Supporting Law-Enforcement to Cope with Blacklisted Websites: Framework and Case Study

Mir Mehedi Ahsan Pritom* and Shouhuai Xu†
*Department of Computer Science, University of Texas at San Antonio
†Department of Computer Science, University of Colorado Colorado Springs

***Abstract*—Cyber attackers have long abused web domains and URLs to carry out various attacks such as Phishing, web scamming, and malware attacks. In order to defend against these attacks, URL blacklisting has been widely used. However, this approach has significant weaknesses, especially from a law-enforcement point of view. In particular, the law-enforcement does not know what to do with a blacklist because it is unclear what needs to be done (e.g., shutting down a host or domain) due to the subtleties associated with the problem. In order to help the law-enforcement in dealing with blacklisted URLs, we propose a novel framework based on Machine Learning (ML) while providing the law-enforcement with probabilistic classification and interpretability of the predictions made by the interpretable model. Our probabilistic classification and interpretability measures provide a basis for law-enforcement trustworthy decision-making and remove the black-box nature of traditional ML-based approaches. Experimental results show that the framework is practical and has further potential to tackle website maliciousness.**

I. Introduction

Websites have been widely abused as a medium for propagating cyberattacks [1]–[3]. One simple defense against these threats is to use URL and domain blacklists, which are client-side interventions and often provided by third-party vendors (e.g., Phishtank, Google Safe Browsing, URLhaus). However, this does not completely eliminate the threat because some users may not use such services and the malicious or compromised domains or hosts are still on the loose. Moreover, these blacklists are far from perfect [4] because they are neither *complete*, meaning that they do not contain all of the malicious websites [5]–[7], nor *accurate*, meaning that they contain many false-positive websites (including the compromise ones that were malicious in the past but have already been cleaned up) [2]. Another defense is to use Machine Learning (ML) models to proactively detect malicious websites (see, e.g., [1], [2], [8]). Moreover, there are also some third-party vendors (e.g., Netcraft [9]) providing takedown services on user requests for protecting against cybercrimes (e.g., cybersquatting). of abusing domains that are imitating a user's brand to provide user protection against cybercrimes.

However, there is one important perspective that has not been investigated in the literature, namely *law-enforcement*. We envision that law-enforcement will be, if not already, authorized to take actions against malicious websites much alike they do in case of botnets [10]. This introduces a new dimension of the problem because the law-enforcement must treat detected malicious websites carefully. For example, the law-enforcement can be authorized to shut down a malicious website owned or operated by a malicious party, but may only be authorized to notify the owner or the operator of a website which itself is compromised and then abused by an attacker to wage further attacks. Moreover, oftentimes we observe that attackers reuse the same domains and hosts for new URL based attacks causing the same domain or hostname to appear in a URL blacklist [11]. This phenomena further encourages us to consider the law-enforcement perspective, as more higher level such as domain or host level intervention is more effective than client-side interventions. This call for studies on helping the law enforcement in distinguishing between malicious (i.e., attacker-owned) and compromised (i.e., legitimate party-owned) websites to take actions.

Our Contributions. In this paper, we make three contributions. First, we initiate the study of the law-enforcement perspective when coping with malicious websites. This turns out to be a challenge because of the dynamic nature of web domains and complexity of web hosting infrastructures. This prompts us to introduce a novel framework to help the law-enforcement to cope with malicious websites. The framework highlights the importance of using *interpretable* (i.e., explainable) ML, while considering the probabilistic *uncertainty* associated with the prediction outcomes. The framework integrates a ML intepretability system, such as InterpretML [12], to provide explanations and probabilistic predictions to the law-enforcement (e.g., why is a website predicted as malicious, and what is the likelihood it is indeed malicious?).

Second, we investigate how to choose the entity for taking action: domain vs. hostname. To our knowledge, this is the first time to propose a principle method for making such decisions.

Third, we conduct a case study on evaluating an instance of the framework with a real-world URL blacklist. Experimental results show that we achieve a 86% accuracy with a 0.92 F-1 score, while providing local explainability (i.e., interpretation) for the individual prediction outcomes for each input blacklisted website.

Paper Outline. Section II presents the problem statement. Section III describes our framework. Section IV presents our case study and results. Section V discusses limitations of this study. Section VI discusses related prior studies. Section VII concludes the paper.

2022 IEEE Conference on Communications and Network Security (CNS) | 978-1-6654-6255-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/CNS56114.2022.9947260

II. Problem Statement

Suppose the law-enforcement is authorized to take actions against malicious websites. The problem is: *Given a URL that is blacklisted (i.e., deemed malicious), what should the law-enforcement do?* While intuitive, there are technical subtleties.

A. Technical Subtleties Encountered by Law-Enforcement

Subtlety 1: URL structure is complicated. Figure 1 illustrates the URL structure, including: a *protocol* name (`https` in this example), a *hostname* (`mail.example.com`), and a *URL path* (`mail/u/0`) possibly along with some queries within the URL path. A hostname consists of a *domain name* (`example.com`, also refereed to as a 2LD) and possibly a *subdomain name* (`mail`, referred to as a 3LD). A hostname is sometimes referred to as a Fully Qualified Domain Name (FQDN) [13] and mapped to one or multiple IP addresses by the DNS server, while noting that one IP address may be mapped to multiple hostnames (i.e., shared hosting [14]). A domain name must contain a *Top-Level Domain* (TLD) (e.g., `.com`) within it. A higher level subdomain, say Fourth-Level-Domain (4LD) (e.g., `z.x.example.com`), could be resolved to the same IP address as 3LD `x.example.com` or 2LD `example.com`. In this paper, we refer to the 3LD or higher level (if present) of a URL as *hostname* and the 2LD as *domain name*, while noting that these two become the same in the absence of a subdomain name within the website URL in question.

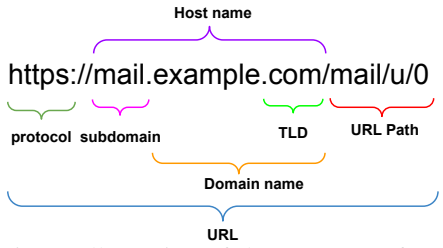

Fig. 1: Illustration of the structure of URLs

The complexity of URL structure makes it unclear what the law-enforcement should do to a malicious website or URL. To see this, consider a URL with a path name. In this case,

- shutting down the specific URL (including the path name), for example by filtering web traffic corresponding to the URL, may not be effective because the attacker can easily create other URL paths with the same hostname;
- shutting down the corresponding port is no good idea because the host (i.e., web server) corresponding to the URL may be malicious (i.e., the host can continue to wage attacks via other ports);
- shutting down the entire domain (e.g., `example.com` in this case) or the entire hostname (e.g., *mail.example.com*) corresponding to the URL without considering its ownership is no good idea because there might be many benign subdomains and URLs blueassociated with the domain or the hostname, which will be affected and deemed as false positives.

To further illustrate the problem, let us look at the structure of a website using web-hosting vs. domain-hosting as shown in Figure 2. In this example, the host `sites.example.com` is not malicious and should not be shutdown even though some URL(s) with the hostname are malicious. In the case of domain-hosting shown in Figure 2(B), multiple hostnames are created under the benign domain `example.com`. If one hostname, say `site1.example.com`, is created by an attacker to publish malicious contents by abusing the hosting service, then `site1.example.com` should be shut down but the other subdomains. However, if domain `example.com` is not associated with any hosting service, then we can safely assume that `example.com` is owned and/or operated by an attacker and therefore should be shut down.

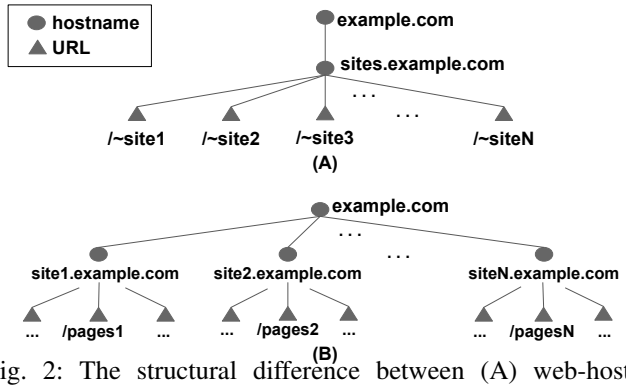

Fig. 2: The structural difference between (A) web-hosting and (B) domain-hosting within an example domain named `example.com` (adapted from [15])

Subtlety 2: Trustworthiness of given malicious websites. Blacklists may contain false-positives [2], meaning that the law-enforcement cannot blindly trust or shut down blacklisted websites. Instead, the law-enforcement must leverage other means to examine whether a blacklisted entity (i.e., host or domain) is indeed malicious before taking any actions.

Subtlety 3: IP Address-based blacklisting is no good idea. Nowadays sharing IP addresses and hosting is very popular. With shared IP addresses, many domains may resolve to the same IP address. If one of the hostnames is malicious, it does not mean the other hostnames that resolve to the same IP address are malicious. This makes it harder to use IP address-based blocking [14], [16]. In order to further highlight the ambiguity, we randomly pick a hostname `mail[.]weddingstaffcompanies[.]com`[1] from the publicly available PhishTank blacklist; the hostname is labeled as *suspicious* by McAffe PC security when accessed from Google Chrome browser. The hostname is resolved to IP address is 207.38.88.153. Then, we do reverse DNS lookup to get a FQDN `usloft5543[.]serverprofi24[.]com`, which is different from the input hostname. In order to see what other domains or hostnames are mapped to IP address 207.38.88.153, we query Robtex.com and find that at least 44 domains or hostnames are associated with it. However, we do not find any other domains or hostnames associated with this IP address in the blacklist. In this case, if the law-enforcement

[1][.] is used to safeguard reader from clicking possibly malicious website

blocks IP address 207.38.88.153, then the other 43 hostnames will be affected, which is not justified. Therefore, IP address-based blocking is no feasible solution.

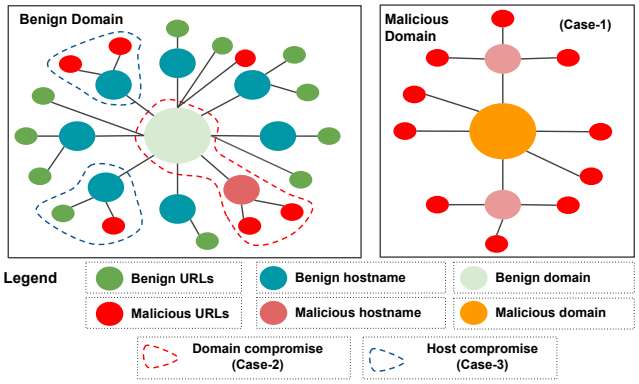

Fig. 3: Structure of websites including domain, hostname, and URL(s)

Subtlety 4: Complications encountered when taking actions against malicious domains or hostnames. Figure 3 highlights the mapping of hostnames and URLs in a domain name. It shows multiple cases: (case-1) The law-enforcement encounters a malicious domain and associated malicious hostnames and URL paths. The law-enforcement can justifiably shut down the domain. (case-2) The law-enforcement encounters a benign domain that has been compromised to create malicious hostname. The law-enforcement needs shut-down the malicious hostname and notify the legitimate domain owners to let them clean up the compromises. (case-3) The law-enforcement encounters a benign hostname associated with a benign domain which is compromised and abused to create malicious URLs. The law-enforcement should notify the hostname owner to clean up and put the URL in blocklist without shutting down the domain or hostname. In summary, it is essential to support the law-enforcement with various kinds of details.

B. Research Questions (RQs)

The preceding subtleties prompts to revise the research problem as as follows: This leads to the following research questions (RQs) regarding a blacklisted URL:

- RQ1 (*URL characterization*): At what entity level(s), such as domain and/or host, should the law-enforcement take actions?
- RQ2 (*quantitative classification*): What is the likelihood that the entity (e.g., domain or hostname) corresponding to a blacklisted URL is malicious or victim (i.e., compromised and abused to wage attacks)?
- RQ3 (*prediction interpretability*): Why an entity associated with a blacklisted URL is predicted as malicious or compromised?
- RQ4 (*law-enforcement actions*): What should the law-enforcement do when the answers to RQ2 and RQ3 are not satisfactory or convincing?

III. The Framework

To address the RQs mentioned above, we propose a framework, which is highlighted in Figure 4. The framework has the following modules: *characterizing*, *labeling*, *feature analysis & extraction*, *training interpretable ML models*, *probabilistic classification*, and *decision-making*. At a high level, these modules work together to address the aforementioned RQ1-RQ4 as follows. To address RQ1, we propose extracting the hostname and domain namefrom a given blacklisted URL, while finding out if the domain or hostname is associated with any known hosting service. In practice, one of the two following scenarios happen often: (i) the law-enforcement is often given blacklisted URLs without being told why and how the URLs are blacklisted; (ii) the law-enforcement is told how the URLs are blacklisted but without being given any explanation on why they are deemed malicious and/or the confidence that the URLs are indeed malicious. This prompts us to propose that the law-enforcement should build their own systems to analyze blacklisted URLs to address RQ2-RQ4. In particular, addressing RQ2 requires to quantifying the probabilistic uncertainty associated with ML models and addressing RQ3 requires to using interpretable ML models. Lastly, addressing RQ4 helps the law-enforcement in dealing with truly malicious or attacker-owned entities.

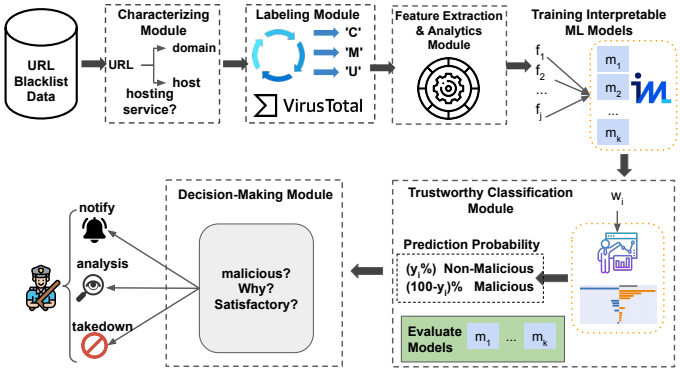

Fig. 4: The Framework with six modules.

Notations. A URL blacklist $\mathcal{L}$ contains n URLs, denoted by $\mathcal{L} = \{u_1, \ldots, u_n\}$. For URL $u_i \in \mathcal{L}$, we denote the associated hostname by h_i and domain name by d_i, a corresponding entity for i-th URL is denoted as en_i. Table I summarizes the main notations used in the paper.

A. Characterizing Module

This module characterizes the URLs on a blacklist to select the appropriate entity en_i based on their domain structure. It should first extract the hostname h_i and the domain name d_i from the input blacklisted URL u_i. Then, we check if the hostname h_i (3LD or higher level subdomain) is associated with any known hosting services, if yes then we notify the hostname for cleaning up and no quantification is necessary; otherwise, we check if the domain name d_i is associated with any known hosting services, if not then the selected entity $en_i = d_i$; otherwise then we further check if $d_i = h_i$ (meaning

TABLE I: Summary of notations used in the paper

Notation	Meaning
$u_i, \mathcal{L}$	i-th URL on blacklist $\mathcal{L}$
d_i, h_i, en_i	Domain, host, and entity for the URL u_i, respectively
$\mathsf{CL} = \{\mathrm{C}, \mathrm{M}\}$	Class label compromised (C) vs. malicious (M)
y_i	Predicted probability for maliciousness of entity en_i
$\mathcal{M}$	Any supervised machine learning model
F_i	Feature vector for i-th entity
E_i	Explanation set for i-th entity
ϕ_j^i	Impact of the j-th feature on classification of i-th entity
$\phi_{j,k}^i$	Impact of the j-th and k-th features together on classification of i-th entity
D_{entity}	Total unique entities (websites) without association to any public/private hosting services
$D_{labeled}$	Labeled ground-truth entities

hostname same as domain or no subdomain in URL), if yes, then we notify the domain and no quantification is required (because domain is legitimate, URL is created with malicious path); otherwise, when $h_i \neq d_i$, we select entity $en_i = h_i$ for quantification. The flow chart is presented in Figure 5. The URLs that are characterized as to quantify hostname or quantify domain names are the ones compiled as the $D_{unlabeled}$, and going as an input to the next module.

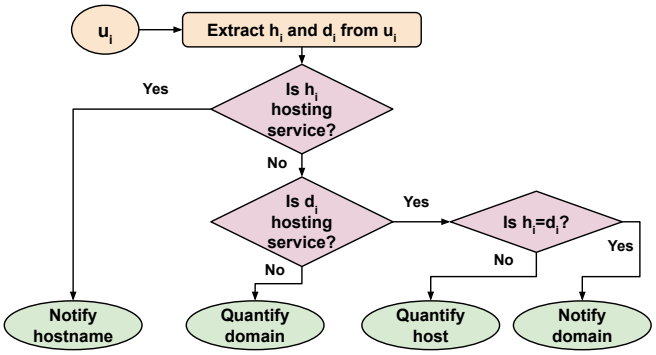

Fig. 5: Flowchart for Characterizing Module.

B. Labeling Module

This module produces a labeled dataset for training ML models, ideally automated. This module takes a hostname or domain name as input, depending on the entity chosen to quantify by the characterizing module. It labels the entity via a third-party threat engine such as the VirusTotal [17]. There may be four possible labels: malicious (M); compromised (C) meaning that the entity is itself a victim, namely compromised and then abused to wage attacks; unknown (U); and not available (N). It may be necessary to preprocess the input from such third-party services, and the preprocessing method may be specific to the third-party services. In any case, we propose only considering the malicious (M) and the compromised (C) labels for analyzing the law-enforcement perspective.

C. Feature Extraction & Analytics Module

1) Feature Extraction: This module extracts feature representations of URLs, including but are not limited to what have been extracted by the *characterizing* module. Denote the resulting feature representation of URL u_i by $F_i = \{f_1^i \dots f_k^i\}$, where k is the total number of features. We propose using the following features:

1) Brand-name in hostname and domain name ($f_1 \in \{0, 1\}$): This feature indicates whether a hostname and domain name string contains popular brand names according to some list(s) of reputable domains, such as Alexa [18] or Tranco [19]. Any domain or hostname involved in the a blacklist should have $f_1 = 0$.
2) Twisted brand-name in hostname ($f_2 \in \{0, 1\}$): This feature indicates if the domain name or hostname contains any twisted string of the top 5k brand names, such as typo-squatting, combo squatting, or homographing. For example, `amaz0n-pay.example.com` contains a twisted version of a top-brand name *amazon* and therefore has been assigned value 1, otherwise 0.
3) Number of dots in a hostname (f_3): This is the number of dot characters (i.e., '.') in a hostname. For example, hostname `ab.c-d.df.com` has 3 dots.
4) Number of hyphens in hostname (f_4): This is the number of hyphen characters (i.e., '-') in the hostname. For example, hostname `ab.c-d.df.com` has 1 hyphen.
5) Digit ratio in hostname (f_5): This is the ratio of the number of digits in a hostname to the length of a hostname. For example, hostname `a12.c34-d1.df.com` has 5 digits, meaning a digit ratio of $5/17$.
6) Number of unique alphabetic-numeric characters in hostname (f_6): This is the number of unique alphabetic characters or digits in a hostname except the TLD part, which is excluded because it often consists of letters. For example, hostname `ab12.c34-d1.df.com` has 9 unique alphabetic-numeric characters (i.e., `a, b, c, d, f, 1, 2, 3, 4`) other than the TLD (`.com`).
7) Hostname length (f_7): This is the length of a hostname.
8) Number of tokens in hostname (f_8): This is the number of tokens in a hostname after tokenizing with hyphen '-' and dot '.', except the TLD part which is excluded because it is typically one token. For example, hostname `ab12.c34-d1.df.com` has 4 different tokens (i.e., `ab12`, `c34`, `d1`, and `df`). This feature highlight the hostnames that contain many '-' and/or '.' characters.
9) Length of the longest token (f_9): This is the length of the longest token in a hostname. For example, the longest token `ab12.c34-d1.df.com` is 4 (i.e., `ab12`).
10) Number of redirects (f_{10}): Attackers often use redirects to deceive victims. This feature reports the number of redirects associated with a hostname.
11) Number of passive DNS queries (f_{11}-f_{12}): These features describe the number of records found for individual DNS record types, such as DNS 'A' and 'NS' records in the global passive DNS database from CIR.CL [20], respectively. For a passive DNS query, an 'A' record and a 'NS' record indicates the corresponding IP addresses and name server. These features report the record counts for each record type. A high number in these records possibly

indicate the website has more historical presence in the Internet, thus deemed more reputable.

12) Presence of self-resolving name servers ($f_{13} \in \{0, 1\}$): It indicates if the associated domain has any self-resolving name server or not. For example, if domain `'example.com'` has name server `ns1.example.com`, the domain is self-resolving and this feature value is 1; otherwise, its value is 0.
13) Domain ranking (f_{14}):This numeric feature measures the reputability of a domain name with respect to some list of reputable websites (e.g., Tranco [19]). In our case study we will present an example.
14) Hostname ranking (f_{15}): This numeric feature measures the reputability of a hostname with respect to some list of reputable hosts (e.g., Tranco [19]). It can be assigned in the same fashion as f_{14}. In our case study we will present an example.
15) Number of subdomains (f_{16}): This is the number of subdomains associated with a domain name corresponding to a URL. For example, given domain name $d =$"`wixsite.com`", one can query all active subdomains of the form of sub`.wixsite.com` and then count the number of such subdomains.

After extracting these features, it is worth mentioning that a preprocess may be needed to eliminate the correlated features (if applicable) before training ML models, because it is well-known that highly correlated features affect model predictions. Removing unnecessary features may also improve ML models' performance. Moreover, other blacklist dataset specific features can be added to complement the existing generic features for extending the framework.

D. Training Interpretable ML Model Module

This module trains for a ML model $\mathcal{M}_x$ to predict any given URL for law-enforcement purposes. Here, x denotes the corresponding interpretable ML model. It takes feature vectors as input to train a ML classifier with uncertainty quantification through probability, while providing interpretability of predictions. We reiterate that one should use the ML methods that (i) are interpretable, so that the law-enforcement can understand why a URL is deemed malicious, and (ii) can quantify the confidence or uncertainty associated with a prediction. Both are important for justifying law-enforcement actions against a blakclisted website.

E. Probabilistic Classification with Interpretation Module

A trained ML model makes probabilistic predictions on URLs, while interpreting its predictions. The probabilistic classification can be evaluated using the standard metrics, such as accuracy, AUC score, precision, recall, and F-1 scores [21]. For a given feature vector representing an entity, the law-enforcement uses the classifier to predict the probability that the corresponding entity (i.e., host or domain) as malicious (M) or compromised (C). Moreover, there will be a explanation set $E_i = \{\phi^i_j\}_j \cup \{\phi^i_{j,k}\}_{(j,k)}$ corresponding to the i-th entity. Both the probabilistic predictions and interpretations will be leveraged by the *Decision-Making* module.

F. Decision-Making Module

This module leverages the output of the probabilistic classification and interpretation module to help the law-enforcement make decisions. At a high level, if an entity associated with a URL, is predicted as malicious (M) with a high probability and a satisfactory interpretation, the law-enforcement should shut down the entity. If it is predicted as compromise (C) with high probability and a satisfactory interpretation, then the law-enforcement should notify the corresponding host or domain owners to clean up. Otherwise, if the probability is not high (e.g., $< 70\%$) or the interpretation is not satisfactory enough then the law enforcement should send the URL to human analysts for further analysis.

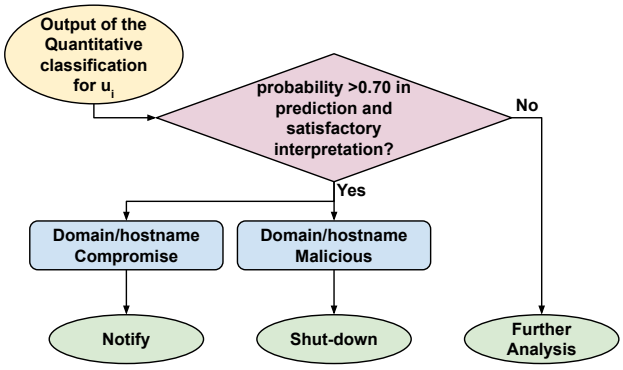

Fig. 6: Flowchart of the decision-making Module.

Figure 6 highlights the flowchart, which can be understood via the following example. If the length of an entity's name string is the main contributor to the maliciousness prediction and the length is less than 10 characters, then it is not conclusive to shut down the entity. If the number of unique alphanumeric characters contained in the entity's name string is an indicator and this number is large, while there are other indicators (e.g., a large digit ratio, a large number of tokens in the entity's name, a large number of redirects, or a large number of passive DNS queries), then it becomes satisfactory that the website is malicious and should be shutdown.

IV. Case Study

Now we present a case study on instantiating the framework. We use a publicly available blacklist, *PhishTank*, because it is open-sourced and widely-used in literature. Even though PhishTank adds new URLs on a regular basis and a community verifies the URLs, the verification does not occur instantly. As a consequence, PhishTank may still past compromised websites that have already been cleaned up. However, PhishTank does not provide any categorization for either malicious or compromised websites, thus making it suitable for the proposed *law-enforcement* perspective. We collect the verified blacklisted URLs for 5/4/2021-5/10/2021, leading to a total of $|\mathcal{L}| = 12,843$ URLs as input to the *characterizing* module.

A. Characterizing Module

This module answers RQ1 by characterizing the URLs to determine the appropriate entities on which the law-enforcement should act upon based on the associated hosting services. In this study, we curate a list of 61 publicly known reputable hosting services. We extract hostnames and domain names from the 12,863 URLs, leading to a total of 7,482 unique hostnames. This means that many URLs have the same hostname but different paths. Among the 7,482 hostnames, there are 8 hostnames that are directly related to some known hosting services and they are notified of the URLs if encountered. For the rest 7,474 (= 7,482-8) hostnames, 2,707 are only domain name (i.e., meaning $h_i = d_i$); among the rest 4,767 hostnames (where $h_i \neq d_i$), 1,590 have their domains associated with hosting services, meaning that the law-enforcement should select entity $en_i = h_i$ to take actions; for the rest 3,177 (=4,767-1,590) hostnames, their domains are not related to any hosting services, meaning that the law-enforcement should select entity $en_i = d_i$ to take actions. In total, we get 6,195 unique entities, where 4,605 are unique domains (derived from the 5,884 = 3,177+2,707 hostnames) and 1,590 are unique hostnames to quantify through the framework.

B. Labeling Module

In our case study we leverage VirusTotal to infer the labels of the selected entities corresponding to the URL in PhishTank blacklist because not all entities are bad. We use the following heuristics to approximate the ground-truth dataset of the URLs on $\mathcal{L}$. For a given URL, we use its entity (i.e., hostname or domain name), which is the output of the *characterizing* module, to query VirusTotal. An entity is deemed malicious (M) if 3 or more VirusTotal detectors say it is malicious; an entity is deemed compromised (C) if no VirusTotal detectors deem it as malicious; otherwise, an entity is disregarded because we consider binary classification. In our experiment, the initial blacklist size is $|\mathcal{L}| = 12{,}863$, from which we obtain a total of 6,195 unique entities, which (equivalently, their corresponding URLs in $\mathcal{L}$) are denoted by D_{entity}. Among the 6,195 entities, 968 are labeled as compromised (C) denoted by D_{com} and 4,017 are labeled as malicious (M) denoted by D_{mal}, leading to a total of 4,985 entities, denoted by $D_{labeled} = D_{com} \cup D_{mal}$; while noting that the other 1,210 entities are disregarded.

C. Feature Extraction & Analytics Module

Extracting Feature Values. For the feature of brand name in hostname (f_1), we curate a list of top 5,000 domains from Tranco [19] on the day of blacklisting and extracted the domain part (e.g., "`example`" from `example.com`) as the brand name. We only consider the brand names with string length greater or equal to 4 because shorter ones can include a lot of noises (e.g., 'fb' is a brandname of Facebook but other benign legitimate domains / hostnames such as 'fbox' could also include 'fb' as a sub-string, which causes ambiguity). If any brand name is present in the hostname or (sub)domain name, we set feature $f_1 = 1$, and $f_1 = 0$ otherwise. For feature f_2, we use dnsTwist [22] to generate typo-squatted domain names based on the top 5,000 brand names mentioned above. This leads to 24,213,971 twisted domains. Among these twisted names, we keep the ones with length greater than or equal to 4 for the same reason as mentioned above and assign $f_2 = 1$ if twisted brand name is present, and $f_2 = 0$ otherwise.

For deriving the values of features f_3 to f_9, we use the hostname and domain name contained in $D_{labeled}$. For feature f_{10}, we use the python `requests` module with an entity en_i as input. If entity is unreachable, then we set $f_{10} = 0$. For features f_{11} and f_{12}, we query the CIR.CL passive DNS database [20] for the 'A' record and 'NS' record corresponding to the entity in $D_{labeled}$, which. gives us a hint on the corresponding host's or domain's past activities. We sum up all the 'A' and 'NS' record counts for f_{11} and f_{12}, respectively. For feature f_{13}, we use the python module `dns.resolver` to resolve the name servers corresponding to the domain name and cross-check if the self-resolving name server is present ($f_{13} = 1$) or not ($f_{13} = 0$).

For feature f_{14}, we extract the Tranco rank list corresponding to the dataset time-frame. We set f_{14} to be the rank of the domain name if it is on the list of Tranco; otherwise, we set is to be the lowest rank or 6,000,000 because it is the size of the Tranco list. For feature f_{15}, we also use the Tranco list and the hostname for determining the value of f_{15}.

For feature f_{16}, we rely on a third-party open-source penetration testing tool *sublist3r*, which uses OSINT [23] to query from search engines (e.g., Yahoo, Bing, Baidu, and Ask) and other threat intelligence feeds (e.g., Netcraft, ThreatCrowd, DNSDumpster, and ReverseDNS).

Feature Values Analysis. We first analyze the correlations between features. By analyzing the Pearson correlation [24] among the numeric features, we find no significant correlations between the numeric features since their correlation coefficient is less than 0.5 between all pair of features.

D. Training Interpretible ML Model Module

We use the labelled data, $D_{labeled}$, for training and testing interpretible ML classification models. We randomly select 80% of data, $D_{train} \in D_{labeled}$, for training and the remaining 20% of data, $D_{test} = D_{labeled} - D_{train}$, for testing. The training set ($|D_{train}|$=3988) contains 3,219 malicious entities and 769 compromised entities, while the test set ($|D_{test}|$=997) contains 798 malicious entities and 199 compromised entities.

In our experiments, we consider the following ML models: Explainable Boosting Machine (EBM) [25], [26], Decision Tree (DT) and Random Forest (RF). Since the last two models are well-known, we only briefly review EBM. EBM a tree-based cyclic gradient boosting model [26], which improves the generative additive mode (GAM) [27]. In a GAM, the model outcome is $f(\mathcal{E}[y_i]) = \beta_0 + \sum \phi_j^i$, meaning that summation of individual feature's contribution is added to the model along with a co-efficient β_0; in EBM, the model outcome is $f(\mathcal{E}[y_i]) = \beta_0 + \sum \phi_j^i + \sum \phi_{j,k}^i$, which contains another summation of the pair-wise interactive feature contributions $\phi_{j,k}$ where $j \neq k$. These feature contributions can be deemed as the explanation set, denoted as $E_i = \{\phi_1, \ldots, \phi_{16}, \phi_{1,2}, \ldots, \phi_{15,16}\}$ where 16 is the total number of actual features for this case

study, which quantifies the contributions of individual features as well as pairs of features for the predicted label of entity en_i. Here, EBM is chosen because EBM provides the probabilistic quantitative classification along with the interpretability at the individual prediction outcome both of which are required in the proposed framework. Moreover, explainability provides transparency and trust in classification of highly imbalanced datasets.

E. Probabilistic Classification With Interpretation Module

This module answers both RQ2 and RQ3 through the use of probabilistic label prediction for a given entity, and the corresponding visualization of an explanation set, respectively. In our experiments, we use the InterpretML platform's visualization to particularly answer the RQ3. Table II presents the EBM model and the other ML models' (i.e., accuracy, AUC score, weighted precision, weighted recall, and weighted F-1 score). We observe that EBM performs better than the two other models. Moreover, EBM offers interpretations for individual predictions, which is important to the law-enforcement for effective decision-making on certain entities. Therefore, we will focus on EBM in the rest of the paper. The following Decision-Making module provide examples for use cases of the framework.

TABLE II: Performance Metrics of the ML Models on D_{test}

Models	Acc	AUC	Precision	Recall	F-1 Score
EBM	$85 \pm 1\%$	0.87	0.86	0.98	0.92
RF	$84 \pm 1\%$	0.83	0.88	0.93	0.86
DT	82 ± 1	0.72	0.88	0.88	0.88

F. Decision-Making Module

This module answers the RQ4 by aiding the law-enforcement with the appropriate decision support based on the satisfaction with the quantitative predictive classification and the visualized explanation set (i.e., interpretation). For blacklisted entity en_i, in this module the law-enforcement receives a prediction probability y_i for the entity to be malicious or compromised, together with an explanation set E_i as visualized through InterpretML platform shows top contributing features for that prediction. In what follows, we use examples to show how the system can indeed support the law-enforcement decision-making process.

Example 1 (Malicious → Takedown). There are maliciously registered websites abusing hosting services such as the `inmotionhosting.com`. Figure 8 shows one example. In this example, the EBM classifier predicts the hostname in question as malicious (M), with the explanation that several indicators—such as the number of name server records (22272), digits ratio (0.10), entity length (29.00), unique alphanumeric characters (15.00) —make significant contribution to the malicious prediction; whereas, a few other features—such as the number of A records, number of dots, domain rank, number of tokens, self-resolving NS, max token length, and number of hyphen—contribute to indicate that the hostname is compromised (C). However, we observe that the prediction by the interpretable EBM classifier is made as malicious (label value 1) with a high probability 0.868. As a result of both the high prediction probability and the strong explanations, the law-enforcement should *takedown* this hostname `secure285[.]inmotionhosting.com` and possibly notify the domain owner `inmotionhosting.com` for clean up because it is a hosting service provider.

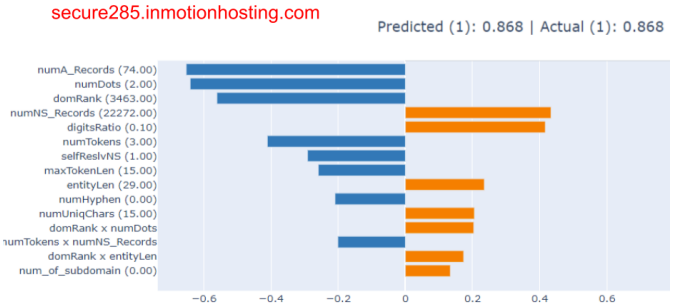

Fig. 7: Example 1: malicous → takedown.

Example 2 (Compromise → Notify). Oftentimes there are many legitimate business website that gets compromised by attackers and abused to wage cyberattacks, which place them into blacklists. But it is very important that law enforcement identify this legitimate entities and try to notify them asap for cleaning up. Figure 8 shows one example of a compromise case where the framework is predicting the input entity `123formbuilder.com`. In this example, the EBM classifier predicts domain name in question as compromised (C) with explanations that indicators—such as hostname ranking (7070), number of A records (96), number of tokens (2.00), number of dots (1.00) —make significant contribution towards the compromise prediction. There are few indicators—such as digit ratio and (domain rank × entity length) —make some contributions towards the malicious (M) prediction. In this particular case, the prediction probability for class C is 0.916 which is very high and trustworthy along with the strong host rank features. Hence, it is quite trustworthy that the domain name is compromised in this case and thus be notified by law-enforcement.

Example 3 (Compromise → Further Analysis). In this example, the domain name `whyymedia[.]com.au`corresponds to available domain under the hosting domain `h2osupportservices.com.au`. It is definitely not a malicious website for now, but we do not know for sure it it is a safe site in the future. It is blacklisted by PhishTank.

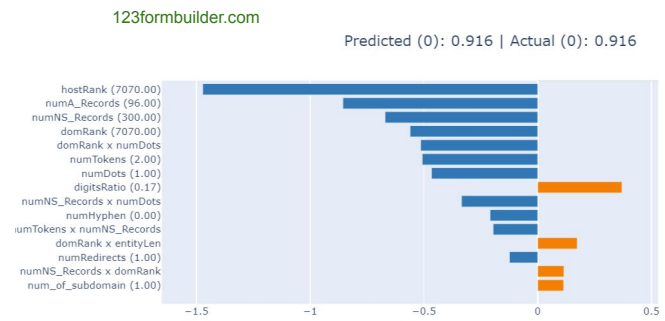

Fig. 8: Example 2: Compromised → Notify.

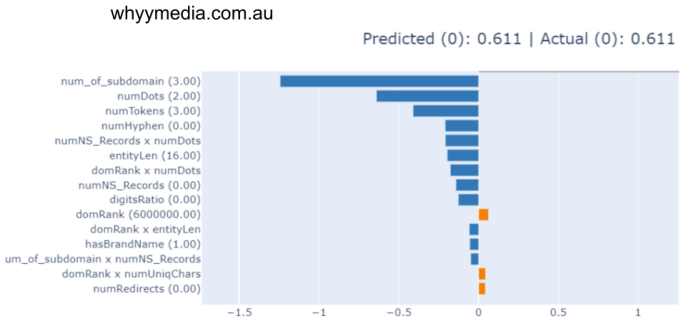

Fig. 9: Example 3: Compromised → Further Analysis.

As highlighted in Figure 9, the EBM classifier predicts it as compromised (C) rather than malicious (M), while only 3 features contribute to supporting the prediction that the domain is M rather than C. However, the probability of class C prediction is 0.611, which infers there is uncertainty associated with this prediction. Thus based on the decision-making module recommendations as shown in the flowchart described in Figure 6, the law-enforcement should not take actions (e.g., notify or takedown) without conducting a further analysis. Our manual verification on a later date, which is more than 10 days after the URL is blacklisted by PhishTank, confirms that the associated domain name is indeed legitimate but may be taken by attackers in future. Hence, further analysis will justify the corresponding action.

Example 4 (Malicious → Further Analysis). Figure 10 shows an example, where `sctrlgin[.]com` is blacklisted by PhishTank. The EBM classifier predicts it as malicious (M) with a probability of 0.675. Moreover, the classifier does not give a convincing interpretation on the prediction. Specifically, only few features—such as the domain rank (6000000) and number of redirects (0.00)—contribute to predicting it as M, which is not convincing enough. This would give the law-enforcement a low confidence in the prediction, suggesting that the law-enforcement should conduct a further analysis on the domain name in question.

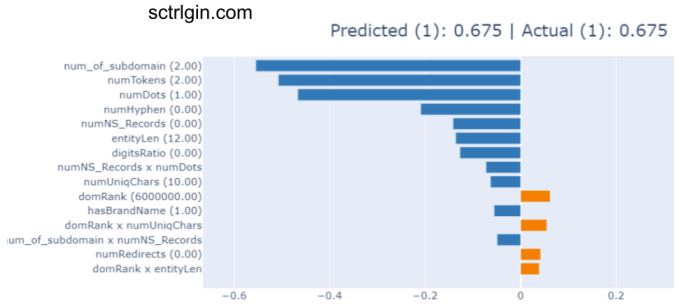

Fig. 10: Example 4: malicious → further analysis.

V. Discussion and Limitations

The case study is presented with EBM model [12] for interpretable ML because we want to provide explanation for individual prediction outcome, which EBM is well capable of. Again, existing literature states that EBM is often more intelligible and high performing than traditional Random Forest model that often successfully used in malicious website detection [28]. However, the present study has several limitations. First, in the current framework we rely on VirusTotal (VT) for labeling websites, but VT is not perfect [29], meaning that the resulting labels are not the try ground truth. Second, we do not use any WHOIS features because of potential concerns in relation to the GDPR [30]. Third, we do not analyze or leverage website contents for this study. Fourth, the quality of explanations and further evaluation need to be quantified in future studies. Fifth, there are still expert decision to make based on the framework outcome, which needs to be addressed in future studies with more automated Decision Support System (DSS) for the law-enforcement.

VI. Related Works

Although the *law-enforcement* perspective is a new dimension in dealing with blacklisted websites, the problem of malicious websites has been extensively investigated (see, e.g., [2], [31]–[33]). While the literature studies are loosely related to ours, it's worth discussion by divided them into the following categories. ***First***, few recent studies mention the notion of *compromised websites* in the same sense as ours (i.e., they are owned by legitimate users) [34]–[36]. However, these studies do not consider the notion of *compromised hostname* and *compromised domains* separately. ***Second***, from an interpretability point of view, most studies deal with the detection of malicious website or phishing URLs via black-box ML models, which often provide highly accurate models but lack interpretability and cannot be used for law-enforcement (e.g., takedown, notify) purposes. Recently, Silva et al. [36] use the LIME explanation method on the random forest model to provide global feature explanations but not individual predictions. In our case study, we use the EBM model for individual prediction interpretations, which the EBM model is also used to detect phishing URLs [37]. ***Third***, there are studies focusing on the detection of phishing webspages or URLs [38], [39]. ***Fourth***, there are studies on leveraging new kinds of information to detect malicious domains. For example, Bilge et al. present EXPOSURE [40], which leverages passive DNS analysis to detect malicious domains; their study inspires us to use the CIR.CL passive DNS dataset and incorporate 'A' records and 'NS' records as features in this study. Another proactive defense tool, PREDATOR [8], aims to detect domain abuses at the time of registration, which is effective against bulk registration events. ***Fifth***, there are studies on sophisticated attacks and defenses. For example, attackers may re-register expired benign domains to exploit their residual trust and evade reputation based detection (i.e., domain drop-catch). studies [5]–[7] investigate how impersonation and combo-squatting can evade detection by blacklists for a long time, further justifying the incompleteness of blacklists. Lastly, the notion of *adversarial malicious website detection* has been studied in [41].

VII. Conclusion

We presented a framework to help the law-enforcement deal with blacklisted websites, namely taking the appropriate data-driven decisions in terms of (i) whether a domain or host should be treated as the entity for action and (ii) whether the

entity should be shut down or notified to the stakeholder. The framework leverages interpretable ML techniques and quantitative classification, which is essential in website maliciousness where datasets are highly imbalanced. Our case study shows that the framework is useful, the results are accurate, but there are rooms for improvement. In the future, the framework can be enhanced to incorporate ways to analyze quality of the interpretations of ML models and to understand why misclassification happens with a more complete law-enforcement decision-support system framework to cope with blacklisted websites.

Acknowledgement. We thank Adham Aytabi and Raymond M. Bateman for their valuable comments. This work was supported in part by NSF Grants #2122631 and #2115134, and by Colorado State Bill 18-086.

REFERENCES

[1] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM TIST*, vol. 2, no. 3, pp. 30:1–30:24, 2011.

[2] L. Xu, Z. Zhan, S. Xu, and K. Ye, "Cross-layer detection of malicious websites," in *ACM CODASPY*, 2013, pp. 141–152.

[3] D. Chiba, T. Yagi, M. Akiyama, T. Shibahara, T. Mori, and S. Goto, "Domainprofiler: toward accurate and early discovery of domain names abused in future," *International Journal of Information Security*, vol. 17, no. 6, pp. 661–680, 2018.

[4] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *International Workshop on Recent Advances in Intrusion Detection*, 2014, pp. 1–21.

[5] K. Tian, S. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: Tracking down elite phishing domains in the wild," in *Proc. Internet Measurement Conference 2018*, 2018, p. 429–442.

[6] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis, "Hiding in plain sight: A longitudinal study of combosquatting abuse," in *Proc. ACM CCS*, 2017, pp. 569–586.

[7] A. Banerjee, M. S. Rahman, and M. Faloutsos, "Sut: Quantifying and mitigating url typosquatting," *Computer Networks*, vol. 55, no. 13, pp. 3001–3014, 2011.

[8] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "Predator: proactive recognition and elimination of domain abuse at time-of-registration," in *Proc. ACM CCS*, 2016, pp. 1568–1579.

[9] "Netcraft site take down service," https://www.netcraft.com/, 2022. Accessed May 1, 2022.

[10] S. Kaur and S. Randhawa, "Dark web: A web of crimes," *Wireless Personal Communications*, vol. 112, no. 4, pp. 2131–2158, 2020.

[11] T. Moore and R. Clayton, "Evil searching: Compromise and recompromise of internet hosts for phishing," in *International Conference on Financial Cryptography and Data Security*, 2009, pp. 256–272.

[12] H. Nori, S. Jenkins, P. Koch, and R. Caruana, "Interpretml: A unified framework for machine learning interpretability," *arXiv preprint arXiv:1909.09223*, 2019.

[13] P. V. Mockapetris, "Rfc1034: Domain names - concepts and facilities," USA, 1987.

[14] A. Niakanlahiji, M. Pritom, B. Chu, and E. Al-Shaer, "Predicting zero-day malicious ip addresses," in *Proc. ACM Workshop on Automated Decision Making for Active Cyber Defense*, 2017, pp. 1–6.

[15] D. Chiba, M. Akiyama, T. Yagi, K. Hato, T. Mori, and S. Goto, "Domainchroma: Building actionable threat intelligence from malicious domain names," *Computers & Security*, vol. 77, pp. 138–161, 2018.

[16] G. C. Moura, R. Sadre, and A. Pras, "Bad neighborhoods on the internet," *IEEE communications magazine*, vol. 52, no. 7, pp. 132–139, 2014.

[17] VirusTotal, "Virustotal," https://www.virustotal.com/, 2020.

[18] A. I. Inc., "Global top sites," https://www.alexa.com/topsites, 2008, Dec.

[19] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proc. NDSS*, 2019.

[20] L. Computer Incident Response Center, "Passive dns," https://www.circl.lu/services/passive-dns/, 2020, accessed on 8/1/2021.

[21] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Comput. Surv.*, vol. 49, no. 4, pp. 62:1–62:35, Dec. 2016.

[22] DNSTwist.it, "Dns twist," https://github.com/elceef/dnstwist, last accessed on 5th October, 2021.

[23] M. A. Masud, "An open source intelligence (osint) framework for online investigations," June 2019. [Online]. Available: http://essay.utwente.nl/78074/

[24] J. Benesty, J. Chen, and Y. Huang, "On the importance of the pearson correlation coefficient in noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 757–765, 2008.

[25] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning," in *Proc. CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[26] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Proc. ACM KDD*, 2012, pp. 150–158.

[27] M. Ying, "Additive models of probabilistic processes," *Theoretical Computer Science*, vol. 275, no. 1-2, pp. 481–519, 2002.

[28] C. Liu, L. Wang, B. Lang, and Y. Zhou, "Finding effective classifier for malicious url detection," in *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, 2018, pp. 240–244.

[29] P. Vallina, V. Le Pochat, Á. Feal, M. Paraschiv, J. Gamba, T. Burke, O. Hohlfeld, J. Tapiador, and N. Vallina-Rodriguez, "Mis-shapes, mistakes, misfits: An analysis of domain classification services," in *Proc. ACM Internet Measurement Conference*, 2020, pp. 598–618.

[30] E. Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016, April.

[31] M. Pritom, K. Schweitzer, R. Bateman, M. Xu, and S. Xu, "Data-driven characterization and detection of COVID-19 themed malicious websites," in *IEEE International Conference on Intelligence and Security Informatics*, 2020, pp. 1–6.

[32] ——, "Characterizing the landscape of COVID-19 themed cyberattacks and defenses," in *IEEE International Conference on Intelligence and Security Informatics*, 2020, pp. 1–6.

[33] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proc. ACM KDD*, 2009, p. 1245–1254.

[34] S. L. Page, G.-V. Jourdan, G. V. Bochmann, I.-V. Onut, and J. Flood, "Domain classifier: Compromised machines versus malicious registrations," in *International Conference on Web Engineering*. Springer, 2019, pp. 265–279.

[35] S. Maroofi, M. Korczyński, C. Hesselman, B. Ampeau, and A. Duda, "Comar: Classification of compromised versus maliciously registered domains," in *2020 IEEE European Symposium on Security and Privacy (EuroS P)*, 2020, pp. 607–623.

[36] R. De Silva, M. Nabeel, C. Elvitigala, I. Khalil, T. Yu, and C. Keppitiyagama, "Compromised or attacker-owned: A large scale classification and study of hosting domains of malicious urls," in *30th USENIX Security Symposium*, 2021, pp. 3721–3738.

[37] P. Hernandes, C. Floret, K. De Almeida, V. Da Silva, J. Papa, and K. Da Costa, "Phishing detection using url-based xai techniques," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 01–06.

[38] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "Deltaphish: Detecting phishing webpages in compromised websites," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 370–388.

[39] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using url and html features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[40] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive dns analysis service to detect and report malicious domains," *ACM Trans. Inf. Syst. Secur.*, vol. 16, no. 4, pp. 14:1–14:28, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/2584679

[41] L. Xu, Z. Zhan, S. Xu, and K. Ye, "An evasion and counter-evasion study in malicious websites detection," in *Proceedings of IEEE 2014 Conference on Communications and Network Security (IEEE CNS'14)*, 2014.