has been on the rise in recent years, according to quarterly reports from *Anti-Phishing Working Group (APWG)* [1].

As mentioned in Section 1, many of these mechanisms are based on hybrid protection, which identifies the threat through blacklists or predicts the threat using a feature-driven heuristic. These blacklists are commonly fed from voluntary reports that are maintained by a community-based platform (Whittaker et al., 2010; OpenDNS, 2019), but this is not all. SafeBrowsing declares that it periodically scans the web (Google, 2019) to investigate patterns and, if it deems a site to be malicious, the page URL will be entered into the list. As a result, hybrid solutions adopt the use of blacklists for performance reasons, and these lists are fed by user complaints and systematic prediction assessments. Figure 2 illustrates a typical scenario for such a solution.

In general, the size of a blacklist to be stored in the web browser can be drastically reduced by converting its component URLs to a collision-resistant hash, such as SHA-256. In this way, every time the browser attempts to render a page, the protection engine will generate the hash prefix of the page in question (the first 8 bits of the hash) and check if the combination exists in the blacklist. If so, a crash test is performed using the hash prefix and suffix. Because the list needs to handle as many incidents as possible, i.e., it needs to store old and recent phishing, the result is a large amount of data in a single file, justifying the use of a two-step hash checking process. If a collision occurs, the user will be alerted of the danger, and some solutions adopt a self-reporting method, reporting the newly-identified threat

[1] https://apwg.org/resources/apwg-reports/

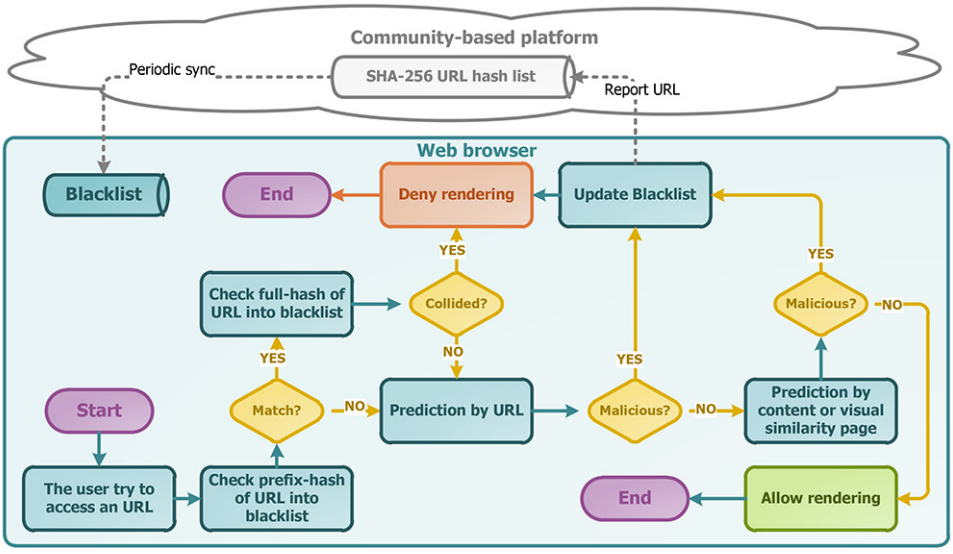

Figure 2: The conventional hybrid-mechanism workflow

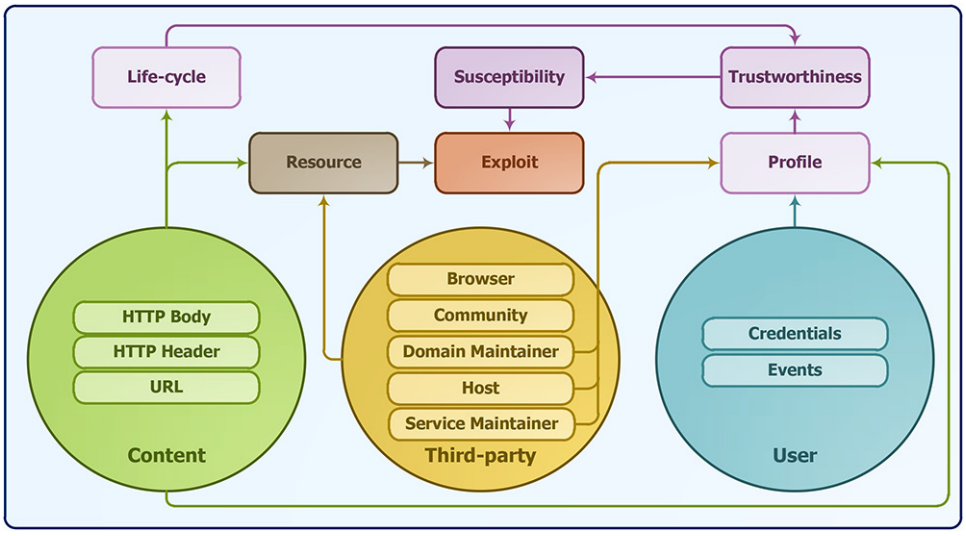

Figure 3: The phishing context-aware approach

to the community for assessment.

An interesting observation that can be made from Figure 2 is the gradual depth strategy that can be adopted to improve performance and mitigate the violation of end-user privacy. The predictive heuristic usually has an initial strategy of investigating only the URL, to make processing less costly and inspection of user navigation less intrusive, unlike an analysis that considers the content or visual patterns of the page itself. However, conclusions limited to the URL can be inaccurate, causing the heuristic to require more context sensitivity to reach a more accurate verdict about the suspicious page and thereby minimize the occurrence of false positives and negatives. When the context is taken into consideration, some elements are related to one another, as in an ecosystem. This approach can be defined as context-sensitive, as shown in Figure 3.

The context-sensitivity perspective divides information for extraction into three groups, namely: content (i), third-party agents (ii), and the end-user (iii). In (i), the HTTP protocol must consider aspects for extraction, such as the URL structure and the header or body of the page. With regard to (ii), there are collaborative actors in the ecosystem, such as the web browser, complaint platforms, domain registrars, hosting services, and other service maintainers. Finally, (iii) has information and actions that may be of interest to malicious users, such as the existence of a 6- or 8-digit password.

It is important to note that groups (i) and (ii) result in features that can be directly exploited by malicious users, i.e., without the need for direct end-user intervention (Whittaker et al., 2010). Examples of this nature are applications with upload forms that allow one to inject a malicious page due to a failure to sanitize entries. Moreover, in (i), the data has

a life cycle, and all groups have an information profile. Such elements have behaviors that can provide a greater degree of trustworthiness to a phishing site information p. Trustworthiness is associated with richness of detail that matches the genuine page that is the target of fraud, and the higher the quality of information extracted from the profile, the greater the trustworthiness. In the end, trustworthiness diminishes suspicions of fraud by increasing end-user susceptibility the purpose, improving the odds of successful exploitation.

The academic literature reports techniques that take context-aware approaches for constructing prediction heuristics. However, there is considerable effort in the literature to promote a satisfactory sensitivity to the point of identifying not only previously defined aspects, as well as perceiving unexpected changes in the performance context, a common behavior in a dynamic scenario such as the performance on phishing environment. Such changes can be defined as Concept Drift (Elwell & Polikar, 2011). Briefly, there are four distinct types of concept drift, as follows:

- **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.

Based on the above, it is interesting to investigate the dynamic behaviors present in the blacklist and prediction-based approaches, in order to observe aspects that contribute to reduce the gaps in the current state of the art in anti-phishing solutions.

3. Methodology

This study aims to investigate phishing behavior patterns in real environments. Because the observation scenario does not intervene, and the variables (dependent and independent) are not controlled, the study adopted the *empirical* methodology for the investigation, as suggested in the studies by Molleri et. al (Molleri et al., 2016), Robson (Robson, 2002) and Babbie (R. Babbie, 2019).

According to Wohlin et al. (Wohlin et al., 2000), a empirical research is an approach to experimental software engineering that investigates evidence by observing behaviors in a real environment. In line with the proposed study, the observations would be directed towards dynamic behaviors of the scenario in question, which would result in dynamic feature candidates that belong to a prediction heuristic. To make the study decisions more accurate, a comparative analysis is planned, analyzing a confirmed phishing site and a genuine page, to verify that the behavior identified only occurs when the site is fraudulent, avoiding chance hypothesis-driven conclusions (Molleri et al., 2016).

3.1. Data Design

This section describes the planning and nomenclature adopted in the methodology, and delimits the scope and objectives. At this stage, processes will be defined as obtaining primary data, extracting behaviors, and defining and structuring features. Using the context-aware approach, it was possible to categorize the features according to their relationships and similarity, resulting in a taxonomy.

3.1.1. Data Terminologies

The data collection process considered some terminology that was used in the remainder of the study. For example, for obtaining and extracting information from URLs, the study was based on the anatomy illustrated in Figure 4, which divides the URL into two distinct parts, based on the strategies and extractions applied to each type of URL feature. A large amount of information can be extracted from a URL (da Silva et al., 2019). Because of this, the study was concerned with formalizing the strategy for obtaining and extracting data.

Part 1 of the URL is its most static structure, i.e., it is less susceptible to attacker manipulation, containing the protocol, host, and port. Among the elements mentioned, the host has several peculiarities in its composition that deserve mention. When an IP address does not reference the host, it must consist of a domain and, optionally, a subdomain. A domain is the combination of a top-level (gTLD), which represents the country or segment, and a second level (SLD), which is an element named by the site owner through DNS name servers. These latter elements are the fuel for attacks because they give the attacker the freedom to exploit the susceptibility of victims using word combinations.

Part 2 of the URL contains more dynamic information, meaning that attacker has a little more freedom to perform malicious manipulations to exploit the susceptibility of end-users. To give an idea, all elements of this part are optional, meaning that the URL can consist of only the first part, while still being functional on the web. This means that all elements that make up the second part of the URL, such as the resource path, file name, querystring, and anchors, are arbitrary for the page owner. This provides attackers with numerous possibilities. Because the blacklist-based approach identifies malicious URLs through their hash values, any modification to the URL characters results in a different hash, rendering this type of approach ineffective.

Behaviors tied to both parts of the URL can be identified through a regular expression, as the HTTP protocol sets syntax patterns in the use of each element in the URL. For example, the protocol always follows ": //", while port values are preceded by ":". Querystring variables are preceded by "?" and when there are multiple variables specified, they

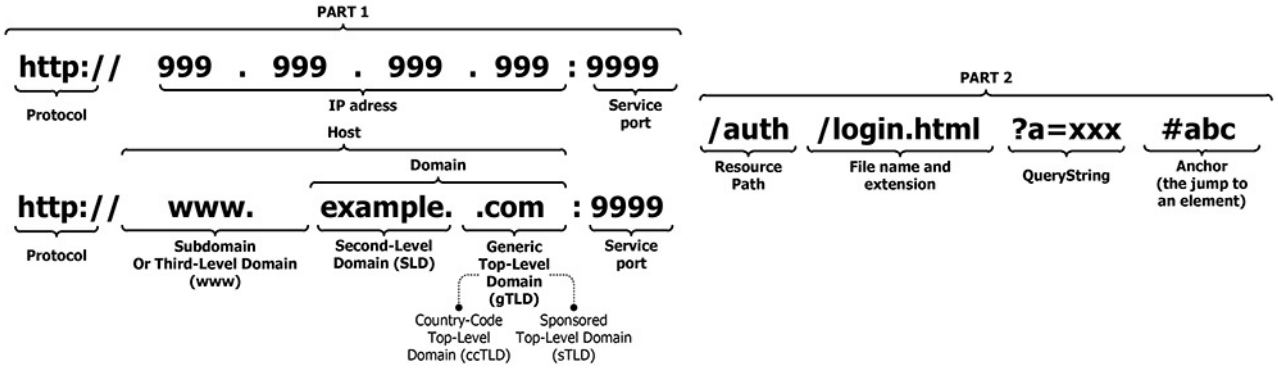

Figure 4: URL anatomy

will be separated by the "&" character. Similarly, path elements are preceded by "/" and finally, page anchors are referenced by "#" and are always found at the end of the URL. However, a difficulty was observed, such as in the case of URL encoding[2]. For example, if whitespace occurs within the URL, the protocol will replace the space with "+" or "%20". Other individual substitutions occur when special characters are present.

3.1.2. Features Design

As mentioned earlier, features are defined by observing behavior on malicious pages. However, in certain situations, the presence or absence of a single feature may not be sufficient to arrive at a verdict on whether or not a particular page is fraudulent. Therefore, prediction models adopt a strategy of grouping one or more features that represent a specific class in the proposed classification model, reflecting the balance between sensitivity and specificity in the heuristic.

Despite this, the study aims to evaluate the features individually, considering behaviors such as: (i) features active in the Web browser involving lexical patterns in the URL or page source code; (ii) features based on reporting platforms that consider interaction and maintainability; (iii) the phishing life cycle, which considers spread and volatility of activity; and, (iv) the profile of phishing incidents, which considers trends and temporal aspects. For practicality, the details of each feature will be described in section 4, where data and its interpretation are presented.

3.1.3. The context-aware Taxonomy

The proposed taxonomy, as shown in Figure 5, is considered context-sensitive. It considers elements of the phishing ecosystem, such as third-party intervention, temporal behaviors, and computational resources involved, highlighting dynamic aspects of the phishing environment. In addition, the taxonomy also defined three different approaches, namely: content-based, third-party based and URL-based. The content-based approach considers all the content provided for the HTTP protocol, as the body and header. In the same line, the third-party based approach considers the actors of this context that provide or shared some information that allows the analysis of certain feature, as the community platform or other third-party services. Lastly, the URL-based approach consider the data that can be extracted from the URL anatomy as describes in the Figure 4.

Each approach operates to 4 distinct categories such as the phishing exploits over the web browser, the behaviors of the phishing approach based on community platforms, the phishing life cycle, and the target profile exploited by malicious users, representing 16 features. The rationale for the distribution of the features in category is supported by the nonparametric and partitional clustering proposal, which defines intrinsic models such that they have similar elements, based on some specific criterion. However, the respective subgroups are not hierarchically related.

In summary, it is a non-supervised classification process that agglomerates data based on specific similarities. To define the taxonomy, four questions were raised, namely: (i) How will the similarity between the features be established? (ii) How will the clusters be segmented? (iii) How many clusters will there be? (iv) How will the clusters defined be evaluated?

To answer these questions, an investigation was carried out in the literature as a basis for structuring, justifying, and testing the proposed taxonomy. This was done using the Lough strategy (Lough, 2001), which compiles arguments extracted from five similar studies (Howard, 1998; Amoroso, 1994; Bishop, 1999; Lindqvist & Jonsson, 1997; Krsul, 1998). These studies enumerate the requirements necessary for the definition of a taxonomy. Such requirements can be translated as directives to achieve a level of efficiency in the taxonomyâĂŹs classification model.

3.2. Populations and Samples definitions

For the observations to be carried out, the next step was to establish the population and respective samples to be defined. The population adopted by the study was the community-based platform known as PhishTank. Although other alternatives exist, such as OpenPhish[3] and SafeBrowsing[4], PhishTank was chosen because its database is open and has a larger volume. With the population determined, the next step was

[2] https://www.w3schools.com/tags/ref_urlencode.asp

[3] https://openphish.com/

[4] https://safebrowsing.google.com

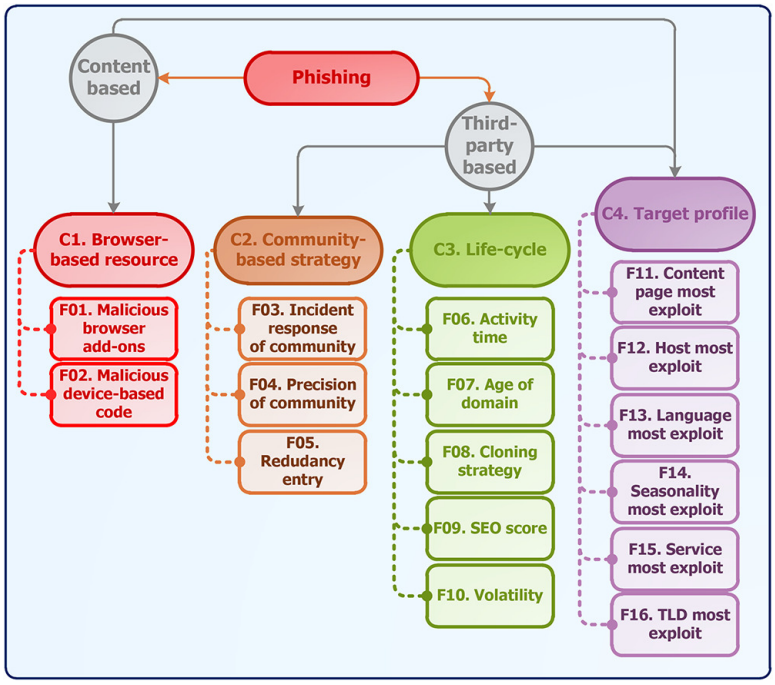

Figure 5: Phishing context-aware taxonomy

to define the samples, which were grouped into two types: pages confirmed to be malicious phishing sites, and pages verified as non-malicious and genuine.

3.2.1. The PhishTank Community

The PhishTank platform is characterized as a community-type platform where people can collaborate on phishing information available on the Web as a voluntary whistleblower prevention measure. This initiative produces an information base that can be shared around the world to help minimize phishing attacks. Although it does not claim to be a protective measure, the information provided by the platform provides support for anti-phishing mechanisms from various organizations, such as *Yahoo!*, *McAfee*, *APWG*, *Mozilla*, *Kaspersky*, *Opera* and *Avira*.

The term **community** refers to the platform having a large number of users that participate collaboratively. This collaborative character allows participants to denounce and opine on the verdict of a suspicious page, contributing to the feeding of the blacklist. This means that each phishing record or genuine page is subject to confirmation by participants through a voting system that determines whether the reported phishing is **valid** or **invalid** (genuine page). Additionally, the platform also identifies threat availability by checking whether the site is **online** or **offline**. The systemic flow of context that considers the platform and its users can be divided into five steps, as shown in Figure 6.

Step 1 starts a cycle of a potential threat once the likely malicious page has been posted to the web. When it is viewed by a platform user (step 2), it may report the suspicious URL on the platform (step 3), suggesting that participants initiate a voting system to reach a verdict on the possible threat (step 4). Finally, based on the votes cast, the platform reaches a final verdict (step 5). It is important to mention that the PhishTank community is not very clear about the number of votes sufficient to reach the decision, only states that much of the results are based on previous reporting patterns [5].

3.2.2. Samples and Data Extractions

To define the sample, it was necessary to obtain a significant amount of "valid" *phishing* sites, either *online* or *offline*. As an alternative, the platform has a web service that provides an hourly JSON file[6]. This file contains an average of 15,000 confirmed phishing sites, that is, sites for which the voting process has been completed to confirm them as phishing. For each site, the file also contains the confirmation date and time, which is the time when the verdict was rendered to categorize the site as fraudulent. This information, titled "PhishTank Info", was extracted following the model proposed in Table 7.

There is also an API that performs periodic queries against the data on the platform. However, this method has difficulties in restricting the number of requests. It is important to note that these queries contain only confirmed phishing sites. Considering that the voting period does not have fixed length of time, the study adopted a collection interval of one month of adjacency, to handle possible delays. In other words, the records for January were collected up until the end of February, with the extraction period continuing until 01/31/2019. The extraction process is illustrated on the left side of Figure 8.

With this scenario in mind, it was initially planned to

[5] https://www.phishtank.com/faq.php#howmanypeoplehavetov
[6] http://data.phishtank.com/data/online-valid.json.bz2

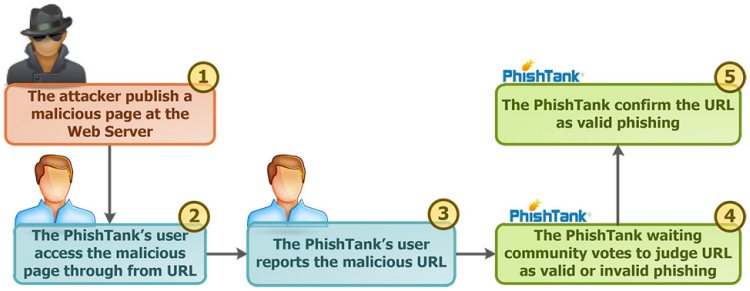

Figure 6: Community life-cycle in PhishTank platform

Samples	HTTP				PhishTank Info							Transitional	Extraction type	
					Submission time	Verification Time	Status		Confirmation		Target			
	URL	Code	Header	Body			Online	Offline	Valid	Invalid			Auto	Manual
Sample #1	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗
Sample #1.1	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓
Sample #2	✓	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗	✓	✓	✗
Sample #2.1	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓
Sample #3	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗

Figure 7: Details of *PhishTank* info

base the sampling process on the JSON file only. However, there are particularities in the process of making this file available that could potentially interfere with the results. First, because this file only considers confirmed and online phishing sites, and because of the volatile nature of phishing, the quantitative result for each month could be compromised, because many phishing sites last less than 1 hour, the JSON file generation interval. This can be confirmed through PhishTankâĂŹs " Phish search " option, which can list all phishing sites published on the platform, whether confirmed or not, including both online or offline. This option, in addition to solving the monthly quantity issue, also makes it possible to measure the accuracy rate of complaints made to the platform, among other possibilities. The extraction process for the "Phish search" is illustrated on the right side of Figure 8. The flow of the sample definition process is shown in Figure 9.

Through this information source, it was possible to obtain a list of 2,156,759 phishing sites from 2009 through 2018. However, for questions of scope, the study was limited to 2018 for many of its statements. Based on the extractions illustrated in Figure 8, it was possible to define five distinct sample types. **Sample #1** contains 189,892 confirmed phishing sites, and provides the following information: URL, HTTP response code, HTTP header, and submission time. **Sample #2** is very similar, but contains 1,384 pages that were confirmed to be genuine (invalid phishing), making it useful for comparative tests against the sites in sample #1. These two samples were created based by extracting the " Phish search " through a WebCrawler. Such samples were reserved for qualitative analysis using an automated algorithm.

It was subsequently observed that more subjective manual analyses needed to be performed, making it necessary to obtain samples that supported qualitative analyses. According to Equations 1 and 2, random **Sample #1.1** (Singh & Mangat, 1996; Lumley, 2011) was defined. Similarly, **Sample #2.1** was defined in order to compare hypotheses be-

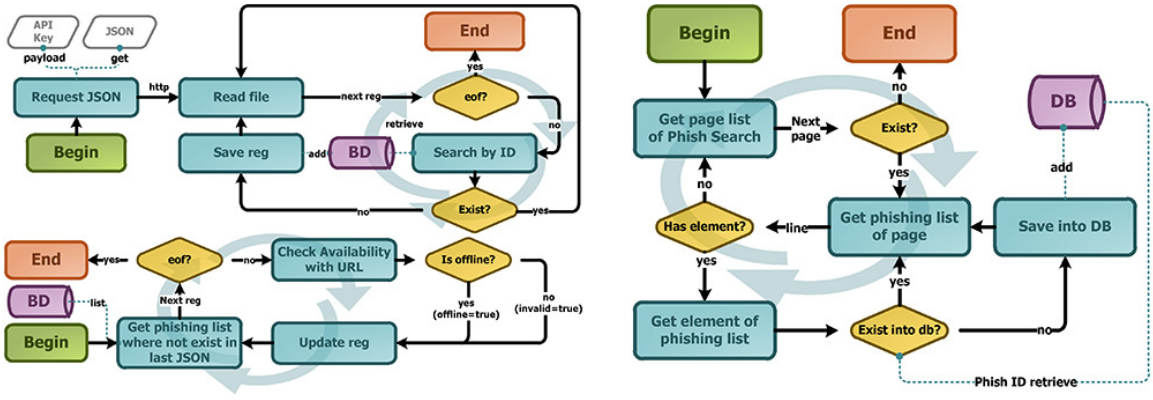

Figure 8: Workflow for "JSON" and "Phish Search" extractions

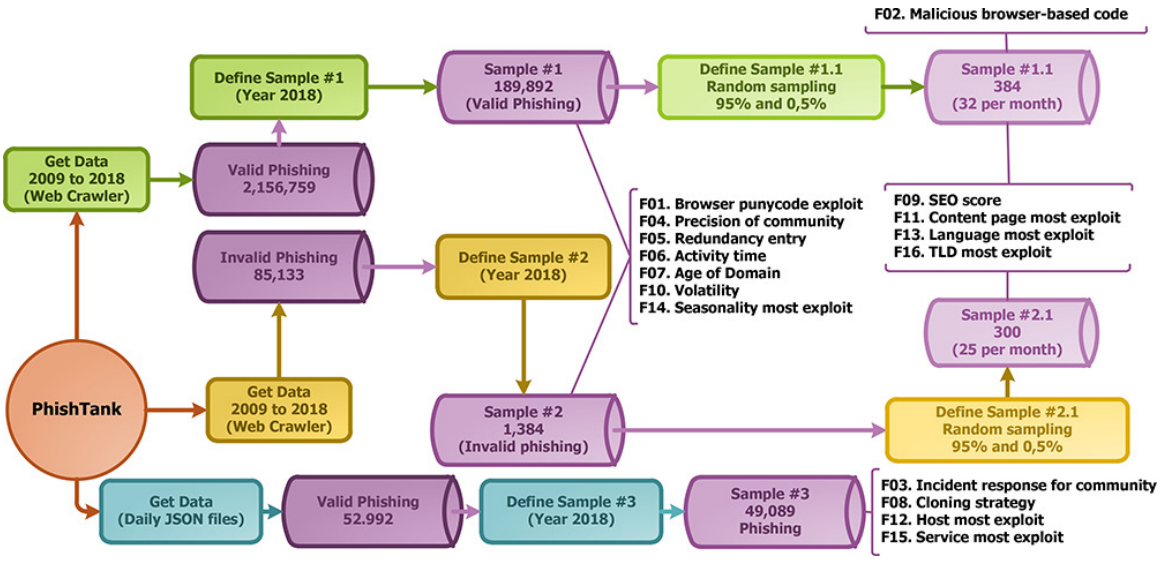

Figure 9: Population and Sample definition strategies

tween valid and invalid phishing sites. Assuming maximum population variability, with a confidence interval of 95% and an accuracy of 5%, the sample size should be 384 for valid phishing and 300 for invalid phishing. As these samples also need to be time-based, the number of records was proportionally divided across the months of the year, resulting in 32 records for each sample month #1.1 and 25 records for each sample month #2.1. These samples were created using a manual process based on samples #1 and #2.

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

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

Finally, other behaviors that needed to be observed, without regard to temporal aspects, such as volatility that could influence the results. For this, it was necessary to develop **Sample #3**, which stored 49,089 confirmed phishing sites, preserving the phishing content for future analysis, such as clone inspection.

4. Results and Analysis

This section describes the outcome of the empirical research as well as its presentation and interpretation. As illustrated in Figure 10, you can see the features, their extraction contexts, and their resulting samples. The order of presentation of the features follows the structure proposed by the taxonomy in Section 3.1.3.

The methodology adopted by the study to present the results was the "Goal Question Metric" (GQM), proposed by Basili et. al (Basili et al., 1994). This approach offers formalism and planning in data presentation by systematically structuring data and answering research questions. 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 defines a **question**, and identified behaviors represents a **metric**, totaling 36. Additionally, the study is also concerned with exposing an analysis of the relevance of features, classifying each as *WEAK*, *MODERATE* and *STRONG*. The analysis considers quantitative and qualitative criteria as definition of the depth level of each relevance.

4.1. Browser-based resource

This category groups features that are directed at flaws in the **browser resources**, that is, possible exploitations by attackers related to the features or functions offered by the end user, such as add-ons (plug-ins and extensions) and browser pre-functionalities that are triggered by protocols used on the Web.

4.1.1. F01. Browser Punycode Exploit

This feature evaluates browser resource exploits that can be done through the page URL using the Punycode attack. According to RFC 3492 (Costello, 2003), Punycode is a protocol that allows the conversion of unicode-specific characters, such as Chinese or Russian, into a compatible version for DNS domain names. This string is always prefixed with the *xn–* prefix and the Web browser is assigned responsibility for conversion.

In practice, domains with accented characters, for example "NetflÃŋx.com", result in "http://xn-netflx-7va.com". However, the options are not restricted to the use of Latin unicode. When using the Cyrillic alphabet (Slavic), the word apple in Cyrillic unicode (Jovanovic, 2009) converted to Punycode results in *xn–80ak6aa92e*, that is, an ASCII version valid for a domain. Another way to make the scam more sophisticated is the possible combination of distinct unicode that results in a string of homographs, that is, characters

C1. Browser-based resource	Extract approach	Extract element	Sample #1	Sample #1.1	Sample #2	Sample #2.1	Sample #3
F01. Browser punycode exploit	URL-based	URL part 1 (unicode)	✓	✗	✓	✗	✗
F02. Malicious browser-based code	Content-based	HTTP Body & Header	✗	✓	✗	✗	✗
C2. Community-based strategy	**Extract approach**	**Extract element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
F03. Redundancy entry	Third-party based	URL part 1 & 2	✓	✗	✓	✗	✓
F04. Incident response for community	Third-party based	Submission & Verification Time	✗	✗	✗	✗	✓
F05. Precision for community	Third-party based	HTTP Status Code	✓	✗	✓	✗	✗
C3. Life-cycle	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
F06. Activity time	Third-party based	HTTP Status Code	✓	✗	✓	✗	✗
F07. Age of Domain	Third-party based	WHOIS	✓	✗	✓	✗	✗
F08. Cloning strategy	Third-party based	HTTP Status Code	✗	✗	✗	✗	✓
F09. SEO score	Third-party based	Alexa Service	✗	✓	✗	✓	✗
F10. Volatility	Third-party based	HTTP Status Code	✓	✗	✓	✗	✗
C4. Target Profile	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
F11. Content page most exploit	Content-based	HTTP Body	✗	✓	✗	✓	✗
F12. Host most exploit	URL-based	URL part 1 (Host)	✗	✗	✗	✗	✓
F13. Language most exploit	Content-based	HTTP Body & Header	✗	✓	✗	✓	✗
F14. Seasonality most exploit	Third-party based	Submission Time	✓	✗	✓	✗	✗
F15. Service most exploit	Content-based	HTTP Body	✗	✗	✗	✗	✓
F16. TLD most exploit	URL-based	URL part 1 (domain)	✗	✓	✗	✓	✗

Figure 10: Relation between categories and samples

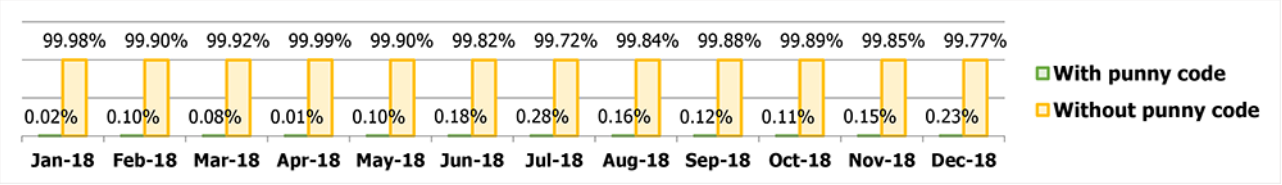

Figure 11: Occurrences of F01. Browser punycode exploit

identical to the name of the genuine site. However, some alphabets, like Cyrillic, do not have characters for lowercase g or l, which makes it difficult for the offender to forge a set of characters that resemble a domain like google.com. The closest approximation would be something like GooGLe.com, different to raise suspicion.

However, nothing prevents the offender from using different combinations of unicode, such as using lowercase g and l from the basic Latin alphabet and the other characters from the cyrillic alphabet, resulting in the string xn-ggl-tdd6ba.com, which, when converted by Punycode, would result in google.com22. This browser "feature" was originally proposed to provide a more user-friendly representation to the end user, but turned out to be an opportunity for the malicious. The attacker often takes steps to leave a fraudulent site even more believable, such as registering the *xn-80ak6aa92e.com* domain and providing tunneling. The data results are shown in Figure 11 and the GQM is described in Table 1.

Current browsers have minimized such problems, since the browser stopped performing the "friendly" exchange in the address bar during the act of Punycode conversion, maintaining the display in the original characters. Another criterion was considered to be defensive in cases where the characters make use of distinct unicode. However, it is still possible to use these techniques in HTML elements such as web pages or in the body of emails, such as the *href* attribute on $< a >$ elements and the use of *HTML unicode*, thus requiring more accuracy on the part of end users in observing such exploitation. Finally, pages were also noticed that requested the installation of plug-ins or extensions to the browser, proving the use of malware. Given this, the features was considered to be of MODERATE.

4.1.2. F02. Malicious Browser-based Code

This feature evaluates exploits performed through the source code of the page, in order to make the site appear more reliable to the end user, persuading them to believe that the fraud in question is a genuine site. Although the source code is part of the page content, the feature is considered static because it describes a lexical analysis of the source code of the page, i.e. the analyzed content is not the same as that presented to the end user, so subjective aspects are not considered.

Patterns of this nature are recurrent through the manipulation of HTTP headers such as *User-Agent* and *Refere*, typically used for *SMiShing attacks*, that is, targeted to mobile devices or to a particular context, such as *Spear Phishing attacks*. Nevertheless, cases of persuading the user to install malicious plug-ins or browser extensions were also investigated. The data result are shown in Figure 12 and the GQM analysis is described in Table 2. Since the behavior of this feature is completely offensive, there was no point in making a comparison between the valid and invalid samples.

As shown in Figure 12, it was possible to identify that most of the pages analyzed used a combination of "User-Agent" and "Force 404 by IP", resulting in a certain page being opened in the smartphone browser. When trying to open this page in a PC browser, the user was presented with a forged 404error, making it clear that it was a *SMiShing*. Given this, the feature was considered to have STRONG.

Table 1
GQM of F01. Browser punycode exploit

Goal 1	Analyze the reliability of the exploits incident response from the browser-based resources viewpoint.				
Question	Q01. Which records make use of the punycode feature?				
Metrics	[M01]: Count of valid phishing records that use punycode.				
	[M02]: Count of invalid phishing records that use punycode.				
Hypotesis	To simulate greater trustworthiness, a malicious URL will make use of punycode, resulting in greater apparent veracity.				
Sample	1 and 2	**Relevance**	MODERATE	**Relations**	F02
Extraction	Get part 1 of the URL and count and analyze the cases where punycode is used.				
Limitations	-				
Observations	The punycode analysis was performed directly on the URL.				
Analysis	Despite the modest numbers, punycode attacks were found to be present during 2018, becoming an attack utilized by malicious users.				

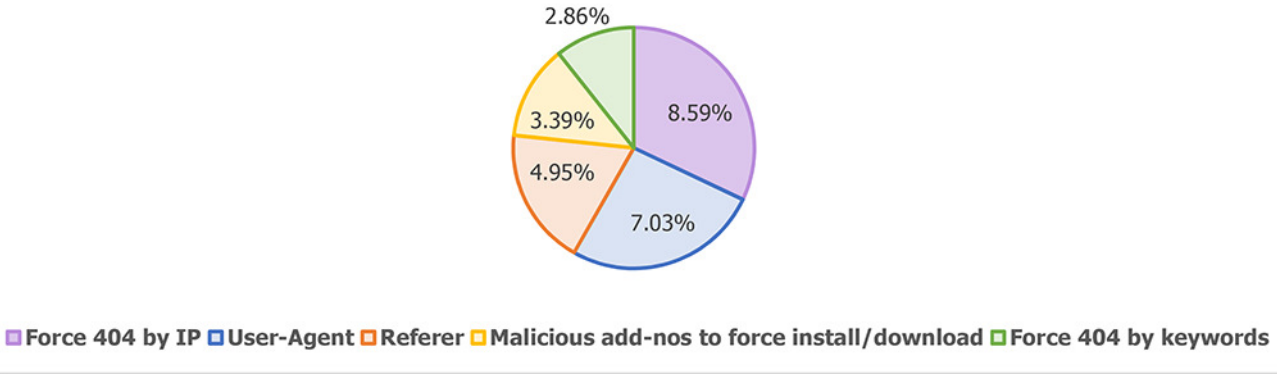

Figure 12: Occurrences of F02. Malicious browser-based code

4.2. Community-based strategy

This category groups features that are directed at flaws in the **strategies adopted by denunciation platforms**, that is, possible exploitations by attackers related to the anti-phishing controls that the platform offers.

4.2.1. F03. Incident response of community

According to the denunciations and voting, presented in Section 3.2.1, the difference between the denunciation date and confirmation date makes it possible to analyze the response time of the community. The community interval flow is illustrated in Figure 13, the data extracted follow in Figure 14, and the GQM analysis is shown in Table 3.

Table 2
GQM of F02. Malicious browser-based code

Goal 1	Analyze the reliability of the exploits incident response from the browser-based resources viewpoint.				
Question	Q02. Which records exploit browser features based on source code?				
Metrics	[M03]: Count of records that use javascript code to manipulate the result of the page load on the client.				
	[M04]: Count of records that use the HTTP header to manipulate the result of the page that loads for the client.				
	[M05]: Count of records that suggest that the end user to download a file.				
Hypotesis	In order to simulate greater trustworthiness, a malicious URL may require the end user to install add-ons to the browser for improved user experience.				
Sample	1.1	**Relevance**	STRONG	**Relations**	F01
Extraction	Obtain the content and page header and analyze the browser features that are scanned.				
Limitations	-				
Observations	The analysis of the metrics was observed by examining the page contents, such as elements and javascript codes.				
Analysis	It was observed that certain records used the HTTP header features to intervene in the page loading result. Headers such as User-Agent and Referer were heavily used, some of which forged a 404 error in cases when opening the page on a device that was not mobile, such as a smartphone, representing a type of SMiShing attack. In addition, JavaScript code was also identified as being used to force 404 errors for a particular region (via IP address), thus representing a type of Spear phishing attack.				

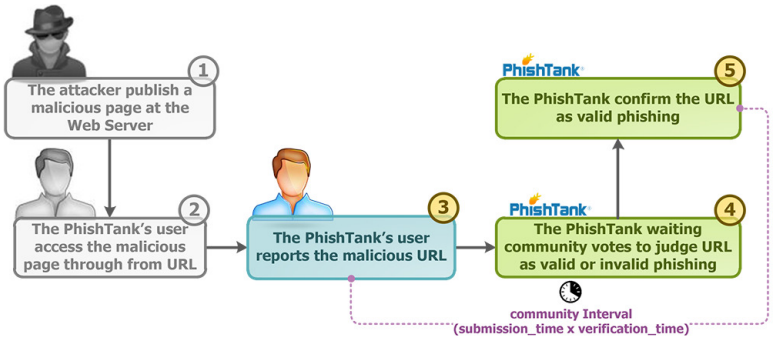

Figure 13: Workflow for community interval

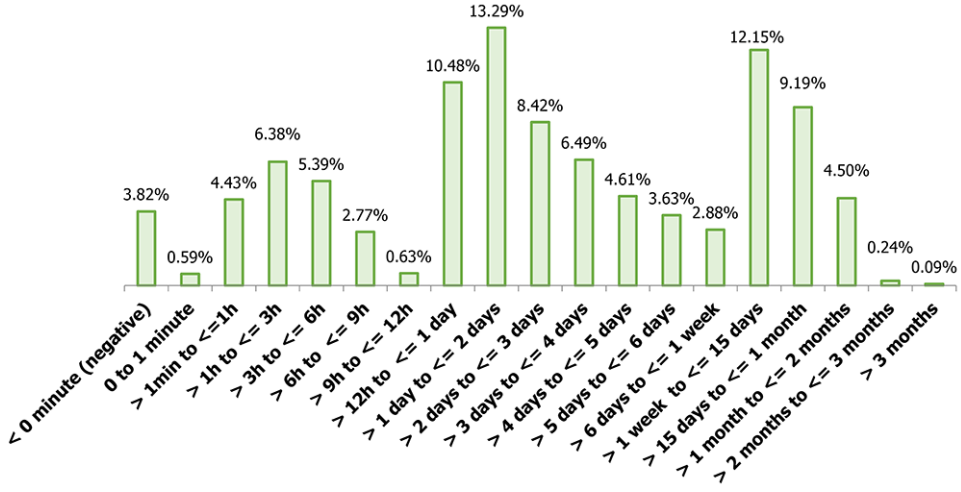

Figure 14: Occurrences of F03. Incident response for community

As shown in Figure 14, the X-axis represents the time intervals as a scale, while the Y-axis represents the number of occurrences by percent of the entire sample. The PhishTank platform, by not establishing a deadline for its voting system, causes the community's response policy to be weak. Of all reported phishing, only 20% are confirmed within the first 6 hours from the time of the complaint, showing that this feature has STRONG relevance.

It was possible to observe that 50% of valid phishing sites take between 9 and 168 hours (one week) to be confirmed, characterizing a considerable window of vulnerability. Almost 4% of the records have a listed confirmation date earlier than the submission date, resulting in outlier values in the chart. Finally, the remaining 26% took more than a week or up to months to be confirmed.

4.2.2. F04. Precision of community

This feature evaluates the precision of the community with regard to false positives for URLs submitted. This as-

Table 3
GQM of F03. Incident response for community

Goal 2	Analyze the reliability of the phishing incident response from the community-based approach viewpoint.				
Question	Q03. What is the transition interval between submission of phishing and confirmation?				
Metrics	[M06] Community response time to complete voting on possible phishing.				
Hypotesis	The community has a considerable window of vulnerability regarding 0-day phishing.				
Sample	3	**Relevance**	STRONG	**Relations**	F06 and F10
Extraction	Subtraction between the submit time and the confirmation time.				
Limitations	-				
Observations	Some results returned negative values. These were discarded, since it does not make sense to be confirmed before being submitted.				
Analysis	It was possible to observe considerable delay on the part of the platform in response to an existing phishing site. Some data are very worrying, as 10.67% take from 12 to 24 hours to receive a voting result. Even worse, 13.48% took 2 days, and 11.78% take between 7 and 15 days to be confirmed.				

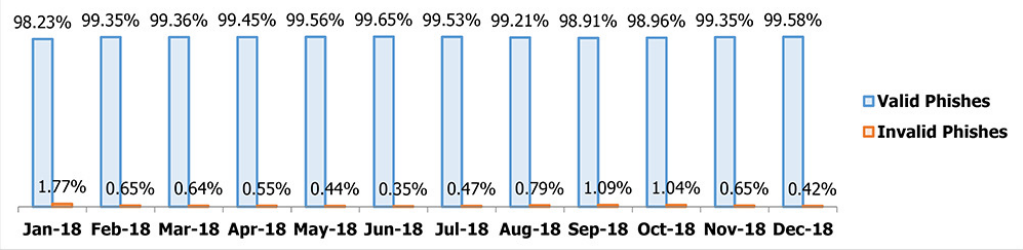

Figure 15: Occurrences of F04. Precision of community

Table 4
GQM of F04. Precision of community

Goal 2	Analyze the reliability of the phishing incident response from the community-based approach viewpoint.				
Question	Q04. What is the index of false positives reported by the community?				
Metrics	[M07] Number of valid phishing reports.				
	[M08] Number of invalid phishing reports.				
Hypotesis	The community submits a considerable number of false positives.				
Sample	1 and 2	**Relevance**	MODERATE	**Relations**	F10
Extraction	Subtract the amount of valid and invalid.				
Limitations	-				
Observations	Some entries appeared as both valid and invalid. These were excluded from the valid phishing during the sample definition.				
Analysis	Of the total number of valid and invalid denunciations, the rate of invalid complaints is relatively low, averaging 0.74%. In addition, the platform base demonstrates constant values over time as to the rate of valid phishing submitted.				

pect is important to the platform as it represents the precision of the valid and invalid phishing data being reported. The data result are shown in Figure 15 and the GQM analysis is described in Table 4.

In Figure 15, the X-axis shows the months during the year 2018, while the Y-axis represents the number of occurrences as a percentage of the entire sample. The platform presented good precision with regard to the complaints received, with a standard deviation of 0.54 on the confirmed phishing results. However, despite being discreet, the platform still does not have a solid process to handle undue reporting, requiring a volunteer to identify, so the weight for this behavior can be considered MODERATE.

4.2.3. F05. Redundancy entry

This feature evaluates URLs that have duplicate denounced entry into a community platform. Some blacklist mechanisms store the hash by considering all characters of the URL, that is, the two parts as shown in Figure 4. This problem has two ways: when the user reports a URL with some character that the browser will discard during the load process, as explicit default port at URL (":80" or ":443") and double slash path ("//"). On this, will cause the resulting hash for URL denounced to be different from a hash in which these characters are not specified, making it platform understand that this is a different URL. Another way is when the user just denounced the same URL at two or more times and the platform not and the platform does not make an efficient treatment. The GQM is described in Table 5.

This feature also evaluates cases where the attacker spreads the malicious URL with an empty path, that is, with the characters "//" in part 2 of the URL. Therefore, in some cases, such as applications that are not RESTful, the browser will redirect the user by neglecting the "empty" path, circumventing the blacklist, as the URL without the double slash would have a different hash. Cases of this nature need to be handled by the blacklist maintainer in order to avoid false negatives. The extracted data are shown in Figure 17.

Lastly, this feature also refers to the duplication of the same URL in the community-maintained blacklist. The intention of this feature is to show that unnecessary voting for certain URLs occurs after they have already been submitted to a vote. The extracted data are shown in Figure 18.

According to Figure 16, the X-axis shows the port used, while the Y-axis represents the number of occurrences (in units and as a percentage). Cases of this feature are therefore quite rare, with only 0.005% of valid phishing sites making use of this exploit. However, its existence does cause the problems described above. SafeBrowsing blocks phishing sites by considering only the domain. Because of this, it was possible to observe that in cases of domain hijacking, all pages of the respective domain were blocked by browsers making use of SafeBrowsing, generating false positives.

As shown in Figure 17, the occurrence of this feature is similar to valid and invalid phishing, which draws attention to the fact that the platform does not predict this situation. This allows for the occurrence of false negatives. As shown in Figure 18, the bar graph indicates that the community has few duplicated phishing records, 0.32%, or 607 duplicates. One behavior that might explain this occurrence would be confirmed phishing cases that later go offline, be-

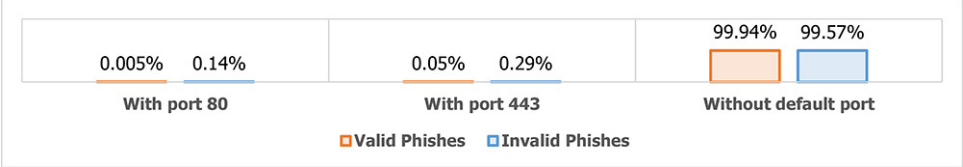

Figure 16: Occurrences of F05. Redundancy entry (default HTTP port exposure)

Figure 17: Occurrences of F05. Redundancy entry (double slash path)

ing denounced again and submitted again to the voting system. In the same figure, the scatter plot shows a count of the individual repetitions of each duplicate phishing. The X-axis shows the quantity as a percent, while the Y-axis displays the number of URLs that have been duplicated, totaling 242. The sum of these individual duplicates results in a total of 607.

Due to the fact that the number of duplicates was not expressive. However, an unnecessary effort is evident on the platform, and as there is no established deadline for voting, this can create unnecessary delays in confirmation. So the feature was considered MODERATE.

4.3. Life-cycle

This category highlights features directed at the **phishing life cycle**, considering aspects such as uptime, the brevity of existence, and propagation through cloning. The data obtained aim to show the **average activity time** of certain phishing behaviors.

4.3.1. F06. Activity time

This feature evaluates the uptime of a phishing site, in order to observe temporal patterns. This is an important indicator for identifying specific cases of phishing with a longer than average shelf life. The extracted data are shown in Figure 19 and the GQM analysis is described in Table 6.

According to the upper bar graph in Figure 19, the X-axis shows the time intervals, and the Y-axis represents a percentage of the entire sample. Based on the data extracted, it was possible to observe that 78.59% of valid phishing had a shelf life of fewer than 2 months. In contrast, 70.81% of invalid phishing has between 7 and 13 months of activity, meaning that phishing has a much shorter activity cycle than legitimate. Given this, the feature was considered to be of STRONG relevance.

4.3.2. F07. Age of Domain

This feature evaluates the age of a given domain. Consequently, it also makes it possible to identify specific phishing cases registered for the purpose of committing fraud or suspicious domains that may have been sequestered. The extracted data follow in Figure 20 and the GQM is in Table 7.

The left bar graph in Figure 20 shows that 33.07% of valid phishing had its own domain. In contrast, 78.07% of invalid phishing had their own domain, providing evidence that invalid phishing commonly has a registered domain address. To obtain the data, the WHOIS[7] protocol was used to observe the age of the domain registration, and the acquisition of this data was automated through the Domage API[8].

[7] WHOIS: https://tools.ietf.org/html/rfc3912
[8] Domage API: https://ipty.de/domage/index.php

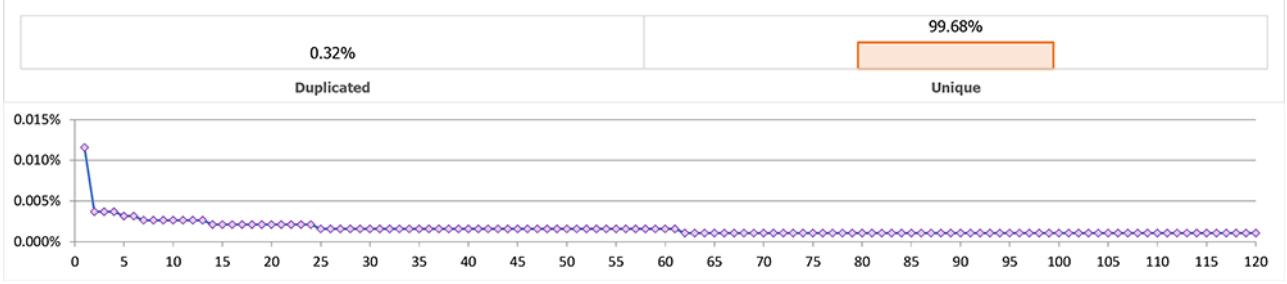

Figure 18: Occurrences of F05. Redundancy entry (duplicate URL entry)

Table 5
GQM of F05. Redundancy entry

Goal 2	Analyze the reliability of the phishing incident response from the community-based approach viewpoint.				
Question	Q05. How does the community handle URL duplicated entries?				
Metrics	[M09] Count of records that explicitly use port 80 in valid phishing.				
	[M10] Count of records that explicitly use port 443 in valid phishing.				
	[M11] Count of records that explicitly use port 80 in invalid phishing.				
	[M12] Count of records that explicitly use port 443 in invalid phishing.				
Metrics	[M13] Number of URLs with double slash path in valid phishing.				
Metrics	[M14] Number of URLs with double slash path in invalid phishing.				
Metrics	[M15] Number of duplicate URLs in the repository.				
Hypotesis	There is an interest on the part of malicious users to propagate malicious URLs that display the default port, in order to make changes to the hash generated or does not bother to check for double slash path exploits at the confirmed URLs. The community does not anticipate this type of situations and expends effort dealing with URLs previously reported.				
Sample	1 and 2	**Relevance**	MODERATE	**Relations**	F08
Extraction	1) Get the URL and parse it for the presence of port 80 or 443. 2) Verify double slash path in the URL String. 3) Verify duplicate URLs through URL String collision.				
Limitations	-				
Observations	There may be cases where the URL differs only from the default port value, which would already change its hash, thus performing a bypass.				
Analysis	Default port exposure is almost non-existent for malicious URLs. More significant occurrences were found with legitimate URLs that exposed port 443. Based on the sample, this is found in a modest number of valid and invalid phishing records, 1.48% and 1.19%, respectively. These data demonstrate that the feature can occur regardless of whether or not it is a threat. The platform does not seem to predict this behavior, however. Based on the sample, there was a modest number of repeated URLs (0.32%). One URL was repeated 22 times in the platform entries.				

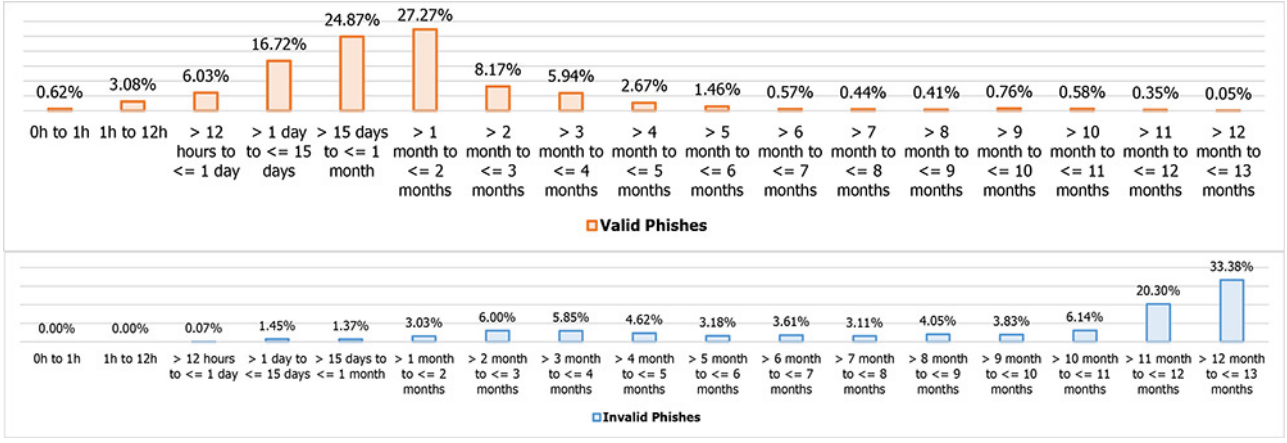

Figure 19: Occurrences of F06. Activity time

Table 6
GQM of F06. Activity time

Goal 3	Analyze the phishing life cycle from the point of view of its activity.				
Question	Q06. What is the average lifetime of a phishing site?				
Metrics	[M16] Time between 01/01/2019 and submission date of valid phishing.				
	[M17] Time between 01/01/2019 and submission date of invalid phishing.				
Hypotesis	Phishing has a short uptime.				
Sample	1 and 2	**Relevance**	STRONG	**Relations**	F07, F10 and F14
Extraction	Subtract the amount of valid and invalid entries.				
Limitations	-				
Observations	The submission date does not provide real-time precision regarding activity, as it only records the time when the phishing in question was reported.				
Analysis	Almost 80% of phishing has a period of less than two months of activity. It seems strange that many phishing sites with a considerable age of activity (1.75% have uptimes between 9 and 13 months) are not banned by their respective hosting servers.				

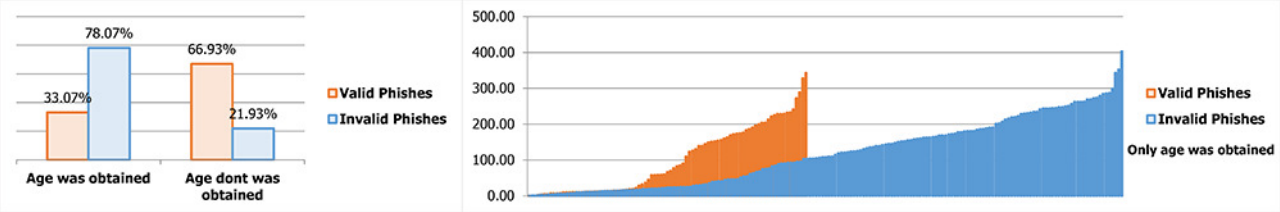

Figure 20: Occurrences of F07. Age of Domain

Table 7
GQM of F07. Age of Domain

Goal 3	Analyze the phishing life cycle from the point of view of its activity.				
Question	Q07. What is the average age of a registered phishing domain?				
Metrics	[M18] Calculate the age of a record used in a valid phishing.				
	[M19] Calculate the age of a record used in an invalid phishing.				
Hypotesis	Phishing domains will have a young age.				
Sample	1 and 2	**Relevance**	STRONG	**Relations**	F09 and F16
Extraction	Subtract the amounts of valid and invalid phishing.				
Limitations	Only 33.07% of valid phishing and 78.07% of invalid phishing were considered.				
Observations	It is important to note that many phishing sites operate on domains provided by third party services. In addition, there are cases where the domain is abducted by the attacker. Such behaviors could skew the results. Considering this, it was necessary to carry out the observation in small samples for a subjective analysis.				
Analysis	feature F07 can be considered an indicator that raises suspicions about a URL. Around two-thirds, or 66.93%, of valid phishing did not have a registered domain, suggesting that the URL in question is more likely to be dangerous when it has a registered domain. However, the analysis also indicated that 1/3 of malicious users invested in their fraud.				

On the right bar graph in the same figure, the Y-axis measures the number of months of activity for each domain, considering only records having a registered domain. It can be noted that the age of the registry for invalid phishing is considerably older than for valid. Given this, the feature was considered to have STRONG relevance.

4.3.3. F08. Cloning strategy

This feature evaluates the occurrence of Web-propagated clones, as shown in Figure 21. It is important that a phishing parameter measure this aspect, because the practice, beyond propagating phishing sites, also aims to circumvent blacklist mechanisms. The GQM analysis is described in Table 8.

As shown in the bar graph in Figure 21, 32.43% of all phishing sites in sample #3 were cloned. In the same figure, the second bar chart shows the 10 most-cloned phishing sites in the sample, differentiated by color. The most-cloned site had a total of 525 occurrences, representing 1.07% of all of the records in the sample.

Regarding the 10 most-cloned sites, it is possible to see in the dispersion chart the continuous action on the part of the malicious in keeping their fraudulent sites up throughout the months of 2018. The X-axis shows the dispersion of the occurrences during the months of 2018, while the Y-axis the number of occurrences in the respective period. It was possible to see that the 525 clones were well distributed throughout the year, and their highest peaks occurred in the month of March. Given this, the feature was considered to have STRONG relevance.

4.3.4. F09. SEO score

This feature evaluates the SEO score of a particular page. This refers to the reputation of a site during its time of activity. For example, situations where a site is built by Content Management System (CMS) share common vulnerabilities that can be exploited by attackers. In this context, whether or not to use a particular CMS may already be a criterion for reducing the SEO of a site. In short, a site that has already been breached or realistically could be, already loses SEO score. The fact is that a phishing site often has a well-defined short-term goal and well-targeted motivations, i.e. it is not an environment in which its owner would invest in digital marketing, thereby representing something that could cast suspicion on a particular site. The data are shown in Figure 22 and the GQM in Table 9.

It is important to note that the Google page rank feature, by having been discontinued, eventually became obsolete and was not addressed by this study Further details on this are found in Section 5. Therefore, Alexa was used to analyzing the SEO score of the pages. As shown in the bar graph in Figure 22, the Y-axis gives the mean scores for valid and invalid phishing sites. Given this, it was possible to observe that the ranking for valid sites was much greater the ranking of invalid sites, with a global average around 6 times greater. In Alexa, a page with a certain reputation and investment in SEO tends to have a low number. For example, the website www.google.com has a score of 1. Unknown websites without SEO strategies tend to have high numbers, for example, the valid phishing site *https://id-apple-account.usa.cc/* has a score of 2,366,789.

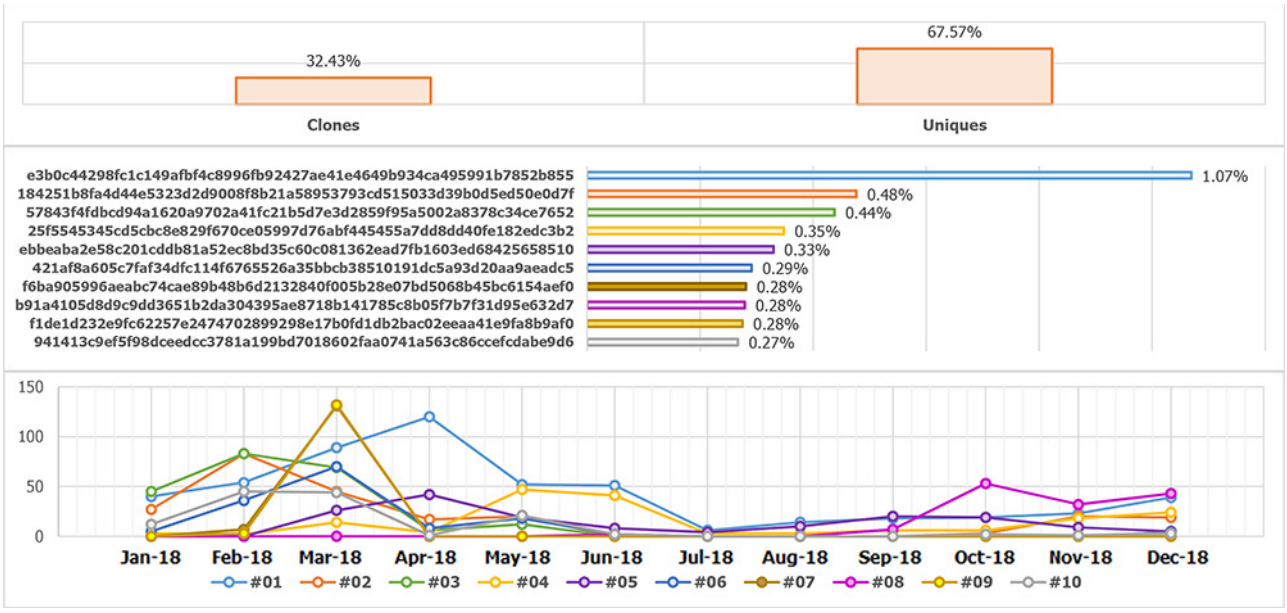

Figure 21: Occurrences of F08. Cloning strategy

Table 8
GQM of F08. Cloning strategy

Goal 3	Analyze the phishing life cycle from the point of view of its activity.
Question	Q08. What is the quantity of cloned phishing?
Metrics	[M20] Quantity of cloned phishing.
	[M21] The 10 most-cloned phishing sites.
Hypotesis	Due to the short time and high transition of the activity, the attacker will adopt cloning for greater propagation.
Sample	3 **Relevance** STRONG **Relations** F05
Extraction	Obtain the http response and transform it into a hash. After that, perform a hash collision to compute.
Limitations	-
Observations	-
Analysis	Based on the sample, 32.42% of phishing sites have one or more clones. For greater precision, the analysis process was based on the page content hash with considerable collision resistance (SHA-256). One specific phishing site was cloned in such a way that its number of clones represented 1.07% of the entire sample analyzed. Together with this feature, the results from F06 and F10 reinforce the volatile and brief nature of phishing. Once the phishing site is captured in a list, the fraudster, in addition to creating others, also ends up reusing the same site on different servers, causing the URL to change and making it 0-day site for the blacklist mechanisms.

The overall average highlights the sharp contrast between valid and invalid pages. However, an Alexa rank could not be extracted for every website. Many returned the error *We do not have sufficient data to rank this website*. This showed that Alexa does not have sufficiently wide coverage to handle the proposed scenario, making the feature of MODERATE relevance. Another factor is that the size of the URL can influence the score. URLs smaller than 50-60 characters are influenced positively, while those with over 100 are influenced negatively.

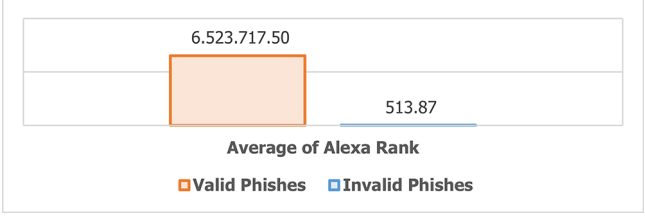

Figure 22: Occurrences of F09. SEO score

Table 9
GQM of F09. SEO score

Goal 3	Analyze the phishing life cycle from the point of view of its activity.				
Question	F09. What are the SEO scores for valid and invalid phishing sites?				
Metrics	[M22]: Counting the rankings of valid phishing sites.				
Metrics	[M23]: Counting the rankings of invalid phishing sites.				
Hypotesis	Because of its nefarious purpose, a phishing site almost never has investments in marketing. Faced with this, suspicious sites are excluded from the search engine index so users are not put at risk.				
Sample	1.1 and 2.1	**Relevance**	MODERATE	**Relations**	F07
Extraction	Enter the full URL for the Alexa[9] page.				
Limitations	-				
Observations	-				
Analysis	It has been observed that the vast majority of malicious URLs have a much lower rank compared to legitimate pages, making Alexa an interesting parameter for raising suspicion about a particular site.				

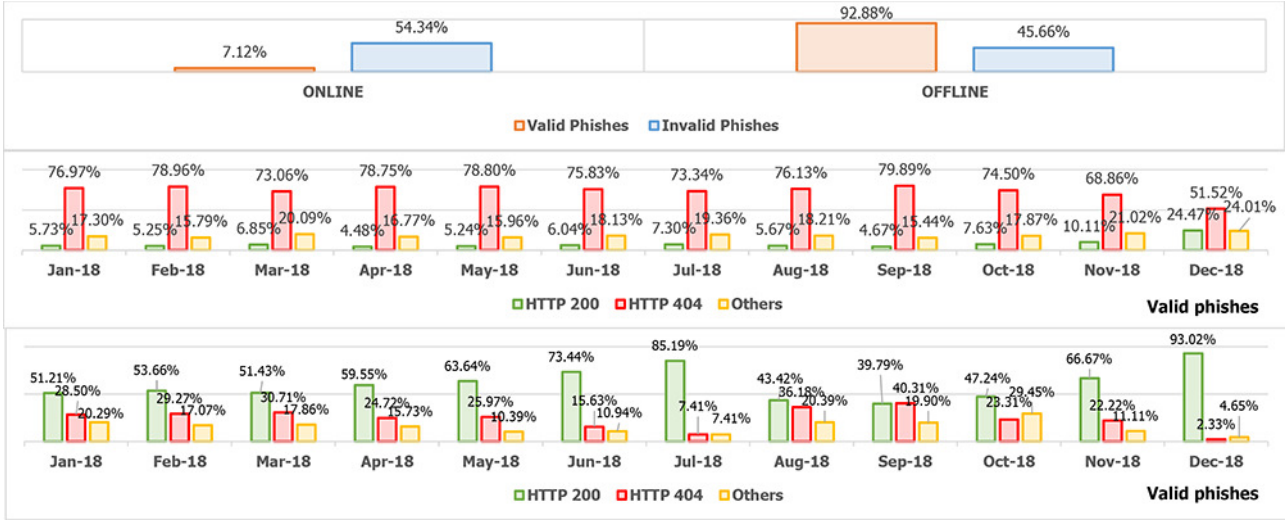

Figure 23: Occurrences of F10. Volatility

4.3.5. F10. Volatility

This feature evaluates the likelihood of phishing status changes, that is, the transition from online to offline. In order to read these data, annual and monthly charts were created as shown in Figure 23 and the GQM analysis is described in Table 10.

As shown in Figure 23, the first bar graph describes the total online and offline incidents for valid and invalid phishing sites over the entire year 2018. The X-axis groups record as either online and offline status, while the Y-axis shows the occurrences as a percentage of the total sample. It quickly became apparent that the volatility of valid phishing cases is much greater than that of invalid cases. Out of all the valid phishing sites of 2018, only 7.12% ended the year online, and most of those were from the last months of the year. Invalid phishing sites ended the year with 54.34% online.

In the same figure, the second and third bar graphs distribute the records for valid and invalid phishing sites, respectively, during the months of 2018, considering the HTTP codes 200, 404, or "Others". The X-axis groups the records by month over the respective year, while the Y-axis presents the values as a percentage of the total sample. Based on the data, it was possible to observe a pattern in the transition between online and offline valid phishing over the months, with a median of 75.98%.

It is important to note that data from the months of November and December are somewhat outliers, 68.86%, and 51.52% respectively, due to the delay in voting as described in Section 3.2.2. As a consequence, the first months of 2019 quantify the records for the last months of 2018, a situation which was observed and minimized by using the collection interval, as described in Section 3.2.2. Given this, the feature was considered to be of STRONG relevance.

4.4. Target profile

This category contains features that are targeted at **inc**. The possession of these data can help identify services, languages, contexts, and other aspects targeted by malicious users.

4.4.1. F11. Content page most exploit

This feature evaluates the type of content most targeted by malicious users. To read this data, **7 content categories were defined**, and the data was segmented by month and respective category, as shown in Figure 24 with the GQM in Table 11.

Table 10
GQM of F10. Volatility

Goal 3	Analyze the phishing life cycle from the point of view of its activity.				
Question	Q10. What is the variation in online and offline frequency during each month?				
Metrics	[M24] Difference between online and offline valid phishing.				
	[M25] Difference between online and offline invalid phishing.				
Hypotesis	By having a short active time, the attacks will have considerable volume in the transition from online to offline.				
Sample	1 and 2	**Relevance**	STRONG	**Relations**	F06
Extraction	Count the number of phishing sites with status 200, 404, and others.				
Limitations	The time aspect is considerably compromised. Older phishing sites tend to have more that are offline, while newer phishing sites are more likely to be online.				
Observations	-				
Analysis	The creator of the fraud uses the strategy of creating a phishing site and using it over a short period of time, aiming to reach the highest number of victims before being caught in the blacklist. The variation of phishing online and offline over the months is small. In this case, December did not follow the standard variance because of the limitations described.				

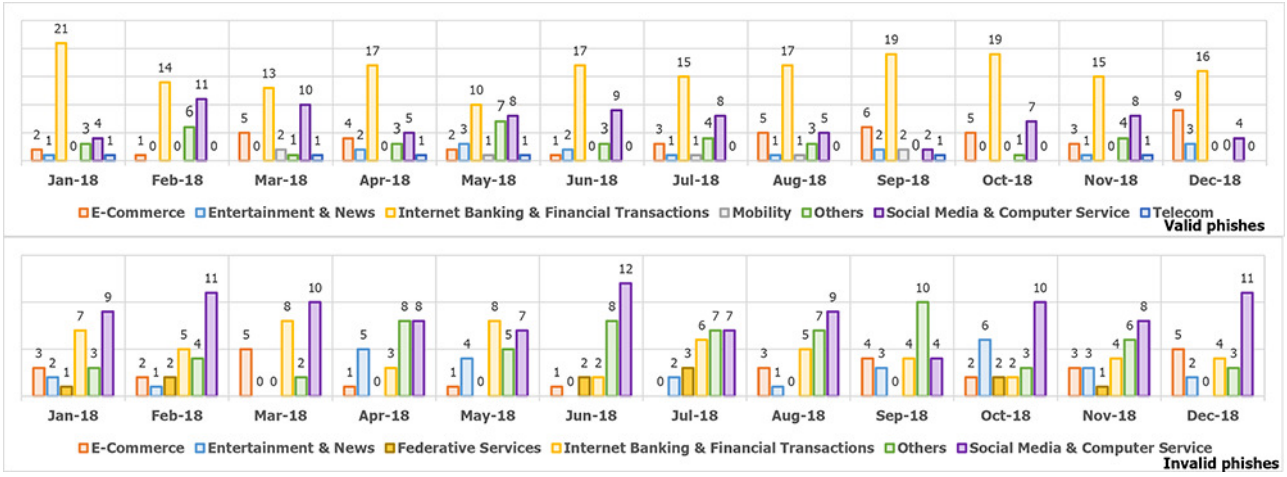

Figure 24: Occurrences of F11. Content page most exploit

In Figure 24, the bar graphs show the totally valid and invalid records segmented into the seven categories defined by this study, and which were based on the target field of PhishTank Info, as previously described in Section 3.2.2. The X-axis groups the data by month for the year 2018, while the Y-axis describes the occurrences for each category in the respective month. The categories are *E-commerce, Mobility, Telecom, Entertainment & News, Others, Internet Banking & Financial Transactions, and Social Media & Computer Services*.

Some segments had many occurrences combined, so these were grouped into a single category, such as *Social Media & Computer Services*. The "Others" represents the group of pages with very mixed content or even a segment with only

Table 11
GQM of F11. Content page most exploit

Goal 4	Analyze trends in the occurrences of phishing.				
Question	Q11. What is the type of content most exploited by phishing attack?				
Metrics	[M26] Count types of valid phishing by category.				
	[M27] Count types of invalid phishing by category.				
Hypotesis	There is a trend regarding the type of content approached, with certain types being more susceptible to attacks.				
Sample	1.1 and 2.1	**Relevance**	MODERATE	**Relations**	F14 and F15
Extraction	Get the http response and parse the content type in question based on categorization.				
Limitations	-				
Observations	The content type was grouped into seven categories, based on the APWG reports[10].				
Analysis	With regard to valid phishing, it is known that most attacks are directed to the segment of internet banking and financial transactions. False reports refer mostly to social networks.				

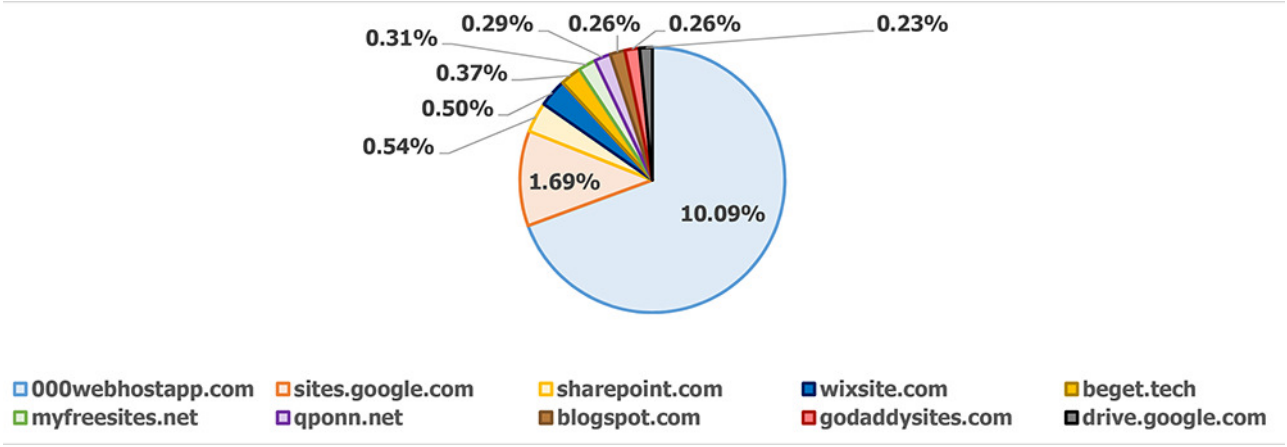

Figure 25: Occurrences of F12. Host most exploit

Table 12
GQM of F12. Host most exploit

Goal 4	Analyze trends in the occurrences of phishing.
Question	Q12. Which hosting service is most exploited by attackers?
Metrics	[M28] Count of the most exploited storage services.
Hypotesis	There is a trend regarding the hosting service, whereby its intrinsic usage policies make it even more susceptible to attacks.
Sample	3 **Relevance** STRONG **Relations** F15
Extraction	Obtain the URL and count the number per domain, verifying if it is a hosting service.
Limitations	Certain URLs had a registered domain name. For such cases, the hosting service in question could only be known through WHOIS. However, not all URLs provide support for obtaining details through WHOIS. For example, the API only gets WHOIS information for .com, .net, and .edu domains.
Observations	An API was used to capture the WHOIS results.
Analysis	10% of all phishing sites in the sample were hosted on the 000webhostapp service. It was observed that the respective service had few registration criteria and a low cost (even a free plan with 1GB of storage). It was noted that in the mentioned service the annual plan ($ 4.49 per month) additionally includes 1 domain registration and SSL certificate. These aspects make the service very attractive to malicious users.

one or two isolated occurrences, where it made more sense to be represented together in a single category.

Based on the data presented, it was clear that *Internet Banking & Financial Transactions* is the segment most exploited by fraudsters, with an average of 16.08% of occurrences each month, which represent 4.19% of all sample records each month. The standard deviation of attacks in this category was 2.97%, in other words, the standard is well-established throughout the year. Given this, the feature was considered to be of MODERATE relevance.

4.4.2. F12. Host most exploit

This feature identifies the most exploited hosting service. Several aspects can make a certain service more or less desirable to malicious users, whether it be location, hosting plans, or the availability of storage. The extracted data are shown in Figure 25 and the GQM analysis is described in Table 12.

As shown in Figure 25, the service offered by *000webhospapp.com* accounted for 10.09% of all records in Sample #3, i.e. it is clear that this service is the most exploited for conducting web fraud. Some possible reasons are the benefits and practicalities offered, such as a generous 1GB of free storage, a low-cost subscription plan without add-ons and with dedicated bandwidth, as well as few criteria required for registration. The second most-targeted service was that of Google, which has the advantage of practicality with its previously-defined template, facilitating the process of creating new pages. Given this, the feature was considered to be of STRONG relevance.

4.4.3. F13. Language most exploit

This feature evaluates the language most exploited by the attackers. Several aspects must be considered in defining the language, for example, cases of the same language being used in several different countries or even pages using more than one language. Because of this, it was necessary to extract the data manually, considering the language of the predominant content and observing some HTML elements that specify the language. This level of refinement also made it possible to consider that this feature, beyond just the language, represents the country to which the attack is directed. The extracted data are shown in Figure 26 and the GQM analysis is described in Table 13.

In Figure 26, the bar graph shows the observed incidents. The X-axis describes the language and the Y-axis shows the total number of occurrences, as a percentage of the total sam-

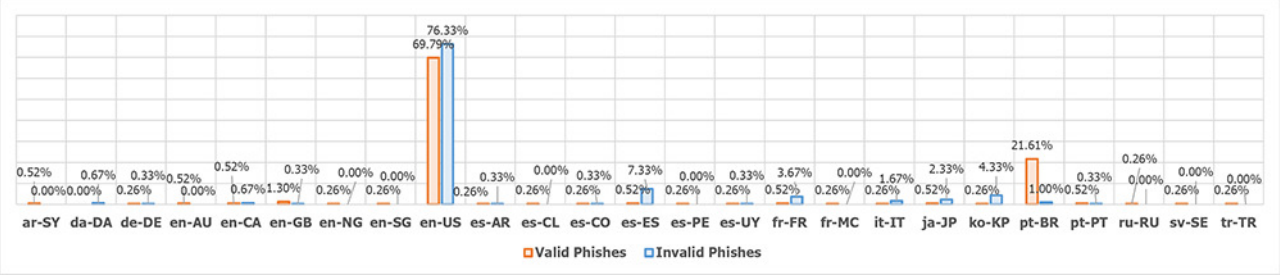

Figure 26: Occurrences of F13. Language most exploit

Table 13
GQM of F13. Language most exploit

Goal 4	Analyze trends in the occurrences of phishing.
Question	Q13. What language is the most exploited by attackers?
Metrics	[M29] Count the language most used in valid phishing, considering the country of origin.
	[M30] Count the language most used in invalid phishing, considering the country of origin.
Hypotesis	There is a trend towards a specific language, that one being more susceptible to attacks.
Sample	1.1 and 2.1 **Relevance** WEAK **Relations** -
Extraction	Obtain the http response and parse the predominant language.
Limitations	Certain URLs had a registered domain name. For such cases, the hosting service in question could only be known through WHOIS. However, not all URLs provide support for obtaining details through WHOIS. For example, the API only gets WHOIS information for .com, .net, and .edu domains.
Observations	-
Analysis	The vast majority of the analyzed attacks used American English. Brazilian Portuguese also stood out significantly compared to the other languages.

ple. It is evident that the American English language is the most exploited by attackers. However, because it is a near-universal language, this does not necessarily mean that the United States is the most exploited country.

What attracts the most attention, however, is the result for incidents directed at Brazil, in second place with 21.61% of total occurrences, more than 16 times the amount for British English, the third most exploited. In addition, the occurrences of Brazilian pages as invalid phishing sites are very low compared to valid ones. For years, Brazil has been a country that stands out for phishing incidents on a global scale. Given this, the feature was considered to have WEAK relevance.

4.4.4. F14. Seasonality most exploit

This feature evaluates the most likely time periods for phishing. Given this, it was necessary to categorize fixed calendar events, such as *Christmas* or *Black Friday*. Considering that there may be atypical events, such as the withdrawals from FGTS done in Brazil, different charts were developed as a means of observing fixed and atypical events throughout the years. In Figure 27, the graphs show the data segmented by months during the year 2018, in order to observe fixed events. In Figure 28, the proposed graphs consider the last 5 and 10 years, in order to observe atypical events. The GQM analysis is described in Table 14.

It is important to note that some events considered fixed may not have the same date in every country, or may not even exist. The criterion for considering an event as "fixed" was to consider the majority of countries in the world. "Atypical" events could only be observed when they represented something very significant, that is, when comparing months over the last five years, those with a considerably greater or lesser number of incidents were analyzed.

As shown in Figure 27, the bar graph shows the occurrences of phishing during the months of 2018. The X-axis groups the data by month for 2018, while the Y-axis shows the occurrences as a percentage of the total sample. It was, therefore, possible to observe a pattern in the number of valid phishing submissions, with a monthly average of 8.33% registrations, based on the entire sample #1, with a standard deviation of 0.02. Regarding the fixed events, the pie chart shows that most of the valid phishing incidents in 2018 occurred during the *Christmas* and *Black Friday* events, periods when the global marketplace heats up.

In Figure 28, the first bar graph shows valid phishing occurrences over the past five years. The X-axis presents the corresponding month for the five years observed, and the Y-axis shows the percentage of occurrences. It is possible to observe atypical cases that increased the number of attacks during certain months, resulting in an asymmetry between columns for the same month. For example, in the last five years, some releases of FGTS have occurred in Brazil, such as in 2014 and 2015, due to the public calamity of flooding[11][12]. Because this was a regionally limited event, the propagation was not significant in terms of phishing incidents.

[11] https://glo.bo/2UunEVt
[12] https://bit.ly/2DAf8yX

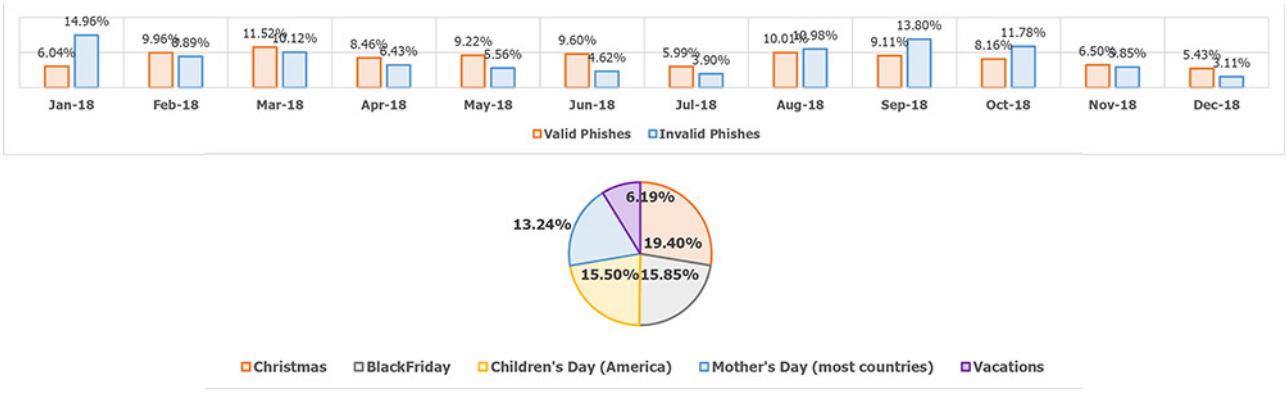

Figure 27: Occurrences of F14 over the year of 2018

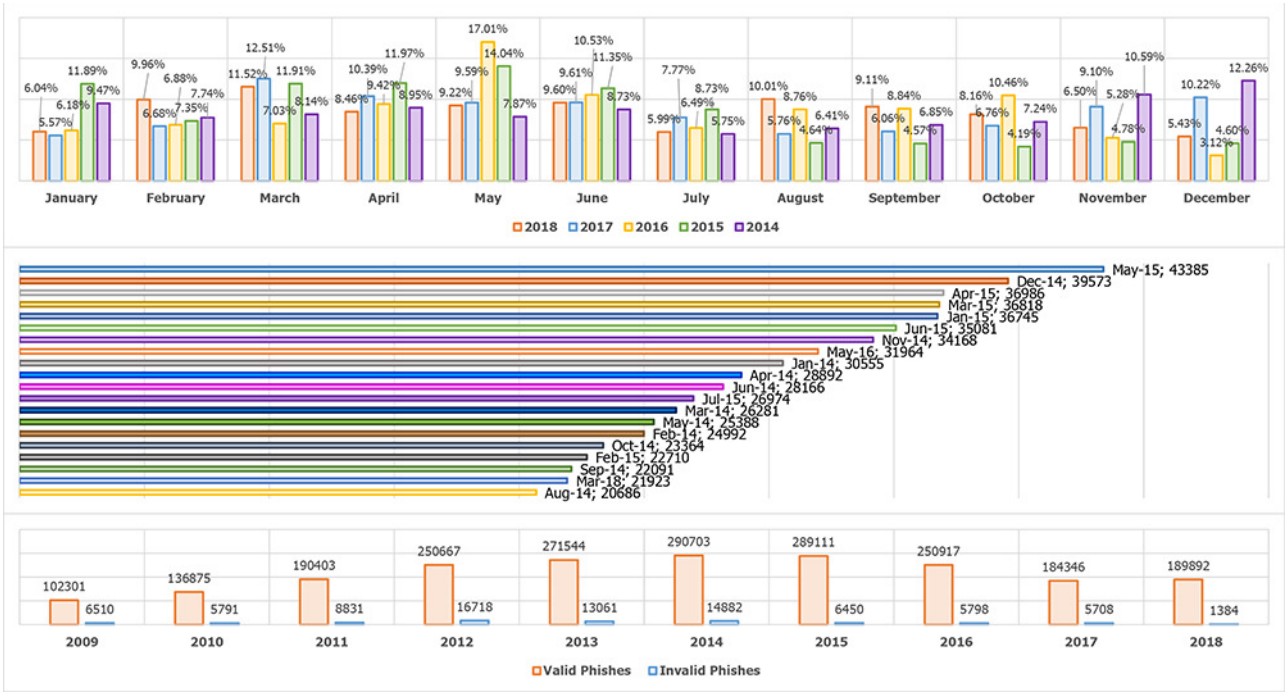

Figure 28: Occurrences of F14 over the last 10 years

However, also in 2015, there was a decree allowing for the withdrawal of FGTS in inactive accounts, causing a considerable increase in incidents that year. This event was repeated in December 2016, meaning that a significant increase was also seen during the first half of 2017. It is important to note that although May 2016 has the highest peak on the graph, at 17.01%, it is necessary to understand that this value is proportional to the total quantity for the respective year, so the value of 14.04% for the same month in the year 2015 is much higher in quantitative terms.

The bar graph in Figure 28 shows the 20 most exploited months over the past 10 years, highlighting a large number of attacks during 2014 and 2015. The atypical FGTS withdrawal event contributed to the occurrence of attacks in 2017 exceeding those in 2018. In this same graph, attention is also raised by the amount of invalid phishing in 2018 compared to other years, which had much lower values, potentially biasing the results between valid and invalid. In the study, an effort was made to keep comparisons proportional in order to minimize this problem.

Finally, as shown in the last bar graph in Figure 28, the X-axis divides the data by year for the last 10 years, with the Y-axis showing the total occurrences for each respective year. It was possible to observe the history of denouncements on the PhishTank for the last 10 years, showing that 2014 was the year with the highest number of attacks. Given this data, the feature was considered to have STRONG relevance.

4.4.5. F15. Service most exploit

This feature evaluates the services most prone to phishing services. For this, it was necessary to make use of the content type categories defined in F10. Data were divided into these categories and identified through automated extraction based on the data reported by PhishTank. The ex-

Table 14
GQM of F14. Seasonality most exploit

Goal 4	Analyze trends in the occurrences of phishing.				
Question	Q14. What is the period most exploited for attacks?				
Metrics	[M31] Count attacks distributed by month for valid phishing sites.				
	[M32] Count attacks distributed by month for invalid phishing sites.				
Hypotesis	There is a seasonality to attacks due to determined or non-calendar events.				
Sample	1 and 2	**Relevance**	STRONG	**Relations**	F11
Extraction	Count the most exploited time periods during the year.				
Limitations	-				
Observations	Events that are scheduled for a certain period of the year, such as Christmas, Mother's Day, and Black Friday were considered.				
Analysis	Events such as Black Friday, Mother's Day, Christmas, New Year's, and other holidays are more appealing to attackers. In addition, certain specific events were identified due to their relevance in the data obtained, such as the case of FGTS withdrawals in Brazil.				

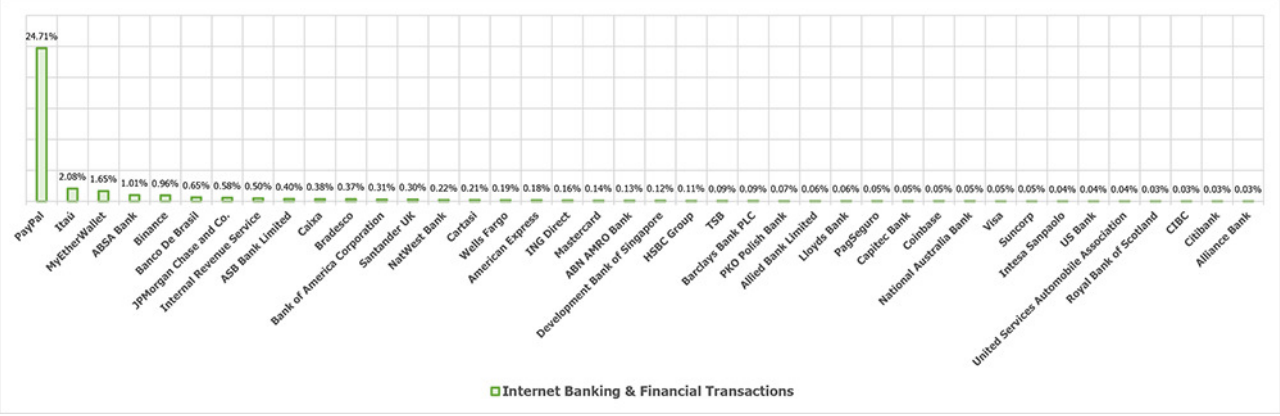

Figure 29: Occurrences of F15 for Internet Banking & Financial Transactions

tracted data are shown in Figures 29, 30, 31, and 32 and the GQM is described in Table 15.

As shown in Figure 29, the X-axis describes the organizations found in the Internet Banking & Financial Transactions segment, while the Y-axis shows the quantities of occurrences as a percentage of the entire sample. The PayPal service was the one most exploited by fraudsters, accounting for almost one-fourth, or 24.71% of all valid phishing sites in Sample #1.1. After PayPal, the most exploited targets are mostly banks operating in Brazil, such as Bradesco, Caixa EconÃťmica Federal, Santander, ItaÃž, Safra, and Banco do Brasil. Because the data was divided into the categories defined in *F10. Content page most exploited*, it was possible to observe, in an individualized manner, considerable exploitation of specific companies in different segments.

For example, as shown in Figure 30, the Telecom and Mobility area had very discrete occurrences. It was possible to observe, however, that consolidated companies, such as Orange and Delta Air Lines, are consistently targeted by fraudsters, others that are still consolidating in the world market, like Uber, have also become targets. Another fact that draws attention is the sheer amount of companies and areas being exploited.

As shown in Figure 31, in the E-commerce segment it was possible to observe the motivation of the fraudsters in making use of brands like Alibaba, eBay, and Walmart. In Social Networks & Computer Services segment, brands like Facebook, Microsoft, Dropbox, and Google are the most exploited. The case of federal government offices also attracted attention, represented by the category Federative Services, which recur frequently and are mostly related to federal taxes and postal services. Finally, in Entertainment & News, as shown in Figure 32, emerging companies such as Steam and Netflix dominate the occurrences of phishing sites within their industry. Given all this, the feature was considered to have STRONG relevance.

4.4.6. F16. TLD most exploit

This feature evaluates the most likely domains for phishing. For this, it was necessary to define categories for the type of domain, such as Commercial, Government, Not Registered, Organizational, and Others. It is important to stress that "Not Registered" refers to domains that use second or third level records, that is, they can be free and are often linked to a top-level domain, making a manual extraction necessary in order to analyze the data. It was also noted whether the URL was with or without "www" and if the third level was used to reference a particular purpose, for example, ftp.example.com, which uses the FTP protocol name to indicate that it would be a download page session. The ex-

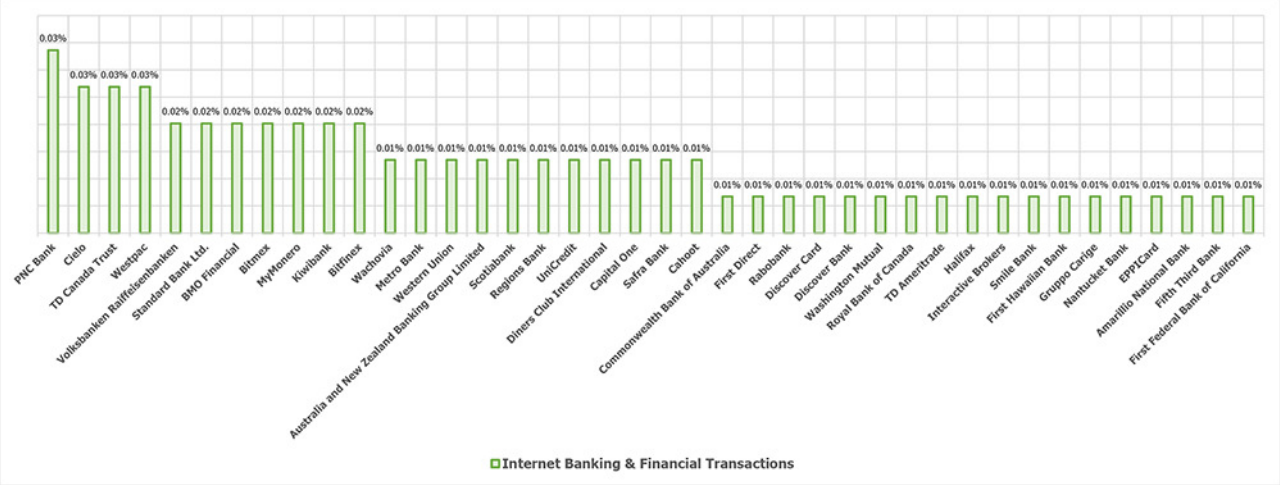

Figure 30: Occurrences of F15 for Internet Banking & Financial Transactions

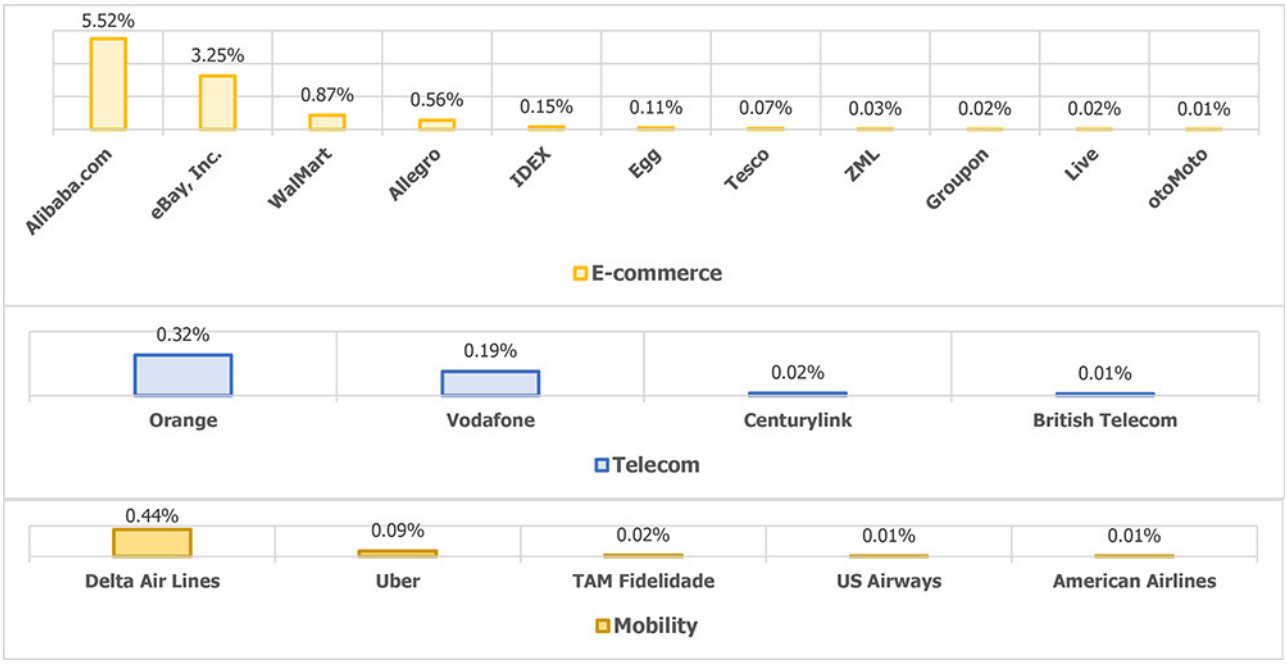

Figure 31: Occurrence of F15 for E-commerce, Telecom, and Mobility

Table 15
GQM of F15. Service most exploit

Goal 4	Analyze trends in the occurrences of phishing.
Question	Q14. What service (brand) is most exploited by attacks?
Metrics	[M33] Count the services most targeted by valid phishing sites.
	[M34] Categorize services based on the categorization defined in F10.
Hypotesis	There is a trend regarding the popularity of a service, and it is more susceptible to attacks.
Sample	3 **Relevance** STRONG **Relations** F11
Extraction	Count the occurrences of brands attacked by looking at the target field from PhishTank Info.
Limitations	The sample obtained in PhishTank had many occurrences of "Other". To minimize this situation, an algorithm was run to identify the mention of a service in the URL. However, records of "Other" with no indication in their URLs had to be discarded.
Observations	The brands were grouped based on the categorization defined in F10.
Analysis	Popular services like Facebook and Google are targets that deserve to be highlighted. However, prominence should also be given to the presence of many Brazilian banks and stores. Finally, PayPal appears as the most exploited service.

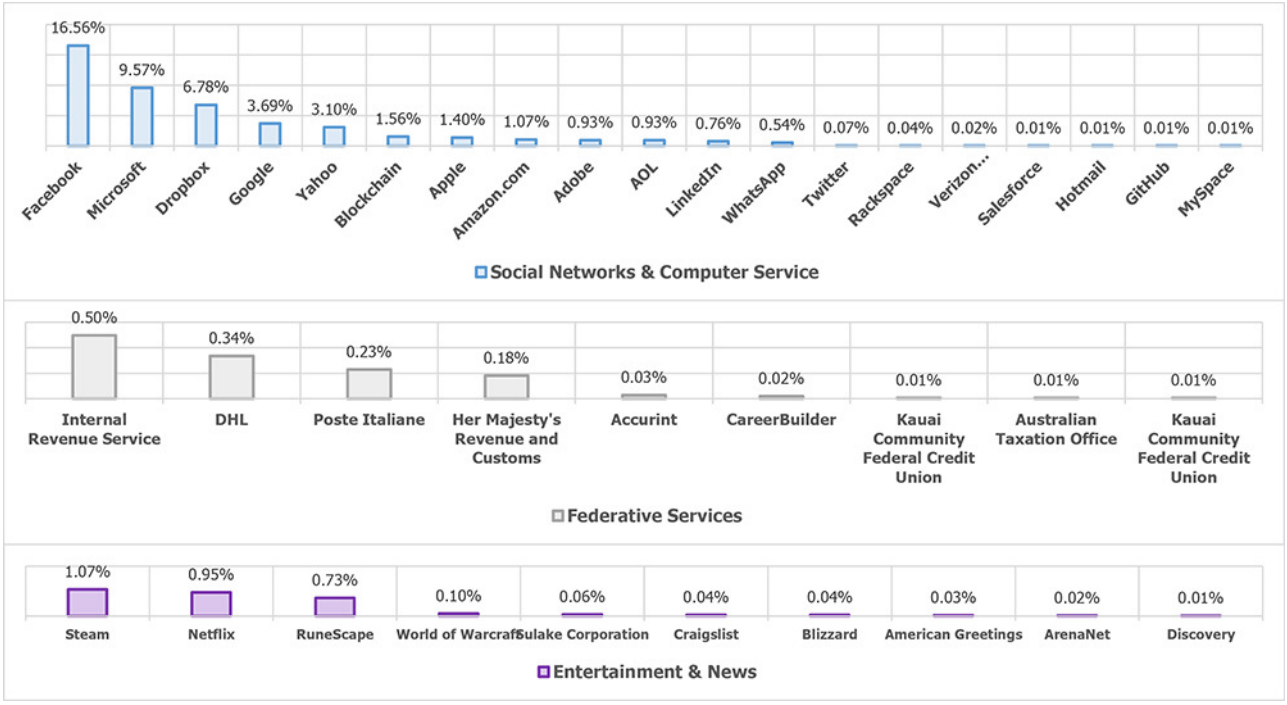

Figure 32: Occurrences of F15 for Social Networks & Computer Services, Federal Services, and Entertainment & News

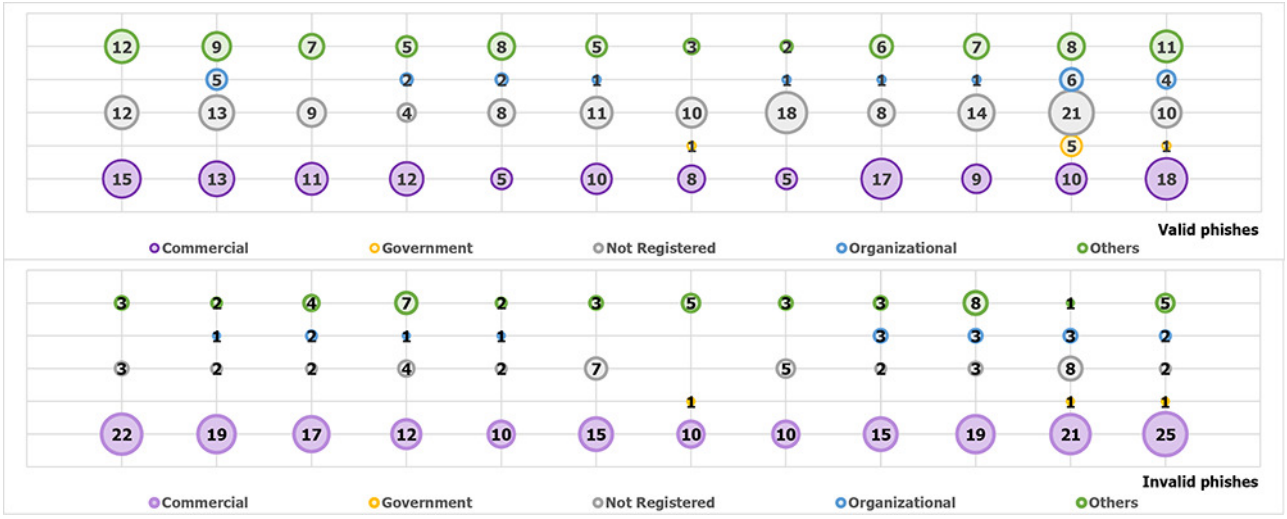

Figure 33: Occurrences of F16 during 2018, by category

tracted data are shown in Figures 33 and 34 and the GQM analysis is described in Table 16.

As shown in Figure 33, the occurrences were grouped by month for 2018 and divided into domain type categories, as shown in the bubble chart, both for valid and invalid phishing sites. It was possible to observe that commercial domains are the most used by attackers. There were more sites in this category than in Others, and almost as many as in Not Registered. Many fraudsters currently register their domains in order give the appearance of veracity to their fraudulent sites thereby requiring a domain registration. In addition, some options, such as .tk, .ga, and .cf, among others, are offered free of charge in the first year or even forever.

Another behavior that occurs is that some of the registered domains are hijacked and used to conduct crimes. Attention should also be drawn to the existence of "Government" and "Organizational" records, which in theory, should be much more difficult to obtain. However, cases of abduction also explain such occurrences. Finally, it was shown that invalid phishing usually has the registered domain of type "Commercial".

As shown in Figure 34, the first graph describes the occurrence of second level domains (SLDs), widely used to describe a section of the site, different from the TLDs that

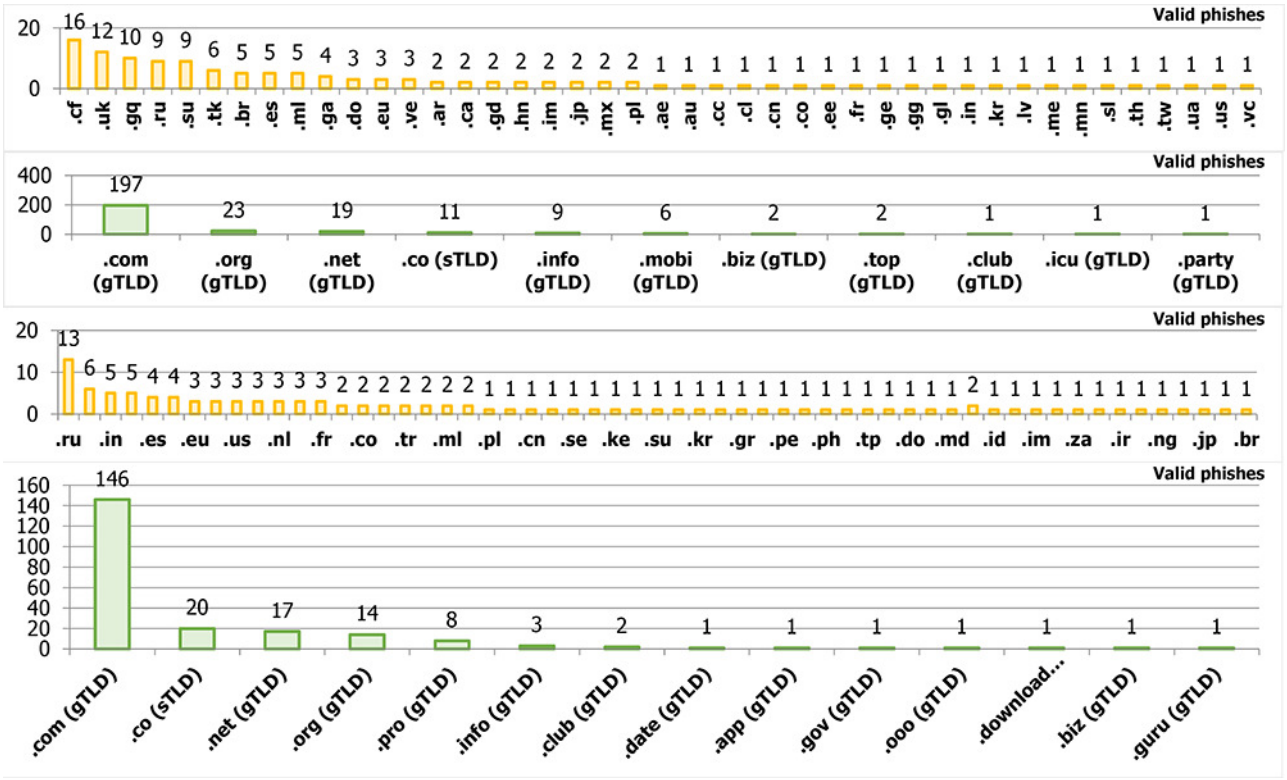

Figure 34: Occurrences of F16 over gTLD and sTLD details

Table 16
GQM of F16. TLD most exploit

Goal 4	Analyze trends in the occurrences of phishing.				
Question	Q15. What is the top domain most exploited by attacks?				
Metrics	[M35] Count the distributed attacks per month for valid phishing sites, considering the use of TLD.				
	[M36] Count the distributed attacks per month for invalid phishing sites, considering the use of TLD.				
Hypotesis	There is a domain level trend, in order to make frauds appear more trustworthy, making a certain level more susceptible to attacks.				
Sample	1.1 and 2.1	**Relevance**	STRONG	**Relations**	F07
Extraction	Obtain the URL and parse the domain manually.				
Limitations	A necessidade de ter apenas registros *online*, de certa forma, pode inviesar o resultado final.				
Observations	It was necessary to perform a manual analysis because some URLs have a TLD, but they have not been registered by the malicious user (for example, cases where a ".com" in the URL refers to the hosting service). In these data, there may be cases where the registry belongs to a domain TLD that has been hijacked by the attacker, for example, when a legitimate server has been invaded.				
Analysis	It was possible to observe many cases where domains were registered, showing a tendency for fraudsters to make their attacks appear more trustworthy. Attention is also called to the cases of use of restrictions such as .gov and .org. It was noted that the vast majority of false positives occurred with .com domain names, of which few are unregistered.				

represent the domain extension. There are techniques to deceive inattentive end users, through which a play of words is used that causes the user to think that the navigation environment belongs to a certain organization.

In the same figure, the second graph shows the generalized purpose domains, which are attractive because they are cheaper or sometimes even free, making it possible for the fraudster to reduce the size of the URL. In the third graph, the domains representing regions are described, showing that ".ru" domains are the most exploited. An interesting fact made clear by this extraction is that, even though an extension may be that of a given country, the fraud will not necessarily be directed to that respective region. In many cases, a page may have an extension of a particular country and have its content in the language of another. Cases of this nature were observed in F13.

Finally, the fourth graph from the same figure displays the number of cases out of the entire sample #1.1, disregarding the categories, therefore making it possible to observe that the .com domains are the most exploited. Given this, the feature was considered to have STRONG relevance.

C1. Browser-based resource	Scale			C2. Community-based strategy	Scale		
	weak	moderate	strong		weak	moderate	strong
F01. Browser punycode exploit	✗	✓	✗	F03. Incident response for community	✗	✗	✓
F02. Malicious browser-based code	✗	✗	✓	F04. Precision for community	✗	✓	✗
				F05. Redundancy entry	✗	✓	✗
C3. Life-cycle	**Scale**			**C4. Target profile**	**Scale**		
	weak	moderate	strong		weak	moderate	strong
F06. Activity time	✗	✗	✓	F11. Content page most exploit	✗	✓	✗
F07. Age of Domain	✗	✗	✓	F12. Host most exploit	✗	✗	✓
F08. Cloning strategy	✗	✗	✓	F13. Language most exploit	✓	✗	✗
F09. SEO score	✗	✓	✗	F14. Seasonality most exploit	✗	✗	✓
F10. Volatility	✗	✗	✓	F15. Service most exploit	✗	✗	✓
				F16. TLD most exploit	✗	✗	✓

Figure 35: The results of features relevance

4.5. Relevance results

Based on the results obtained, the study grouped the relevance obtained as described in Figure 35. The study weighted the quantitative results since, in certain situations, some subjective aspects (such as content and time) were determinant to establish the solidity of a particular behavior. As an example, it was observed that most of the most relevant features were concentrated in the " Life cycle " and " Target profile " categories.

"Life-cycle" features have brought to the fore critical data such as activity, domain age, cloning, and volatility, behaviors that can be critical in classifying a suspected page as fraudulent. Similarly, the "Target profile" category showed that aspects such as host, seasonality, service type, and domain registration can trace out trends in attacks over time. With regard to the "Community-based strategy" category, these are specific problems found in a specific scenario, in other words, it is likely that their consequences will not be propagated on a global scale or may not even occur on another platform, thereby justifying their low average relevance.

4.6. Relationship and similarities results

This section describes the relationships observed between features. These behaviors can have a direct or indirect influence on the result of each feature, in addition to the impact on related features. However, the relationship can also be crossed with different categories, which describes greater sensitivity of cause and effect to similar behaviors among the other features.

Certain behavior combinations describe the abuse of a specific language, as the english language, to attract the exploit the susceptibility (F13), uses port number (F05) or typosquatting exploit into registered domains (F13), for scamming popular services (F15). As Typosquatting occurrences describes when the URLs with spelling errors that appear to be accidental, such as "faceb00k", "Netfliiix", or "dr0pbox" (Stout & McDowell, 2012). In addition, due to careless domain maintainers, malicious domains register to exploit end-user susceptibility (F07). One way would be to induce a URL length (F12, F16), either a long length to obfuscate the malicious address to user views in the browser status bar, or a short length to increase the SEO score (F09). Another noteworthy point is the number of domains registered holographically similar to famous brands, a practice defined as emphcybersquatting citepPatent.

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. In some cases, the report sent to the platform is a URL that has already been reviewed and has gone offline but is currently returning activity with a new uptime. For example, feature (F03) showed that 12.15% of valid phishing sites take between 7 and 15 days to be confirmed on the platform. However, feature (F06) showed that 24.87% of valid phishing sites have an activity time of between 15 days and 1 month, that is, this relationship presents a chronic problem. Consequently, this implies that the blacklist ends up storing a large stack of offline phishing sites (F06, F10) and that community responsiveness needs to be improved (F03, F04).

In addition, it would be interesting for community-based platforms to adopt best practices regarding complaints to avoid unnecessary voting, such as in the cases of duplicity (F05) that were observed, as well as other features (F08) that end up increasing cases of features. For example, when generating a hash for occurrences of features F05, the PhishTank platform considers the two parts of the URL. As discrete as it may be, any change to the second part of the URL already becomes the same as being susceptible to deviation, resulting in duplicity. The ability to parameterize values on the way or in the query makes it a common practice among malicious users, generating a lot of unnecessary effort in the community platform review system.

Other combinations are designed to exploit a brand's reputation, such as social engineering practice on phishing attack profiles (F11, F13), as well as the use of techniques to increase the reliability of fraudulent pages (F01). This practice strengthens attacks that are directed at a specific organization (F02), known as Spear Phishing, or a specific period (F14). However, instead of striving to register a malicious domain, the attacker can make use of a legitimate URL that redirects the user to an address stored via QueryString.

However, it is also possible for an attacker to hijack an

existing domain through some vulnerability, such as injecting files into an upload session. The attacker uses the domain and benefits from the same advantages mentioned above as well as others such as uptime (F06), and low volatility (F10), thus contributing to the SEO score (F09) and user confidence in certain periods of the year (F11) in cases where the domain owner has prestige in a particular activity, such as e-commerce.

Other observations refer to malicious strategies that consider dynamic aspects like time and context, such as maintaining a fraudulent activity on different servers (F06, F08), using a short phishing life (F06, F10) and allowing it to be quickly abandoned, with the certainty that the site will be caught by some blacklist and its efforts only last for a short period. Another aspect is the content (F11, F14) and service (F14, F15) that is exploited during a certain time of year, directing the attacks by seasonality. Finally, other threats refer to aspects that exploit browser features that could be minimized by their respective maintainers, such as the policy of certain javascript codes or plug-in and extension installations (F02).

5. Threats and limitations

This section describes some threats and limitations that need to be considered in the study protocol, ie points that may skew the results if you reproduce the research in another scenario.

5.1. Threats from dynamic features

Some features may be founded in the literature but these not present in this study. Firstly, through the 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 (da Silva et al., 2019). An example of a feature not covered by the study is the Google page rank, because it has been unavailable since April 18, 2016 (Dunlop et al., 2010).

In addition, certain features, such as F07, had to query data through the *WHOIS* protocol, a process that may have some obstacles, such as private domains or domains that do not belong to the malicious user, such as cases of domain hijacking or even domains that are offered by commercial web hosts, such as sites.google.com. Due to this behavior, it was necessary to perform a more subjective extraction, analyzing the content manually, thereby justifying the use of a reduced-size sample (da Silva et al., 2019).

5.2. Threats from context-aware taxonomy

In Figure 5, the categories are classified for the purpose to analyze the context of phishing ecosystem. 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 (da Silva et al., 2019). As an example, features F07, F10 and F14 when combined, can justify feature F06. Another interesting point is that the composition of the 3 features does not overlap their respective behaviors, as structured by the taxonomy. This phenomenon suggests a good adherence to the taxonomy proposed by the study.

5.3. Threats from Sampling and Data definition

The process of receiving newly created JSON files identified that the PhishTank **removed** or **added** platform in a given time frame. Although it is not possible to confirm the reason for these activities, some justifications can be suggested.

Regarding the removal activities, it is possible to suggest the hypothesis of the platform to carry out removals to mitigate false positive problems registered on the platform. The interesting thing is that the platform emphasizes the possibility for the user to be able to report these types of failures [13], something seen as very positive. During the research of the current study, it was possible to observe that the resource works well and the user who notifies has quick feedback on his observation.

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 (da Silva et al., 2019). That is, the moment of transition between " invalid " to " valid " suggests the possibility that new entries will gradually appear in later JSON.

An observed behavior was that a JSON obtained on 01/15/2019, the entries for January and February 2019 had 358 and 617 records respectively (da Silva et al., 2019). In contrast, by accessing the " phish search " link provided by the PhishTank platform, it was possible to observe that January and February 2019 had 11,503 and 18,953 entries (da Silva et al., 2019). Such behavior suggests that files from previous months are more susceptible to removals than the most recent months, which justified periodic monitoring in each month of extraction.

5.4. Threats from Results

Due to its subjective nature, the scale proposed by the study (*WEAK*, *MODERATE* and *STRONG*) (da Silva et al., 2019) suggests statements that are based on interpretations supported by the result of the empirical research. Details on each limitation will be presented below.

5.5. Threats from Browser-based resource results

A limitation of this diagnosis occurs with pages displaying a 404 error or account suspended by the hosting server, making it impossible to obtain the source code of some pages present in the sample, which then had to be discarded from the analysis, resulting in a phishing rate discreetly lower than that actually practiced by malicious users. It is also important to highlight the difficulty in retrieving evidence regarding SMiShing attacks. Because only the Web side was an-

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

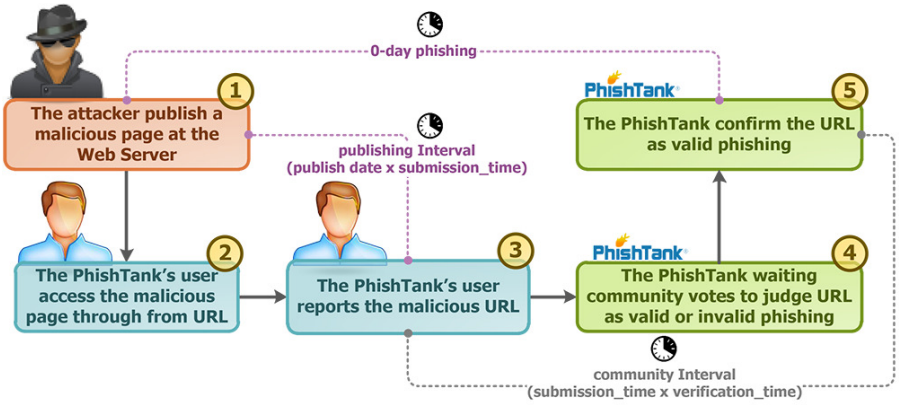

Figure 36: Checkpoints for the phishing life-cycle

alyzed, such as the URL and malicious page content, other features belonging to the attack cycle, such as the SMS messages, could not be intercepted, which would provide more evidence to test the hypothesis.

5.6. Threats from Community-based strategy results

As shown in Figure 36, if there was a way to extract information from the domain, some important milestones could be analyzed, such as the interval between steps 1 and 3, which would be the window of time when the phishing site was published on the Web until being denounced on the platform, thus estimating the **average denunciation time**. In the same vein, the interval between steps 1 and 5 would be the **0-day phishing period**, that is, the average time that a phishing site stays immune from the blacklist.

However, for greater accuracy regarding denunciation time and *0-day*, it would be necessary to look at the date on which the phishing activity began. One way to get this information would be through the "Creation Date" obtained from the *WHOIS* protocol. However, this cannot guarantee accuracy as to the extent of the malicious activity, i.e. the "Creation Date" does not suggest a timeframe for a page's illicit practices, because there could have been idle time before the fraudulent activities began.

Another interesting feature for investigation would be the **synchronization between the platform database and the respective browser that uses the platform as support for the protection mechanism**. For example, on the PhishTank page, certain phishing sites were confirmed but not recognized as a threat when accessed through the Opera browser. The reason for this was the delay in synchronization of the browser blacklist with the records of the respective repository, giving evidence for a potentially chronic problem that deserves to be investigated (da Silva et al., 2019). However, for greater accuracy regarding the synchronization delay, it would be necessary to compare for each phishing site the date and time of access against the date and time of confirmation in the repository, obtaining the difference.

5.7. Threats from Life-cycle results

Once a phishing site assumes the status "offline", it is disregarded by future JSON. However, the reason for its unavailability may be temporary, and if it returns to activity, the platform will continue to consider as "offline". It is likely that the same malicious URL will have to be submitted to the complaint and voting process again in order to be added again to the JSON. Faced with this, tools that operate together with the API need to store numerous records of old phishing to avoid the possibility of false negatives. This results in an immense phishing database that must be continually consulted, even though many are inactive, creating unnecessary and avoidable effort. Such behavior could be investigated in a future study.

Another behavior would be to evaluate the frequency of phishing changes over time through the WHOIS protocol's "Updated Date" field. However, the scope of these operations refers to the DNS and not to the page itself, i.e. it is only applicable to phishing cases that have a registered domain. It would also lose some precision because the illicit activity can occur both before and after domain registration. Likewise, a change in "Update Date" does not imply that the contents of the page have been modified, making it possible to skew the results.

With regard to the monitoring of changes undergone by phishing sites, Figure 20 describes the emergence of clones throughout the year using the hash but does not observe changes in the respective clones. An alternative would be to observe the changes and also monitor the phishing activity by hash, however, this was not contemplated in the current state of the study, for reasons of scope.

However, in certain cases, tracking phishing events through their hash can have limitations. For example, any additional information, such as an advertising banner, which would not necessarily be inserted by the fraudster, but rather by the hosting server, could be enough to modify the hash of the response. In addition, related, dynamic information can also modify the hash, such as informing the current date or data extracted from cookies. Such behaviors may reduce the number of clones that are captured. In contrast, there is also the chance of a hash collision, that is, separate pages that have

Studies	Year	Proposed Taxonomy	C1. Browser-based resource		C2. Community-based strategy			C3. Life-cicle					C4. Target profile					
			F01	F02	F03	F04	F05	F06	F07	F08	F09	F10	F11	F12	F3	F14	F15	F16
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	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 37: Related Works

the same hash, erroneously increasing the number of clones. However, the chances of this happening are very low when using the SHA-256 algorithm.

5.8. Threats from Target profile results

An opportune factor for the commission of cybercrimes is the diversity in conduct with regard to punishments for digital crimes caused by the policies of a particular state or country. In light of this, it would be interesting to analyze the region of activity most exploited by the ill-intentioned. Several servers that store malicious pages are geographically located in countries where there is almost no legislation that responds to crimes of this nature. This explains why there can be phishing sites that continue to be active for years.

Although the study analyzed the language and TLD of the region, thereby identifying the target population, it does not yet mean that the fraudulent site is operating in the respective target region. It is very likely that there are several phishing cases targeting a locality that is hosted on a server operating in another region. Due to scope issues, it was not possible to identify the region of phishing activity. Finally, feature F12, which evaluates the most exploited hosting service, had cases in which a phishing site had a registered domain configured with masking[14]. Such cases made it impossible to know the hosting server of the fraudulent site, so they ended up not being considered in the study.

6. Related Works

In this section, papers with correlated approaches will be presented, that is, studies published in the literature that present reflections on observation-based phishing behaviors, either through empirical researches or other study types. As illustrated in Figure 37, beyond those studies with a correlated approach, each study that made use of a taxonomy was also identified. Finally, the number of similar features between the studies was taken into account.

Some studies in the literature present phishing detection techniques that take into account the human factor, such as Khonji et al (Khonji et al., 2013). This study considers attack and defense techniques, as well as preventions and countermeasures, through a classification structured by a taxonomy. The study by AlEroud and Zhou (AlEroud & Zhou, 2017) has a proposal based on a taxonomy of attacks built through evidence extracted from the literature, in order to propose countermeasures against new types of phishing, and guide teams that need to offer anti-phishing solutions in the current scenario using the taxonomy.

Other studies, such as Sharma et al. (Sharma et al., 2017), aim to compare eight anti-phishing tools to evaluate their performance in a controlled experiment. In addition to PhishTank, the study featured a phishing sample extracted from APWG reports. Also, similar to the present study, it presented a comparison between real phishing and legitimate sites, in order to observe any differences present. Similarly, the study by Goel and Jain (Goel & Jain, 2018) evaluated anti-phishing mechanisms active on mobile devices and proposed a methodology divided into four phases. The phases consider the context, variations of attacks, countermeasures, and reflections on the challenges of fighting phishing in the mobile context.

Other studies also present discussions of approaches to executing a phishing attack. Chiew et al. (Leng Chiew et al., 2018) conduct a systematic review to assist researchers in understanding the features present in the attacks, resulting in a taxonomy. Similar to the present study, this study also highlights gaps in the heuristic solutions segment. In the same vein, the study by Qabajeh et al. (Qabajeh et al., 2018) presents a reflection that takes into consideration aspects such as awareness, training, computational intelligence, and legal issues involved in heuristic-based anti-phishing solutions. Like the present study, it also analyzed the pros and cons of heuristic solutions. The reflection presented is also intended to serve as a foundation for new solutions.

In light of the related studies, none except for F07 and F09 addressed the remaining 14 features between F01 and F16, indicating a potential for further reflections in the literature based on the present study. Because they are more intrinsic to context, these features tend to be explored less in the literature. This behavior demonstrates the tendency of proposals that are skewed towards heuristics to be restricted to statistical strategies. While such approaches have their benefits, they can be more susceptible to problems of context, such as the challenge of concept drift discussed. Promoting higher sensitivity to context can reduce the gaps provided by dynamically changing scenarios.

[14] http://support.godaddy.com/help/article/422/forwarding-or-masking-your-domain-name

7. Conclusion and Further Works

This study presented a empirical research 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 Browser-based resource

With regard to browser-based features, was describes a chronic problem regarding the targeted phishing attacks. It was possible to observe a large number of attacks on the end user through Web browser resources that were triggered by the URL itself and javascript source-code.

7.2. Challenges of the Community-based strategy

With regard to community-based features, a chronic problem regarding the incident responses by PhishTank was observed. Because the voting system does not have an estimated deadline or report the number of votes needed for completion, there is a delay of one to seven days before verdicts are rendered in 49.80% of the records, indicating a window of vulnerability. It would be reasonable for the protection mechanisms that use PhishTank, such as Opera, to consider phishing sites that have not yet been confirmed to be suspicious, a warn that the environment may be hostile. Despite the gap, the reports submitted to the platform presented a high index of accuracy, with a standard deviation of 0.54.

7.3. Challenges of the Life-cycle

Regarding life-cycle-based features, the short life of phishing sites was evident. When the activity times of valid phishing sites were compared with invalid phishing sites, a much shorter lifespan was observed for the fraudulent sites, with 74.89% of them have an activity time between 12h and 2 months. Consequently, the volatility of phishing in the transition from "online" to "offline" is quite high. Almost 80% of existing phishing sites tend to go offline. For disabled sites, 59.82% are concentrated between 10 and 13 months of activity. Nonetheless, there are a number of phishing sites that are cloned, demonstrating that the practice is well-exploited and its treatment could reduce effort on the part of community-based strategies.

7.4. Challenges of the Target profile

With regard to the features based on the target profile, this category dealt with the resources most exploited by attackers, making it possible to analyze aspects having greater susceptibility to attack. It was possible to observe that banking information and financial transactions content is the most often exploited. Another aspect is that the hosting service *000webhostapp* accounted for 10.09% of all fraudulent sites cataloged in sample #3, indicating it to be the most exploited service. Some results were predictable, such as American English being the most exploited language, but the fact that the second most exploited language was Brazilian Portuguese was interesting, indicating that Brazil is a region widely exploited by fraudsters.

Another behavior that reinforces the theory regarding attacks directed at Brazil was the seasonality of phishing incidents during the calendar year. The latter months of the year have significant occurrence spikes around fixed and scheduled events, such as "Black Friday" and "Christmas". However, sporadic cases, such as the withdrawals from FGTS in Brazil brought alarming numbers. Similarly, an examination of the most exploited services made it clear that Brazilian banks are constant targets. There are also a considerable number of ".br" domains being registered or even hijacked.

7.5. The Road Ahead for the Heuristic Strategies

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 (da Silva et al., 2019).

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 (da Silva et al., 2019). The designer of the classification model would have the responsibility to establish weights for each feature considered, which can be supported by the relevance analysis. The relevance grouping, shown in Figure 35, suggests support in maturity in the listing which features may be more or less relevant. Proposing a classification model has its challenges, however, results of this study, such as (i) **categorization**, (ii) **relevance analysis**, and (iii) **grouping of features** can mitigate the effort.

Based on (i), a feature suggests a categorical continuous value variable. A categorical variable would be whether phishing is a clone of another (F08). A continuous variable, on the other hand, would be the time in minutes that phishing took to be confirmed (F03). As the classification model needs to deal with categorical values, it would be necessary to transform the continuous variables into categorical ones. Through descriptive statics, the data obtained from the empirical research results from a perspective that helps in the conversion of continuous variables to categorical ones. For example, the graph in Figure 14 segments continuous results at pre-established intervals into 19 fixed variations on the X-axis, thus resulting in a variable that could have categorical values.

When proposing an anti-phishing mechanism, it is important to establish which features have greater or lesser weight for the prediction, as already stated in Section 4.5. However, the phishing environment, being very dynamic, is very sensitive to changes, which can hinder the process of relevance because new trends emerge, and the direction of phishing behavior patterns are very varied (da Silva et al., 2019). This means that establishing a heuristic based on lexical patterns has its risks, either in the URL or in the content of the page.

Therefore, there is a motivation to observe other patterns that are part of the context of phishing activities, such as browser resources, SEO rank, weather aspects such as uptime and takedown, as well as seasonal events, such as Black Friday. In the same line, the analysis of this study is expected to assist in the challenge of balancing features in a classification model, as mentioned in Section 1. In addition, the charts resulted in Section 4 can guide input data during the classification model construction, as well as aid in preprocessing of model training.

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 that ensures greater sensitivity to features similarities in the phishing context, as discussed in Section 4.6. Therefore, a grouping avoids overlapping attributes that work with a similarities evaluation (da Silva et al., 2019).

Targeted phishing attacks, such as *Spear Phishing* or *SMiShing*, for not having a generalized scope, suggest a richer elaboration in details on the profiles of the parties involved, which results in greater wealth on the fraud's reliability, with the intention of to explore the user's susceptibility. Therefore, strategies for **brand protection** need to be adopted. Attacks already mentioned, like **cybersquatting** and **typosquatting**, have a lot of adherence in the strategy of targeted attacks since they target domains, subdomains and use of keywords in SEO (da Silva et al., 2019).

However, in its current state, the present study cannot be undertaken, due to issues of scope. Strategies of analyzing **textual and visual identity** can be adherent to the prediction models and are considered as future studies by this study. Finally, pillars (i), (ii) and (iii) focus on providing assistance in **responsiveness** and **response time** for new classification models or the robustness of existing models.

References

Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, *41*(13), 5948 - 5959. doi: https://doi.org/10.1016/j.eswa.2014.03.019

ACFE. (2018). Report to the nations: 2018 global study on occupational fraud and abuse. *Available on: http://bit.ly/2MJ4zgm*.

Adebowale, M., Lwin, K., SÃąnchez, E., & Hossain, M. (2019). Intelligent web-phishing detection and protection scheme using integrated features of images, frames and text. *Expert Systems with Applications*, *115*, 300 - 313. doi: https://doi.org/10.1016/j.eswa.2018.07.067

AlEroud, A., & Zhou, L. (2017). Phishing environments, techniques, and countermeasures: A survey. *Computers & Security*.

Alkhozae, M. G., & Batarfi, O. A. (2011). Phishing websites detection based on phishing characteristics in the webpage source code. *International Journal of Information and Communication Technology Research*.

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

Amoroso, E. G. (1994). *Fundamentals of computer security technology*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of software engineering*. Wiley.

Bishop, M. (1999). *How attackers break programs, and how to write programs more securely* (Tech. Rep.). Technical Tutorial Session T1, University of California, Davis, August 24, 1999: University of California at Davis.

Chaudhry, J. A., Chaudry, S. A., & Rittenhouse, R. G. (2016). Phishing attacks and defences. *International Journal of Security and its Application*.

Chelliah, G. A., & Aruna, S. (2014). Preventing phishing attacks using anti-phishng prevention technique. *International Journal of Engineering Development and Research*.

Costello, A. M. (2003). Punycode: A bootstring encoding of unicode for internationalized domain names in applications (idna). *Available in: https://tools.ietf.org/html/rfc3492*.

da Silva, C. M. R., Feitosa, E. L., & Garcia, V. C. (2019). Heuristic-based strategy for phishing prediction: A survey of urlbased approach. *Computers & Security*. doi: https://doi.org/10.1016/j.cose.2019.101613

Dunlop, M., Groat, S., & Shelly, D. (2010, May). Goldphish: Using images for content-based phishing analysis. In *2010 fifth international conference on internet monitoring and protection* (p. 123-128). doi: 10.1109/ICIMP.2010.24

Elwell, R., & Polikar, R. (2011, Oct). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *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. *Computers & Security*, *73*, 519–544. Retrieved from https://doi.org/10.1016/j.cose.2017.12.006 doi: 10.1016/j.cose.2017.12.006

Google. (2019). Safe browsing. *Available at: https://safebrowsing.google.com/*.

Gowtham, R., & Krishnamurthi, I. (2014, 02). A comprehensive and efficacious architecture for detecting phishing webpages. *Computers & Security*, *40*, 23âĂŞ37.

Howard, J. D. (1998). *An analysis of security incidents on the internet 1989-1995* (Unpublished doctoral dissertation). Carnegie Mellon University, Pittsburgh, PA, USA. (UMI Order No. GAX98-02539)

Jovanovic, G. (2009). Standardization of the old church slavonic cyrillic script and its registration in unicode.

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 Communications Surveys and Tutorials*, *15*(4).

Kirda, E., & Krugel, C. (2005). Protecting users against phishing attacks. *The Computer Journal*.

Krsul, I. V. (1998). *Software vulnerability analysis* (Unpublished doctoral dissertation). Purdue University, West Lafayette, IN, USA. (AAI9900214)

Leng Chiew, K., Yong, K., & Tan, C. L. (2018, 03). A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, *106*. doi: 10.1016/j.eswa.2018.03.050

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using url and html features for phishing webpage detection. *Future Generation Computer Systems*.

Lindqvist, U., & Jonsson, E. (1997, May). How to systematically classify computer security intrusions. In *Security and privacy, 1997. ieee symposium on* (p. 154-163).

Lough, D. L. (2001). *A taxonomy of computer attacks with applications*

to wireless networks (Unpublished doctoral dissertation). Virginia Polytechnic Institute and State.

Lumley, T. (2011). *Complex surveys: A guide to analysis using r*. Wiley. Retrieved from https://books.google.com.br/books?id=L96ludyhFBsC

Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. *Expert Systems with Applications*, *53*, 231 - 242. doi: https://doi.org/10.1016/j.eswa.2016.01.028

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* (pp. 58:1–58:6). New York, NY, USA: ACM. doi: 10.1145/2961111.2962619

Moore, T., & Clayton, R. (2007). Examining the impact of website takedown on phishing. In *Proceedings of the anti-phishing working groups 2nd annual ecrime researchers summit* (pp. 1–13). New York, NY, USA: ACM.

Naresh, U., Sagar, U. V., & Reddy, C. V. M. (2013). Intelligent phishing website detection and prevention system by using link guard algorithm. *IOSR Journal of Computer Engineering*.

OpenDNS. (2019). Phishtank. *Available at: https://www.phishtank.com/*.

Qabajeh, I., Thabtah, F., & Chiclana, F. (2018, 08). A recent review of conventional vs. automated cybersecurity anti-phishing techniques. *Computer Science Review*, *29*, 44-55. doi: 10.1016/j.cosrev.2018.05.003

R. Babbie, E. (2019, 05). 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* (second ed.). Malden: Blackwell Publishing.

Sharma, H., Meenakshi, E., & Bhatia, S. K. (2017, May). 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)* (p. 1437-1442). doi: 10.1109/RTEICT.2017.8256835

Singh, R., & Mangat, N. S. (1996). Stratified sampling. In *Elements of survey sampling* (pp. 102–144). Dordrecht: Springer Netherlands. Retrieved from https://doi.org/10.1007/978-94-017-1404-4_5 doi: 10.1007/978-94-017-1404-4_5

Srinivasa, R., Alwyn, R., & Pais, R. (2019). Jail-phish: An improved search engine based phishing detection system. *Computers & Security*.

Stout, B., & McDowell, K. (2012, 10). *United states patent* (Tech. Rep.). Citizenhawk, Inc., Aliso Viejo, CA (US). Retrieved from https://patentimages.storage.googleapis.com/2c/d7/19/1b58c99bb246c4/US8285830.pdf

Vayansky, I., & Kumar, S. (2018, 01). Phishing âĂŞ challenges and solutions. *Computer Fraud & Security*, *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. In *Ndss '10*.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in software engineering: An introduction*. Norwell, MA, USA: Kluwer Academic Publishers.