From Attachments to SEO: Click Here to Learn More about Clickbait PDFs!

Giada Stivala
Cispa Helmholtz Center for Information Security
giada.stivala@cispa.de

Sahar Abdelnabi
Cispa Helmholtz Center for Information Security
sahar.abdelnabi@cispa.de

Andrea Mengascini
Cispa Helmholtz Center for Information Security
andrea.mengascini@cispa.de

Mariano Graziano
Cisco Talos
magrazia@cisco.com

Mario Fritz
Cispa Helmholtz Center for Information Security
fritz@cispa.de

Giancarlo Pellegrino
Cispa Helmholtz Center for Information Security
pellegrino@cispa.de

ABSTRACT

Clickbait PDFs are PDF documents that do not embed malware but trick victims into visiting malicious web pages leading to attacks like password theft or drive-by download. While recent reports indicate a surge of clickbait PDFs, prior works have largely neglected this new threat, considering PDFs only as accessories of email phishing campaigns.

This paper investigates the landscape of clickbait PDFs and presents the first systematic and comprehensive study of this phenomenon. Starting from a real-world dataset, we identify 44 clickbait PDF clusters via clustering and characterize them by looking at their volumetric, temporal, and visual features. Among these, we identify three large clusters covering 89% of the dataset, exhibiting significantly different volumetric and temporal properties compared to classical email phishing, and relying on web UI elements as visual baits. Finally, we look at the distribution vectors and show that clickbait PDFs are not only distributed via attachments but also via Search Engine Optimization attacks, placing clickbait PDFs outside the email distribution ecosystem.

Clickbait PDFs seem to be a lurking threat, not subjected to any form of content-based filtering or detection: AV scoring systems, like VirusTotal, rank them considerably low, creating a blind spot for organizations. While URL blocklists can help to prevent victims from visiting the attack web pages, we observe that they have a limited coverage.

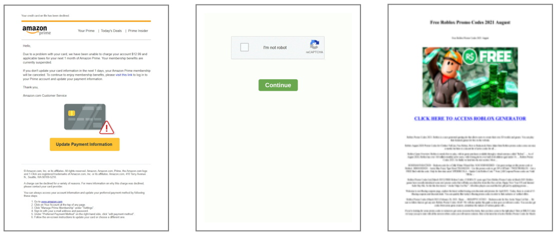
1 INTRODUCTION

Phishing emails are one of the major online threats [63], where the attacker sends fraudulent emails often attaching PDF files with embedded exploit code or malware [29, 30, 49, 58], which compromise victims' computers upon opening the attachments. Recent reports [38, 45] have shown another malicious use of PDF files, which stands out due to a surge in their numbers (estimated in the order of five million files only in 2020 [45]) and to the increased effectiveness of the deceitfulness of visual baits [38, 45]. Such PDFs, hereinafter clickbait PDFs, do not embed any malware or exploit code but only clickbait images which, when clicked, take the victim to an attack webpage stealing passwords, user identities, or compromising victims' computers via drive-by downloads [38, 45].

Although these reports show that large amounts of PDFs lead to attacks on the Web rather than installing malware, the scientific community has largely neglected the threat posed by clickbait PDF files and, to the best of our knowledge, did not investigate the role of PDFs outside classical email phishing attacks. Prior works have thoroughly explored classical phishing attacks, from empirical measurements of email phishing campaigns' number, volume and temporal dynamics (e.g., [49]), to studying the duration of phishing attacks (e.g., [44]), including the characteristics of their baits (e.g., [61]), and their effectiveness (e.g., [9, 48]). Such works only considered PDFs in the context of email phishing campaigns, however, it is unclear whether clickbait PDFs are part of them and, if so, to which extent. This paper aims to fill this knowledge gap by presenting the first comprehensive study centered on clickbait PDF files. We study this phenomenon and discuss its evolution, distinctive characteristics, and distribution channels, including their distribution as email attachments.

Our study. We start from a dataset of 176,208 PDFs by identifying and clustering PDFs that exhibit meaningful visual similarities. For this analysis, we prioritize content in the first page of the PDF, which, being displayed first to victim users, is most likely to embed an attack bait. When the first page also contains a URL, we verify its maliciousness by using a URL analysis service and by manual inspection. Having identified which clusters contain PDFs leading to attacks on the Web, we study their temporal and volumetric properties, as well as visual baits and geographical reach. Finally, the inspection of the structure and visual baits of PDFs leading to Web attacks leads us to hypothesize about two possible distribution vectors, namely email attachments and SEO attacks. We show that clickbait PDFs analogous to those in our dataset can be found on two search engines, and that online scoring services (i.e., VirusTotal) struggle in clearly separating benign from clickbait PDFs.

Findings. Overall, the main finding of our study is providing evidence that PDF files are no longer only ancillary tools of email phishing campaigns. Starting from a dataset of 176,208 PDF files—collected from Dec. 16th, 2020 to Jun. 23rd, 2021 by two industrial partners—we identified 44 out of a total of 80 clusters of clickbait PDFs whose documents lead to attacks like credential phishing and malware download. Among clickbait PDFs, we discovered three clusters with significantly different features than the rest, demonstrating the ongoing activity of a new kind of Web-based threat. These three clickbait PDF clusters are large in volume and persistent in time, accounting for 89% of the total dataset and lasting for the entire duration of our data collection. Also, they exhibit



Your credit card on file has been declined.

Hello,

Due to a problem with your card, we have been unable to charge your account $12.99 and applicable taxes for your next 1 month of Amazon Prime. Your membership benefits are currently suspended.

If you don't update your card information in the next 1 days, your Amazon Prime membership will be canceled. To continue to enjoy membership benefits, please visit this link to log in to your Prime account and update your payment information.

Thank you,

Amazon.com Customer Service

© Amazon.com, Inc. or its affiliates. All rights reserved. Amazon, Amazon.com, Prime, the Amazon.com logo and 1-Click are registered trademarks of Amazon.com, Inc. or its affiliates. Amazon.com, 410 Terry Avenue N., Seattle, WA 98109-5210.

A charge can be declined for a variety of reasons. For more information on why this charge was declined, please contact your card provider.

You can always access your account information and update your preferred payment method by following these steps:

1. Go to www.amazon.com
2. Click on Your Account at the top of any page.
3. Click "Manage Prime Membership" under "Settings".
4. Sign in with your e-mail address and password.
5. Under "Preferred Payment Method" on the right-hand side, click "edit payment method".
6. Follow the on-screen instructions to update your card or choose a different one.

Free Roblox Promo Codes 2021 August

Amazon message reCAPTCHA In-game currency

Figure 1: Examples of clickbait PDF files.

significantly different volumetric and temporal features when compared to email campaigns. Finally, this paper shows that, while many clickbait PDFs are distributed as email attachments, the three large clusters of clickbait PDFs are distributed via search engines, exploiting SEO attacks—a new insight placing almost all our files outside the email delivery ecosystem[1].

In this paper, we make the following contributions:

- We create and present the first large-scale, pre-labeled dataset of 176,208 clickbait PDFs, featuring 80 document categories.
- We identify 44 clusters out of 80 whose documents lead to Web attacks.
- We present the first characterization of clickbait PDF clusters, covering different aspects, i.e., volume, duration and activity, visual deceits, and targeted languages.
- We show that the vast majority of the documents in our dataset is distributed via SEO attacks, i.e., at least three clusters, covering 89% of the dataset, with a 60-day study of searching clickbait PDFs on Google and Bing.
- We release file hashes, file screenshots, class labels and URLs to the research community.

2 BACKGROUND AND METHODOLOGY

Before presenting our study, we define clickbait PDF attacks (§ 2.1) and outline our methodology (§ 2.2).

2.1 Background

Previous works discussed PDF files solely as a tool in email phishing attacks, where the deception was in the email body and the exploit occurred via the malicious code embedded in the attached PDF (hereinafter MalPDFs) [29, 30, 49, 58].

Unlike MalPDFs, clickbait PDF files do not embed malware nor do they contain exploits, but they are designed to trick victims into performing an action that can result in landing on malicious web pages that are stealing passwords or user identities, or compromising victims' computers via drive-by downloads [38, 45]. Clickbait PDFs rely on a wide variety of visual deceits to lure users into clicking on specific areas of the documents. Figure 1 shows a few examples of clickbait PDFs taken from our dataset, using classical phishing patterns, e.g., fake Amazon messages, as well as clickbait messages, e.g., in-game currrency generators.

2.2 Problem Statement and Methodology

The threat posed by clickbait PDFs has been object of concern by leading security teams in industry [38, 45]. Despite this anecdotal evidence, the scientific community has largely neglected the threat posed by clickbait PDFs. We follow a strict methodology, performing an array of analyses aimed at providing the first characterization of this phenomenon, based on measurable properties such as volume, activity and duration. We analyze visual baits and structure of clickbait PDFs looking for signs of diverse exploitation contexts and investigate distribution vectors used by attackers to reach their victims.

Achieving our overarching goal involves addressing both technical challenges and research questions. First, we tackle the technical challenge of analyzing PDFs at scale. The characterization of clickbait PDFs starts with the inspection of the PDFs that our partners receive daily. This daily procedure involves hundreds of documents and is expensive and inefficient, motivating the development of an assistive clustering module. We observe that clickbait PDFs contain remarkable visual similarities, which we leverage as a clustering feature to drastically reduce the number of PDFs to inspect manually. Identifying and enumerating such clusters is key to characterize both the general phenomenon and individual clusters.

We now turn to our first research question, which requires to *identify and characterize clickbait PDFs linked to malicious activity.* We extract all URLs from our PDFs, identifying *bait* URLs—URLs reachable by clicking on visual or textual baits in the first page—that might lead to malicious activity on the Web. We determine maliciousness through a third-party URL analysis service (i.e., VirusTotal) and confirm these results via manual inspection. Next, we focus on those clusters whose clickbait PDFs evidently lead to attacks on the Web and proceed with their characterization. Our analysis focuses first on measurable properties, such as cluster size, duration, activity and temporal dynamics (similarly to prior works, e.g., [22, 49, 61]) as well as their reach, by measuring the number and distribution of languages across and within clusters. Additionally, we discuss the visual baits of clickbait PDFs, searching for indications of attackers' reliance on different exploitation contexts other than the email distribution ecosystem (e.g., Web).

Then, we *investigate two possible distribution vectors.* Understanding the origin of clickbait PDFs is a key component in characterizing this phenomenon. Previous works only discussed PDFs as part of email phishing campaigns. We quantify how many clusters are distributed as email attachments by matching files on a corporate spam trap and by leveraging VirusTotal metadata. Beyond that, empirical observations on the structure of clickbait PDFs suggest another distribution mean: the documents of the three largest clusters share the common traits of Search Engine Optimization (SEO) attacks, i.e., keyword stuffing [43], cross-linking resources [68], and use of benign websites for linked resources [23]. We hypothesize that attackers poison search engine results to increase the visibility of these files to reach their victims. We verify this hypothesis by inspecting search results of popular search engines, such as Google and Bing [55], for 30 days.

[1] While working on this study, Microsoft warned (Tweet: https://twitter.com/MsftSecIntel/status/1403461397283950597) that the operators of the malware SolarMarker Jupyter are using PDF documents stuffed with SEO keywords to reach victims, further strengthening the importance of our study, indicating a change of distribution strategy.

3 DATASET AND CLUSTERS

Our analysis relies on a dataset of 176,208 PDF documents with unique SHA256 signature, collected from Dec. 16th, 2020 to Jun. 23rd, 2021. In this section, we describe the sources of data and data collection procedures (§ 3.1). Then, we report the procedure we followed to extract clusters of visually similar documents (§ 3.2).

3.1 Dataset

Data Sources. The sources of our dataset are two industrial partners, i.e., Cisco and InQuest Labs[2], who provided us with daily feeds of PDF files. Cisco started sending us data on Dec. 16th, 2020. To increase the diversity and coverage of the dataset, we introduced a second industrial partner, InQuest Labs, starting from Mar. 3rd, 2021. We were concerned that Cisco's sampling policy regarding the least number of AV flags (see § *Data Collection* below) might have introduced a bias towards documents with a higher number of AV flags. We sought to counter-balance this effect by including documents with lower AV scores, as a minimum threshold was not imposed by InQuest Labs. Figure 6 shows daily uploads aggregated per week until the end of this study, Jun. 23rd, 2021, highlighting the contribution of each partner; the respective areas are stacked to highlight the total weekly amount. The contribution of Cisco and InQuest Labs to the dataset is of 55% and 43%, respectively, with a negligible fraction of shared samples over the total, i.e., 0.02%.

Data Collection. Cisco retrieves data from VirusTotal [65] (VT), fetching PDF files uploaded on the previous day and flagged as malicious by at least nine antivirus (AV) engines by using search modifiers, a VT feature to filter files on properties such as file type, size, and the number of engines flagging the file as malicious. InQuest Labs receives feeds of malicious documents from multiple sources, one of which is VT, and shares with us those samples which are also confirmed from a second source. InQuest Labs retrieves samples from VT using selectors specified via YARA rules [46], a rule-based approach designed for the description of malicious files. InQuest Labs's rules search for unseen PDFs tagged as phishing, flagged as malicious by at least one AV engine, with encrypted PDF objects, or tagged with embedded JavaScript. The list of the rules used by InQuest Labs is publicly available [20]. We receive samples from InQuest Labs on the day they are uploaded on VirusTotal.

Data Preprocessing. At first, we rule out the possibility that our dataset contains PDFs with exploits or malicious JavaScript. We look for PDFs tagged by VT with `js-embedded`, `file-embedded`, `exploit`, `cve-xxxx`, and `launch-action`, which indicate the presence of exploit code or malware, and find that MalPDFs are a negligible fraction of our dataset (0.24% or 440 files).

3.2 PDF Clustering

The first challenge we address is grouping PDF documents using an appropriate similarity metric. As exhaustively inspecting all documents manually is not scalable, our goal is to implement a procedure for grouping documents whose content is visually similar, with the aim of using this by-product to speed up human inspection of the daily PDF feed. A common clustering approach for phishing messages relies on Natural Language Processing (NLP), where the similarity metric is calculated using the text in the message (e.g., [22, 49, 61]). However, PDF documents in our dataset do not exclusively rely on text to convey the fraudulent message, e.g., the fake reCAPTCHA documents, making it challenging for NLP-based clustering to produce meaningful clusters. Another approach to determine document similarity is by using raw document screenshots and supervised learning (e.g., [1]). Unfortunately, supervised learning techniques rely on a pre-existing labeled training set, which is unavailable in our case, making supervised learning unsuitable for our goal. We thus resort to unsupervised learning techniques to assist the identification of clusters of visually-similar PDF files.

Clustering. Previous work shows that replacing raw images with Convolutional Neural Networks (CNNs) features can lead to better clustering performance [17, 18]. Thus, we utilize the DeepCluster framework [5], a recent work in unsupervised representation learning, that jointly trains a CNN with k-means clustering. In each epoch, the training alternates between training the CNN and clustering and computing the pseudo-cluster-labels. We adopt the same DeepCluster setup (AlexNet architecture [27]) with mainly two changes: (i) We keep color information, as it can be a distinguishing factor; (ii) We decrease the number of clusters from 10,000 to 900, as we have a smaller dataset with a lower expected number of clusters.

We generate a raw screenshot of the first page of a PDF using `pdftoppm` [41] with 150 dots per inch (DPI) and obtain 176,208 screenshots. As a pre-processing step, we remove images with the same p-hash value (obtained from documents with different SHA256), lowering the number of samples to 20,671. Once we trained and ran DeepCluster on the screenshots with unique p-hash values, we validate the 900 clusters by randomly selecting 10 documents per cluster (9,000 samples in total) and determining the screenshot similarity considering text and image positions. As an output of this step, we identify 635 homogeneous clusters covering 18,557 (90%) of the input samples. This clustering step split large clusters into many smaller, fine-grained ones, therefore we merge homogeneous clusters containing similar documents. At the end of this step, we obtain 15 distinct clusters of documents.

To cluster similar documents in the remaining 2,114 (10%) samples, we run DBSCAN [12], using the learnt embeddings as distance metric (as in, e.g., [5]): we obtain 120 clusters and 1,135 noise points. We subsequently confirm that 87 clusters (610 samples) of the 120 are homogeneous and identify 29 new clusters obtained by merging similar homogeneous clusters. As a refinement step, we manually cluster the remaining 1,504 documents, discovering another 36 clusters, and group 389 spurious documents in the *Outliers* cluster. Table 5 (Appendix) reports the amount of documents involved at each clustering step. Finally, we assign each cluster an arbitrary name of our choice, with the only purpose of helping the authors remember the outlook of each of them, and redistribute the 155,535 samples that we filtered out by means of perceptual hash, assigning them to the cluster of their matching sample. The final number of PDF clusters observed in the dataset is 80, including *Outliers*. The interested reader can find more details on the clustering procedure and validation in Appendix A.

[2] Cisco is a global corporation in the field of networks, telecommunications, and security, with a number of employees in the order of tens of thousands. InQuest Labs is a SME in the field of packet inspection, network security, and threat intelligence.

4 ESTABLISHING MALICIOUSNESS

PDF documents, including clickbait PDFs, may contain URLs in any page. More importantly, clickbait PDFs exhibit the specific feature of embedding a URL leading to a Web attack in the first page. We use the presence of such *malicious* URLs as a discriminating factor to identify clickbait PDFs. In this section, we first present the extraction methodology for the URLs embedded in all 176,208 documents. Then, we identify PDFs linked to an ongoing malicious activity on the Web. Finally, we detail the observed attacks and motivate the soundness of our findings.

4.1 URL Extraction

Although trivial at a first glance, URL extraction from clickbait PDFs poses a few challenges. First, PDF files can contain encoded (e.g., base 64), compressed (e.g., deflate), or encrypted objects and streams, removing the string markers characterizing URLs, such as `http://`. Next, automated PDF generation from attackers may lead to corrupted or invalid permutations of the PDF structure where, e.g., URL-bearing PDF objects are disconnected from the PDF graph and thus not clickable, or they have a null clickable area. Below, we detail the URL extraction procedure, which ensures the extraction of clickable, well-formed first-page URLs (*bait* URLs) at scale.

We produce a normalized representation of each PDF file by removing any encoding or compression. Decrypting streams and objects was not possible because we did not have the encryption key. Then, we extract a graph-like representation of the normalized PDF with `peepdf` [11], a popular tool for analyzing malicious PDFs. We traverse the graph-like structure starting from the root element (the `Catalog` node) using a breadth-first algorithm to avoid loops, searching for those nodes containing links. Using regular expressions to extract URL-looking string text may increase the number of false positives. Accordingly, we leverage the semantic of the graph-like structure, searching for the PDF elements used to implement document areas that result in visiting a URL upon a mouse click. Such an area is implemented as a node containing a `URI` node having ancestors with the attribute `Subtype Link`, the attribute `Rect`, and either the `Type Annot` or `Type A` attribute. Further, we remove ill-formed URLs (e.g., the top-level domain is invalid, the URL network location is `127.0.0.1` or the URL scheme is not HTTP or HTTPS) and URLs pointing to static resources such as images or JSON files, which do not present a threat to users.

We verify that PDFs in our dataset are more likely to include links in the first page rather than in following pages by plotting the distribution of unique *bait URLs* per page, shown in red in Figure 2. We observe that 86% of all the extracted URLs are first-page URLs, covering 99% of the PDFs. This distribution confirms our intuition that first-page URLs are relevant features of our PDFs and that they are worth analyzing. First-page URLs, being displayed first to victim users, are more likely to lead to an attack. We thus discard URLs in pages after the first, obtain 157,623 unique URLs, and focus the next steps of our analysis on first-page bait links.

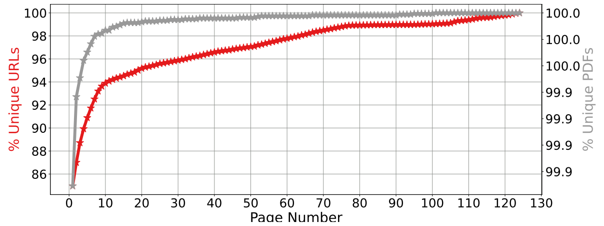


Figure 2: Distribution of *bait URLs* per PDF page (red) and number of unique PDFs embedding them (grey). The graph shows the .95 quantile of PDF pages (max: 524) for visibility reasons.

4.2 URL Analysis

After the extraction step, we determine which URL points to a malicious webpage. A common technique to determine the maliciousness of URLs is using URL blocklists, such as Google Safe Browsing (GSB) [16]. Blocklists like GSB intend to offer a live protection mechanism for browsers to warn users visiting a malicious website at the time of the visit. As a result, URLs that are no longer malicious or no longer exist are evicted from the blocklist, reducing our ability to determine maliciousness after a short period of time. We empirically observed that in some cases the time interval between the start of the malicious activity of a webpage and our reception of the PDF via VirusTotal is non-negligible, especially when considering web attacks such as phishing, whose malicious activities last on average 21 hours [44]. Such malicious bait links might already be offline or evicted from the blocklist by the time we look them up. A better option for our case study is using URL analysis services with historical data, e.g., VirusTotal or `urlscan`[60]. Thanks to Cisco's availability of 20K URL analysis requests on VT, we randomly sampled an equal number of URLs from each cluster, until either the entire cluster was covered or the cap was reached. To ensure validity of our approach, we inspected its coverage by cluster. Our sampling offers a high coverage, of 100% for all clusters except for 14, where we covered from 1.28% (or 1,000 files) of the *reCAPTCHA* cluster up to 99.69% (or 765 files) of the *NSFW 'Find'* cluster. Table 6 (Appendix) shows the coverage per cluster.

We also perform a manual inspection of 722 randomly-sampled first-page well-formed clickable URLs (*bait links*) to determine maliciousness. We label a URL as malicious if we observe any of the following behaviours: prompting file download, user interaction (click), asking for permissions, modifying the browser settings, leading to a phishing page, a Google SafeBrowsing warning, or to other types of unwanted content. Otherwise, we label the URL as benign.

4.3 Observed Malicious Activity

Cisco fetched a total of 19,935 distinct URL reports, where 89% of the URLs were unknown to VirusTotal, 7% were flagged as benign, and 4% (868) were flagged as malicious. The reasons behind this low number of URLs known to VT are unclear to us, and studying the AV inner workings goes beyond the scope of our research questions. We empirically observed that VT may have no knowledge of links embedded in PDFs even when one or more of its partner AVs flags the binary file as malicious. We discuss this observation in § 7.2. The 868 malicious URLs flagged by VirusTotal belong to 52 clusters. Our manual analysis validated both URLs that were labelled as malicious (32% of the manually-analyzed URLs) and URLs that were

Cluster Identifier	Attack Type	Volume	# Unique Phash	Avg/day	First seen	Last seen	% Active
reCAPTCHA	○	78,854	157	436	16.12.20	23.06.21	95.8%
ROBLOX Text	○	59,348	16,399	667	06.03.21	23.06.21	81.7%
ROBLOX Picture	○	18,065	192	278	05.03.21	23.06.21	59.1%
NSFW ‘Play’	∗	9,797	274	55	17.12.20	23.06.21	94.7%
reCAPTCHA Drive	○	1,693	15	18	12.02.21	23.06.21	73.3%
Download Torrent	○	1,121	112	18	15.02.21	23.06.21	48.4%
Ebooks	○	795	458	7	17.12.20	22.06.21	61.5%
NSFW ‘Find’	▼	322	45	4	20.01.21	20.06.21	58.3%
CLICK-HERE	○	286	58	3	09.03.21	21.06.21	81.7%
PDF Blurred	●	228	27	3	11.01.21	23.06.21	44.2%
Coin Generator	○	167	115	3	23.12.20	23.06.21	28.0%
Russian Forum	○	167	12	3	23.12.20	21.06.21	29.4%
AS PDF / File #1	●	134	17	2	24.12.20	22.06.21	40.0%
Elon Musk BTC	○	82	17	4	06.02.21	22.06.21	14.7%
Try Your Luck	▲	79	25	7	29.12.20	17.06.21	6.5%
Play Video	○	70	56	2	05.03.21	22.06.21	38.5%
Access Online Gen.	○	55	6	4	20.12.20	04.05.21	9.6%
NSFW ‘Click’	▲	44	15	3	12.02.21	02.06.21	11.8%
Lottery 25th Ann.	▲	43	23	2	19.01.21	28.05.21	20.2%
AS PDF / File #4	●	41	12	1	23.12.20	04.06.21	18.4%
Apple receipts	●	30	21	1	20.12.20	11.06.21	15.6%
Download Btn	○	19	19	1	19.12.20	26.05.21	11.4%
Fake SE	○	18	17	1	01.02.21	05.05.21	19.4%
Amazon scam	▼	14	11	1	20.01.21	11.06.21	8.5%
NSFW ‘Dating’	▼	14	13	5	17.04.21	07.06.21	5.9%
Download PDF	◇	13	13	1	14.02.21	17.06.21	10.6%
AS PDF / File #11	□+ ▼	11	6	1	03.02.21	08.06.21	8.8%
AS PDF / File #3	□	11	7	1	11.03.21	25.05.21	10.7%
Sigue Leyendo	⊠	10	7	1	27.02.21	03.06.21	10.4%
Web Notification	□	8	2	1	10.03.21	04.05.21	12.7%
Link farm	▲	7	6	2	17.01.21	04.04.21	5.2%
AS PDF / File #10	□	6	3	1	26.12.20	07.06.21	3.7%
AS PDF / File #8	□	6	4	1	25.03.21	14.04.21	25.0%
AS PDF / File #6	●	5	2	1	18.03.21	03.06.21	6.5%
Netflix scam	▲	5	2	3	21.12.20	23.12.20	100.0%
Get Your Files	●	4	2	1	10.03.21	17.03.21	42.9%
QR code	●	3	3	2	21.01.21	22.03.21	3.3%
Click Here TShirt	○	3	3	1	26.03.21	17.04.21	13.6%
Download File	○	3	3	2	03.09.21	21.05.21	2.7%
AS PDF / File #7	●	3	3	1	16.04.21	18.05.21	9.4%
AS PDF / File #13	▼	2	2	1	10.02.21	07.06.21	1.7%
Adobe Click	●	2	2	1	26.01.21	09.06.21	1.5%
SharePoint	●	2	2	1	04.05.21	02.06.21	6.9%
Shared Excel	▲	2	2	1	06.01.21	12.02.21	5.4%

Table 1: The 44 clusters associated with malicious activity. Clusters in italics were validated by manual inspection only. Dates are in `dd.mm.yy` format.

flagged as benign or never scanned (61%), and confirms 44 of the malicious clusters reported by VT. Conversely, we observed that URLs belonging to eight clusters were not malicious, containing documents about phishing training, generic text documents or ebooks, invoices, articles about security, reports by a security firm, flyers about events, or screenshots of a tool by Netcraft. Further details are reported in Appendix B. The manual analysis also flagged URLs, not flagged by VT and belonging to nine clusters, as malicious, and identified benign URLs belonging to five clusters.

Overall, the URL analysis returned eight different outcomes, reported in Table 1 and detailed in the following. *Malicious advertisement and Data harvesting (16 clusters, symbol: ○)*: in this attack, the user is redirected to a personalized advertisement page or is prompted to provide personal data to receive a reward (similarly to, e.g., [25]). *Google SafeBrowsing warnings (10 clusters, symbol: ●)*: GSB warned against either phishing or harmful content. *Malware (five clusters, symbol: □)*: the web page prompts to download a file (e.g., Office documents) or suggests to install additional software. We observed one cluster delivering multiple attacks and classified it accordingly. *Phishing (four clusters, symbol: ▼)*: these pages delivered classic phishing attacks. *VirusTotal (six clusters, symbol: ▲)*: the evidence of malicious activity was provided by VirusTotal results. *Various attacks (three clusters)*, which include: *Drugs promotion* (symbol: ⊠), where one cluster led to a blog promoting diet pills; *Fake search engine* (symbol: ◇), describing one cluster leading to a page pretending to be a search engine; *Adult content (symbol: ∗)*, describing one cluster leading to an adult website.

4.4 Summary of Findings

The goal of this section was to analyze representative URL samples for all the clusters obtained in § 3.2, investigating whether these URLs lead to Web attacks. In 44 clusters all analyzed active URLs led to an attack webpage, where the attack types are consistent. This pattern of homogeneity in attack types among the clusters suggests that they may be linked to malicious activity. Conversely, URLs from nine other clusters showed signs of malicious activity as well as of benign activity (at least one malicious and one benign URL). We excluded them from the rest of the analyses, as we conservatively select clusters linked to malicious activity only.

5 CLUSTERS CHARACTERIZATION

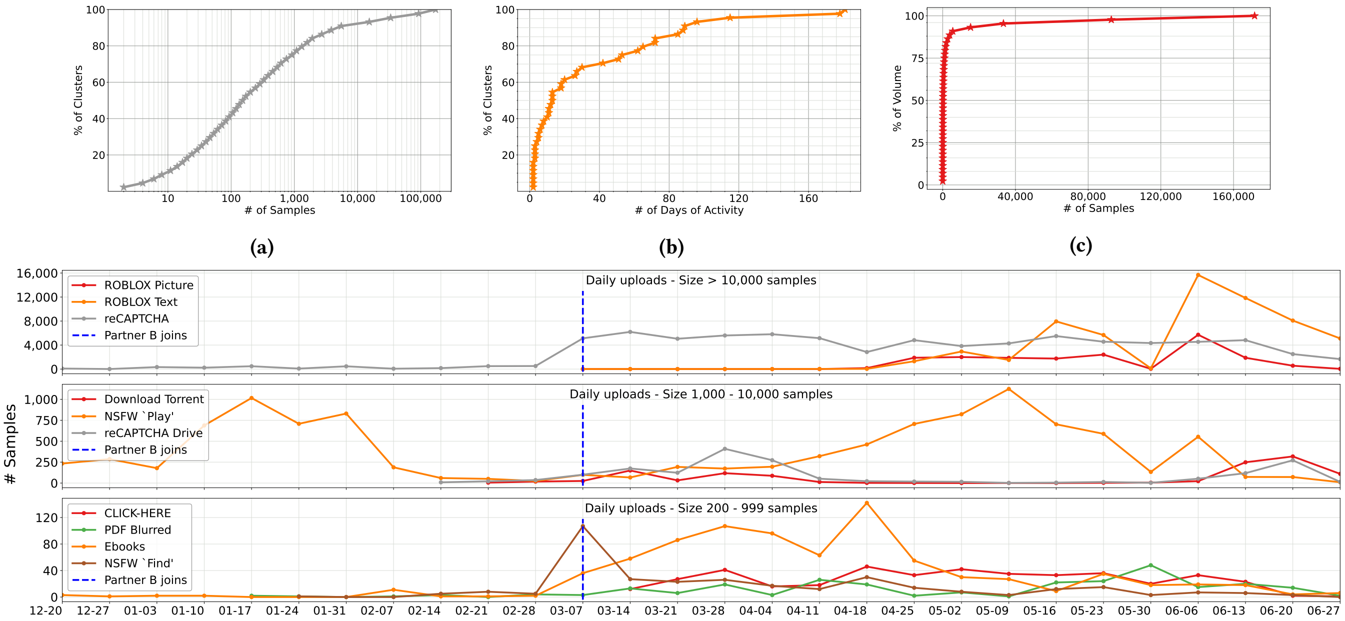
We now characterize each of the 44 clusters identified in § 4.3. First, we look at volumetric and temporal properties of each cluster (§ 5.1). Second, we analyze the visual deceits of each cluster (§ 5.2), providing a categorization of the type of fraudulent activities and their visual elements. Then, we explore the effectiveness of the VirusTotal maliciousness score (§ 5.3). Finally, we study the geographical reach of each cluster by observing the languages used in their text (§ 5.4).

5.1 Volumetric and Temporal Dynamics

Volume. Clickbait PDF files are not evenly distributed over the 44 malicious clusters. Cluster sizes are skewed, with the top 5% of malicious clusters (i.e., three clusters) corresponding to about 89% of the dataset, while 78% of the clusters contain fewer than 1,000 documents and 42% contain fewer than 100 (see Figure 3a and 3c).

Duration and Activity. The temporal dynamics of the clusters are diverse. For example, clusters like *reCAPTCHA* tend to be constant, without notable peaks. We speculate that the absence of patterns and peaks may indicate that their discovery and upload on VirusTotal may be automated. In contrast, other clusters, e.g., the two *ROBLOX* clusters, all clusters with sizes between 1,000 and 10,000 samples, and *NSFW 'Find'*, have a less regular evolution, indicating periods of low and high activity. Figure 3d shows the temporal dynamic of the clusters by number of daily uploads, grouped by the total size of the cluster (200 - 999 samples, 1,000 - 10,000 samples, and more than 10,000 samples).

We observe that most clusters are active for a period between one and two months, where specifically 28% of them are active for up to five days and 77% of them are active for at most 60 days (see Figure 3b). While few clusters operate for 60 days or more



(a)

(b)

(c)

(d) Clusters volumes grouped by cluster over time.

Figure 3: Cumulative Distribution Function of: (a) The volume of clickbait PDF documents over number of clusters. (b) The cluster activity in days over number of clusters. (c) The contribution of cluster volumes over the total dataset.

(11 clusters), their total size covers 99% of the entire dataset, with three clusters lasting more than 100 days (i.e., *reCAPTCHA*, *NSFW 'Play'* and *Ebooks*). These activity periods are considerably long, especially in comparison with email-based phishing campaigns, which last one day on average [49]. Table 1 shows size, prevalence, duration and temporal location for all the 44 malicious clusters.

5.2 Visual Deceits

Attackers use visual deceits to lure victims into clicking [3, 9]. We enumerated the types of visual baits and clickbait messages conveyed by the document text and identified two types of deceits. If a document includes logos, images or phrases reproducing existing entities (e.g., a company), processes (e.g., sharing of a document) or situations (e.g., receiving a money transfer), we categorize it as *Impersonation*. Otherwise, when a document entices the victim into clicking in order to obtain paid goods, illegal goods, or other unwanted content (e.g., adult content), we categorize it as *Promotion*. Also, we consider whether visual elements in clickbait PDFs may be similar to those found in different contexts. In particular, we look for PDFs resembling invoices, cloud or email notifications, and documents with UI elements used in web pages. The clusters distribute evenly between the two types of deceit.

Promotion. Promotion clusters can be further divided into four sub-clusters: in-game currencies or pirated content (15 clusters), material goods, e.g. electronic devices or money (two clusters), adult content (four clusters), and drugs (one cluster). With two large-size clusters, this deceit category covers 45% of the dataset.

The layout of these documents is usually not elaborate: 64% of them have a bare structure including an image for the advertised product, a catchphrase or bait (e.g., "Click here for free BTC") and a button, 18% are very text-heavy, employing techniques such as keyword stuffing and randomization, and five clusters show with varying levels of detail renown visual elements such as video players, hubs for content sharing, or threaded discussions.

Impersonation. The clusters in this category disguise their content as legit, mimicking existing commercial services, communications or people by means of typographic and visual elements, and ask to review the status of a process, access a shared document or prove their identity. In 17 cases documents reproduce parts of communications (e.g., emails from colleagues, friends or firms) or behaviors of viewer programs, prompting for valid credentials to access a protected file. In the remaining cases, the documents mimic established and widely recognized Web UI components or processes, like search engine results or CAPTCHA challenges by including key textual and graphical elements. For example, they display search results on the initial PDF page just like in a web browser, feature a reCAPTCHAv2 challenge image at the center of the first page, or show a browser popup requesting permissions. We note that attackers overlay large clickable areas around them. These UI elements are familiar and linked to authentic services, which operate by briefly halting user interaction with the page until they are removed with a click. Clusters displaying such visual baits likely exploit the notion that such an interruption is inconspicuous, as it aligns with typical behavior, and can be dismissed through a click. This characteristic of clickbait PDFs strikes a difference from conventional attack scenarios focused on attachments, opening up alternative possibilities such as employing the PDF as an intermediary step within a redirection chain.

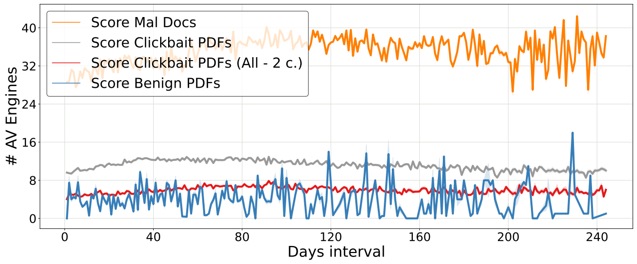


Figure 4: VirusTotal score comparison between MalDocs and clickbait PDFs. Data collection until Aug, 18th.

5.3 VirusTotal Score for Maliciousness

Prior studies on malware programs have relied on the VirusTotal scoring system, i.e., the number of AV engines flagging a sample, to select relevant samples to create a dataset. Recent studies [74] show that defining a threshold on the score for sample selection is challenging, mainly because the score of the same sample can change unpredictably over time. Figure 4 shows the variation of the VT score after x days following the upload date in four scenarios: (i) the score of malicious Microsoft Word (MS) documents with malware provided by our partners for this analysis; (ii) and (iii) the score of clickbait PDFs in our dataset, respectively with and without the two largest clusters; (iv) the score of PDFs in benign clusters. The data is collected as follows: every day d_i, we randomly select up to 500 files per provider–including malicious MS documents–from our dataset up to the day d_{i-1} and submit the selected hashes to VirusTotal to retrieve their VT score. Each file is selected only once.

We observe that documents in the two largest clusters, *reCAPTCHA* and *ROBLOX Text*, significantly influence the average score by increasing it to almost twice its value. Without considering these two clusters, the overlap between the scores of malicious and benign PDF documents is significant (a histogram of the scores is shown in Figure 8), making it more challenging to determine an appropriate threshold that could separate them. Finally, we note that, after 150 days, variance increases, most likely due to the fewer points for older documents.

5.4 Languages

We further investigate whether clusters target specific geographical areas by using language information obtained via the Google Vision API [15] when processing the first page of each document. We preferred this approach over the extraction of text from the PDF file itself, as the latter approach may lead to incomplete results due to the lack of text in embedded images. Google Vision processed 174,298 images, identifying in total 62 different languages, 15 of which with a high confidence threshold (0.90 or higher). Google Vision could not detect text in 678 documents and could not identify the language in 131 documents. Results are in Tables 2, 7 and 8.

All clusters

Lang.	Vol.	# of clusters
en	167,475	40
ru	1462	9
es	757	11
fr	211	5
pt-PT	154	5
de	74	10
it	64	5

(a)

reCAPTCHA only

Lang.	Vol.
en	77,425
es	567
fr	165
pt	137
id	96
it	51

(b)

Table 2: (a) Distribution of documents per language code (with # of clusters ≥ 5); (b) Distribution of languages in the *reCAPTCHA* cluster (with # of documents ≥ 50).

We observe that all large-size clusters are multi-regional, targeting users in different countries, and that languages are not evenly distributed across documents and clusters. English is by far the most common language, covering 95% of the dataset and 40 clusters, followed by Russian (0.8% and nine clusters) and Spanish (0.4% and 11 clusters). Small and medium-sized clusters tend to focus on one or two languages only (mostly English, 37 clusters, Russian and Spanish, eight clusters), except for *CLICK-HERE*, *NSFW 'Play Button'* and *Ebooks* which target, respectively, 17, nine and eight languages. When comparing with the distribution of languages on the Internet (see, i.e., [24]), we observe that highly-represented Internet languages are virtually not represented in our dataset: Chinese, the second most used language on the Internet, with about 19.4%, is absent from our malicious documents.

6 DISTRIBUTION VECTORS

In this section, we present two experiments to confirm the use of two distribution vectors. In § 6.1, we look at the VirusTotal tags of our files, and we search for our file hashes in a corporate spam trap to identify which clusters may be distributed as attachments. Then, in § 6.2, we go through search engine results looking for clickbait PDFs distributed via Search Engine Optimization (SEO) attacks.

6.1 PDFs as Attachments

Methodology. The ideal means to determine if our clickbait PDFs are attached to phishing emails is by using large phishing email datasets, e.g., the Gmail dataset used by Simoiu et al. [49], which is hard to get in practice, or subscribing to services specialized in malicious email feeds, e.g., MX Mail Data [8], which costs tens of thousands of dollars.

As email phishing campaigns target a large number of addresses at once [49], we speculate that spam traps might also contain phishing emails with attachments. Based on this observation, we asked our collaborator at Cisco to search for our file hashes inside their spam traps. Also, a closer look at the VT Public API reveals that VT users can upload samples and use the `attachment` and `email-spam` tags to indicate the source of the sample [64]. Accordingly, we use VT tags as an additional data source in this analysis.

Results. Table 3 shows the result of our experiments. The total number of matches in Cisco's spam trap is 106 for 57 unique PDF files, covering 11 clusters. Using a more conservative threshold of at least two matches per file, we have 68 matches for 19 files, covering seven clusters. Next, we look at VT tags and use the same data we collected in § 5.3, i.e., 106,062 files (60.19% of our dataset). In total, we found 65 files with the `attachment` tag and no files with the `email-spam` tag, covering eight clusters. Using the same conservative threshold (of two matches) as in the previous analysis,

	Spamtrap		attachment
	# hits	# PDFs	# PDFs
AS PDF / File #1	8	2	0
Shared Excel	3	1	0
Amazon scam	15	5	0
Apple receipts	5	4	0
PDF Blurred	8	4	10
Fake SE	16	1	0
NSFW 'Find'	6	6	2
NSFW 'Play'	42	31	9
Try Your Luck	1	1	22
NSFW 'Click'	0	0	18
Web Notification	0	0	2

Table 3: Clusters with at least two documents marked as `attachment` or found in a spamtrap by Cisco.

we count six different clusters. Overall, our analysis identified 11 clusters where at least one of the two methods identified at least two PDF files as attachments. Two of these 11 clusters are identified by both methods.

6.2 SEO Attacks

A closer look at the PDF documents of the three largest clusters (i.e., *reCAPTCHA*, *ROBLOX Text* and *ROBLOX Picture*, covering about 89% of our dataset) reveals that they share distinguishing characteristics with SEO attacks. The first characteristic is *keyword stuffing* [43], where the resource content is filled with keywords that are relevant to popular searches, ranking the page higher within search results for the included terms. We also observe that our PDF files use keywords that are related to the document titles. For example, the keywords used in a document with the title `Windows xp iso 32 bit file download` can be `Microsoft`, `ISO_Windows_XP_SP3`, and `crack`. The second characteristic is *cross-linking resources* [68], which exploits the link-based ranking algorithms of search engines. Attackers craft a network of ad-hoc resources and cross-link them to influence the ranking of target resources. A manual inspection of a sample of documents of the three main clusters revealed a consolidated structure of these PDFs, where the first page usually embeds one *bait* link, while the following pages include a list of URLs pointing to other PDFs of the same cluster. The third characteristic is the *use of benign websites* to host the cross-linked resources [23], as search engines tend to rank them more quickly than newly registered domains. We verified via GSB [16] that the URLs to these PDFs and the hosting website are not flagged as malicious.

Based on these three observations, we hypothesize that the three largest clusters are distributed via SEO attacks and perform a number of experiments to confirm our hypothesis. We verified that document types that are typically utilized in phishing attacks to infect victims' machines (e.g., [29, 30]) do not present the same SEO-oriented document structure by inspecting 225 MS Word, Excel and OLE2 documents, provided by Cisco. We first present the methodology we followed and then our findings.

Methodology. The goal of our experiments is to verify if victims can find clickbait PDFs belonging to the three largest clusters in our dataset via search queries on popular search engines. We use as search query the exact string of the document title since we aim at finding direct matches with the clickbait PDFs in our dataset. A challenge to the formulation of appropriate search queries is the popularity of the search terms. Search terms for poisoned search results usually have a lifespan of at most five days, with few exceptions (median: 19 days) [31]. Because titles extracted from VT clickbait PDFs might not be popular search terms anymore, or the PDFs corresponding to those search queries might have been taken down, we create effective queries with the title of fresh clickbait PDFs. The freshness property is ensured through daily selection of newly-uploaded clickbait PDFs from a new source, i.e., large PDF directories, which we discover by inspecting URLs in clickbait PDFs in the VirusTotal feed. Specifically, we observe that the URLs in clickbait PDFs in pages after the first, in the three largest clusters, point to `.pdf` files. Many of these URLs share the domain and path, suggesting the existence of large directories hosting cross-linked PDFs. We identify the precise URL of the directory starting from a link pointing to a `.pdf` file by, first, removing the file name and then, gradually, by removing URL path segments. This procedure identified 898,450 potential URLs of open directories. We verify that the directory index page exists and, if so, that it lists other PDF files hosted on the same directory. Then, we ensure that these PDFs are actually clickbait PDFs. We download each newly uploaded PDF and check if it contains a similar cross-link structure, i.e., if it contains at least 11 URLs, where 10 end with `.pdf` but the first one does not. The reason for this threshold is to include as many documents as possible (the average number of URLs ranges from 16 for *reCAPTCHA* to 30 for *ROBLOX Picture*). If the PDF file matches our criteria, we extract the title string by parsing the PDF structure. Appendix C provides additional details on our query search terms.

We monitor index pages daily recording new uploads of PDF files, observing a total of 13,012 PDF files from Dec. 1st, 2021 to Jan. 30th, 2022. In total, we found 426 index pages online during the whole duration of the analysis, with a few exceptional downtimes of 1-2 days. However, only 137 of them had new files uploaded during our study period. We point out that *we do not store any new PDF files on disk.* Instead, we perform the entire analysis in memory to minimize the risk of fetching documents that are not part of the three targeted clusters. We manually verified the accuracy of our heuristic by inspecting a daily sample of ten URLs to determine if the corresponding PDF files belong to the three clusters. We conclude that our heuristic is accurate and that all files belong to one of the three clusters.

Finally, we use the title string to search for PDF files via web APIs of search engines. In this experiment, we used the web APIs of the two most popular search engines, Google and Bing [55]. Each query returns the first top ten results, which we analyze in two ways to determine if an entry contains a PDF file belonging to one of the three clusters. First, we check if the result set contains the exact URL of the PDF file. Second, we download the PDF files, checking if they meet the cross-link structure criteria.

Results. Table 4 shows the number of matches obtained either by exact URL match or by examining the cross-link structure of PDFs. In total, we submitted 47,795 queries to each search engine, with differing results depending on the matching heuristic. In total, we successfully retrieved 3,469 documents via exact URL match and 6,947 via cross-link heuristic match, confirming our hypothesis that SEO attacks are used in practice. However, results vary across

Search engine	Type of match	Total	Daily Avg
Google	Exact match	0	0
Bing		3,469	59.81
Google	Cross-link heuristic	925	15.95
Bing		6,022	103.83

Table 4: Search engines results.

search engines. In general, we observe that finding these PDFs via Google search queries is more challenging than via Bing. In particular, we were not able to retrieve documents on Google via exact URL match, but only via the cross-link heuristic.

After confirming our hypothesis, we measure the effectiveness of SEO attacks, looking at the ranking of the query results. Figure 5 shows the weekly number of newly discovered PDFs and their result rank as a box plot. Overall, almost all clickbait PDFs are ranked high in the query results. Also, we notice a different behavior of Bing and Google, where the average position of PDF files is more stable and higher for Bing than for Google.

7 DISCUSSION

This study presents the first categorization of clickbait PDFs, including an analysis of their distribution vectors. In this section, we summarize our main findings, evaluate existing defenses, and discuss how to move forward. Finally, Appendix D further discusses possible limitations of our study.

7.1 Main Findings

The main finding of our study is providing sufficient evidence that clickbait PDFs are not just simple tools within phishing email campaigns. In fact, among clickbait PDFs, we discovered three clusters with unique features in terms of size, duration, and distribution means, indicating the rise of a new kind of web-based clickbait PDF attacks. Below, we present our main results.

Many Well-defined Clickbait PDF Clusters. Our study identifies 44 clickbait PDF clusters, covering nearly all documents in our dataset: 97% of the documents are part of a malicious cluster. Most of the clusters are small, with few notable exceptions, e.g., *reCAPTCHA*, *ROBLOX Text* and *ROBLOX Picture*, with 78k, 59k, and 18k files, respectively. Also, we found that large clusters tend to be more persistent, with daily uploads.

More Clusters Than Previous Study. When comparing our results to the Unit 42 blog post [45], our study found 39 additional clusters, including two large ones, i.e., *ROBLOX Text*, and *ROBLOX Picture*.

The Distribution Vector: SEO Attacks. Our study confirms that, just as for MalPDF, clickbait PDFs can be distributed as attachments, by finding files of 16 clusters in a corporate spam trap or flagged as malicious attachments by VirusTotal. However, our study also shows that the three largest clusters (i.e., *reCAPTCHA*, *ROBLOX Text*, and *ROBLOX Picture*), covering 89% of our dataset, are distributed via SEO attacks. As we observed, these attacks rely on cross-linked PDF files, requiring the generation of many files for the attack to be effective and explaining the large imbalance of sizes between the top three clusters and the others.

Clickbait PDFs Exploit the Web Context. Ten clusters include UI controls and visual signals commonly observed in webpages, e.g., reCAPTCHA, Google Drive search bar, threaded forum discussions, online repositories for files and torrents, and Web video players. The use of these elements suggests that attackers may expect victims to visualize these documents inside a browser, tricking them into interacting with these elements as with normal web pages.

7.2 Existing Defenses and Future Directions

We observed that clickbait PDFs distributed by SEO attacks represent a persistent threat for victim users. In this section, we consider existing in-browser defenses (i.e., blocklists) and evaluate the level of protection they offer against attacks delivered by clickbait PDFs. Our inspection shows that blocklists offer partial protection against clickbait PDFs, both in terms of the observed attacks and of the URLs known to the blocklist. We discuss possible roadblocks and future directions for research.

URL Blocklists. A quick evaluation of two popular protection systems, Google SafeBrowsing [16] and the rule-based ad-blocking provided by EasyList and EasyPrivacy [13] shows that blocklists offer partial protection against attacks conducted via clickbait PDFs, with a higher success for websites with malicious advertisements.

Google SafeBrowsing offers a lookup API returning the current blocklist status for a URL and does not provide historical records. However, VirusTotal includes GSB records of the last URL scan in its reports. We observed a low number of matches by using the reports fetched in § 4.2, where 155 of 868 URLs (18%) were blocklisted by GSB, with 22 labeled as *malicious* and 133 as *phishing*.

Ad-block based blocklists provide an additional defense to users by blocking requests to resources matching URLs or patterns in the blocklist. We logged all outgoing requests when loading the page as we manually inspected websites in § 4.2. Then, we retrieved EasyList and EasyPrivacy blocklists via the Wayback Machine [21], considering the closest available day to the processing date of the PDF file. By matching the collected URLs to the blocklists, we observed that 40% of the malicious URLs had at least one blocked request. These URLs mostly deliver malicious ads or lead to adult sites. We further inspected the impact, in terms of potential breakage, of blocked background requests and observed that 50% of these websites were affected, either not loading or stripped of their advertisements. While effective against malicious advertisement and data harvesting sites, ad-blockers fail to protect users against other attacks delivered by clickbait PDFs.

PDF Detection via Structural Features. We also evaluated the effectiveness of existing open-source state-of-the-art malicious PDF detectors [6, 53] in our context. Established techniques [51, 53] leverage the identification of groups of PDF objects (or "subtrees") that are common among malicious PDFs but absent in benign files, often embedding malicious code such as exploits or JavaScript. Recent advancements [6] offer flexibility in this similarity metric, allowing variations such as N differing PDF objects.

We evaluated Hidost's [53] ability in detecting malicious PDFs or identifying structural similarities among PDFs in the same cluster. We manually inspected graphical representations and raw PDF objects of sampled files, observing differences in the number, type, and

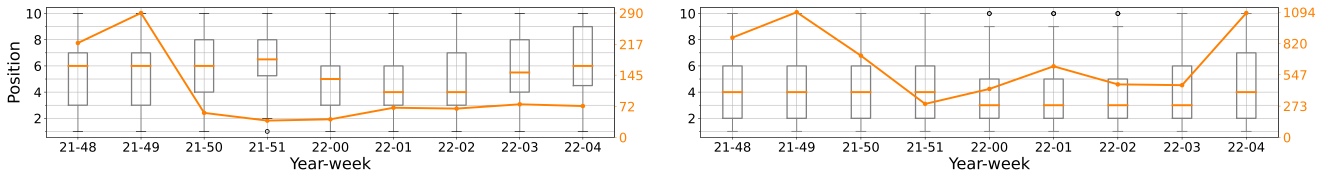


Figure 5: Number and position of PDFs found on Google (on the left) and Bing (on the right) over time.

connections of PDF objects across samples, despite visual similarities. Our analysis of the subtrees identified by the feature selection procedure revealed that they encode specific rendering instructions or metadata objects, which we deem to be a byproduct of the specific PDF generation tool. The feature selection algorithm likely did not identify representative subtrees encoding malicious functionality as MalPDFs are a negligible fraction of our dataset (see § 3.1). The detection result seemed to only loosely correlate with both features of the attack, i.e., the URL leading to malicious activity and the visual bait. This was evident in two ways: first, we could craft proof-of-concept clickbait PDFs with known URLs and identical visual bait that remained undetected. Second, it successfully identified shared subtrees in PDFs with different visual baits generated with the same tool. The improvements presented in [6] did not lead to better results, as they concern the similarity metric and not the feature selection. In conclusion, although existing methods such as [6, 53] effectively group PDFs based on structural similarities, they are not suited to our context, as the features of PDF structures lack the necessary discriminatory power to distinguish between benign and clickbait PDFs, or effectively differentiate clickbait PDFs belonging to different clusters.

Domain-Specific Detection Features. Our insights show that existing detection methods for MalPDFs are sub-optimal (see above), and also that existing commercial solutions lag behind (see § 5.3 and above). Nonetheless, our study highlights other distinctive features of clickbait PDFs that could be integrated into existing detection systems. For example, the three largest categories all include, in pages after the first, a large number of URLs pointing to similar clickbait PDF files, hosted on benign websites (see § 6.2). One solution could be the joint use of multiple indicators, such as the presence of cross-linked PDFs when they also exhibit visual similarity to known clickbait PDF clusters. This information could be used by, e.g., anti-phishing entities or search engines to either maliciously flag or reduce the rank of clickbait PDFs distributed via SEO attacks. A lower rank in search results could help reduce the number of victim users exposed to clickbait PDFs as result of queries containing poisoned search terms.

Coverage. Our findings in § 6.2 show the result of an ongoing malicious activity, where clickbait PDFs can be found on popular search engines when querying for specific popular keywords. We thus investigated if those PDFs had already been discovered by an anti-phishing entity and uploaded on VirusTotal by looking for SHA256 matches between the clickbait PDFs found on search engines (3,112 files) and those in our dataset. A total of 44 PDFs were already known to VT among those found on search engines, 17 of which were known to VT from 10 days to eight months prior. These empirical observations are in line with the findings presented in § 5.1, i.e., the activity of most clusters lasts for a long time, even extended to the online availability of single PDF files. Conversely, 27 PDFs observed in our search results later appeared in our partners' feeds, with an average delay of 22 days. The reasons for the limited overlap may lie in different concurrent causes, e.g., the PDFs were not flagged as malicious on their first submission or did not receive a 'phishing' label (a criterion of InQuest Labs). Alternatively, they may have been uploaded after the end of our data collection period.

Nonetheless, the crowdsourced nature of VT and the filtering rules employed by our partners may have introduced a source of bias in our data collection. We believe this bias may be evident in the amount of data, i.e., the size of this phenomenon may be bigger than our measurements report. Conversely, independent studies, like the one of Palo Alto Networks [45], report results similar to ours in terms of discovered clusters, which corroborates our findings.

Future Directions. Overall, we observed that the coverage of the phenomenon of clickbait PDFs is not exhaustive. This may be due to the combined medium of PDF binary and web page delivering the attack, and to the diverse nature of the attacks clickbait PDFs lead to. The low coverage of the inspected URL blocklists may be due to their incompleteness, given by the inability of ecosystem players to extract URLs from PDF files and feed them back to blocklists. In fact, the few URLs flagged as malicious (by GSB or VT) may be attributed to manual submissions. This shortcoming may result from the good reputation held by hosting providers, which can make blocklisting challenging. Nonetheless, a closer look at the autonomous system names hosting the 868 URLs flagged as malicious suggests the opposite, as they include popular providers such as Cloudflare, AWS, and Google Cloud Platform. This conflicting observation reaffirms the need for more research in this field to determine the role, reach and limitations of anti-phishing ecosystem players.

7.3 Data Sharing and Ethics

Two industrial partners provided the samples of our dataset. While we are not allowed to share the raw PDF files, we can publish the metadata of our dataset allowing researchers to reproduce and build on our results. We will share all file hashes of the PDF files (allowing to retrieve them from VirusTotal), PDF file screenshots, clustering labels and URLs. The data and supporting scripts can be found at https://www.kaggle.com/datasets/emerald101/from-attachments-to-seo.

This study did not involve human subjects, and we did not seek IRB involvement. However, we discuss a few ethical considerations of our study. One concern of our study is that VirusTotal files may contain private data. While VT allows the removal of private files, there is a possibility that they ended up in our dataset. Our manual evaluations exclude that clickbait PDFs (98.94% of the files) contain private information; still, the non-malicious ones might contain such information. Before releasing the dataset, we will manually inspect the remaining 1,862 benign PDFs, removing those with private information.

Another concern is that the SEO attack experiments may have downloaded files with private information. We addressed this concern at the design time, enforcing two strict rules: (i) we process PDF files only in memory, and (ii) we use our cross-link heuristic to guarantee that we store the metadata, e.g., URLs and file hash, only of those files fitting the heuristic. Finally, we retrieved contact points for those websites hosting direct clickbait PDF matches (observed in § 6.2) and raised awareness of the ongoing threat following the state of the art for vulnerability notifications [32, 57].

8 RELATED WORKS

We now review works closely related to our study.

Clickbait PDFs and MalPDFs. The closest study to ours is the non-peer-reviewed analysis [45] performed by Unit 42 of Palo Alto Networks. Unit 42's analysis indicated a surge of clickbait PDF files, illustrating the existence of five clusters and analyzing the landing pages of the *reCAPTCHA* one. In comparison, our study relies on a dataset that is dwarfed by the 5.2 million files of Unit 42 (about 0.033%). Nevertheless, we not only confirm the presence of the five clusters but discover 39 new malicious ones, including two large clusters, *ROBLOX Text* and *ROBLOX Picture*, which were not found by Unit 42. In addition, our results help build a better picture of the clickbait PDF ecosystem, showing that VirusTotal scores are of little help, as opposed to scores for malicious Office documents. Last but not least, our study tackles the question of distribution, confirming the use of attachments and showing the use of another distribution vector, i.e., SEO attacks.

The analysis of MalPDFs is also a research area close to our work. Several works (e.g.,[37, 51, 53]) proposed MalPDFs detection via machine learning, leveraging features derived from the internal structure of PDF documents, or relying on the analysis of embedded JavaScript [4, 28, 59]. Other works show how to evade existing classifiers (e.g., [4, 54, 70]) or how to improve their robustness (e.g., [6, 52]). Müller et al. [40] crafted MalPDFs exploiting caveats in the PDF specification, also without using JavaScript or exploit code. In our work, we do not focus on this type of documents, and we estimate their presence in our dataset to be very low.

Phishing Attacks. Many works tackled the detection of phishing messages and web pages, i.e., detection of malicious emails (e.g., [7, 10, 14, 19, 26]), URL and page content (e.g., [33, 67, 69, 73]), and passive DNS data (e.g., [2]). Recently, new ideas proposed using visual features of web pages to find phishing attacks (e.g., [1, 35, 36]). As opposed to these works, our paper does not present a technique to detect phishing attacks nor does it evaluate anti-phishing techniques. Our paper provides the first characterization of the threat posed by clickbait PDF files. Other studies focused on characterizing victims of phishing emails [49] and why they fall for phishing [3], on measuring the effectiveness of such campaigns [44], on visual features of malicious links in social networks (e.g., [56]), on victims' characteristics on social networks, such as gender, age, and country (e.g., [47]), and cognitive response to malicious emails (e.g., [61]).

Email Phishing Campaigns. Simoiu et al. [49] study phishing campaigns delivered by email and measure their volume and duration. In this respect, our works are related. The clickbait PDF clusters of this paper show significant differences compared with email-based campaigns in terms of volume, duration, and temporal features. Clickbait PDF clusters are lower in number and larger in size; they last longer and they do not happen in bursts, but they are rather constant or show less frequent and well-distanced peaks.

SEO Attacks. SEO attacks are used to game page rankings to expose users to a variety of Web attacks and campaigns (see, e.g., [23, 31, 66, 71]). Our work shares similarities with, e.g., [43, 62, 68], as clickbait PDFs also employ document cross-linking or keyword stuffing to game SE ranks. Even if an update in the Google Pagerank algorithm made keyword stuffing-based attacks largely ineffective [39], Liao et al. [34] demonstrated how malicious players can still find new ways into search results.

9 CONCLUSION

In this paper, we presented the first comprehensive study and categorization of clickbait PDFs, quantifying the threat posed by this kind of malicious PDF documents. We identified 44 clickbait PDF clusters in a real-world dataset of 176,208 PDFs and studied their volumetric and temporal properties. We observed large-size, long-lasting clusters, active for almost the entire duration of our study, and highlighted their difference with respect to email phishing clusters. Further, we studied the visual baits in clickbait PDFs and observed that several clusters include visual elements typical of web pages, e.g., fake reCAPTCHA buttons. In addition, we assessed the usefulness of online scoring systems such as the one provided by VirusTotal. Finally, we performed a series of experiments studying the distribution vectors used by attackers. Overall, our main finding consists in providing enough evidence that clickbait PDFs mainly spread through SEO attacks (89% of our dataset), while we observe their usage as part of email campaigns on a much lower scale. We publicly release the screenshot dataset, metadata and labeling performed during this study to foster new research on this subject.

ACKNOWLEDGMENTS

We thank Pedram Amini (InQuest) for providing the data of this study, and thank the InQuest team for their valuable feedback.

REFERENCES

[1] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. Visualphishnet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security.*

[2] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. 2011. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *NDSS.*

[3] Mark Blythe, Helen Petrie, and John A Clark. 2011. F for fake: four studies on how we fall for phish. In *Proceedings of the SIGCHI conference on human factors in computing systems.*

[4] Curtis Carmony, Xunchao Hu, Heng Yin, Abhishek Vasisht Bhaskar, and Mu Zhang. 2016. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors. In *NDSS.*

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV).*

[6] Yizheng Chen, Shiqi Wang, Dongdong She, and Suman Jana. 2020. On training robust PDF malware classifiers. In *29th USENIX Security Symposium*.

[7] Asaf Cidon, Lior Gavish, Itay Bleier, Nadia Korshun, Marco Schweighauser, and Alexey Tsitkin. 2019. High precision detection of business email compromise. In *28th USENIX Security Symposium*.

[8] MX Mail Data. [n.d.]. MXMAILDATA: Email Threat Data.

[9] Rachna Dhamija, J Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*.

[10] Sevtap Duman, Kubra Kalkan-Cakmakci, Manuel Egele, William Robertson, and Engin Kirda. 2016. Emailprofiler: Spearphishing filtering with header and stylometric features of emails. In *IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*.

[11] Jose Miguel Esparza. 2016. peepdf.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD)*.

[13] fanboy, MonztA, Famlam, Khrin. [n.d.]. EasyList. (01/22/2022).

[14] Ian Fette, Norman Sadeh, and Anthony Tomasic. 2007. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*.

[15] Google. 2022. Vision AI | Google Cloud. (01/22/2022).

[16] Google. 2022. Safe Browsing – Google Safe Browsing. (01/22/2022).

[17] Joris Guérin and Byron Boots. 2018. Improving Image Clustering With Multiple Pretrained CNN Feature Extractors. In *British Machine Vision Conference BMVC*.

[18] Joris Guérin, Olivier Gibaru, Stéphane Thiery, and Eric Nyiri. 2017. CNN features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700* (2017).

[19] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. 2017. Detecting credential spearphishing in enterprise settings. In *26th USENIX Security Symposium*.

[20] InQuest. [n.d.]. yara-rules.

[21] Internet Archive. [n.d.]. About the Internet Archive.

[22] Danesh Irani, Steve Webb, Jonathon Giffin, and Calton Pu. 2008. Evolutionary study of phishing. In *2008 eCrime Researchers Summit*.

[23] John P John, Fang Yu, Yinglian Xie, Arvind Krishnamurthy, and Martin Abadi. 2011. deSEO: Combating Search-Result Poisoning. In *USENIX security symposium*.

[24] Joseph Johnson. 2021. Most common languages used on the internet as of January 2020, by share of internet users.

[25] Amin Kharraz, William Robertson, and Engin Kirda. 2018. Surveylance: Automatically detecting online survey scams. In *2018 IEEE Symposium on Security and Privacy (SP)*.

[26] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. 2012. Enhancing phishing e-mail classifiers: A lexical url analysis approach. *International Journal for Information Security Research (IJISR)* (2012).

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012).

[28] Pavel Laskov and Nedim Šrndić. 2011. Static Detection of Malicious JavaScript-Bearing PDF Documents. In *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11)*.

[29] Stevens Le Blond, Cédric Gilbert, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and David R Choffnes. 2017. A Broad View of the Ecosystem of Socially Engineered Exploit Documents. In *NDSS*.

[30] Stevens Le Blond, Adina Uritesc, Cédric Gilbert, Zheng Leong Chua, Prateek Saxena, and Engin Kirda. 2014. A look at targeted attacks through the lense of an NGO. In *23rd USENIX Security Symposium*.

[31] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. 2014. A nearly four-year longitudinal study of search-engine poisoning. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*.

[32] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Michael Bailey, Damon McCoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *USENIX Security Symposium*.

[33] Bin Liang, Miaoqiang Su, Wei You, Wenchang Shi, and Gang Yang. 2016. Cracking classifiers for evasion: a case study on the google's phishing pages filter. In *Proceedings of the 25th International Conference on World Wide Web*.

[34] Xiaojing Liao, Chang Liu, Damon McCoy, Elaine Shi, Shuang Hao, and Raheem Beyah. 2016. Characterizing long-tail SEO spam on cloud web hosting services. In *Proceedings of the 25th International Conference on World Wide Web*.

[35] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. 2021. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In *30th USENIX Security Symposium*.

[36] Ruofan Liu, Yun Lin, Xianglin Yang, Siang Hwee Ng, Dinil Mon Divakaran, and Jin Song Dong. 2022. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In *31st USENIX Security Symposium*.

[37] Davide Maiorca, Giorgio Giacinto, and Igino Corona. 2012. A pattern recognition system for malicious pdf files detection. In *International workshop on machine learning and data mining in pattern recognition*. Springer.

[38] Microsoft Defender Security Research Team. 2017. Phishers unleash simple but effective social engineering techniques using PDF attachments.

[39] Tyler Moore, Nektarios Leontiadis, and Nicolas Christin. 2011. Fashion crimes: trending-term exploitation on the web. In *Proceedings of the 18th ACM conference on Computer and communications security*.

[40] Jens Müller, Dominik Noss, Christian Mainka, Vladislav Mladenov, and Jörg Schwenk. 2021. Processing Dangerous Paths. In *NDSS*.

[41] Derek Noonburg and Albert Astals. 2021. Poppler, a PDF rendering library.

[42] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749* (2021).

[43] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*.

[44] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *29th USENIX Security Symposium*.

[45] Palo Alto Networks Unit 42. 2020. 2020 Phishing Trends With PDF Files.

[46] Yara Project. [n.d.]. YARA: The pattern matching swiss knife for malware researchers (and everyone else).

[47] Elissa M Redmiles, Neha Chachra, and Brian Waismeyer. 2018. Examining the demand for spam: Who clicks?. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*.

[48] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. 2010. Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI conference on human factors in computing systems*.

[49] Camelia Simoiu, Ali Zand, Kurt Thomas, and Elie Bursztein. 2020. Who is targeted by email-based phishing and malware? Measuring factors that differentiate risk. In *Proceedings of the ACM Internet Measurement Conference*.

[50] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*.

[51] Charles Smutz and Angelos Stavrou. 2012. Malicious PDF detection using metadata and structural features. In *Proceedings of the 28th annual computer security applications conference*.

[52] Charles Smutz and Angelos Stavrou. 2016. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In *NDSS*.

[53] Nedim Šrndic and Pavel Laskov. 2013. Detection of malicious pdf files based on hierarchical document structure. In *Proceedings of the 20th Annual Network & Distributed System Security Symposium*.

[54] Nedim Šrndic and Pavel Laskov. 2014. Practical evasion of a learning-based classifier: A case study. In *2014 IEEE symposium on security and privacy*.

[55] Statcounter. 2022. Search Engine Market Share Worldwide | Statcounter Global Stats.

[56] Giada Stivala and Giancarlo Pellegrino. 2020. Deceptive previews: A study of the link preview trustworthiness in social platforms. In *NDSS*.

[57] Ben Stock, Giancarlo Pellegrino, Frank Li, Michael Backes, and Christian Rossow. 2018. Didn't you hear me?—Towards more successful web vulnerability notifications. (2018).

[58] Kurt Thomas, Danny Huang, David Wang, Elie Bursztein, Chris Grier, Thomas J Holt, Christopher Kruegel, Damon McCoy, Stefan Savage, and Giovanni Vigna. 2015. Framing dependencies introduced by underground commoditization. (2015).

[59] Zacharias Tzermias, Giorgos Sykiotakis, Michalis Polychronakis, and Evangelos P Markatos. 2011. Combining static and dynamic analysis for the detection of malicious documents. In *Proceedings of the Fourth European Workshop on System Security*.

[60] urlscan. 2022. urlscan.io. (01/22/2022).

[61] Amber Van Der Heijden and Luca Allodi. 2019. Cognitive triaging of phishing attacks. In *28th USENIX Security Symposium*.

[62] Tom Van Goethem, Najmeh Miramirkhani, Wouter Joosen, and Nick Nikiforakis. [n.d.]. Purchased Fame: Exploring the Ecosystem of Private Blog Networks. In *Proceedings of the 2019 ACM Asia CCS*.

[63] Verizon Inc. 2021. 2021 Data Breach Investigations Report.

[64] VirusTotal. 2022. File search modifiers – VirusTotal. (01/22/2022).

[65] VirusTotal. 2022. VirusTotal - Home. (01/22/2022).

[66] David Y Wang, Stefan Savage, and Geoffrey M Voelker. 2013. Juice: A Longitudinal Study of an SEO Botnet. In *NDSS*.

[67] Colin Whittaker, Brian Ryner, and Marria Nazif. 2010. Large-scale automatic classification of phishing pages. In *NDSS*.

[68] Baoning Wu and Brian D. Davison. 2005. Identifying link farm spam pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*.

[69] Guang Xiang, Jason Hong, Carolyn P Rose, and Lorrie Cranor. 2011. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)* (2011).

[70] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically evading classifiers. In *NDSS*.

[71] Ronghai Yang, Xianbo Wang, Cheng Chi, Dawei Wang, Jiawei He, Siming Pang, and Wing Cheong Lau. 2021. Scalable Detection of Promotional Website Defacements in Black Hat SEO Campaigns.

[72] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, et al. 2021. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In *Proceedings of the IEEE Symposium on Security and Privacy.*

[73] Yue Zhang, Jason I Hong, and Lorrie F Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web.*

[74] Shuofei Zhu, Jianjun Shi, Limin Yang, Boqin Qin, Ziyi Zhang, Linhai Song, and Gang Wang. 2020. Measuring and modeling the label dynamics of online anti-malware engines. In *29th USENIX Security Symposium.*

A PDF Clustering

In this section, we expand on the procedure used to cluster visually similar documents, described in § 3.2. First, we report the evaluation on the embeddings returned by DeepCluster [5]. Then, we explain which parameters were used to run DBSCAN [12] and why. Finally, we report on our validation of the obtained clusters.

DeepCluster Embeddings Evaluation. As the dataset does not have ground-truth labels, we evaluate the embeddings by visually inspecting the nearest neighbours (using L_2 distances) for a small subset; the top nearest neighbours for a document should ideally be visually similar. Our evaluation further covers three criteria: (i) Outliers (documents which lack any similarity to others) should have larger closest distances compared to documents that have similar counterparts (intra-cluster). (ii) There should be a consistent distance threshold beyond which the samples are no longer similar to the query (a “cluster-flipping” point). (iii) The number of similar images returned before the “flipping” point should be as high as possible. We use the results of the k-means clustering to inspect and select representative samples from different visual clusters (51 samples) and from outliers (48 samples of documents that were not grouped with similar ones). We compare three methods that can be used as a metric: the trained DeepCluster network, pre-trained VGG [50], and perceptual hashing. The first row in Figure 7 shows the comparison between outliers and intra-cluster documents. As observed, the trained DeepCluster has a better separation between the two cases, followed by VGG, while perceptual hashing has a very poor one. For intra-cluster documents, the second row shows the distance thresholds at which the clusters flipped, in comparison with the smallest distances. Ideally, there should be enough separation on average between the flipping points and the closest distances in order to select thresholds that allow the formation of clusters. This is, again, better accomplished by DeepCluster rather than by VGG and hardly accomplished by perceptual hashing. These three images also show the count of correct images retrieved at the flipping point: while VGG can retrieve more samples than DeepCluster, the threshold distances for VGG are less consistent, making it less useful for further clustering. On the other hand, the retrieved samples in the case of perceptual hashing are relatively few.

Overall, this analysis shows that the embeddings obtained by training DeepCluster are more useful in terms of nearest neighbours analysis than perceptual hashing and off-the-shelf pre-trained CNNs. It also gives insights into the possible distance thresholds that could be used in a next clustering step.

Parameters of DBSCAN. The first parameter needed to run DBSCAN is representative of the minimum number of samples in a cluster. To be able to capture even very small clusters, we select a minimum number of samples per cluster of 3, leveraging the insights gained during the previous step of manual inspection. When selecting a distance threshold for DBSCAN, we keep into consideration the insights observed during the clustering procedure (see Figure 7(b)). The last intra-cluster sample is located at a relative distance of 70, although several outliers are already present at this threshold. We keep a conservative approach to reduce the number of outliers as well as to maximize the number of correctly classified instances and choose a distance threshold of 50. This procedure returned 120 clusters and 458 noise points, identifying nine new clusters, however, including 68% non-homogeneous clusters. We therefore finally used a lower distance threshold, i.e. 35, where most of the samples in the “Closest” group are located. In this case, DBSCAN outputted 120 clusters and 1,135 noise points. We manually validate clusters’ coherence by inspecting all samples, obtaining 87 homogeneous clusters, for a total of 610 documents. It is important to note that while the overall clustering procedure we followed might involve some tuning steps to select the parameters, it drastically reduces the time to label all samples individually and identify all clusters in the dataset.

Clusters Validation. We estimated the overall clustering effectiveness by selecting at most 20 random PDF files for each of the 80 clusters[3], collecting 1,071 samples, and checking for labeling errors. The fraction of mislabeled samples is 3.27%, which, in perspective, is about half of the error in popular datasets, e.g., 6% of ImageNet[42].

B False Positives in Maliciousness Validation

In this section, we examine the conflicts that arose when the manual validation procedure did not confirm a ‘malicious’ label in VirusTotal reports. In particular, we examine those clusters where not only no malicious activity was observed, but also the visual content lacked any form of deceit. These clusters are: *Book cover*, *Document Layout*, *Invoice-like*, *AS PDF / File #12*, *Boletín de Noticias*, *Excel tables*, *Informative Flyer*, *Netcraft.*

No Sign of Malicious Activity. Four clusters, i.e., *AS PDF / File #12*, *Boletín de Noticias*, *Excel tables*, *Netcraft* show no sign of malicious activity. The first cluster groups PDFs designed for phishing training (e.g., within a company) and specifically crafted to be flagged by AVs. In fact, clicking the link embedded in the PDFs leads to a webpage hosted by the organizing company, which reveals that the document was a test and includes educational content on phishing. The second and third clusters include links to security-related resources, as they promote educational material. Similarly, PDFs in the fourth cluster include rich-text dumps of the URL-scanning tool from the security company Netcraft, reporting on malicious sites.

Outliers Flagged as Malicious. Three clusters, i.e., *Book cover*, *Document Layout*, *Informative Flyer* include documents whose URLs have been correctly flagged by VirusTotal. Upon manual inspection,

[3] Clusters can contain fewer than 20 files.

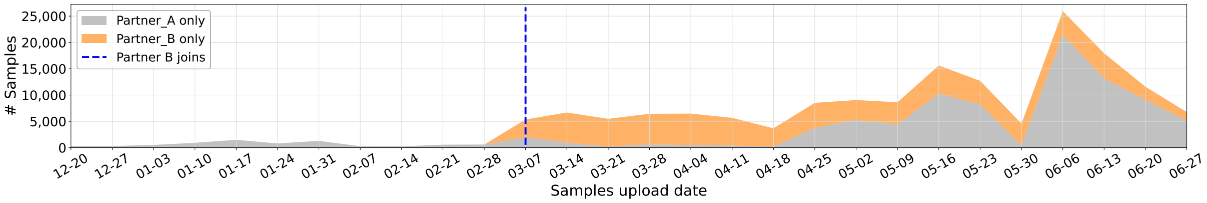


Figure 6: Weekly sum of daily uploads of the two datasets, stacked.

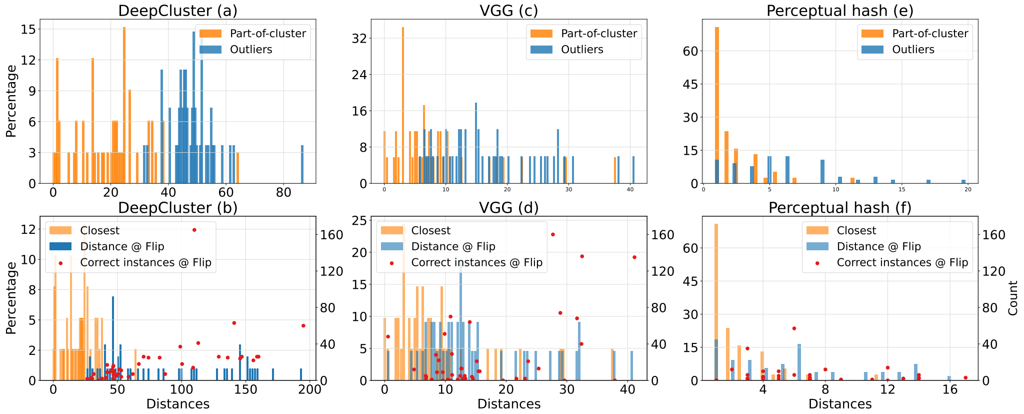


Figure 7: Comparing: (above) closest distances for outliers and 'part-of-cluster' samples, (below) closest distances of 'intra-cluster' samples and distances at cluster-flipping points. Scatter plots display correct instances (same cluster) at flipping points.

Step	Input Size	Clusters Clust. No.	Docs. No.	Incr.	New clusters Largest Campaing Name	Rank	Smallest Campaing Name	Rank
SHA256 dedup.	185,575	-	-	0	-	-	-	-
phash dedup.	176,208	-	-	0	-	-	-	-
k-means	20,671	635	18,557	+15	reCAPTCHA	1	Fake SE	35
DBSCAN	2,114	87	610	+29	reCAPTCHA Drive	6	Download File	47
Full-manual	1,504	-	-	+36	AS PDF / File #12	10	Shared Excel	48
Outliers	389	-	-	0	-	-	-	-

Table 5: PDF cluster identification: overview of the input/output properties of each step.

we verified that the cluster label of these documents is not correct—in other words, they are improperly assigned to these clusters and, as such, do not contribute to making the cluster a malicious cluster.

One URL Flagged as Malicious. In two cases, i.e., *Invoice-like* and *Document Layout*, one URL per cluster was flagged. Upon manual inspection, the URL in *Document Layout* appeared to be flagged by Google SafeBrowsing, while the URL in *Invoice-like* pointed to the main page of a hosting provider. We speculate the reason for this may be that these clusters aggregate a few documents with larger intra-cluster distances, which alternatively could have been split in sub-clusters or moved to the *Outliers* cluster, as the manual validation procedure did not raise any flag.

	URL Coverage		Document Coverage	
AS PDF / File #1	286	100.00%	285	99.65%
Book cover	252	94.59%	248	98.41%
Document Layout	322	100.00%	320	99.38%
Download File	3	66.67%	2	66.67%
PDF Blurred	274	100.00%	273	99.64%
Ebooks	789	97.98%	765	96.96%
NSFW 'Find'	397	99.69%	396	99.75%
NSFW 'Play'	9,783	49.37%	4,827	49.34%
Netcraft	298	100.00%	281	94.30%
ROBLOX Picture	12,497	14.04%	1,829	14.64%
ROBLOX Text	36,919	9.59%	2,120	5.74%
Crawler trap	4,917	99.35%	1,738	35.35%
reCAPTCHA	77,988	1.28%	1,000	1.28%
reCAPTCHA Drive	1,692	34.79%	589	34.81%

Table 6: Number of bait URLs submitted to VT and respective number of PDFs. Missing clusters have 100% coverage.

Regional clusters	Vol.	Lang.
reCAPTCHA Drive	1,693	en
Download Torrent	1,120	ru
AS PDF / File #1	134	en
Access Online Gen.	55	en
Lottery 25th Ann.	43	ru
AS PDF / File #4	41	en
Apple receipts	30	en
NSFW ‘Dating’	14	en
AS PDF / File #11	11	en
AS PDF / File #3	11	en

Table 7: Clusters targeting one language (Vol. $>$ 10 docs).

Multi-regional clusters	Vol.	# Lang.s
reCAPTCHA	78,852	52
CLICK-HERE	286	17
NSFW ‘Play’	9,126	9
ROBLOX Text	59,345	9
Ebooks	795	8
ROBLOX Picture	18,065	6
Download Btn	19	5
PDF Blurred	228	3
AS PDF / File #8	6	3
Play Video	70	3
Download PDF	13	3
Coin Generator	167	2
Amazon scam	14	2
Elon Musk BTC	82	2
Web Notification	8	2
Try Your Luck	79	2
Russian Forum	167	2
Fake SE	18	2
NSFW ‘Click’	44	2
NSFW ‘Find’	322	2
Sigue Leyendo	10	2
AS PDF / File #6	5	2

Table 8: Multi-regional clusters.

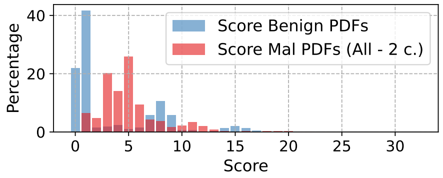


Figure 8: Histograms of the VT scores of benign and clickbait PDFs, without the two largest clusters, for the first 30 days.

C Search Engine Queries

Our queries use 15,436 individual keywords. The most frequent keywords are English words, and among the top five we have `pdf` (9,270), `free` (3,732), `guide` (2,233), `template` (1,822), and `manual` (1,740). When looking at their effectiveness, `pdf` is used to find 2,036 new documents, followed by `answers` (356), `free` (332), `guide` (316), and `movie` (288). We also look at the frequency distribution of query bigrams, with the top five most effective words being `answer key` (156 files), `pdf free` (116 files), `how to` (109 files), `full movie` (89 files) and `edition pdf` (80 files).

D Limitations

This study should be considered alongside certain limitations. Due to an accidental cap limiting the number of PDFs in their feed, Cisco sent us a maximum of 300 samples per day, until March 3rd, when this cap was removed. Until then, the dataset accounted for 7,787 unique samples, i.e., 4.41% of the entire dataset, affecting 30 of the clusters leading to attack pages. While this may influence the size of the clusters, it did not prevent us from observing clusters with samples linking to malicious activity. Among them, four clusters (*Netflix scam*, *Shared Excel*, *Download Btn*, *AS PDF / File #13*, *Adobe Click*, *AS PDF / File #10*, *Apple receipts*) saw a contribution of 50% or higher of their entire volume, and, three clusters entirely take place before March 3rd. Moreover, the *reCAPTCHA* cluster has received +2,897 samples, which is a marginal increase when considering the size of this cluster. Similarly, the *NSFW ‘Play’* cluster has seen a contribution of +4,262 samples, corresponding to 44% of its volume. The remaining clusters received a very limited number of samples, on average 16 samples each. Nevertheless, including the data points before March 3rd gave us the invaluable opportunity to place the starting date of each cluster at a much earlier point in time (45 days on average).

Before implementing our clustering procedure, we evaluated a series of possibilities. Using URLs as an additional feature may have improved accuracy, but two main challenges make this inadequate in practice. First, as the same cluster uses different URLs, URL string matching would have resulted in clusters that are too fragmented. Second, using maliciousness scores from online services is also a weak signal for clustering. § 4.3 shows that URL analysis services are incomplete, making them more suitable for determining the maliciousness of a cluster via random sampling rather than a feature for clustering. Finally, visiting all landing pages and detecting attacks requires tackling non-trivial challenges, e.g., bypassing client-side cloaking and detecting malicious pages. CrawlPhish [72], by Zhang et al., tackles both challenges. Unfortunately, this tool is not available in practice[4], making it challenging to analyze URLs at scale for our purpose.

[4] The authors could not share the code with us because it relies on a third-party component that they are not authorized to share.