Constructs of Deceit: Exploring Nuances in Modern Social Engineering Attacks

Mohammad Ali Tofighi, Behzad Ousat, Javad Zandi, Esteban Schafir, and Amin Kharraz

Florida International University, USA

Abstract. Despite the increasing effort in the defense community in developing robust security solutions, social engineering attacks are getting more prevalent every year. Detecting fraudulent websites has been a concurrent task of both academia and industry in combating this type of attack. A common approach is to use supervised methods and labeled data to locate suspicious cases. In this paper, we evaluate a set of more common features related to the development and deployment aspects of websites that have been widely used in detecting scam and phishing websites over the years. As threat actors and the defense community are in a cat-and-mouse game, we aim to investigate whether such features are still prevalent or how to move forward in determining signs of malice when looking at the problem space at scale. Our study challenges the efficacy of deployment-based features, such as infrastructure providers or certificate issuers, in detecting fraudulent websites. Additionally, we perform an empirical analysis of the development aspects of websites that can be utilized in the detection pipeline.

1 Introduction

Social engineering attacks persist as a significant security threat. The impact of these attacks is often deep and consequential. Modern social engineering attacks have evolved to deliver different classes of malicious code while collecting extensive financial and personal information [1–5]. Moreover, these attacks result in significant collateral damage by harming the reputation and necessitating substantial effort to mitigate abuse. Over the years the security defense community has developed various methods and tools to fight against social engineering attacks [6–10]. The core insight in a large number of prior works is that the majority of adversaries behind social engineering attacks are *cost sensitive*. That is, adversaries aim to minimize their costs to develop and distribute their social engineering web attacks. This observation has been translated into several detection heuristics on the defense side. For instance, prior work incorporated features that extracted the type of domain name or the network address to which they were resolved based on the intuition that adversaries are more likely cheaper domain names or network addresses or that they would use free services such as Let's Encrypt [11] more frequently to develop realistic websites.

Given that adversaries are continuously adapting their techniques to evade detection, a research question that arises is to what degree these features are still relevant in today's threat landscape. This paper aims to revisit the corresponding insights, aiming to answer what measures are still effective and can be insightful, and what measures are losing their effectiveness in this evolving landscape. Our work is guided by three primary research questions. First, we aim to investigate what has changed in network infrastructure to host fraudulent websites and if there still exists any distinguishing patterns in the usage of network addresses. Second, we investigate how premium certificate services, as a key component of the deployment mechanism, are being used in fraudulent website development. Lastly, we evaluate the adoption of software development techniques in the design and development of fraudulent websites and how they can be translated into defense mechanisms. To answer these questions, we partnered with a well-known security company and received daily access to their URL seeds for six months – from August 2022 to February 2023. We crawled these websites using an instrumented browser, collecting development and deployment-related information when visiting these websites. In the following, we describe the main findings of the experiments by analyzing 9.5 TBs of data collected over this period. It is also notable that to reduce false positives and evade websites, we removed all scans that resulted in an HTTP redirection to another domain. We also collected a list of the top 1 million websites from Chrome's UX Report (CrUX) and scanned them using our crawler, collecting the same set of artifacts.

The underlying infrastructure of fraudulent websites are getting more diverse and distributed. We identified 1,995 infrastructure providers utilized in fraudulent websites. 1,764 (89%) of the providers are being also used in the top legitimate websites. We observed that 20 ASNs were present in over 1,000 cases on both fraudulent and legitimate websites. Notably, *Cloudflare* was the top infrastructure provider in both categories with 461,913 (51.22%) instances in legitimate URLs and 66,258 (30.96%) times observed for the fraudulent group. The second common provider was *Amazon* with 279,135 (30.95%) and 31,043 (14.5%) instances in legitimate and fraudulent groups respectively.

The adoption of certificate services in adversarial settings is getting significantly more diverse. While we observed that Let's Encrypt, a free certificate authority, is the most prevalent certificate issuer for both legitimate and fraudulent websites (38% and 36% respectively), premium SSL certificate services are also prevalent in the fraudulent websites. In particular, we observed that over 84,289 (40%) of the certificates used in fraudulent websites belong to premium services such as DigiCert, GoDaddy, and Sectigo.

The usage of major web technologies is often very similar among fraudulent and legitimate websites. Similar to legitimate websites, modern fraudulent websites also use popular Javascript libraries (e.g., core-js, lodash), or development frameworks (e.g., bootstrap, animate-ccs). However, the difference is visible for technologies from advertising and analytics categories. Specifically, we observe that over 600,000 (67%) of legitimate websites utilize *analytics* tools but the usage is diminished to nearly 24,000 (11%) on fraudulent webpages. We used Wappalyzer to detect web technologies used in websites; based on the version we used in our analysis, we were able to detect more than 4,500 technologies across 102 categories. Among other findings, perhaps the main takeaway of this paper is to provide practical evidence in modern social engineering attacks. The features that were historically being used in classifications are not likely to be effective for detecting and attributing cost-sensitive attacks anymore. We claim that the definition of cost-sensitive attacks needs to be adapted to stay effective, given the dynamic changes that are taking place in the adversarial landscape.

This paper makes the following contributions:

- We built a dataset containing various artifacts, including the run-time behavior artifacts, web technologies being used, certificate details, and underlying network infrastructure for 213,958 fraudulent and 901,817 legitimate websites over 12 months.
- We performed a large-scale longitudinal analysis of fraudulent websites, including phishing and scam websites, on how they differ from legitimate websites in development and deployment
- We provided two case studies using the collected dataset. The dataset is accessible upon request containing over 9TBs of artifacts for legitimate and fraudulent websites.

2 Background and Related Work

A substantial volume of research has been dedicated to the analysis and identification of fraudulent websites. We specifically focus on supervised approaches, wherein diverse features are extracted from both legitimate and fraudulent websites. Typically, a machine learning model is trained to classify between the two categories effectively. To comprehensively analyze the related studies, we categorize the features into two groups: development and deployment. Development features are the ones that are determined during the development of the website such as the structure of the page or the utilized technologies. On the other hand, deployment features are considered the ones that come from the deployment decisions such as the infrastructure provider or SSL certificate. Table 1 includes a summary of common features in each of these categories and shows whether a related study has used them in their classifier or not. In the following, we briefly describe the features and the related works that have utilized them for detection purposes. Despite the merits associated with each of these approaches, they all exhibit vulnerability to evasion tactics employed by attackers.

Page DOM. DOM of the Page can play a crucial role in fraud detection as it provides insights into the structure and content of a page. Pan et al. [12] utilized features related to a web page's identity, including the page title, destination of HTTP requests, anchor elements, and content of the HTML body. Similarly, Rosiello et al. [13] focused on the graph representation of the page DOM to extract features related to the number and structure of HTML tags. Other studies such as [15, 17, 20, 21] also leveraged page DOM features in their classifiers.

Remote Resources. The utilization of remote resources is another aspect considered in fraud detection models. Whittaker et al. [14] discussed Google's ML-powered classifier, analyzing the URL of the website and considering external

Table 1: Related Studies on Fraudulent Website Detection

Study	Venue	Year	Selected Features								
			Development						Deployment		
			Page DOM	Remote Resources	Technologies	HTML Content	CSS	Script Tags	SSL Cert	ASN/IP Address	Page URL
[12]	ACSAC	2006	✓	✓		✓					
[13]	SecureComm	2007	✓								
[14]	NDSS	2010	✓	✓						✓	✓
[15]	ACM TOPS	2011	✓	✓							✓
[16]	NDSS	2011								✓	
[17]	INCoS	2013	✓				✓				
[18]	APWG eCrime	2015							✓		
[19]	USEC	2016	✓								
[20]	ESORICS	2017	✓	✓		✓	✓				✓
[21]	ASIA CCS	2017	✓	✓							
[22]	IEEE ISI	2018			✓			✓	✓		
[23]	AISec	2018							✓		
[24]	IEEE S&P	2018	✓	✓					✓		
[25]	EURASIP	2019					✓				
[26]	ESWA	2022	✓	✓	✓						✓
[27]	IEEE S&P	2023	✓	✓						✓	
[28]	ISCC	2023	✓								

links and images included in the web page to make a decision. Studies like [26, 15] also explored remote resource features in their detection mechanisms.

Web Technologies. The technologies employed in a website can be indicative of its legitimacy or fraudulent intent. Studies including [26] considered the presence of specific web technologies in their classifiers. Niakanlahiji et al. [22] detected web technologies used in the website using HTTP headers and used them to detect the targets.

HTML Content. Features extracted from the HTML content of websites provide valuable information for detection. Pan et al. [12] and Corona et al. [20] utilized features related to HTML content to distinguish between legitimate and fraudulent websites. These features include the content of the HTML body, page title, and stylesheet-related elements in the DOM.

CSS Features. The CSS of a page can also serve as a discriminative feature in classification. Studies such as [20, 25, 17] extracted features from the CSS of web pages to build their classifiers.

Script Tags. The presence and characteristics of script tags in the HTML code were considered by related studies. In [22], authors check the code complexity of the included JavaScript as features for detecting fraudulent websites.

SSL Certificate. SSL certificate-related features have been utilized in scam detection. Torroledo et al. [23] and Dong et al. [18] based their work solely on SSL certificates, using various metadata for detection.

ASN/IP Address. Analyzing ASN/IP addresses is another aspect considered in scam detection. Whittaker et al. [14] used the ASN/IP address as one of the features in their model. Additionally, Bilge et al. [16] focused on DNS traffic and their resolved IP addresses.

Page URL. In some cases, the URL of the website is used to decide without actually visiting the website. Studies including [15, 27, 26], incorporated features derived from the page's URL in their classifiers.

Despite all of the advancements in the last two decades, the core techniques used by the defense community to detect malicious entities on the Web remain the same. This paper aims to see an adaptation of adversaries to these approaches and whether they are still good indicators or whether there is a need for novel data points to distinguish between the two groups.

3 Methodology

In this section, we introduce the three research questions we seek to answer. We then elaborate on how we conducted our research to collect forensically relevant artifacts to answer these questions. We also explain our approach to clean the dataset before running the large-scale analysis.

3.1 Research Questions

Our research seeks to answer the following research questions:

RQ1: How do adversaries abuse network infrastructure? There have been several studies on the use of free hosting services or abusing legitimate infrastructure to publish fraudulent websites. This research question aims to answer how the usage of underlying infrastructure has changed over time and if there is a distinguishing pattern in the underlying network structure in legitimate and fraudulent websites in modern social engineering attacks.

RQ2: How do adversarial campaigns make use of premium certificate services? There is no lack of evidence that adversaries have been using certificates to look legitimate when developing their fraudulent websites. The question we seek to answer is to investigate the distribution of legitimate/premium certificate providers in fraudulent web development and how the distribution differs between legitimate and fraudulent websites.

RQ3: How do fraudulent websites differ from legitimate websites when considering web development aspects of the websites? Our intuition is that, unlike legitimate and high-profile websites, adversaries are less likely to prioritize best practices and software development details of their fraudulent websites. We take a deeper look on if this hypothesis holds and how it would be different among different forms of websites.

Scope of the Study. There have been several work on different classes of social engineering attacks. Scareware [29, 30], Survey and technical scams [24, 31], themed-based attacks [32] are just a few examples of social engineering attacks. The term fraudulent website we are using in this paper covers websites such as phishing as well as those types of scam websites designed to trick incoming users and incorporate certain elements of social engineering such as deception to deliver the attack. That said, the term does not cover other forms of social engineering attacks that do not require development of a website like common forms of spear-phishing or technical scams [31].

3.2 Datasets

Fraudulent Websites We collaborated with a security company and subscribed to their premium URL seed, which exclusively provides URLs confirmed to direct users to fraudulent web pages. We collected the longitudinal list of 313,110 fraudulent websites over the period of six months from August 2022 to February 2023. This dataset was continuously updated at five-minute intervals, initiating scanning immediately upon capture. Throughout the scanning process, we encountered certain failures, primarily stemming from two key sources. The first category of failed scans arose from non-responsive URLs, which often result from the short lifespan of phishing websites. The second category encompassed scans that appeared to evade the scanner by redirecting to known legitimate web pages. After removing the failed scans including non-responsive URLs, possible evasions, and duplicate URLs, we ended up with 213,958 successful scans of phishing, scams, and other types of fraudulent websites. It is noteworthy to mention that the fraudulent dataset covers malicious websites in 50 different languages which shows the diversity of the targets.

Legitimate Websites. Our scanning pipeline is not limited to fraudulent websites. To have a more comprehensive overview, we incorporated the top 1 Million URLs included in Chrome's User Experience Report [33]. The complete dataset which is publicly accessible includes over 15M URLs. The dataset provides many attributes including the origin (URL) of the pages per month [34]. We used the data from February 2023 in our crawling pipeline. An overview of the datasets and the number of successful scans on each of them is presented in Table 2.

Table 2: URLs Scanned Using Different Data Sources

Type	**Legitimate**	**Fraudulent**
Source	CrUX Top 1M	Daily Seed
Date	Feb 2023	Aug 2022 - Feb 2023
Scans	901,817	213,958

3.3 Collected Artifacts

We built the forensics layer on top of the Chrome Devtools Protocol [35] to minimize the source code modifications and avoid possible crashes. The Chrome Debugging Protocol offers programmatic access to the browser engine and allows the code to attach to open windows and interact with the loaded JavaScript and the DOM tree created for the window. We used Chrome's Lighthouse [36] which is an open-source tool to monitor the performance, quality, and correctness of web applications. This tool is offered as a library that includes all of the extracted data from the website. We based our crawler on this tool and took advantage of custom data gatherers to collect additional data from different aspects of each page. These custom gatherers include but are not limited to *DOM Graph Gatherer* which adds the graph representation of the DOM of the page to be used in graph analysis; The *Network Metadata Gatherer*, which captures all DNS A records that the domain resolves to, along with ASN information of the

Table 3: Top infrastructure providers used in legitimate and fraudulent websites. 153,286 (71.64 %) of the fraudulent websites share the same infrastructure provider to deliver their page.

#	Legitimate Websites			Fraudulent Websites		
	Name	**Presence**		**Name**	**Presence**	
		#	%		#	%
1	Cloudflare	461,913	51.22	Cloudflare	66,258	30.96
2	Amazon	279,135	30.95	Weebly	38,356	17.92
3	Google	32,950	36.53	Amazon	31,043	14.50
4	Microsoft	31,975	35.45	Fastly	22,652	10.58
5	Akamai	31,450	34.87	Google	16,335	7.63
6	OVH	31,226	34.62	Microsoft	10,527	4.92
7	Fastly	29,614	32.83	Unifiedlayer	9,092	4.24
8	Hetzner	23,852	26.44	Dedipath	8,121	3.79
9	DigitalOcean	15,269	16.93	Namecheap	6,862	3.20
10	Incapsula	7,845	8.69	Network Solutions	5,355	2.50
11	Azion Technologies	6,832	7.57	DigitalOcean	3,796	1.77
12	XServer	6,689	7.41	Protocol	2,713	1.26
13	Hostinger	5,829	6.46	OVH	2,675	1.25
14	Ionos	5,186	5.75	Quantline Networks	2,597	1.21
15	Korea Telecom	4,737	5.25	Quadranet Global	2,093	0.97

associated IP addresses for the website's domain. Additionally, we implemented the *SSL Certificate Gatherer*, which captures the SSL certificate of the domain and stores it in a PEM file for future use. Furthermore, we incorporated a fingerprinting technique to detect used libraries in the target web applications [37]. While Lighthouse includes some JavaScript libraries in the result, the service does not cover a large list of web technologies used at the internet scale. In each scan, we collect the source code of the page, the loaded JavaScript files and their execution, resources and technologies, the entire HTTP request and responses, and the certificate information loaded into the browser while visiting the target website. The result of each scan is a JSON object that is archived based on the date of the scan. A sample of the collected data for both legitimate and fraudulent websites is publicly available in a GitHub repository [1]. Details to access the full dataset is provided in the repository.

4 Deployment Aspects of Modern Web Ecosystem

In this section, we empirically study the deployment of fraudulent websites and compare them with legitimate ones by looking at the implementation details in three different ways: (1) the underlying infrastructure used to deploy fraudulent websites, (2) the certificate services used to issue certificates for fraudulent pages, (3) the common web technologies used to deliver these websites while comparing them with legitimate websites.

[1] https://github.com/phishvsbenign/phishvsbenign

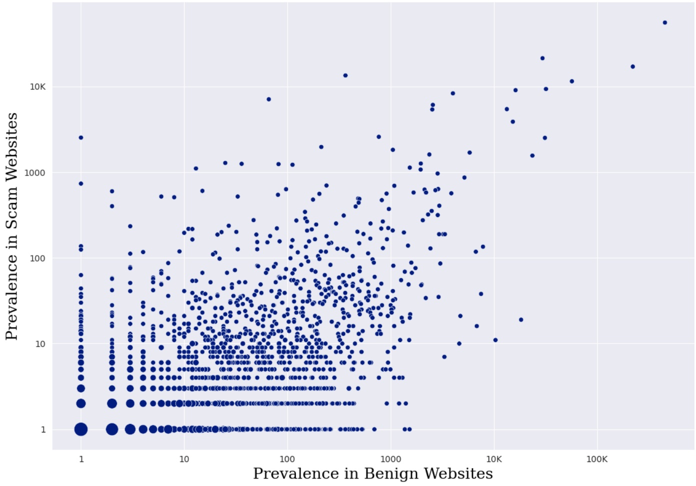


Fig. 1: The use of ASNs in legitimate and fraudulent websites. 1,764 (89%) ASNs observed in the fraudulent websites are present on the legitimate side as well. X and Y axes are log-scaled for visualization.

4.1 Deployment Distribution

To gain insights into the hosting environment of both fraudulent and legitimate websites, we analyzed the deployment of these websites in the wild by studying their network addresses. Our methodology involved identifying the top ASNs associated with fraudulent websites and examining their prevalence in the benign context. The results of this investigation are depicted in Figure 1, which illustrates the frequency of each target ASN appearing in both fraudulent and legitimate websites. The analysis shows that 1,764 of 1,995 (89%) ASNs observed in the fraudulent websites deployment are being used in both fraudulent and legitimate websites. Table 3 shows the top cloud service providers observed in our analysis. Nearly 72% of the fraudulent websites share the same infrastructure services (e.g., Cloudflare, Amazon, Google, Fastly, and OVH) being used in legitimate websites. This observation suggests that modern social engineering attacks are increasingly using infrastructures that were historically serving legitimate services – influencing the effectiveness of security mechanisms that monitor the hosting reputation.

4.2 Certificate Analysis

In addition to the deployment infrastructure, we analyzed how certificate authorities are being abused in fraudulent websites. In particular, we analyzed the certificate issuers and certificate validity periods in the collected datasets and compared them with each other. Aligned with prior work[38–40], usage of SSL certificates by phishing and scam websites has become increasingly prevalent to imitate. We observed that 176,397 (82.4%) of the fraudulent websites are using

Table 4: Top certificate issuers used in legitimate and fraudulent websites. 175,363 (81.96 %) of the fraudulent websites use the same top certificate authorities that legitimate websites use.

Legitimate Websites						Fraudulent Websites					
Name	Presence		Validity Days			Name	Presence		Validity Days		
	#	%	min	max	mean		#	%	min	max	mean
Let's Encrypt	328,207	38.3	90	90	90	Let's Encrypt	78,855	36.8	90	90	90
Cloudflare	140,651	16.4	15	366	365	DigiCert	27,662	12.9	91	825	371
DigiCert	90,507	10.5	19	826	365	Google Trust Services	17,306	8.0	84	91	88
Sectigo Limited	68,790	8.0	26	397	376	cPanel	13,344	6.2	91	366	92
Amazon	49,396	5.7	141	396	394	Cloudflare	12,322	5.7	365	366	365
GlobalSign	41,796	4.8	22	825	372	Sectigo Limited	9,767	4.5	91	397	376
Google Trust Services	25,698	3.0	31	91	88	GoDaddy.com	3,888	1.8	39	398	380
GoDaddy.com	23,705	2.7	26	398	380	Amazon	3,842	1.7	338	396	394
cPanel	22,611	2.6	91	366	92	ZeroSSL	2,717	1.2	91	366	114
Entrust, Inc.	7,650	0.8	79	397	376	Entrust, Inc.	2,138	0.9	277	397	376
Unizeto Technologies	5,670	0.6	48	395	364	Microsoft Corp.	1,563	0.7	182	366	347
ZeroSSL	5,599	0.6	91	366	114	GlobalSign nv-sa	1,462	0.6	34	826	372
Starfield Technologies	4,898	0.5	28	398	374	Starfield Technologies	497	0.2	187	398	374
GEANT Vereniging	4,759	0.5	117	396	365	D-Trust GmbH	198	0.0	233	394	366
SECOM Trust Systems	2,692	0.2	131	397	385	TrustAsia	178	0.0	91	397	359

SSL certificates to deliver their content. This number for the legitimate websites was 856,144 (94.9%) which shows a very narrow gap between the two.

Certificate Authorities. In our datasets, we identified a total of 147 certificate authorities involved in issuing SSL certificates. From these, 48 (33%) of the CAs had been abused in generating at least one certificate for a malicious website. Table 4 shows the top certificate issuers in legitimate as well as fraudulent websites. While the analysis shows that fraudulent websites use a smaller set of certificate issuers to generate certificates compared to legitimate websites, the number of certificates and frequency at which we observed in the legitimate dataset do not reveal any evident patterns, That said, it is not likely that certificate information, including certificate issuance, lifespan helps to distinguish these adversarial cases compared to legitimate websites.

Certificate Validity Periods. Additionally, we analyzed the validity period by calculating the number of days between issuance and expiration dates. The results of this analysis are presented in Table 4, revealing analogous trends in both scenarios, where the average duration for both categories of websites exhibits a high degree of similarity.

5 Development Practices in Fraudulent Websites

In this section, we present an analysis of the development aspects of fraudulent websites, from the complexity of their resulting HTML pages to UX and security practices that are expected to be seen in professionally developed legitimate websites. Specifically, we analyze the attributes of the Document Object Model (DOM), the prevalence of distinct HTML tags, the utilization of CSS/JS properties, two of the most common security practices, and web technologies used in the websites. This comparative study aims to highlight distinctive patterns and characteristics that can aid in the identification and differentiation of fraudulent websites from their counterparts.

5.1 DOM Stats Analysis

Analysis of a small set of data showed us that if we create the DOM graph of the page, the size of the graph is significantly different for legitimate versus fraudulent pages. Figure 2 presents a sample login form and its corresponding DOM graph using the graph drawing tool available at [41]. The Figure illustrates the elements of the page as well as width and height.

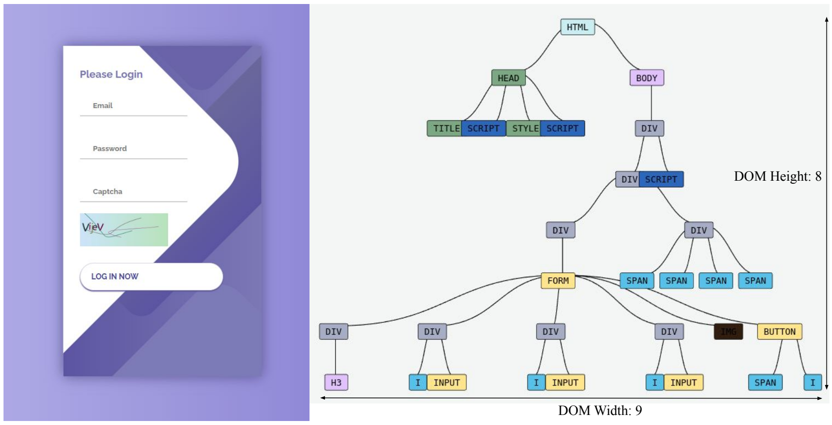


Fig. 2: Sample of a login form and its corresponding DOM graph

We investigated the idea by extracting the DOM statistics from the collected data and comparing the results for each category. Figure 3 showcases the CDF graph of DOM statistics for legitimate and fraudulent scans. The number of total body elements has a significant difference in the two cases. The mentioned reasons made us believe that DOM stats may be a useful feature to include in our analysis. We aim to identify distinguishing metrics that could effectively differentiate between the two by fingerprinting the usage of different resources in the document and the statistics of DOM elements in the pages. By examining these aspects, we try to uncover potential patterns that could aid in accurately distinguishing between the two types of websites. Figure 3 showcases the CDF graph of DOM statistics for legitimate and fraudulent scans. The number of total body elements has a significant difference in the two cases. We can see that a larger percentage of fraudulent websites have a low number of elements. For example, 60% of fraudulent websites have total elements of only 100 or less.

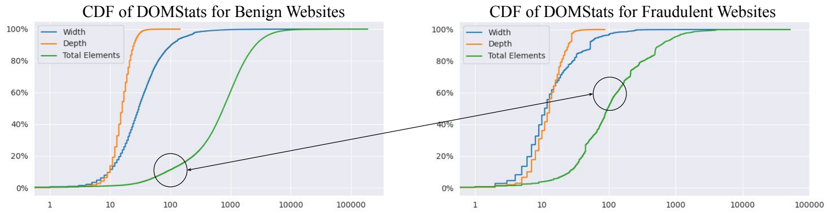


Fig. 3: DOM Statistic Comparison Between Legitimate and Fraudulent Websites. 60% of fraudulent websites have total elements of only 100 or less.

5.2 Resource Inclusions

The hypothesis is that developers of scam and phishing websites focus mainly on the look and feel of their target websites to defraud victims and focus less on common programming best practices that a legitimate website would consider. This part of the analysis aims to see whether fraudulent websites are less complex than legitimate ones in terms of included resources, or if they do they are most likely come with errors or not used at all when observing at the run-time behavior of the website. Figure 4 shows the comparison between the two.

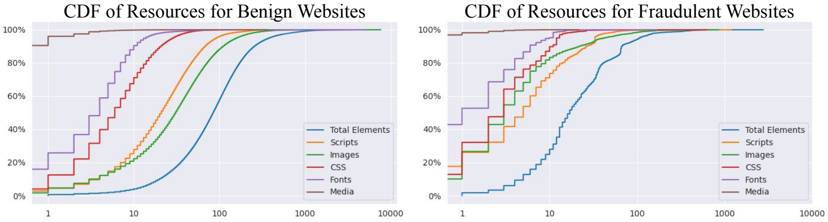


Fig. 4: Resource inclusion comparison between legitimate and fraudulent websites. Legitimate websites are more complex in terms of each resource type.

5.3 Scripts Analysis

In our investigation, we analyzed the sourced JavaScript in the collected websites and focused on evaluating the implementation of best practices, such as minification and the removal of unused JavaScript. As anticipated, the results revealed a notable discrepancy between the practices employed by developers of fraudulent websites and legitimate ones. Specifically, we found that developers of fraudulent websites tend to neglect these optimization techniques. When comparing the overall percentage of potential savings (based on total JavaScript size), legitimate websites demonstrated a modest 0.82% for unmodified JS and a more substantial 14.17% for unused JS. In contrast, these figures increased significantly for fraudulent websites, reaching 3.59% for unmodified JS and an impressive 29.15% for unused JS. Our findings underscore the importance of adhering to best practices in JavaScript development, as they not only optimize website performance but also serve as potential indicators for distinguishing between legitimate and fraudulent websites. By recognizing and addressing these disparities, we can gain better insights for developing protection solutions against social engineering attacks.

5.4 Security Mechanisms

One of the hypotheses we explored in our research was the disparity in security focus between fraudulent websites and legitimate ones. To investigate this, we employed two essential criteria: (1) Content Security Policy (CSP) headers, and (2) The usage of HTTPS. Upon analyzing the data, striking differences emerged between fraudulent and legitimate websites. We found that only 17% of legitimate websites had enforced CSP policies, while this number drastically dropped to less than 8% for fraudulent websites.

In terms of HTTPS analysis, we observed that the majority (82.4%) of the fraudulent websites were equipped with an SSL certificate. However, the main question would be to check whether these websites exclusively operated under HTTPS or not. This entails determining if plain-text HTTP requests are made while the primary page is presented through HTTPS, or if the website effectively redirects all HTTP traffic to HTTPS. We observed that 11% of legitimate websites lacked full HTTPS support, but this figure skyrocketed to 45% for fraudulent websites, which shows that while they are paying attention to obvious notations to use an SSL certificate, they do not care about the follow-up security practices. It is crucial to note that while these factors alone may not be conclusive evidence of a website being fraudulent or legitimate, incorporating them alongside other relevant indicators can significantly improve the accuracy of detection. By considering these features in combination with other factors, we can enhance our ability to identify and combat potential phishing and scam threats more effectively.

5.5 Technologies Analysis

One of the crucial aspects of analyzing websites from a development perspective is examining the employed technologies. We investigated the adoption of software packages on fraudulent and legitimate websites. Specifically, we queried the category of the observed packages and marked the top five most widely utilized ones for each category. Several categories, including JS libraries, programming languages, and web servers, emerged as consistent front-runners within both groups. Yet, a noteworthy observation emerged—categories like advertising and analytics exhibited a pronounced prevalence primarily within legitimate websites. We present the outcomes of our investigation into the utilized technologies in each website group. Our findings indicate that the technologies used in the development of fraudulent websites differ from those utilized in legitimate websites. While there are certain similarities in the technology stacks of both categories, we observe that scam websites are less likely to employ technologies from *analytics, advertising,* and *tag managers* categories. Table 5 provides a comprehensive

Table 5: Top Five Mostly Observed Technologies in Different Categories for Legitimate and Fraudulent Websites

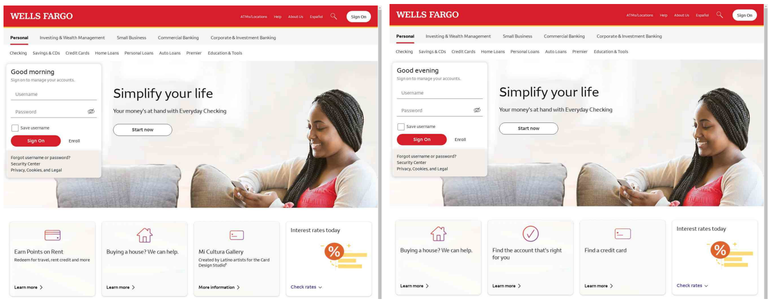
Category	Rank	Legitimate Websites			Fraudulent Websites		
		Name	Presence		Name	Presence	
			#	%		#	%
JavaScript Libraries	1	jquery	682,617	75.69%	jquery	110,275	51.54%
	2	core-js	352,214	39.05%	core-js	33,726	15.76%
	3	lodash	102,086	11.32%	lodash	15,907	7.43%
	4	lazysizes	82,983	9.20%	fancybox	9,886	4.62%
Programming Languages	1	php	435,297	48.26%	php	77,904	36.41%
	2	java	41,587	4.61%	node-js	6,352	2.96%
	3	node-js	34,504	3.82%	typescript	4,503	2.10%
	4	python	14,836	1.64%	java	4,036	1.88%
Analytics	1	google-analytics	599,485	66.47%	google-analytics	24,906	11.64%
	2	facebook-pixel	192,854	21.38%	snowplow-analytics	19,505	9.11%
	3	hotjar	48,276	5.35%	datadog	9,249	4.32%
	4	yandex-metrika	35,223	3.90%	ads-conversion-tracking	3,424	1.60%
Font Scripts	1	google-font-api	314,761	34.90%	google-font-api	34,935	16.32%
	2	font-awesome	189,689	21.03%	font-awesome	25,024	11.69%
	3	twitter-emoji-twemoji	91,451	10.14%	typekit	1,682	0.78%
	4	typekit	18,170	2.01%	twitter-emoji-twemoji	1,209	0.56%
Video Players	1	youtube	50,279	5.57%	videojs	9,820	4.58%
	2	mediaelement-js	12,320	1.36%	vimeo	9,493	4.43%
	3	vimeo	11,599	1.28%	mediaelement-js	9,488	4.43%
	4	videojs	8,082	0.89%	youtube	1,521	0.71%
Tag Managers	1	google-tag-manager	497,073	55.11%	google-tag-manager	15,744	7.35%
	2	adobe-launch	7,321	0.81%	adobe-launch	820	0.38%
	3	tealium	3,747	0.41%	ensighten	543	0.25%
	4	matomo-tag-manager	1,363	0.15%	tealium	528	0.24%
Advertising	1	google-adsense	102,534	11.36%	microsoft-advertising	1,143	0.53%
	2	google-publisher-tag	48,804	5.41%	33across	1,053	0.49%
	3	twitter-ads	45,319	5.02%	twitter-ads	1,016	0.47%
	4	microsoft-advertising	39,519	4.38%	dtscout	910	0.42%
Web Servers	1	nginx	225,808	25.03%	apache	63,116	29.49%
	2	apache	203,441	22.55%	nginx	53,948	25.21%
	3	iis	88,416	9.80%	litespeed	13,056	6.10%
	4	litespeed	30,493	3.38%	openresty	8,144	3.80%
CMS	1	wordpress	184,452	20.45%	weebly	19,175	8.96%
	2	drupal	18,580	2.06%	wordpress	5,478	2.56%
	3	joomla	8,182	0.90%	godaddy-website-builder	2,836	1.32%
	4	1c-bitrix	5,970	0.66%	adobe-experience-manager	2,134	0.99%
Frontend	1	bootstrap	293,081	32.49%	bootstrap	51,112	23.88%
	2	animate-css	47,785	5.59%	tailwind-css	5,120	2.39%
	3	zurb-foundation	16,606	1.84%	animate-css	3,442	1.60%
	4	tailwind-css	16,167	1.79%	marko	2,417	1.12%

overview of the top 10 categories observed in fraudulent and legitimate websites, showcasing the prevalence of top packages in each category. These disparities in technology adoption between fraudulent and legitimate websites serve as valuable indicators that can aid in distinguishing between the two and contribute to more effective identification and mitigation of social engineering attacks.

6 Case Studies

Throughout our analysis of different features between legitimate and fraudulent websites, we found interesting cases revealing possible future avenues of analysis in the fraudulent websites. The first case study discusses how utilizing development best practices on the defense side can potentially lead to more robust detection mechanisms where analyzing visual representation, certificate details, and infrastructure may not be sufficient. The second case study relies on the idea that cost-sensitive adversaries rely on code reuse practices, similar to legitimate developers, to reduce the cost of development. We investigate how this intuition can be translated into threat detection and attribution.

6.1 Case Study 1: Cloning Legitimate Websites for Fraud



(a) Original Wellsfargo Website (b) Fake Wellsfargo Website

Fig. 5: Example of scam page developed by duplicating the target legitimate counterpart. The two pages look identical and use the same set of technologies.

In numerous instances, we observed a disconcerting trend where fraudulent/phishing websites closely mimic their legitimate counterparts. Our investigation suggests that these deceptive pages are crafted by duplicating the entire target page and hosting it on a different domain with minimal alterations. Consequently, many of the criteria discussed in Section 5 prove inadequate in differentiating between authentic and fraudulent web pages. Figure 5 exemplifies one of these cases, showcasing two nearly identical pages of *Wellsfargo Bank*. Besides the visual similarity, we verified that the identified technologies were too similar to be reliable discriminators. However, a more in-depth analysis revealed a crucial distinction— the prevalence of errors, such as console messages, during the loading process of the page. Notably, the authentic Wells Fargo website exhibited zero console messages, whereas the fraudulent counterpart generated 12 network and JavaScript errors resulting from failures in accessing the original Wells Fargo servers. This is due to the fact that the scam page includes scripts that directly interact with the original servers. The act of duplicating these scripts without modification led to the observed errors. The detailed breakdown of errors for this specific case is available in Table 6. It is essential to note that the infrastructure features, including domain, certificate, and ASN, differ between the two websites; however, they cannot be deemed as dependable identification factors. As outlined in Section 4.1, scam pages strategically leverage the same deployment environment as authentic websites. Consequently, it becomes feasible to migrate the exemplified fraudulent website to the identical environment as the original target page, effectively obfuscating any differentiating elements in the infrastructure analysis. This case study underscores that despite adversaries duplicating their target websites to manipulate end-users, there are discernible differences that can be identified. With the right tools and perspectives, the defense side can effectively safeguard users from falling victim to such manipulative tactics.

Table 6: Console Errors Generated from Scam Wellsfargo Website

Error Source	Error Text
javascript (*6)	Access to XMLHttpRequest at https://connect.secure.wellsfargo.com/... from origin www–wellsfargo–com–... has been blocked by CORS policy: The ’Access-Control-Allow-Origin’ header contains the invalid value ’connect.secure.wellsfargo.com’.
network (*6)	Failed to load resource: net::ERR_FAILED https://connect.secure.wellsfargo.com/.../...638a.js

6.2 Case Study 2: Code Reuse for Development

Among our cases, we found fraudulent pages that were identical in several aspects. One of these cases is presented in Figure 6 where the three pages have the same DOM structure with minimal changes in the logo image. Table 7 presents the comparison of these three pages based on their domain, technologies, DOM stats, certificate issuer, and IP address. This particular case highlights the tendency of adversaries, much like legitimate developers, to employ code reuse, facilitating the widespread deployment of their fraudulent pages. An application of the aforementioned features discussed earlier in this manuscript is the potential to identify such scam campaigns, which may originate from the same source. We believe this is an interesting opportunity for defenders to formulate robust defense mechanisms for similarity testing. That is, customized similarity hashing mechanism similar to prior work in malware analysis research [42] can assist us to better catalog similar fraudulent websites without relying on the visual representation, infrastructure, and certificate which can be evaded.

Table 7: Comparison between a set of visually-identical fraudulent websites

Domain	*creditinternationalbank.com*	*wealthpathbank.com*	*moonstandcu.com*
Technologies	jQuery, jQuery-Fast, FastClick, Drupal		
DOM Stats	DOM Height: 20, DOM Width: 31, DOM Elements: 409		
Certificate	cPanel Inc.	Let’s Encrypt	cPanel Inc.
IP Address	82.163.176.106	192.185.46.77	162.241.176.106

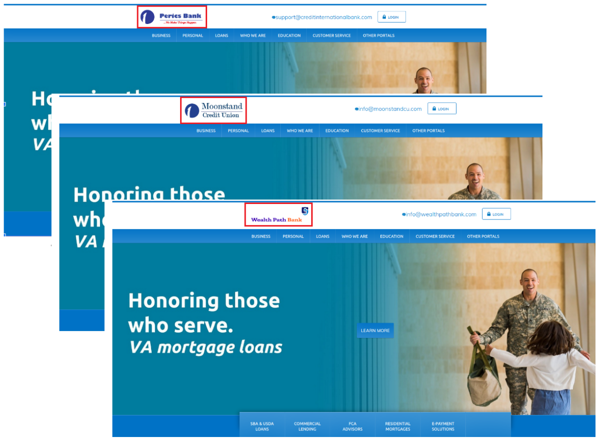


Fig. 6: Screenshots of different domains using the same design, but changing the bank name and logo.

7 Discussion and Future Works

In this section, we discuss the key findings derived from various analyses conducted during this research. Additionally, we put forth prospective avenues for further exploration and investigation.

Detecting Fraudulent Websites. Across various sections, we delved into multiple feature categories obtained from analyzing fraudulent web pages. Our observations indicate that although some of these features have been employed in prior phishing and scam detection efforts, their reliability may not be consistent across all cases. Notably, our analysis unveiled similarities between the deployment environment and certificate issuers of the fraudulent and legitimate websites. While past studies have found success in leveraging these features for phishing and scam detection, we believe that depending exclusively on such features diminishes the accuracy of the approach when applied in real-world scenarios. Additionally, our examination extended to the HTML features of the target pages, revealing a notable difference in complexity between scam pages and legitimate websites. Primarily, fraudulent pages exhibit a significantly reduced usage of fonts, images, and scripts, along with a smaller total count of DOM elements. This disparity can be attributed to the fact that adversaries prioritize functionality over aesthetics, concentrating solely on obtaining visitor credentials with minimal effort, thereby ensuring the scalability of their deceptive sites. Furthermore, we analyzed the development aspects of the pages to compare the utilized technologies and best practices. We observed differences in the usage of specific technology categories and libraries. Overall, we conclude that effective phishing and scam detection solutions necessitate a comprehensive approach, considering various factors and analyzing multiple facets of the target website to arrive at a reliable decision.

Runtime Features Analysis. In our investigations, we encountered instances where the static features of fraudulent pages closely mirrored those of their legitimate counterparts. This resemblance is especially prevalent in phishing websites, which can adeptly clone the front-end of authentic pages. Section 6.1 provides a detailed exploration of one such case. However, despite the visual similarity, we identified that the generated error messages on the scam page can serve as a robust indicator for detection. This insight underscores the potential effectiveness of analyzing runtime features in distinguishing between fraudulent and legitimate pages. The generated error messages represent a subset of other runtime features, encompassing console messages, network traces, and logs from development tools. Each of these features includes valuable information that holds promise for enhancing fraud detection. We leave the comprehensive analysis of runtime features to future research endeavors.

Campaign Analysis. Throughout our research, our focus was on identifying distinctive features of fraudulent and legitimate websites. During this process, we encountered instances where the same scam page was found across different domains, maintaining an identical design while altering the name and logo of the bank. Intriguingly, our brief analysis illuminated that the clusters of websites were operational under disparate domain names and distinct networks, characterized by distinct IP addresses and ASNs. While a subset of these clusters displayed visual resemblances, it was fascinating to find groups that exhibited entirely different designs while focusing on the same target group. These examples underscore the significant influence that integrating multiple page attributes can wield in enhancing the detection of phishing websites and conducting thorough campaign analysis. Discovering campaigns can help find scam pages at a higher rate and even prevent similar scam pages from being deployed from scratch which helps the defense side in the arm-race against adversaries.

8 Conclusion

In this study, we present an overview of techniques for detecting fraudulent websites. To investigate the details of the features prevalent in the contemporary fraud ecosystem, we collected artifacts by crawling a live feed of fraudulent URLs over six months. Our experiments indicate that the deployment aspects of fraudulent websites closely resemble those of legitimate ones, making them an unreliable source for distinction. However, observable differences emerge in the development features of the two groups. Notably, we found that fraudulent pages exhibit significantly lower complexity compared to legitimate ones. Furthermore, the adoption of best practices is largely overlooked in the fraud ecosystem. Additionally, we showcase case studies on how the collected features can serve purposes beyond mere detection, emphasizing the broader utility of our findings.

9 Acknowledgments

We would like to thank the anonymous reviewers for their thoughtful feedback. This project was supported by Microsoft AI Security and CITES/IUCRC Grant No. 2113880. Opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Microsoft Incorporation or NSF.

References

1. Business Insurance. Ransomware victims paid $18 billion ransom in 2020. https://www.businessinsurance.com/article/20210429/STORY/912341506/Ransomware-victims-paid-$18-billion-ransom-in-2020, 2021.
2. PurpleSec Inc. 2021 Ransomware Statistics, Data, & Trends. https://purplesec.us/resources/cyber-security-statistics/ransomware/, 2021.
3. Sushruth Venkatesha, K Rahul Reddy, and BR Chandavarkar. Social engineering attacks during the covid-19 pandemic. *SN computer science*, 2(2):1–9, 2021.
4. Verizon Inc. Data Breach Investigation Report 2020. https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf, 2020.
5. Security Magazine Inc. Microsoft warns of Russian Nobelium phishing campaign. securitymagazine.com/articles/95328-microsoft-warns-of-russian-nobelium-phishing-campaign, 2021.
6. Gianluca Stringhini and Olivier Thonnard. That ain't you: Blocking spearphishing through behavioral modelling. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 78–97. Springer, 2015.
7. Tushar Bhardwaj, Tarun Kumar Sharma, and Manu Ram Pandit. Social engineering prevention by detecting malicious urls using artificial bee colony algorithm. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving: SocProS 2013, Volume 1*, pages 355–363. Springer, 2014.
8. Ajaya Neupane, Nitesh Saxena, Keya Kuruvilla, Michael Georgescu, and Rajesh K Kana. Neural signatures of user-centered security: An fmri study of phishing, and malware warnings. In *NDSS*, 2014.
9. Kavinga Yapa Abeywardana, Eckhard Pfluegel, and Martin J Tunnicliffe. A layered defense mechanism for a social engineering aware perimeter. In *2016 SAI computing conference (SAI)*, pages 1054–1062. IEEE, 2016.
10. Zikai Alex Wen, Zhiqiu Lin, Rowena Chen, and Erik Andersen. What. hack: engaging anti-phishing training through a role-playing phishing simulation game. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.
11. Let's Encrypt. Let's Encrypt Certificate Provider. https://letsencrypt.org/, 2024.
12. Ying Pan and Xuhua Ding. Anomaly based web phishing page detection. In *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pages 381–392, 2006.
13. Angelo P. E. Rosiello, Engin Kirda, Christopher Kruegel, and Fabrizio Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, pages 454–463, 2007.

14. Colin Whittaker, Brian Ryner, and Marria Nazif. Large-scale automatic classification of phishing pages. In *NDSS '10*, 2010.
15. Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Trans. Inf. Syst. Secur.*, 14(2), sep 2011.
16. Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. Exposure: Finding malicious domains using passive dns analysis. In *Ndss*, pages 1–17, 2011.
17. Jian Mao, Pei Li, Kun Li, Tao Wei, and Zhenkai Liang. Baitalarm: Detecting phishing sites using similarity in fundamental visual features. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 790–795, 2013.
18. Zheng Dong, Apu Kapadia, Jim Blythe, and L. Jean Camp. Beyond the lock icon: real-time detection of phishing websites using public key certificates. In *2015 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–12, 2015.
19. Calvin Ardi and John Heidemann. Auntietuna: Personalized content-based phishing detection. In *NDSS Usable Security Workshop (USEC)*, 2016.
20. Igino Corona, Battista Biggio, Matteo Contini, Luca Piras, Roberto Corda, Mauro Mereu, Guido Mureddu, Davide Ariu, and Fabio Roli. Deltaphish: Detecting phishing webpages in compromised websites. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security – ESORICS 2017*, pages 370–388, Cham, 2017. Springer International Publishing.
21. Routhu Srinivasa Rao and Alwyn R. Pais. Detecting phishing websites using automation of human behavior. In *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, CPSS '17, page 33–42, New York, NY, USA, 2017. Association for Computing Machinery.
22. Amirreza Niakanlahiji, Bei-Tseng Chu, and Ehab Al-Shaer. Phishmon: A machine learning framework for detecting phishing webpages. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 220–225, 2018.
23. Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. Hunting malicious tls certificates with deep neural networks. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, AISec '18, page 64–73, New York, NY, USA, 2018. Association for Computing Machinery.
24. Amin Kharraz, , William Robertson, and Engin Kirda. Surveylance: Automatically detecting online survey scams. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 70–86. IEEE, 2018.
25. Jian Mao, Jingdong Bian, Wenqian Tian, Shishi Zhu, Tao Wei, Aili Li, and Zhenkai Liang. Phishing page detection via learning classifiers from page layout feature. *EURASIP Journal on Wireless Communications and Networking*, 2019(1), 2019.
26. Manuel Sánchez-Paniagua, Eduardo Fidalgo, Enrique Alegre, and Rocío Alaiz-Rodríguez. Phishing websites detection using a novel multipurpose dataset and web technologies features. *Expert Systems with Applications*, 207:118010, 2022.
27. Marzieh Bitaab, Haehyun Cho, Adam Oest, Zhuoer Lyu, Wei Wang, Jorij Abraham, Ruoyu Wang, Tiffany Bao, Yan Shoshitaishvili, and Adam Doupé. Beyond phish: Toward detecting fraudulent e-commerce websites at scale. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2566–2583, 2023.
28. Yuxin Zhang, Xingyu Fu, Rong Yang, and Yangxi Li. Drsdetector: Detecting gambling websites by multi-level feature fusion. In *2023 IEEE Symposium on Computers and Communications (ISCC)*, pages 1441–1447, 2023.
29. Amin Kharraz, William Robertson, Davide Balzarotti, Leyla Bilge, and Engin Kirda. Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks.

In *Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 07 2015.
30. Amin Kharraz, Sajjad Arshad, Collin Mulliner, William Robertson, and Engin Kirda. UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware. In *25th USENIX Security Symposium*, 08 2016.
31. Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: Analyzing and detecting technical support scams. In *22nd Annual Network and Distributed System Security Symposium (NDSS 16*. NDSS, 2016.
32. Behzad Ousat, Mohammad Ali Tofighi, and Amin Kharraz. An end-to-end analysis of covid-themed scams in the wild. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, ASIA CCS '23, page 509–523, New York, NY, USA, 2023. Association for Computing Machinery.
33. Google. Chrome UX Report. https://developer.chrome.com/docs/crux/, 2023.
34. Zakir Durumeric. Cached chrome top million websites. https://github.com/zakird/crux-top-lists, 2023.
35. Google. Chrome Devtools Protocol. https://chromedevtools.github.io/devtools-protocol/, 2024.
36. Google. Chrome's Lighthouse. https://developer.chrome.com/docs/-lighthouse/overview/, 2024.
37. Wappalyzer. Technology Profiler. https://www.wappalyzer.com/, 2024.
38. Vincent Drury and Ulrike Meyer. Certified phishing: Taking a look at public key certificates of phishing websites. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 211–223, Santa Clara, CA, August 2019. USENIX Association.
39. Josh Aas, Richard Barnes, Benton Case, Zakir Durumeric, Peter Eckersley, Alan Flores-López, J. Alex Halderman, Jacob Hoffman-Andrews, James Kasten, Eric Rescorla, Seth Schoen, and Brad Warren. Let's encrypt: An automated certificate authority to encrypt the entire web. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2473–2487, New York, NY, USA, 2019. Association for Computing Machinery.
40. Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer. Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ARES '21, New York, NY, USA, 2021. Association for Computing Machinery.
41. Edward. Dom visualizer. https://0xedward.github.io/dom-visualizer/, 2024.
42. Omid Mirzaei, Roman Vasilenko, Engin Kirda, Long Lu, and Amin Kharraz. Scrutinizer: Detecting code reuse in malware via decompilation and machine learning. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2021.