Received 22 March 2023, accepted 9 April 2023, date of publication 18 April 2023, date of current version 28 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3268027

RESEARCH ARTICLE

PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification

FELIPE CASTAÑO, EDUARDO FIDALGO FERNAÑDEZ, ROCÍO ALAIZ-RODRÍGUEZ, AND ENRIQUE ALEGRE
Department of Electrical, Systems and Automation Engineering, University of León, 24071 León, Spain
Spanish National Institute of Cybersecurity (INCIBE), 24005 León, Spain

Corresponding author: Felipe Castaño (felipe.castano@unileon.es)

This work was supported by the framework agreement between the University of León and the Spanish National Cybersecurity Institute (INCIBE) under Addendum 01.

ABSTRACT Recent studies have shown that phishers are using phishing kits to deploy phishing attacks faster, easier and more massive. Detecting phishing kits in deployed websites might help to detect phishing campaigns earlier. To the best of our knowledge, there are no datasets providing a set of phishing kits that are used in websites that were attacked by phishing. In this work, we propose PhiKitA, a novel dataset that contains phishing kits and also phishing websites generated using these kits. We have applied MD5 hashes, fingerprints, and graph representation DOM algorithms to obtain baseline results in PhiKitA in three experiments: familiarity analysis of phishing kit samples, phishing website detection and identifying the source of a phishing website. In the familiarity analysis, we find evidence of different types of phishing kits and a small phishing campaign. In the binary classification problem for phishing detection, the graph representation algorithm achieved an accuracy of 92.50%, showing that the phishing kit data contain useful information to classify phishing. Finally, the MD5 hash representation obtained a 39.54% F1 score, which means that this algorithm does not extract enough information to distinguish phishing websites and their phishing kit sources properly.

INDEX TERMS Cybersecurity, cybercrime, cyber threats, phishing, social engineering, phishing kits.

I. INTRODUCTION

The Internet has become more and more accessible over the world in the last decades, going from 20% of the world population with Internet access in 2005 to 63% in 2021 [1]. This amount represents 4.9 billion people using the internet. With this exponential growth, protecting internet users and their data has become a concern textcolorbluefor Law Enforcement Agencies (LEAs), research centres, and people in general.

As a textcolorblueresult, researchers have focused on important topics related to cybersecurity. Recent relevant works include spam identification or classification [2], [3], bots detection to early response and Distributed Denial of Service (DDoS) attacks [4], [5], algorithms to classify suspicious content on the darknet [6], [7], [8], [9], and even image analysis as a forensic tool to detect criminal activity on videos [10], [11].

Phishing is a cybercrime that uses social engineering and aims to deceive people and steal their financial account credentials or other sensitive data [12]. Phishers imitate websites to impersonate well-trusted companies and request victims' personal and sensitive information, as shown in Figure 1.

Phishing has become one of the most common cyberattacks due to the exponential growth of the Internet [13]. The Anti-Phishing Working Group (APWG) reported a huge increase in the second quarter of 2022 [14], finding 1, 097, 811 phishing attacks, a record number, making that quarter the worst ever observed. This increase in the number of attacks has also been motivated by the changes that have taken place during the pandemic, as the studies by Hijji et al. [15] and Alzahrani et al. [16] suggest.

The associate editor coordinating the review of this manuscript and approving it for publication was Jemal H. Abawajy.

APWG also reported that Financial institutions, Software as a Service (SaaS) websites, webmails, social media, payments, and e-commerce are the target of around 74% of the phishing attacks [14]. That indicates that phishers are defining their targets to deploy massive attacks to achieve higher profits.

A phishing website deployment is composed of two steps: first, set up a server where the attacks can be deployed, which may include an SSL certificate configuration. And second, the definition of a URL and HTML source code that will imitate the website [17].

Recently, researchers have found that phishing attacks have changed, and cyber criminals are using phishing tool kits to deploy attacks in a faster, easier and more massive way against defined targets [18], [19], [20]. Phishing kits contain ready-to-deploy phishing websites, scripts to automatically save stolen data, and other functionalities to help them deceive people effortlessly. Phishers can release attacks in a short time on different domains to the same target by using these kits.

One of the most important functionalities of a phishing kit is to save the stolen data [21], [22]. There are different approaches according to their complexity: (i) saving the stolen credential into plain text, (ii) sending the stolen data to a specific email each time a victim is deceived, (iii) phishing kits that contain control panels and use databases to show the stolen credentials to the phishers [20].

Other phishing kit functionalities found by researchers are: Obfuscation [23] and server-side traffic filters (cloaking) techniques [20], [24]. In the first case, obfuscation techniques make phishing website attack analysis more difficult for researchers and LEAs. In the second case, cloaking techniques block or redirect unwanted connections to avoid third parties or automatic algorithms from crawling the attack. The use of phishing kits has changed the phishing attack process and, consequently, the phishing lifecycle has changed as well.

Authors have begun to study phishing kits more intensively because of the potential danger of massive phishing attacks. Having a dataset available is one of the first requirements, acknowledging that its creation usually supposes a significant challenge for studying phishing kits. Authors currently collect their data to evaluate methods using well-known phishing sources. These sources present phishing kits but do not present their relationship to phishing website attacks. Therefore, researchers use the collected samples to create phishing websites [25]. As a result, the evaluation is limited since the data to test the methods could be affected by the decisions taken by the researchers at the moment of collecting and creating the samples.

Other researchers collect phishing website samples on the Internet using a different process from those followed during the phishing kit collection one [20], [23], [25]. In this way, the authors do not interfere with the data collection and get phishing website samples directly from the wild under more realistic conditions. However, it is impossible to correctly associate the previously collected phishing kit with the phishing website samples directly collected from the Internet. This means that there is no clear ground truth. Without a reliable relationship between the phishing kits and the real phishing websites collected on the Internet, the results presented by the researchers do not use the common metrics for a classification problem, and they need to check results manually to assess the performance of the models.

This paper overcomes the previous drawbacks by presenting a methodology for collecting datasets where the phishing websites are clearly associated with their phishing kit source. Using this methodology, we created and made publicly available PhiKitA, a dataset containing phishing kits, phishing websites created by them and even traces of a phishing campaign. We also evaluated and compared the performance of several classification and clustering algorithms from the literature in our presented dataset.

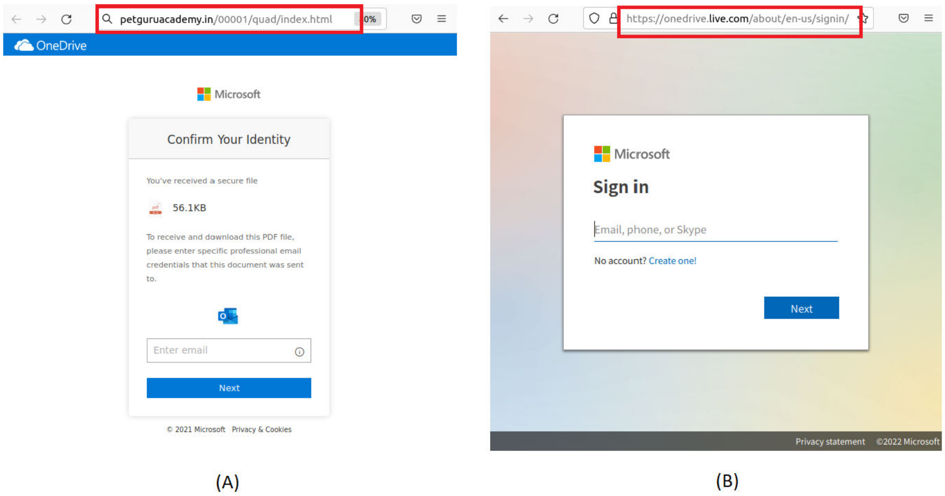
The main contributions of this paper can be summarized as follows:

- We propose a methodology for collecting datasets that guarantees that the provided phishing websites are related to their phishing kit source. Using this methodology, we avoid the particular conditions introduced to the data by the decisions made by authors when creating phishing websites. We also guarantee the relationship between phishing kits and phishing website attacks as they are collected in the same process.
- We present PhiKitA, the first dataset, up to our knowledge, with a ground truth that is correct, presenting an accurate association between phishing kits and real phishing websites on the Internet. PhiKitA contains 510 phishing kit samples, 859 phishing website attacks and 1141 legitimate samples, and traces of a phishing campaign.
- We evaluate three different algorithms from the literature comparing their results on PhiKitA. For the first time, we evaluate the performance of these algorithms in three different experimental setups: familiarity analysis, phishing detection and multi-class classification to detect the source of a phishing website.

The organization of the paper is as follows. Section II introduces the literature review of phishing kits. Later, Section III describes the proposed methodology to collect phishing kit samples and phishing website attacks, and it presents the content of the collected dataset. Section IV describes the experimental setup, the proposed experiments on the data and the algorithms evaluated. The results are discussed in Section V. Finally, the main conclusions and our future work are presented in Section VI.

II. STATE OF THE ART

Literature about phishing kits could be divided into two groups. The first group includes approaches focused on analyzing phishing kit behavior, which will be reviewed in Section II-A. This analysis contributes to the understanding of any process of phishing website attacks.


petguruacademy.in/00001/quad/index.html
OneDrive
Microsoft
Next
(A)
https://onedrive.live.com/about/en-us/signin/
Microsoft
Next
(B)

Confirm Your Identity

You've received a secure file

56.1KB

To receive and download this PDF file, please enter specific professional email credentials that this document was sent to.

Enter email

© 2021 Microsoft Privacy & Cookies

Sign in

Email, phone, or Skype

No account? Create one!

Privacy statement ©2022 Microsoft

FIGURE 1. Example of a phishing website and a legitimate website. (A) A phishing website for a OneDrive login. (B) The real microsoft OneDrive login.

Second, we consider a different situation in which researchers use phishing kit analysis to support phishing identification, which we present in Section II-B. This last group includes approaches that directly use the phishing kit information to classify or group phishing attacks. These approaches are based on the idea that phishing attacks are an automatic product of phishing kits, and because of this, they are related or share certain patterns.

A. PHISHING KITS BEHAVIORAL STUDIES

Cova [21] analyzed phishing kits by tracking the destination of the stolen information. First, they gathered the phishing kits from distribution sites or downloaded them by checking the directory contents of already reported phishing websites. The authors collected around 500 phishing kits, and after the analysis process, they discovered that many samples contained backdoors that send the stolen data back to the phisher and the original author.

Oest et al. [24] studied the time response of anti-phishing groups' blocklists against evasion techniques using filters found in real phishing kits. The authors measured how cloaking techniques on phishing kits affect the timeliness of blocklisting phishing websites using sterilized phishing that contains different cloaking methods. The phishing websites were reported to anti-phishing groups and wait blocklisting time response. The dataset used in this experiment contained 2.380 spoofed PayPal login pages, and the authors reported that only 23% of cloaked websites were blocklisted against 49, 9% of websites without cloaking.

In a later work, Oest et al. [26] found that phishing kits are a key component of phishing attacks when they studied their life cycle. The authors monitored web events over the internet, processing the ones related to phishing websites. Finally, the authors reported that a phishing campaign takes 21 hours, and at least 7, 42% of the victims provide their credentials in that time window. The results presented in this work were extracted from a dataset with 19.359.676 events related to 404.628 different phishing URLs.

B. PHISHING KITS ANALYSIS AS A SUPPORT FOR PHISHING IDENTIFICATION

Britt et al. [27] proposed one of the first methods that use phishing kits as a resource of information to identify phishing attacks. The authors used MD5 values to represent the similarity between the two samples by counting the overlapped files inside them. Later, they created groups of phishing website samples by comparing the samples' similarity to a specific phishing kit. The clustering algorithm found 22.904 clusters, where 14.129 of those clusters contain phishing websites assigned to a brand, showing a highly consistent brand grouping. The University of Alabama at Birmingham (UAB) phishing Data Mine group collected the dataset used, which contains 265.611 potential phishing websites. Although this work does not use information about phishing kits, it is based on the idea that these phishing websites were deployed using phishing kits and therefore have similar patterns and characteristics.

To detect phishing website attacks, Orunsolu and Sodiya [23] presented an approach that uses phishing kit features. The method comprises a Signature Detection Module (SDM) that relies on 18 extracted features. These features are divided into HTML source, URL source, and phishing kit-related information. The phishing kit features include information such as hexadecimal obfuscation, toolkit names or URLs. Once the feature vector is extracted, the authors used it as an input to a Naive Bayes classifier reporting 98% accuracy on a dataset of 258 kits generated by websites.

To perform these experiments, Orunsolu and Sodiya [23] manually built the dataset, which involved two steps. First, ethical hackers and computer security students used five phishing kits to create 258 phishing websites that did not represent the conditions of a real attack. Then, in the second step, the authors collected 200 samples of phishing and legitimate websites on the internet between September and December 2014.

As a phishing identification technique, Tanaka et al. [25] used a website structure signature of phishing kits. This signature is created by analyzing the Web Access Log generated when users access a landing page. A sample is classified as a phishing website if it reaches a structural similarity score of 0.5 or higher using the Jaccard coefficient compared to the previously collected phishing kit structural scores. The dataset was built following two steps: First, the authors generated 49 phishing websites using phishing kits for the comparison base. Second, the authors collected 18.798 samples from July 2019 to March 2020 on PhishTank. They did not report any matching results, such as accuracy or F1-Score, since there is no way to relate the samples used for the comparison base with the samples collected in the second step. Instead, authors reported 1.742 phishing sites with similar structures to the comparison base, and after a manual revision, they determined that 95% of those samples were indeed similar.

Following a different strategy, Bijmans et al. [20] proposed a fingerprint representation based on file names, paths and strings in the phishing kits. They extracted the fingerprint from seventy Dutch phishing kits. After that, they collected phishing websites using a crawler, collecting information from about 500.000 websites and analyzing their fingerprint. The authors reported that the 70 phishing kits could be grouped into ten different families based on their similarities, and 89% of the samples that their algorithm actually identified were made from a phishing kit belonging to the uAdmin family. As in the previous work, there is no way to relate the phishing kits to the samples collected by the crawler. Instead, the authors reported the result after a manual revision and did not report metrics about the false negatives of the algorithm.

Feng et al. [28] used web structure analysis from HTML sources to identify phishing websites. The authors addressed this problem using a clustering technique since phishers use phishing kits to deploy many phishing attacks. For this reason, the attacks from the same phishing kit may contain similar web structures. The method consists of three steps. First, the extraction of a feature vector with the HTML Document Object Model (DOM) information. Second, the authors grouped the samples by similarity and generated a feature vector from all the samples belonging to a single group. Finally, the feature vector for each group is compared against the fingerprint of websites to obtain a binary classification.

To evaluate their method, they collected a dataset of 10.992 legitimate websites and 10.994 phishing websites. They concluded that this method could identify phishing website familiarity and detect phishing attacks more efficiently than other methods. However, they did not report any comparison results, such as accuracy or F1-Score, since their dataset does not contain a ground truth between phishing kits and phishing websites.

C. DATA COLLECTION METHODS

Phishing attacks are constantly evolving; due to this, data collection is an important issue that has been tackled in several works. Below we will introduce the main collection techniques found in the literature.

Distribution Sites offer a pool of phishing kits to be downloaded. It has been used in [21] and [25] to evaluate the proposed methods' performance. Distribution sites can be found on Internet Relay Chat (IRC) channels, underground communities, GitHub or web forums.

Telegram[1] is an app that allows encrypted messages, secure communications and even secret chats. Phishing kit programmers take advantage of these features to sell their phishing kits [20]. Some researchers followed suspicious channels on Telegram, looking for phishing kits and information to lead them to other channels to expand the search parameters. This approach, called snowball, allows them to collect hidden or hard-to-reach samples [29].

Checking phishing website directories can help collect phishing kits since it is a common mistake for phishers to forget the phishing kit file in the web server. Therefore it can be downloaded from the server using checking directory contents [20], [21]. This technique has an advantage over the others since phishing kits can be collected from phishing attacks, so it is possible to relate information about the attack to the phishing kit or source itself.

Server honeypots are servers that deceive phishers by pretending to provide vulnerable services. Previous studies have found that phishers use compromised domains to host phishing attacks [30]. Authors publish on the internet a sandbox programmed to look vulnerable and, at the same time, isolate the phishing kit keeping it functional to the phisher's perception. This strategy was used by Han et al. [31].

III. PhiKitA: PHISHING KIT ATTACKS DATASET

A. CONTEXT OF DATASET GENERATION

Phishing detection methods are complex to test due to the difficulty of obtaining representative datasets. This is related to the changing nature of phishing attacks and the sensitivity of the data itself. Authors usually collect the data by themselves, considering the requirements of their proposed method. Then, they present the performance of the algorithm but do not release the collected data. All these reasons make comparing the performance of the literature methods a complex task, as they could be tested under certain conditions introduced by the decisions made in the creation process of the dataset.

The problem outlined above also affected the creation of phishing kit datasets. Authors collect their data to evaluate methods using well-known phishing kit sources. Then, they use the phishing kit samples to create phishing website attacks [25]. Researchers make several decisions in the phishing website creation process, which could generate particular conditions in the dataset. It also affects the capability of the

[1] telegram.org/

dataset to represent the phishing attack in real conditions since the authors do not know the phishers' modus operandi.

Other authors have proposed collecting phishing kits, and phishing website samples online as an alternative to the early collection methodology [20], [23], [25]. This methodology consists of two stages: First, collect phishing kit samples and use this data to feed the algorithm or extract features. Then, collect phishing website samples on the internet and use them to evaluate the performance of their method. In this way, authors do not interfere with the data collection and get phishing website samples directly from the wild under more realistic conditions. However, it is not possible to determine a trusted correspondence between the phishing kit with the samples collected in the second stage. Therefore, there is no ground truth for samples of phishing websites deployed with a specific kit, and the results can not be presented using the common metrics for a classification problem.

In this paper, we propose a new methodology to collect samples that aims to fill the above-mentioned gaps in the literature. For that reason, we present a methodology for collecting datasets in which phishing websites are correctly linked to the phishing kits used to generate them. Having a correct association between phishing websites and their phishing kit source will allow researchers to evaluate and compare the performance of their algorithms for detecting phishing using the same dataset.

B. COLLECTION METHODOLOGY

Our proposed phishing collection methodology is divided into four stages: (i) source definition; (ii) phishing kit collection; (iii) phishing website collection, and (iv) post-processing of collected samples and final filtering, as shown in Figure 2.

1) SOURCE DEFINITION

Phishing attack sources are typically websites that allow users to report and expose new phishing URLs. Several groups are working on this topic, and we have selected four of the most popular ones to use as sources of information for our methodology: (i) PhishTank[2] is a website and free community operated by Cisco Talos Intelligence Group. (ii) OpenPhish[3] identifies phishing websites by collecting them from external resources such as blocklists. (iii) Phishing.Database[4] collect URLs from different sources, and then they are checked using testing software. Finally, (iv) PhishStats[5] offers a free list updated every 90 minutes.

As a source of legitimate samples, we used Quantcast Top Sites8[6] and The Majestic Million9[7] following the proposed by Sanchez-Paniagua et al. [32]. These websites rank other websites according to their referring subnets; since phishing websites have a short life cycle, they can not be in that ranking.

We created a script to generate newly reported phishing every hour. This script retrieves information from all the sources, joins all the reported URLs, and deletes the duplicated ones. It also deletes the ones already processed in an early run. Later, this list is sent to the phishing kit collector.

2) PHISHING KIT COLLECTOR

Kitphishr tool is a script developed by a cyber security expert.[8] Kitphishr receives a URL list or looks for one online, then iterate the list to check for zip files inside the downloadable resources of a website. Once the iteration is finished, all the suspicious zip files are saved into a folder. We also can find extra information, such as the URL where the tool found a phishing kit.

In this stage, we saved the phishing kit, the domain and the URL where the phishing kit was found. Figure 3 shows an example of this.

3) URL FILTER AND CRAWLER

Using the domains from the previous stage and the original list of reported phishing websites, we created a new list of reporting websites where a phishing kit was found. Then, we sent the new list to the phishing website crawler.

We developed a crawler following the process proposed by Sanchez-Paniagua [32]. The main idea is to collect enough information to evaluate any method on the dataset. We used Python3, the Wappalyzer tool,[9] Selenium and WGET requests. This crawler takes the list generated early and renders the websites on the Firefox web browser. Then, we downloaded the available information, such as the final URL, the HTML content and the technologies present on the target, using Wappalyzer. Finally, we also downloaded a local copy of the target using WGET.

4) SAMPLES POST-PROCESSING

The sample post-processing is divided into three steps: (i) we applied the phishing kit filter, and here all the phishing kits' content is checked, looking for website file structures. This filter aims to discard zip files containing other kinds of files, such as images or videos. Then, (ii) we used a second filter in the phishing website samples. Each sample is checked, to ensure that it contains all the files we intended to download. We also match the phishing websites samples against the phishing kit domain list obtained in Section III-B2. This filter is necessary due to the cloaking techniques present in some phishing attacks. These techniques trick the crawler, redirecting it to another website, which will be downloaded even if it is not a phishing website.

In the last step, (iii) the dataset is built by matching the phishing website samples with their respective source or phishing kit. We follow the three scenarios shown in Figure 4.

[2] https://phishtank.org/
[3] https://openphish.com/index.html
[4] https://github.com/mitchellkrogza/Phishing.Database
[5] https://phishstats.info/
[6] https://www.quantcast.com/products/measure-audience-insights/
[7] https://majestic.com/reports /majestic-million
[8] https://github.com/cybercdh/kitphishr
[9] https://www.wappalyzer.com/

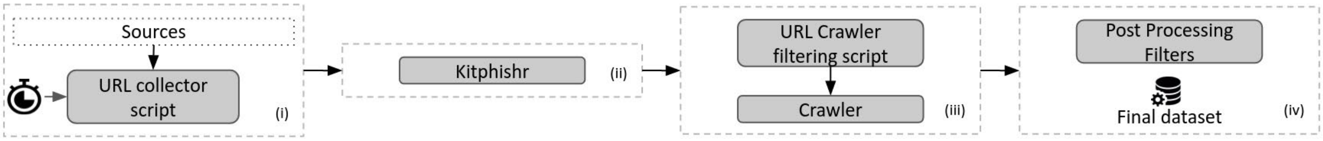
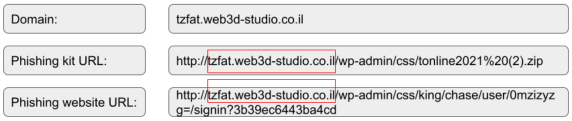


FIGURE 2. Pipeline of the methodology for collecting datasets: (i) URL collection, (ii) phishing kit collection, (iii) URL filter and crawler script, and (iv) post-processing filters.



Domain: tzfat.web3d-studio.co.il

Phishing kit URL: http://tzfat.web3d-studio.co.il/wp-admin/css/tonline2021%20(2).zip

Phishing website URL: http://tzfat.web3d-studio.co.il/wp-admin/css/king/chase/user/0mzizyzg=/signin?3b39ec6443ba4cd

FIGURE 3. URLs involved in a phishing attack: (i) phishing website Domain, (ii) URL where the phishing kit was found, and (iii) phishing websites landing URL.

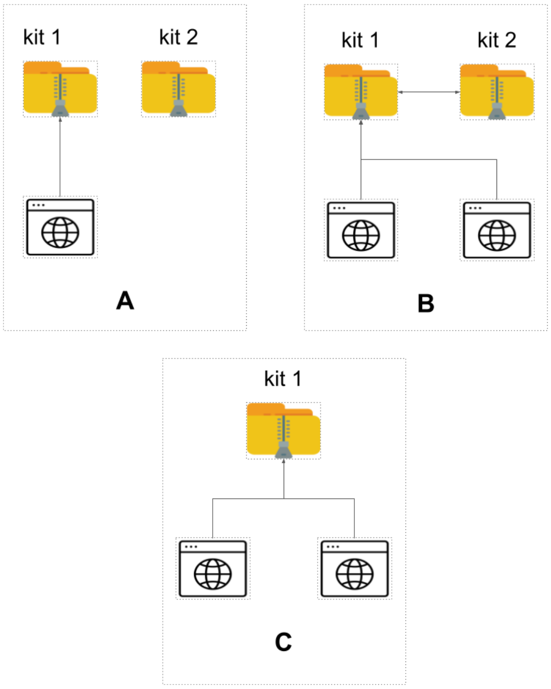


FIGURE 4. Phishing kit scenarios: (A) phishing kit samples with one or zero related websites, (B) duplicate phishing kits, (C) phishing kit samples with multiple phishing websites (designed using resources from Flaticon.com).

On the one hand, we found two simple options in the **scenario A**. First, a phishing kit sample has only one phishing website attack related. Second, a phishing kit does not have any phishing website attack related due to cloaking techniques that affect the crawling process.

On the other hand, there is only one option in the **scenario B**. We collected two or more phishing kit samples using two or more reported phishing website attacks. However, after a post-processing check, we found that the phishing kit samples are the same file. This scenario results in only one phishing kit with two different phishing website-related attacks. Those attacks could have distinct configurations and URLs.

Finally, in the **scenario C**, multiple phishing websites are reported under the same domain but with different URLs. As a response, we saved all the reported phishing websites and related all of them to the collected phishing kit.

C. DATASET INFORMATION

The sample distribution was built around the phishing kit samples. That is why phishing kits are at the highest folder level in phishing sample folders. A phishing kit sample containing information about the URLs from a sample was found, as a phishing kit zip file itself and a folder called 'deployed_webpages', which includes any phishing website attacks related to the sample.

The phishing website samples are active attacks at the moment of the collection. Thanks to that fact, we can collect a lot more information than with phishing kit samples. In the same way, legitimate samples are websites completely functional. Therefore, we collected the same information to compare phishing websites and legitimate samples using any phishing identification method. Both phishing websites and legitimate samples contain the following information:

- **html_content.txt** is the source code from the target website. It contains the HTML, JS, and CSS used to display the website in the browser.
- **info.json** collects information about the process, containing the URL of the phishing attack, the date when the data was retrieved and the class of the sample (phishing or legitimate).
- **tech.json** contains the report generated by Wappalyzer.[10] Wappalyzer identifies different technologies in a website base on certain fingerprints.
- **url.txt** contains the URL at the moment of the sample collection. This information could differ from the URL inside the info.json file because some phishing websites use URL redirections.
- **website_resources** is a local copy of the phishing website. It includes Images, CSS, JS, HTML, PHP files and all the resources necessary to recreate the website.

D. FINAL DATASET

We follow the methodology explained above to collect our dataset, called Phishing Kit Attacks or PhiKitA-500. The data was collected between 19th June and 8th August 2022.

[10] https://www.wappalyzer.com/

During this period, we collected 928 phishing kit samples and 3457 phishing website samples related to the domain of the phishing kits already collected.

Then we applied the three steps of the post-processing. In the first filter, we found 88 phishing kit samples that did not match the defined structure. Those samples were discarded since they contained file extensions such as .exe or .xml without any phishing kit structure. We checked the phishing website samples using the second filter, 2598 samples were discarded because they were incomplete. This is due to the cloaking techniques mentioned previously.

Finally, we applied the scenarios of the final step. We found that 330 were repeat samples. We join the samples using scenario B, leaving 510 unique phishing kits samples. Then, we found that 253 of the unique phishing kit samples do not contain any phishing website related due to the cloaking techniques.

We also collected legitimate samples to complete the dataset, allowing phishing website identification tests. We crawled 1141 from the above sources to obtain a balanced dataset. As a result of the collection, we obtain a dataset that contains 510 phishing kits, 859 phishing websites related and 1141 legitimate samples. PhiKitA-500 dataset is available in our website.[11]

IV. EXPERIMENTAL SETUP

A. ALGORITHMS IMPLEMENTATION

Few works have been presented in the literature to deal with phishing kits. In this paper, we use three of the five methods presented in Section II-B. We implemented the methods that have a detailed description and do not require a large training dataset or use third-party services. The methods implemented were: (i) MD5 to find shared files (Britt et al. [27]), (ii) fingerprint representation using path files (Bijmans et al. [20]). (iii) HTML DOM analysis (Feng et al. [28]). The implementation objective is to evaluate and compare the performance of the literature methods on the same dataset.

We did not consider the method proposed by Tanaka et al. [25] since it depends on third-party services. Third-party services can change over time and leave the method out-date. Therefore the method would require constant updates. We also discarded the method proposed by Orunsolu et al. [23] due to the need to train a machine-learning model. We concluded that a larger dataset is necessary to do this process properly.

B. METRICS

We used two metrics to compute the similarity of two samples in the implemented algorithms. On the one hand, we used the metric proposed by Britt et al. [27] (Equation 1) to measure the similarity in the MD5 and the HTML DOM. This is because both algorithms make comparisons between two sets of files.

$$Score = 0,5\,x\frac{|A \cap B|}{|A|} + 0,5\,x\frac{|A \cap B|}{|B|} \tag{1}$$

where A and B are sets, in this case, phishing kit samples or phishing attack samples. And $|A \cap B|$ is the number of files that are present in both sets at the same time.

On the other hand, we used the Jaccard similarity Coefficient on the fingerprint algorithm since we are comparing strings and we only have one fingerprint for each sample.

Current works on phishing kits do not present consistency in their evaluation methodology due to the dataset limitations, as we explained in Section II. Therefore, we used the phishing attack classification approaches in the state of the art as a reference to select the metrics used in the evaluation process for this work. For that reason, we used Accuracy, F1-Score, Precision, and Recall as other works in the classification problem have used [33], [34], [35].

C. EXPERIMENT DEFINITION

We proposed three experiments based on the available data of the dataset and the implemented algorithms: First, in *Experiment 1*, we tested the relationships between phishing kits samples. Then, the *Experiment 2* is a traditional phishing identification problem where an algorithm classifies phishing websites and legitimate ones. However, we classified the samples based on the information extracted from the phishing kits. Finally, we analyzed phishing kits and phishing websites in the *Experiment 3*. The idea is to use the ground truth of phishing websites and their phishing kit source to evaluate the performance of these algorithms clustering the samples according to their source.

1) EXPERIMENT 1: FAMILIARITY ANALYSIS

Obtaining information about the phishers, the phishing kit developers and how attacks evolve with their interaction is relevant for phishing attack identification. Phishing kit familiarity analysis can provide information about that interaction and how phishing attacks spread over the internet. We analyzed the phishing kits using the MD5 hashes algorithm, comparing all the files of two samples simultaneously. Once we obtained the number of the files found in both of them, we recognized them as familiarity related if they share above 75% of files following the approach proposed by Bijmans et al. [20].

2) EXPERIMENT 2: PHISHING DETECTION

We used phishing websites and legitimate websites for this experiment. The dataset contains 2.000 samples where 859 are phishing website attacks, and 1141 are legitimate. We also used the three implemented algorithms and fed them with the information extracted from the phishing kits. Then, we classified samples as phishing or legitimate according to their similarity with the phishing kit base information.

In this scenario, the experimental results represent the similarity of a sample with the phishing kit base information.

[11] htttps://gvis.unileon.es/dataset/phikita-500/

For this reason, it is necessary to set a threshold to determine if a sample belongs to the phishing class or the legitimate class.

We divided the dataset into two parts to set the thresholds of each algorithm. The 20% of the data was used to find the thresholds. Then, we used the remaining 80% of the data to evaluate the performance of each algorithm. We designed a grid search and divided it into intervals of 0.01, taking into account that the result of the algorithms is a float from 0 to 1, representing the similarity of the sample. Then we selected the threshold value where the algorithm achieved the best performance on the 20% of the data and evaluated it on the remaining samples to report the actual result.

3) EXPERIMENT 3: MULTI-CLASS CLASSIFICATION OF PHISHING KITS

It is essential to get insight into the phishing kits and how they help to deploy phishing websites massively, considering the increasing number of attacks in the second quarter of 2022. For this reason, we have designed this experiment. The idea here is to find the relationships between the phishing kit source and a phishing website to detect campaigns of attacks.

As we explained in Section II, to the best of our knowledge, it is the first time a multi-class classification approach has been tested on a dataset with ground truth. We defined two conditions while we designed this experiment: (i) phishing kit samples that do not contain related phishing websites were discarded in this test. (ii) phishing kit samples identified as family related in the first experiment were considered a one-only class. This way, we reduced the number of classes in the dataset and guaranteed that the classes were separate. After applying the above-mentioned conditions, we obtained an unbalanced dataset containing 257 phishing kits (classes) and 859 phishing websites. The sample distribution can be seen in Figure 5.

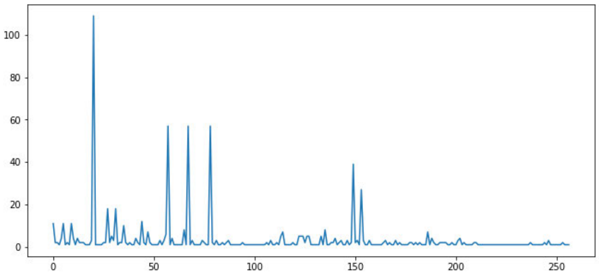
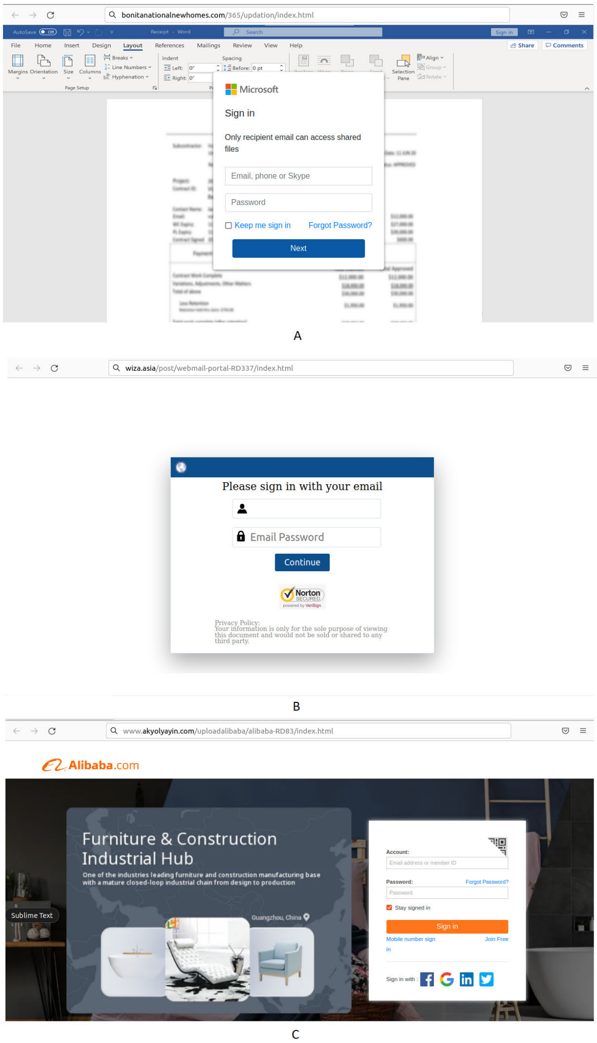


FIGURE 5. Distribution of the samples on the multi-class classification dataset.

V. RESULTS AND DISCUSSION

1) EXPERIMENT 1: FAMILIARITY ANALYSIS

In this experiment, we found 50 relationships in the dataset samples. These relationships consist of several familiarity groups of different sizes. One of the most significant groups consists of 37 samples, and we found that this group targets different companies, but those attacks were developed under the same phishing kit structure, as shown in Figure 6.



Only recipient email can access shared files

Please sign in with your email

Privacy Policy:
Your information is only for the sole purpose of viewing this document and would not be sold or shared to any third party.

One of the industries leading furniture and construction manufacturing base with a mature closed-loop industrial chain from design to production

FIGURE 6. Phishing websites that belong to one family. In this case, all examples use the same attack vector: an image background to imitate the legitimate website and a simple form to steal the victims' credentials.

Phishing kits in this group share the attack structure. After reviewing the phishing attacks related to those phishing kits, we found that they use a basic form with a background image. Furthermore, they share the same file distribution as can be seen in Figure 7 and also contain a similar pattern in their phishing kit name, which could relate them as a software product of the same author. This can be seen in Figure 7 highlighted with a red box.

Another significant group found in this analysis contains only seven phishing kit samples. However, the number of phishing attacks related to those kits represents the 20% of all the phishing websites of the dataset. In detail, 172 phishing website samples were related to this family. Unlike the previous family, this family targets only one company which is Standard Bank.[12]

[12] https://www.standardbank.co.za

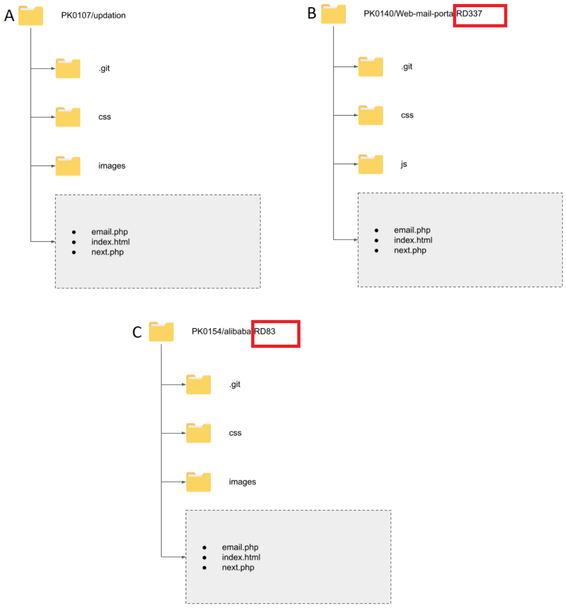


FIGURE 7. Phishing kit samples distribution in the dataset. A phishing kit ID was highlighted with a red box.

The seven phishing kits of this family target the same company. However, there are differences between them; they possess different functionalities, i.e., four of them have cloaking functionalities, and we could not collect phishing websites related to them. The number of phishing websites related to this family may indicate that it is a phishing campaign. Early detection of those kinds of campaigns is important for anti-phishing groups like CERTs to help companies and users deal with the threat of phishing attacks.

After analyzing the groups generated by the familiarity experiment, we found two kinds of phishing kit families: (i) Phishing kit families that share the same functionalities and file distribution, but their target is different. The same structure is used, but each phishing kit is slightly modified to attack a distinct target. (ii) Phishing kits families that only attack one target, the difference between phishing relatives to one of these kinds of families reside in the functionalities. This can be explained under two scenarios. First, phishers keep using the first version of the phishing kit, while other versions with more functionalities are available. Second, the phishing kit programmers sell those phishing kits differentiating the functionalities and changing their price accordingly.

2) EXPERIMENT 2: PHISHING DETECTION

Table 1 shows the best threshold and accuracy for the methods evaluated in Experiment 2. The results show that the Graph Representation and MD5 Hashes algorithms achieve the highest performance in binary classification. With a threshold of 0.46 and 0.18, the algorithms obtained an accuracy of 92.50% and 91.69%, respectively. The Fingerprint Representation algorithm obtains the lowest performance as shown in Table 1.

TABLE 1. Results of experiment 2. The threshold values were selected using the grid search.

Method	Threshold	Accuracy	Precision	Recall	F1-Score
MD5 Hashes	0.18	91.69	**95.06**	84.73	89.60
Fingerprint Representation	0.11	83.25	78.96	83.52	81.17
Graph Representation	0.46	**92.50**	91.83	**90.50**	**91.16**

The Graph Representation and MD5 Hashes algorithms achieved the highest performance in this experiment. These results are low compared to other state-of-the-art algorithms which do not use phishing kit information. An example of this is the method proposed by Sanchez-Paniagua et al. [32]. Although the results are lower than using the mentioned approach, the algorithms achieved a significant accuracy, which indicates that there is important information in phishing kits that actually helps in classifying phishing attacks. This data can be used as additional input in other phishing detection approaches to provide features that are otherwise not considered. And, what is more important, the detection of the phishing kit used allows for the detection of campaigns of phishing attacks from the same attackers, which could help in attributing the responsibility of the attacks to the corresponding phishers.

3) EXPERIMENT 3: MULTI-CLASS CLASSIFICATION OF PHISHING KITS

Table 2 shows the best performance for the third experiment. The MD5 Hashes algorithm achieved the highest accuracy and F1-Score with 34.92% and 39.54%, respectively. In contrast, the Fingerprint Representation algorithm achieved the lowest performance with 9.03% F1-Score. Although the MD5 hashes algorithm performed best in the experiment, its performance in multi-class classification is poor.

TABLE 2. Results of experiment 3. Results of the multi-class classification of phishing kits for each implemented algorithm.

Method	Accuracy	Precision	Recall	F1-Score
MD5 Hashes	**34.92**	**38.78**	**45.21**	**39.54**
Fingerprint Representation	7.57	9.11	11.52	9.03
Graph Representation	31.08	29.40	39.38	31.11

These algorithms were evaluated using binary classification tests or clustering approaches with no ground truth between phishing kits and phishing websites. For this reason, the results may not show the full range of performance of the algorithms. The algorithms achieved higher performance in phishing detection experiments, but their performance in multi-class classification is low. This suggests that the algorithms extract enough information to distinguish between legitimate websites and phishing websites. However, they do not extract enough information to distinguish phishing websites and their phishing kit sources.

VI. CONCLUSION AND FUTURE WORK

In this paper, we explore how phishing kit information can be used to support the identification of phishing websites.

For this purpose, we proposed a novel methodology to collect data where we have crawled phishing kits and phishing websites generated with the related kit. Following our methodology, we proposed and made publicly available PhiKitA, a dataset containing 510 phishing kits, 859 phishing website samples, and 141 legitimate websites. With PhiKitA, we release a ground truth where researchers can evaluate their proposals for phishing kit analysis, phishing binary classification and multi-class classification experiments.

In this paper, we evaluated three phishing classification and clustering algorithms from the literature and tested them in three experiments in PhiKitA. First, the *familiarity* experiment showed two phishing kit families: one with the same functionalities and file distribution but different targets, and the other with the same target but different functionalities, ranging from the simplest to the most complex sample. This may indicate that phishing kit programmers design their products to be attractive to phishers and price them according to the range of functions they offer.

After analyzing PhiKitA and its *familiarity*, we found a family containing seven phishing kits related to 172 different phishing attacks, whose target was Standard Bank. Phishing websites related to this family represent 20% of all phishing website samples in the dataset, which could indicate that PhiKitA might also contain a phishing campaign against Standard Bank.

Of the three algorithms evaluated for phishing detection, the Graph Representation algorithm achieved the highest performance, with an accuracy of 92.50% in PhiKitA. Although this performance is lower than other state-of-the-art approaches in phishing classification, the results show that the information obtained from phishing kits helps to determine whether a sample is phishing. This could be used as supporting information in other approaches.

Finally, in the last experiment, the *multi-class classification*, we found that the MD5 hash algorithm shows the best performance, with 39.54% of the F1-score. Due to the increasing number of phishing attacks and the use of phishing kits for their deployment, it is worth exploring how to cluster these deployments according to the source of the phishing kit. This clustering could be helpful for both binary classifications and identifying common targets of phishing attacks and phishing campaigns.

A. FUTURE WORK

PhiKitA was collected between 19th June 2022 and 8th August 2022. In our future work, we aim to extend PhiKitA by adding more samples in a larger collection period and adding additional data that could be interesting for other approaches, such as screenshots of the samples. We will also work on modifying the collection process to take into account the cloaking techniques detected in this work. Although we collect 510 of phishing kits, 253 of them do not contain phishing websites related due to cloaking techniques that block the crawler.

Finally, it may worth analyzing how to cluster these deployments according to the source of the phishing kit. Therefore, we will explore machine learning techniques that can be applied to the multi-class classification problem, such as authorship analysis between phishing kits and phishing website attacks.

REFERENCES

[1] I. T. Union. (2021). *Measuring Digital Development: Facts and Figures*. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2021.pdf

[2] R. M. A. Mohammad, "A lifelong spam emails classification model," *Appl. Comput. Informat.*, Jul. 2020. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1016/j.aci.2020.01.002/full/html

[3] F. Jáñez-Martino, E. Fidalgo, S. González-Martínez, and J. Velasco-Mata, "Classification of spam emails through hierarchical clustering and supervised learning," 2020, *arXiv:2005.08773*.

[4] J. Velasco-Mata, V. Gonzalez-Castro, E. F. Fernandez, and E. Alegre, "Efficient detection of botnet traffic by features selection and decision trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021.

[5] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for Internet of Things using looking-back-enabled machine learning techniques," *Comput. Electr. Eng.*, vol. 98, Mar. 2022, Art. no. 107716, doi: 10.1016/j.compeleceng.2022.107716.

[6] E. Fidalgo, E. Alegre, L. Fernández-Robles, and V. González-Castro, "Classifying suspicious content in Tor darknet through semantic attention keypoint filtering," *Digit. Invest.*, vol. 30, pp. 12–22, Sep. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287619300027

[7] P. Blanco-Medina, E. Fidalgo, E. Alegre, and F. Janez-Martino, "Improving text recognition in Tor darknet with rectification and super-resolution techniques," in *Proc. 9th Int. Conf. Imag. Crime Detection Prevention (ICDP)*, 2019, pp. 32–37.

[8] E. Figueras-Martín, R. Magán-Carrión, and J. Boubeta-Puig, "Drawing the web structure and content analysis beyond the tor darknet: Freenet as a case of study," *J. Inf. Secur. Appl.*, vol. 68, Aug. 2022, Art. no. 103229, doi: 10.1016/j.jisa.2022.103229.

[9] C. A. Murty, H. Rana, R. Verma, R. Pathak, and P. H. Rughani, "Building an AI/ML based classification framework for dark web text data," in *Proc. Int. Conf. Comput. Commun. Netw.* Cham, Switzerland: Springer, 2022, pp. 93–111.

[10] D. Chaves, E. Fidalgo, E. Alegre, R. Alaiz-Rodríguez, F. Jáñez-Martino, and G. Azzopardi, "Assessment and estimation of face detection performance based on deep learning for forensic applications," *Sensors*, vol. 20, no. 16, p. 4491, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/16/4491

[11] L. Zhu, Q. Zhang, and W. Wang, "Residual attention dual autoencoder for anomaly detection and localization in cigarette packaging," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 475–480.

[12] S. Minocha and B. Singh, "A novel phishing detection system using binary modified equilibrium optimizer for feature selection," *Comput. Electr. Eng.*, vol. 98, Mar. 2022, Art. no. 107689, doi: 10.1016/j.compeleceng.2022.107689.

[13] E. Zhu, Z. Chen, J. Cui, and H. Zhong, "MOE/RF: A novel phishing detection model based on revised multi-objective evolution optimization algorithm and random forest," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4461–4478, Dec. 2022.

[14] Anti-Phishing Working Group. (2022). *Phishing Activity Trends Report 2 Quarter*. [Online]. Available: https://apwg.org/trendsreports

[15] M. Hijji and G. Alam, "A multivocal literature review on growing social engineering based cyber-attacks/threats during the COVID-19 pandemic: Challenges and prospective solutions," *IEEE Access*, vol. 9, pp. 7152–7169, 2021.

[16] A. Alzahrani, "Coronavirus social engineering attacks: Issues and recommendations," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 5, pp. 154–161, 2020.

[17] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and critical analysis of phishing websites methods," *Comput. Sci. Rev.*, vol. 17, pp. 1–24, Aug. 2015, doi: 10.1016/j.cosrev.2015.04.001.

[18] Q. Cui, G.-V. Jourdan, G. V. Bochmann, and I.-V. Onut, "Proactive detection of phishing kit traffic," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Cham, Switzerland: Springer, 2021, pp. 257–286.

[19] P. Zhang, A. Oest, H. Cho, Z. Sun, R. Johnson, B. Wardman, S. Sarker, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupe, and G.-J. Ahn, "CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 1109–1124.

[20] H. Bijmans, T. Booij, A. Schwedersky, A. Nedgabat, and R. van Wegberg, "Catching phishers by their bait: Investigating the Dutch phishing landscape through phishing kit detection," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 3757–3774.

[21] M. Cova, C. Kruegel, and G. Vigna, "There is no free phish: An analysis of 'free' and live phishing kits," in *Proc. 2nd USENIX Workshop Offensive Technol. (WOOT)*, 2008, pp. 1–8.

[22] F. Castaño, E. Fidalgo-Fernández, and F. Jañez-Martino, *Creation of a Phishing Kit Dataset for Phishing Websites Identification*. León, Spain: TFM, Univ. León, 2022.

[23] A. A. Orunsolu and A. S. Sodiya, "An anti-phishing kit scheme for secure web transactions," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, Jan. 2017, pp. 15–24.

[24] A. Oest, Y. Safaei, A. Doupe, G. J. Ahn, B. Wardman, and K. Tyers, "PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *Proc. IEEE Symp. Secur. Privacy*, May 2019, pp. 1344–1361.

[25] S. Tanaka, T. Matsunaka, A. Yamada, and A. Kubota, "Phishing site detection using similarity of website structure," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Jan. 2021, pp. 1–8.

[26] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G. J. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 361–377.

[27] J. Britt, B. Wardman, A. Sprague, and G. Warner, "Clustering potential phishing websites using DEEPMD5," in *Proc. 5th USENIX Workshop Large-Scale Exploits Emergent Threats, (LEET)*, 2012, pp. 1–8.

[28] J. Feng, Y. Qiao, O. Ye, and Y. Zhang, "Detecting phishing webpages via homology analysis of webpage structure," *PeerJ Comput. Sci.*, vol. 8, p. e868, Feb. 2022.

[29] R. Atkinson and J. Flint, "Accessing hidden and hard-to-reach populations: Snowball research strategies," *Social Res. Update*, vol. 33, no. 1, pp. 1–4, 2001.

[30] D. Canali and D. Balzarotti, "Behind the scenes of online attacks: An analysis of exploitation behaviors on the web," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2013, pp. 1–18.

[31] X. Han, N. Kheir, and D. Balzarotti, "PhishEye: Live monitoring of sandboxed phishing kits," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1402–1413.

[32] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-Rodríguez, "Phishing websites detection using a novel multipurpose dataset and Web technologies features," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 118010, doi: 10.1016/j.eswa.2022.118010.

[33] M. Sanchez-Paniagua, E. F. Fernandez, E. Alegre, W. Al-Nabki, and V. Gonzalez-Castro, "Phishing URL detection: A real-case scenario through login URLs," *IEEE Access*, vol. 10, pp. 42949–42960, 2022.

[34] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," *Neural Comput. Appl.*, vol. 35, no. 7, pp. 4957–4973, Mar. 2023, doi: 10.1007/s00521-021-06401-z.

[35] U. Ozker and O. K. Sahingoz, "Content based phishing detection with machine learning," in *Proc. Int. Conf. Electr. Eng. (ICEE)*, Sep. 2020, pp. 1–6.

FELIPE CASTAÑO received the B.Sc. degree in computer science from Universidad del Valle, Colombia, in 2019, and the M.Sc. degree in cybersecurity research from the University of León, in 2022. He is currently a Researcher with the Group for Vision and Intelligent Systems (GVIS), University of León. His research interests include anti-phishing solutions, natural language processing, computer vision, and machine learning.

EDUARDO FIDALGO FERNAÑDEZ received the M.Sc. degree in industrial engineering and the Ph.D. degree from the University of León, in 2008 and 2015, respectively. He is currently a Coordinator of the Group for Vision and Intelligent Systems (GVIS), whose objective is researching and developing solutions to cybersecurity-related problems for INCIBE (https://www.incibe.es/en) by using artificial intelligence. His current research interests include natural language processing, computer vision, machine learning, and deep learning.

ROCÍO ALAIZ-RODRÍGUEZ received the B.Sc. degree in electrical engineering from the University of Valladolid, Spain, in 1999, and the Ph.D. degree from the Carlos III University of Madrid, Spain, in 2005. She is currently a Full Professor with the University of León, Spain. Her research interests include machine learning, neural networks, quantification, and dataset shift problem.

ENRIQUE ALEGRE received the M.Sc. degree in electrical engineering from the University of Cantabria, in 1994, and the Ph.D. degree from the University of León, Spain, in 2000. He is currently the Head of the Research Group for Vision and Intelligent Systems (GVIS) and a Full Professor with the Department of Electrical, Systems and Automation Engineering, University of León. His research interests include computer vision, machine learning in general, deep learning, and natural language processing, specially oriented to cybersecurity and crime control and prevention problems.