Saarland University
Department of Computer Science

Deceptive Attacks in Modern Web

Dissertation
zur Erlangung des Grades
der Doktorin der Ingenieurwissenschaften
der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

von
Giada Martina Stivala

Saarbrücken, 2025

Tag des Kolloquiums:	18. Dezember 2025
Dekan:	Prof. Dr. Roland Speicher

Prüfungsausschuss:

Vorsitzender:	Prof. Dr. Martina Maggio
Berichterstattende:	Dr. Giancarlo Pellegrino
	Prof. Dr. Christian Rossow
Akademischer Mitarbeiter:	Dr. Yuqing Yang

Zusammenfassung

Täuschung im Web bleibt eine anhaltende Bedrohung, da Angreifer*innen zunehmend nicht nur psychologische und visuelle Manipulationen nutzen, sondern auch die technologische Komplexität moderner Plattformen. Trotz wachsender Aufmerksamkeit für Phishing und Schadsoftware besteht nur ein begrenztes Verständnis dafür, wie neue Technologien neuartige, schwerer zu erkennende Formen der Täuschung ermöglichen–was sowohl die Erkennung als auch die Abwehr erschwert. In dieser Arbeit wird untersucht, wie aktuelle Entwicklungen im Web, insbesondere Browser-Funktionalitäten und soziale Plattformen, für täuschungsbasierte Angriffe ausgenutzt werden können, und die operativen Herausforderungen bei deren Eindämmung werden analysiert.

Zunächst werden Link-Vorschauen auf 20 sozialen Plattformen untersucht und aufgezeigt, wie Inkonsistenzen bei der Generierung von Vorschauen und das Fehlen von Schutzmechanismen es Angreifer*innen ermöglichen, harmlos wirkende Vorschauen für schädliche Links zu erstellen und zu verbreiten. Anschließend werden Clickbait-PDF-Dokumente analysiert, die Webinhalte nachahmen und zu Webangriffen führen. Anhand von 176.208 realen Beispielen wird festgestellt, dass Angreifer*innen hauptsächlich auf den Missbrauch von SEO setzen und dass die meisten Dateien nicht von Antivirensoftware erkannt werden. Um zu verstehen, wie Angreifer*innen solche Kampagnen aufrechterhalten, werden 4.648.939 PDF-Links auf 177.835 Domains überwacht und systematischer Missbrauch von Infrastruktur aufgedeckt. Abschließend zeigen Interviews mit 24 Hosting-Anbietern, dass Missbrauchsbekämpfung oft nachrangig behandelt wird – nicht aufgrund technischer Beschränkungen, sondern wegen unklarer Zuständigkeiten, Kostenüberlegungen und der Wahrnehmung, dass kompromittierte Kundeninstanzen ein geringes Risiko darstellen.

Abstract

Deception on the Web remains a persistent threat, as attackers increasingly exploit not only psychological and visual manipulation but also the technological complexity of modern platforms. Despite growing attention to phishing and malware, there is limited understanding of how emerging technologies enable new, harder-to-detect forms of deception–posing challenges for both detection and remediation. In this thesis, we examine how recent web advancements, specifically browser capabilities and social platforms, can be exploited to mount deception-based attacks, and analyze the operational challenges in mitigating them.

We first study link previews on 20 social platforms, uncovering how inconsistencies in preview generation and a lack of safeguards allow attackers to craft and distribute benign-looking previews for malicious links. We then investigate clickbait PDFs, documents mimicking web content and leading to web attacks. Analyzing 176 208 real-world samples, we find that attackers primarily rely on SEO abuse and that most files evade antivirus detection. To understand how attackers maintain such campaigns, we monitor 4 648 939 PDF links hosted on 177 835 domains, revealing systemic infrastructure abuse. Finally, interviews with 24 hosting providers show that abuse remediation is often deprioritized, not due to technical limits but to misaligned business responsibilities, cost considerations, and the perceived lack of risk posed by compromised customer instances.

Acknowledgments

Looking back on the past few years, this PhD journey has been an incredible experience filled with both professional growth and personal development. It has been a time of learning, collaboration, and discovery. I am deeply grateful to all the people who supported me along the way, both inside and outside the academic world.

First of all, thank you to my advisor Giancarlo Pellegrino for always believing in me and giving me this amazing opportunity. Thanks also to my previous advisors, Rebecca Montanari and Marco Prandini, for encouraging me to pursue this path in security and showing me that opportunities are there for those who want to pursue them.

This journey would not have been possible without my family, who consistently supported me through the ups and downs of the PhD, and who patiently adapted holiday plans that always shifted because of "the next deadline".

A big thank you to my awesome lab mates: Soheil Khodayari, Aleksei Stafeev, Andrea Mengascini, Gianluca De Stefano, Xuenan Zhang, Jakub Pružinec, and Yuqing Yang for the insightful discussions, unwavering support, everyday coffee machine chats, and enjoyable after-work meetups. I'm also glad I had the chance to connect with our visiting researchers, interns, and student assistants: Anthony Gavazzi, Lorenzo Cazzaro, Yigit Sever, Angelo Sotgiu, Raoul Scholtes, and Prerak Mittal.

Many thanks to my co-authors Sahar Abdelnabi, Mario Fritz, Pedram Amini, and Mariano Graziano for their valuable input and effort in the long process behind two of the papers in this dissertation. Thank you to Rafael Mrowczynski and Maria Hellenthal for introducing me to the world of interviews and qualitative studies, and for your guidance and helpful feedback throughout the project. Thanks also to Mika Meyer for trusting me with his master's thesis, and for the hard work that led to a conference paper. A big thank you to Andrea Mengascini, Gianluca De Stefano, and Soheil Khodayari for their ongoing support as colleagues and co-authors, and to Bhupendra Acharya for the collaboration and shared enthusiasm for phishing-related research.

My time at CISPA also allowed me to connect with and learn from many other talented researchers. Thanks to Matthias Fassl, Lea Gröber, Carolyn Guthoff, Alexander Ponticello, Divyanshu Bhardwaj, Simon Anell, and Dañiel Gerhardt for all the feedback on usable security and ethics in research involving human participants, as well as the spontaneous and thought-provoking discussions.

Thanks to Marius Steffens, Aurore Fass, Sebastian Roth, Shubham Agarwal, Jannis Rautenstrauch, and Florian Hantke, for always knowing the nitty-gritty details of browser and JavaScript security, for your insights, and for the reality checks during the COVID period. Also, thank you to Abdullah Alhamdan and Masud Bhuiyan for the many engaging and spontaneous conversations.

I'm also incredibly grateful for the support and advice of faculty members, group leaders, and postdocs who were always available to listen and help: Cas Cremers, Katharina Krombholz, Ben Stock, Jonas Hielscher, Ninja Marnau, Bhupendra Acharya, Cristian Staicu, Maximilian Golla, and Sascha Fahl.

A heartfelt thank you goes to all my friends, who made this time more enjoyable. To the "Italian family": Andrea Mengascini, Gianluca De Stefano, Riccardo Zanotto, Giacomo Santato, Eugenio Paracucchi, Matteo Leonelli, Valentino Dalla Valle, as well as Antonio Cinà and Lorenzo Cazzaro. To my first office-mates Daniel Frassinelli and

Sohyeon Park, thank you for helping me settle in at the beginning of this journey. To the dear ones outside the academia bubble: Riccardo, Antea, Andy, Darja, Elisabetta, Lorenzo, Jacopo, Samuele, Dora, Maria, Anna, Sarah, Lisa, Ariel, Francesca, Marina, Giulia, Marc, Valeria, thank you for listening to the ins and outs of paper writing, and for all the fun moments together.

I'm also grateful to have spent time with Silvia Sebastián, Aurora Naska, Yu De Lin, Elia Geretto, Sanam Lyastani, Faezeh Nasrabadi, Hamed Rasifard, Anne Müller, Stella Wohnig, Keno Hassler, Addison Crump, Daniele Antonioli, Yann Bourreau, Jesko Dujmovic, and Lukas Gerlach, for all the fun activities and great conversations, at work, at conferences, and beyond.

Finally, a sincere thank you to the CISPA administrative staff, not only for their support in making this PhD possible, but also for the friendly and much-appreciated coffee machine chats: among them, Georg Demme, Sibylle-Annette Dümont, Tobias Ebelshäuser, Andrea Haas, Thorsten Helfer, Wolfgang Herget, Sebastian Klöckner, Felix Koltermann, Markus Kötter, Avian Krämer, Ecem Kus, Simon Lenau, Michael Schilling, Charlotte Schwedes, Annabelle Theobald, Sebastian Weisgerber, and Maximilian Wolf.

Last but not least: thank you to Riccardo Zanotto and Shubham Agarwal for proofreading this thesis, and to Anna, Andy, and Antea for their input on the German abstract.

To all of you: thank you. This six-year Saarbrücken journey would not have been the same without your support, encouragement, and companionship.

Contents

List of Figures

List of Tables

1 Introduction

The Web is a powerful tool for sharing and accessing information, facilitating the creation of websites and blogs, enabling fast connections with other users, and creating opportunities for businesses and consumers. This same interconnectedness and accessibility unfortunately also benefit malicious actors, who exploit the ease of content creation to deceive users, launch attacks, compromise victims' machines, or steal sensitive information [150]. Deception-based attacks have a long history, with one of the earliest examples being emails impersonating banks or trusted agents [148]. Despite advances in security measures, these attacks continue to cause significant financial losses, amounting to almost 19 million dollars in 2023 alone [65].

Deception techniques come in many forms, combining psychological and technical strategies to manipulate users. These can range from visual tricks, like imitating logos and design elements of trusted entities [2, 134, 136, 4], to psychological cues such as urgency or authority [240, 265, 161]. More recently, deception has evolved to include contextual tailoring, where message content, communication channels, or file attachments are crafted to closely match the target's environment [22, 20]. The diversity of such techniques is not only a result of the various tricks attackers use to deceive victims, but also of their evolution over time, driven by factors such as advancements in security defenses (e.g., evasion via cloaking [249, 264]), new functionalities of technologies (e.g., exploiting modern web browsers [254, 151]), shifts in user behavior (e.g., leveraging social media interactions and content sharing [68]), and increased user awareness of traditional phishing tactics (e.g., highly tailored deception strategies [87, 11]). This research is thus motivated by the need to systematically explore how technological advancements also create opportunities for attackers to develop more deceptive and harder-to-detect threats. Understanding these developments is essential for designing more effective countermeasures and reducing the growing impact of deception-based web threats.

Among the many technological advancements that have shaped the modern Web, two stand out for their impact on how users interact with online content: the increasing sophistication of browser environments and the widespread adoption of social media platforms. Web browsers, for instance, have transformed from simple HTML viewers into powerful, feature-rich environments, with capabilities such as built-in document rendering. We examine whether these added features enable new forms of deception, focusing on PDF rendering as a case study. In particular, we observe that malicious PDFs can be crafted to visually mimic legitimate web content, exploiting the trust users place in the browser interface. Similarly, the rise of social media has brought about new dynamics in how information is shared and consumed, with users now accessing more content through social media referrals than through traditional search engines [9]. Content shared on social media takes the form of link previews, that offer a quick visual summary of external webpages. While intended to aid comprehension and engagement, these previews can be manipulated to misrepresent the true nature of a linked page, deceiving users before they even choose to click. To investigate *whether new integrations, technological developments, or communication platforms can be exploited to mount deception-based attacks (RQ1)*, we examine the cases of misleading link previews [P1] and deceptive PDFs [P2]. We refer to the latter as clickbait PDFs, highlighting a class of malicious documents designed to attract clicks and mislead users through visual,

textual, or contextual manipulation. Both cases serve as emblematic examples of how evolving web technologies can be used to craft subtle and effective forms of deception.

Focusing solely on how technology creates new opportunities for deception-based attacks only provides a partial view of the broader problem. While technological advancements play a significant role, they do not fully account for how such attacks are carried out in practice. A more comprehensive analysis must also take into account other enabling factors, such as the web infrastructure that attackers exploit and the defensive measures available to mitigate these threats. To investigate *how attackers leverage web infrastructure to support deception-based attacks (RQ2)*, we focus on the case of clickbait PDFs, where the broad availability of websites and web services is critical not only to the success of the attack but also to its large-scale distribution.

The persistent availability of web infrastructure to attackers over extended periods raises substantial questions about the role and responsiveness of those responsible for maintaining the infrastructure–such as hosting providers–whose hosting space is being abused. Security researchers routinely notify identified abuse or detected vulnerabilities to affected entities. However, these notifications often result in little to no remediation [127, 216, 30, 215], a pattern we also observed in the case of clickbait PDF campaigns [P3]. Despite the formalization of notification practices [215, 127, 31, 237, 30, 140, 141, 216, 23], it remains unclear what actions are actually taken after a report is received, or what factors shape the decision to act. To investigate *how vulnerability notifications are processed within hosting provider organizations (RQ3)*, we examine the internal characteristics that may influence their response and overall handling of reported abuse.

1.1 Contribution

This thesis contributes by showing how attackers leverage web-based technological innovations to develop novel deception-based attacks, and by providing a comprehensive analysis and systematization of these phenomena. It also examines the operational aspects of these attacks and highlights the shortcomings in the vulnerability remediation process, which ultimately benefit attackers.

We start by analyzing new forms of deception (RQ1) from the perspective of victim users, through two case studies: deceptive previews on social platforms and clickbait PDFs in web browsers. The second case study serves as a foundation for a broader investigation into additional, yet equally important, aspects of web-based attack campaigns. First, we examine the hosting infrastructure that attackers rely on to carry out these attacks (RQ2). Then, we analyze the remediation process, focusing on vulnerability notification as an approach to removing malicious content, and investigate the obstacles that hinder its effectiveness (RQ3).

1.1.1 RQ1: New Forms of Deception

Our first research question investigates *whether new integrations, technological developments, or communication platforms can be exploited to mount deception-based attacks.* To explore this, we adopt a case study approach, examining two distinct scenarios. The

first case illustrates a potential future vector for abuse, while the second focuses on a current, real-world threat.

The first case study focuses on social networks and instant messaging applications, which we collectively refer to as social platforms. Here, we investigate the misuse of link preview functionality—that is, the ability to create benign-looking previews that point to harmful or deceptive content. This case study highlights how seemingly innocuous features in communication platforms can introduce new attack surfaces.

The second case study concerns clickbait PDFs, PDF documents that exploit the seamless rendering functionality of modern web browsers. These documents embed user interface elements typical of the web context and deceive users into clicking, by appearing similar to regular webpages. In this study, we analyze the visual baits, the properties of clickbait PDF distribution, and the delivered attacks.

Deceptive Malicious Link Distribution in Social Platforms The first case study [P1] refers to the misuse of link preview functionality in social platforms, i.e., the creation of benign-looking previews for harmful links. Link previews are a key factor in users' decision-making when choosing whether to click on a link, to the point where Facebook took action [188] by restricting link preview edits to prevent the spread of fake content.

While previous research has examined spam campaigns on social networks [219, 81], the accounts distributing harmful content [230, 52], and the characteristics of users more likely to engage with such links [186], our study shifts the focus to link previews, and the connection they establish between users and the actual landing pages. We dissect and systematize the link preview creation procedure in 20 social platforms, observing how adversaries can create benign-looking previews for malicious links, and platforms' mechanisms against the distribution of malicious content.

Our findings reveal several critical issues, from design creation processes hiding security indicators (which may deceive even tech-savvy users) to the lack of countermeasures against the distribution of malicious links. These issues are not technical flaws limited to a few platforms but represent a systematic problem in how previews are designed across all platforms in our study. We conclude by reflecting on our findings and outlining actionable recommendations, ranging from short-term technical fixes to the establishment of standardized rules for link preview design and generation processes.

Deceptive Malicious Link Distribution via Clickbait PDFs In the second case study [P2], our research objective is to comprehensively investigate the phenomenon of clickbait PDFs, starting from visual and textual baits (reCAPTCHA logos, forum discussions) to the PDF distribution context (search engine results) which strenghtens the credibility of deception. Previous research has primarily focused on malware-bearing PDFs, for example in the context of email phishing campaigns, investigating malicious payloads [120, 121, 229, 205] or attack volume and temporal trends [205]. However, anecdotal evidence preceding the beginning of our investigation reported the existence of PDFs that do not embed malware but lead to Web attacks [176, 147, 1]. This method of delivering attacks is still common at the time of writing [5]. Despite these observations, a comprehensive scientific study on the risks posed by clickbait PDFs had yet to be conducted.

To bridge this gap, we decompose the problem of studying the clickbait PDF phenomenon into three key steps. First, we conduct a large-scale analysis of 176 208 real-world clickbait PDFs, examining their temporal and volumetric properties, visual bait techniques, and geographic reach. Second, through a systematic study, we uncover that clickbait PDFs are mostly distributed via search engine optimization (SEO) attacks (89% of our dataset), employing tactics such as keyword stuffing, backlinking, and hosting on reputable domains. Finally, we observe that commercial anti-viruses struggle in detecting clickbait PDFs, creating a blind spot for organizations.

While the use of PDFs for deception has traditionally been associated with email phishing [205, 229], our findings reveal a broader trend: PDFs are no longer merely auxiliary tools in phishing emails, but also spread via poisoned search engine results, exploiting the web context to enhance their effectiveness, and exhibit distinct patterns of volume and persistence over time.

1.1.2 RQ2: Web infrastructure supporting attacks: the case of Clickbait PDFs

The results of our clickbait PDF study lead us to our second research question on *how attackers leverage web infrastructure to support deception-based attacks.* Building on our initial findings, we investigate the enabling factors behind large-scale attack campaigns, focusing on the role of websites and web hosting services in facilitating the distribution of clickbait PDFs.

The distribution of clickbait PDFs, and thus the effectiveness of the linked attack pages, depends on the large daily uploads of interlinked PDFs to legitimate hosting services, making these hosts critical to the persistence and impact of the threat. Prior research has examined infrastructure-related properties for threats such as drive-by download [155], phishing pages [170], and spam [133], as well as the role of compromised servers in attacks [170]. However, these findings do not directly apply to clickbait PDFs, as they address threats with different operational characteristics or focus on infrastructure in a limited scope.

This research addresses three key questions, including the identification of exploited hosting services, attackers' file upload capabilities, duration of abuse, and reaction of affected parties, through a large-scale measurement study involving 4 648 939 clickbait PDFs hosted on 177 835 hosts. Through this seventeen-months analysis, we identify three distinct types of abused hosting services, with each entity participating in the distribution of clickbait PDFs for an average of nine months. The methods attackers use to upload these files vary, making mitigation efforts challenging. We alerted affected parties of the ongoing abuse observing a moderate but positive initial response; however, clickbait PDF uploads on endpoints belonging to notified parties persisted over time, underscoring the persistent nature of the problem.

1.1.3 RQ3: Roadblocks in Web Vulnerability Remediation

Our third and final research question investigates *how vulnerability notifications are processed within hosting provider organizations.* Obtaining significant remediation rates

when notifying web vulnerabilities is a known open issue [215, 216, 127, 31, 237, 30, 23], which we observed exists for clickbait PDFs as well [P3].

To investigate the root causes of persistent remediation failures, we conducted a qualitative study involving IT professionals working at hosting providers. Our goal was to understand how vulnerability notifications are processed and how various factors, including the characteristics of the notification message and the hosting provider's operational structure, influence the outcomes. We interviewed 24 operators from eleven countries, representing organizations of varying sizes that offer services ranging from shared hosting to complex enterprise solutions.

Our findings show that while most providers are aware of and routinely handle VNs, remediation is often deprioritized due to clearly defined service boundaries, operational cost considerations, and the perception that responsibility lies with customers. Although reachability issues persist in some multi-layered hosting setups, most providers can be contacted through established channels. Ultimately, our results suggest that the limited effectiveness of VNs is not due to technical or procedural obstacles, but to a misalignment between the goals of security researchers and the business logic of hosting providers.

1.2 Corresponding Publications and Tools

This dissertation is based on the following papers:

[P1] Stivala, G. and Pellegrino, G. Deceptive previews: A study of the link preview trustworthiness in social platforms. In: *Network and Distributed System Security Symposium (NDSS)*. 2020.

[P2] Stivala, G., Abdelnabi, S., Mengascini, A., Graziano, M., Fritz, M., and Pellegrino, G. From Attachments to SEO: Click Here to Learn More about Clickbait PDFs. In: *Proceedings of the 39th Annual Computer Security Applications Conference*. 2023.

[P3] Stivala, G., De Stefano, G., Mengascini, A., Graziano, M., and Pellegrino, G. Uncovering the Role of Support Infrastructure in Clickbait PDF Campaigns. In: *IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. 2024.

[P4] Stivala, G., Mrowczynski, R., Hellenthal, M., and Pellegrino, G. Behind the Curtain: How Shared Hosting Providers Respond to Vulnerability Notifications. In: *Under submission.* 2026.

In addition to the first-author papers, the author of this thesis co-authored two other papers [S1, S2]:

[S1] Beluri, M., Acharya, B., Khodayari, S., Stivala, G., Pellegrino, G., and Holz, T. Exploration of the Dynamics of Buy and Sale of Social Media Accounts. In: *Proceedings of the 2025 ACM on Internet Measurement Conference (IMC)*. 2025.

[S2] Stivala, G., Meyer, M., and Pellegrino, G. An Analysis of Malicious File Distribution in Free Hosting Providers. In: *Under submission.* 2025.

The author of this dissertation contributed to papers [P1, P2, P3, P4, S2] as the leader and main author. Below is a detailed description of the contributions of the co-authors for each paper.

In [P1], Giancarlo Pellegrino provided guidance during the execution and contributed to the writing of the paper in his role as the author's supervisor.

In [P2], Sahar Abdelnabi was responsible for developing the machine learning model used for clustering and wrote the corresponding section of the paper, and Mario Fritz contributed with inputs and guidance during our clustering-related meetings. Andrea Mengascini performed the SEO experiments and gave concrete inputs for the corresponding paper section. Giancarlo Pellegrino provided guidance during the execution and contributed to the writing of the paper in his role as the author's supervisor.

In [P3], Gianluca De Stefano developed the pipeline fetching URLs and the new clustering module, and gave concrete input for the corresponding text in the paper. Andrea Mengascini performed the experiments verifying that the SEO campaign was still ongoing during the study period. Giancarlo Pellegrino gave feedback on the idea and advised us along the way to submission.

In [P4], Rafael Mrowczynski and Maria Hellenthal gave substantial feedback concerning the methods, participated in part of the interviews, and gave guidance on data analysis. Both contributed to paper writing by providing specific sections and with overall feedback and improvements. Giancarlo Pellegrino gave feedback on the idea and advised us along the way to submission.

In [S2], Mika Meyer led the implementation of the technical infrastructure required for our methodology and provided concrete input on the paper's content. Giancarlo Pellegrino gave feedback on the idea and advised us along the way to submission.

Differently, in [S1], the author of this dissertation was not the main lead. The author of this dissertation investigated underground forums, handled data collection and analysis, and contributed with the corresponding paper section.

Moreover, we contributed to the release of several open-source datasets and tools. These resources were produced as part of the works [P2, P3] mentioned above to ensure reproducibility and support future research. Our investigation into clickbait PDFs [P2] resulted in a dataset containing first-page screenshots and metadata (such as SHA256 hashes and embedded links), which we made publicly available [53]. Additionally, we released the machine learning model and scripts [55] to reproduce the clustering procedure developed in our study, enabling the identification of clickbait PDF clusters. These dataset and code were evaluated as part of the "Artifact evaluation" procedure of ACSAC '23 and awarded the *Artifact Reusable* and *Artifact Reproduced* badges. We also provided the pipeline modules [54] designed to analyze websites and web services hosting clickbait PDFs in [P3].

Finally, the codebook and interview guide from our study on roadblocks to vulnerability remediation [P4] will also be open-sourced. Interested readers can find them in sections 8.3.2 and 8.3.3 of the Appendix. A permanent repository for these resources will be established upon publication.

1.3 Thesis Outline

This thesis is organized into seven chapters. Chapter 1 outlines the scope of our research and highlights its key contributions. Chapter 2 provides the necessary technical background and terminology used throughout our studies, together with a discussion of relevant related works.

In Chapter 3 we analyze the first case study of technology-enabled malicious deception: the creation of misleading link previews when sharing links on social platforms. Chapter 4 introduces clickbait PDFs, the second example of deception-based attacks enabled by technological integrations and development. In Chapter 5, we shift focus to the operational aspects of the clickbait PDF threat, examining the websites and hosting services attackers use to distribute these files. Chapter 6 addresses the challenge of low success rates in vulnerability notification campaigns, a problem we observed in our large-scale effort to notify stakeholders about clickbait PDFs. We analyze the human, organizational, and technical obstacles that hosting providers face when receiving such notifications.

Finally, Chapter 7 reflects on our contributions and outlines potential directions for future research.

2

Related Works and Technical Background

This thesis is centered around deception-based attacks on the Web. In this chapter, we introduce the foundational knowledge necessary to understand this work. First, we formally introduce deception-based attacks and the concept of social engineering in Web attacks (Section 2.1). Then, we give an overview of the most common Web attacks, which are often delivered after the deception of the victim. Following, we introduce the concept of Web Hosting and the most widespread commercial hosting services (Section 2.2), also giving an overview of the studies on web infrastructure abuse correlated with ongoing attack campaigns. Finally, we conclude by introducing the topic of vulnerability notification and the roadblocks in this procedure (Section 2.3).

2.1 Social Engineering and Deception-based attacks

Social engineering represents a class of security threats that primarily exploit human psychology rather than technical vulnerabilities [116]. Unlike purely technical attacks, which target flaws in software or hardware systems, social engineering attacks manipulate individuals into performing actions or divulging confidential information [250, 11]. These attacks rely on deception, often leveraging visual and contextual deceits as well as psychological levers as trust, urgency, or authority to subvert users' decision-making processes [34, 240, 44, 76].

Deception-based attacks typically unfold over four main phases [171, 21]. In the first phase, the attacker defines their objectives, such as identifying a target group or determining the scale of the operation, and may perform reconnaissance to gather information about potential victims. This information is used to craft a convincing bait and choose a suitable delivery method. In the second phase, the attacker sets up the necessary web infrastructure to deliver the attack, for example a fraudulent webpage [171], a malicious payload [161], or a compromised social media account [25]. Once this infrastructure is in place, the attacker distributes the deceptive bait through ad hoc channels such as email [205, 121, 120], voice calls, SMS [154], or social media platforms [219, P1, 78]. During this phase, the attacker awaits victim interaction with the decoy. Finally, in the fourth phase, the attack may be detected by automated systems, such as Google Safe Browsing [74], which can lead to the takedown of the infrastructure or blacklisting of the malicious content [169].

Phishing is one of the most well-known examples of such an attack, but it represents only a subset of the broader deception landscape. Other methods include romance scams [253], technical support scams [135], and malicious advertisements [238], as well as more sophisticated tactics such as deepfakes [149] or multi-stage social engineering campaigns [86]. Phishing remains a dominant web-based attack vector, counting hundreds of thousands of victims in the last years [66].

2.1.1 Deception in Web-based Attacks

Phishing is a deception-based attack, often including an additional technical or engineering component, where the deception is often used as stepping stone to deliver multiple other attacks, as for example stealing data, infecting the victim's machine, or performing actions on behalf of the victim. Among the various phishing distribution

methods, two vectors are particularly relevant to this thesis: email and social media platforms. Additionally, we introduce search engines as a distribution method, which attackers abuse to deliver a multitude of web attacks, including phishing attacks.

2.1.1.1 Email-based Attacks

Phishing emails have been one of the oldest attack vectors, most often used to share malicious links or harmful attachments that would compromise the victim's machine. Over the years, email-based phishing has been extensively studied, with prior work examining various aspects such as the scale, frequency, and temporal patterns of phishing campaigns (e.g., [205]), the characteristics of their baits (e.g., [240]), their effectiveness (e.g., [203, 44]), and methods for detecting them [67, 3].

The effectiveness of phishing campaigns depends on factors such as the bait and the scale of the operation. Baits often combine visual, textual, and contextual elements. Visuals, such as logos or familiar interface components, are used to mimic trusted brands or user interfaces [44, 134]. Textual cues may create a sense of urgency or authority to prompt action [240, 34]. Contextual elements adapt the message to the recipient, for instance by sending a fake invoice to someone in the finance department to enhance credibility [20, 76].

The bait might be unspecific and targeted to a very large group of victim users (*large-scale campaigns*) [205, 171] or more specific to single users or restricted groups of victims (*spearphishing attacks*) [87, 121]. Large-scale campaigns typically span one to three days, during which attackers may distribute up to 10,000 emails containing malicious links, or up to 1,000 emails with harmful attachments such as malicious documents [205]. Unlike large-scale phishing, spearphishing attacks are more targeted and sophisticated. Rather than relying on volume, they focus on precision, tailoring messages to a few victims to exploit specific vulnerabilities or access sensitive information [116].

As attackers continuously refine their techniques, so too have detection mechanisms evolved. Initially, detection relied on static features such as the URL, email content, or webpage layout [67, 139, 24, 165]. However, as phishing methods grew more sophisticated, machine learning-based detection systems emerged, offering a more dynamic approach by analyzing complex combinations of phishing indicators [259, 220]. A key area of research focuses on the detection of phishing pages via their visual appearance [2, 129, 137, 136], under the assumption that attackers often seek to impersonate legitimate and recognizable brands to deceive their targets.

2.1.1.2 Social Platforms Abuse

Social platforms provide fertile ground for social engineering due to their emphasis on interpersonal connections and content sharing. Attackers exploit these features to craft convincing personas [4], establish trust with potential victims [253], and distribute deceptive content at scale [68]. As a result, social platforms have become central vectors for deception-based attacks, serving both as initial entry points and as enablers of more complex, multi-stage campaigns.

Attackers have long used social platforms to spread harmful content. A range of attack types has been observed. Some attackers focus on distributing spam via public

content or private messages, often through fake or automated accounts [219, 78]. Other campaigns aim to distribute phishing links or malware, typically relying on compromised real accounts to increase trust and engagement [68]. A significant amount of research has focused on detecting this kind of abuse, focusing on the detection of account spreading malicious content. One common line of work looks at account-level features, as the types of posts made, the age of the account [219, 51] or its activity patterns [25, 247]. Other approaches use graph-based methods, analyzing how suspicious accounts connect to the broader social network to identify anomalies or detect clusters of malicious activity [19]. Detection can also focus on the content being shared, especially URLs. Some systems analyze the links directly, using crawlers to follow redirection chains [123], while others classify links based on features of the linked webpages or of the URL itself [24, 139, 228].

2.1.1.3 Abuse of Search Engines

Search engine poisoning is a deceptive strategy where attackers manipulate search rankings to drive users toward malicious or unwanted websites. Unlike traditional Search Engine Optimization (SEO), which typically focuses on boosting the visibility of specific websites within relevant search terms, poisoning attacks disregard keyword relevance entirely and instead target popular or trending queries to maximize user exposure. Successful attacks often rely on three key tactics: the selection of multiple high-traffic keywords [165], the automated generation of large volumes of fake, cross-referenced pages [258] optimized for different search terms, which tricks the ranking algorithm of search engines, and the use benign websites to host the cross-linked resources [97] as their good reputation positively influences the final ranking. Once users click on poisoned search results, they are often redirected through chains of intermediary sites toward malicious destinations.

This technique of malicious redirection has been shown to support a wide range of criminal activities, including the distribution of fake antivirus software, the operation of rogue pharmacies, phishing, click fraud, and the delivery of drive-by downloads [133, 138, 126, 97]. To better understand the infrastructure behind such attacks, researchers have developed methods like PoisonAmplifier [263], which actively discovers compromised websites by analyzing linking patterns and common vulnerabilities shared across attacker-controlled sites, or approaches such as deSEO [97], which focus on identifying clusters of suspicious URLs exhibiting unnatural patterns, allowing the detection of entire SEO campaigns without needing to crawl individual pages. Longitudinal studies have further highlighted the adaptive nature of search poisoning operations, noting that while the average persistence of compromised websites has declined, attackers have compensated by compromising a larger number of sites and funneling traffic through centralized broker infrastructures [126, 248]. Recent investigations have also revealed the growing use of cloud platforms to host long-tail SEO spam, with thousands of doorway pages successfully ranking among the top search results for niche keywords, showcasing attackers' ability to leverage scalable and cost-effective hosting resources for large-scale campaigns [133].

2.1.2 Deception in File-based Attacks

Deception also plays a considerable role in attacks involving the infection of computer systems and networks. This happens through malicious programs ("malware"), which are often delivered or executed thanks to social engineering tricks manipulating victims [191, 12]. Malware distribution frequently involves techniques that bypass user awareness entirely, relying on silent installation methods such as drive-by downloads or vulnerability exploitation [77, 184].

Potentially unwanted programs (PUPs), adware, and stealthy installers operate differently but are similarly grounded in deception. Unlike traditional malware, which installs without user interaction, PUPs typically require some form of user interaction or consent, where sers are tricked into installing these programs through misleading prompts, bundled installations, or fabricated warnings, with the true malicious nature of the software only becoming apparent after execution [114].

Among the various forms of deceptive malicious files, malicious documents have emerged as a particularly effective attack vector [191, 226]. Unlike executables, which may immediately arouse suspicion, documents such as PDFs, Word files, and Excel spreadsheets appear benign and are routinely exchanged in personal and professional contexts. Attackers exploit the trust placed in these formats by embedding malicious code within document structures [194, 191, 121].

Malicious Documents. Microsoft Office documents, in particular, have been a long-standing target, with attackers historically relying on Visual Basic for Applications (VBA) [167] macros and, more recently, formats such as Excel 4.0 (XL4) macros [191, 179]. The diversity of document formats, including OLE and OOXML [50], combined with the variety of macro types, presents significant challenges for both detection and analysis, often forcing analysts to rely on time-consuming manual methods when automated tools fail to handle obfuscated or nonstandard samples [194, 191].

Recent trends have seen a rise in the use of sophisticated obfuscation techniques within malicious documents, designed to bypass security mechanisms and delay detection. To address these challenges, research efforts have proposed techniques such as symbolic execution engines for Excel 4.0 macros [191], which can systematically explore multiple execution paths and reveal hidden payloads across different environmental conditions. Empirical analyses of targeted attack campaigns have confirmed that malicious documents continue to be a dominant delivery mechanism, particularly against high-profile targets such as NGOs, news organizations, and governmental entities, with frequent adaptation of exploits to evade emerging defenses [121].

PDF documents also play a significant role in the distribution of malware, exploiting weaknesses in document parsers [261] or leveraging embedded scripts to trigger malicious behavior upon opening [118]. Early detection methods focused on extracting and analyzing JavaScript, but reliable extraction remains a technical challenge given the format's flexibility and support for deep object nesting [118]. To address these limitations, subsequent work shifted toward static analysis techniques that rely on structural features and metadata rather than direct script inspection. Studies have demonstrated that characteristics such as object layout, image count, and metadata length can be used to train machine learning models that generalize well across previously unseen malware

samples [208]. Building on this, more recent approaches have eliminated the need for JavaScript awareness altogether by analyzing the structural side-effects of embedded malicious behavior [211].

Distribution of Malicious Files. The distribution of malware, and in particular malicious documents, occurs through a range of vectors, with email and web channels being predominant. Email-based distribution often involves phishing campaigns that entice recipients to open infected attachments or click on malicious links [205, 121, 191]. Notably, a large-scale analysis observed that while malware email campaigns were initially rare, the resurgence of the Emotet botnet led to a peak of 224 million malware emails in a single week, relying heavily on Office and PDF documents as initial infection vectors [205]. Complementary work demonstrated the effectiveness of publicly available tools for detecting exploit documents submitted to VirusTotal [245], enabling the identification of hundreds of attacks based on malicious Office and PDF files [120].

Web-based distribution includes the use of compromised websites, malicious advertisements, or purpose-built landing pages designed to deliver malware upon interaction. These web resources may employ drive-by download techniques [77, 144] or social engineering strategies [161] to convince users to download infected files. A large-scale measurement study reported that over a period of 10 months, more than 3 million malicious URLs were found to initiate drive-by downloads, and that approximately 1.3% of Google search queries returned at least one malicious link in the results [144]. Further investigations revealed that a significant portion of file-based attacks are delivered through online advertisements, often served by lower-tier ad networks [161] or pay-per-install (PPI) services [114].

2.2 Web Hosting

Web hosting is a fundamental service that enables websites to be accessible on the internet. Hosting providers allocate server space and resources to customers, allowing them to store and publish website content. Customers typically interact with these services by purchasing hosting plans, uploading website files, and managing site functionality through control panels or dedicated interfaces. These services are critical for the online presence of businesses, organizations, and individuals, providing the backbone for websites ranging from small personal blogs to large e-commerce platforms.

2.2.1 Types of Web Hosting

There are several types of web hosting services tailored to different customer needs. *Shared hosting* is the most common and economical option, where multiple websites share the same server resources. *Virtual Private Servers* (VPS) offer instead dedicated portions of server resources to each user, ensuring better performance and scalability. Both of these shared-resource approaches are cost-effective and require minimal to moderate technical knowledge, making them accessible to non-tech-savvy users. In both cases, customers retain control over the client-side applications of their websites.

Website builders, by contrast, are tailored to users seeking simplicity, as they provide not only hosting services but also web application software as part of the package. These platforms combine hosting with intuitive tools for website creation, making them particularly appealing to users with little or no technical expertise.

At the other end of the spectrum, *Dedicated Servers* offer customers complete control over the server machine, granting them greater flexibility but also more responsibility for server management. *Colocation* represents a specialized form of dedicated hosting, where customers own the server hardware, and the hosting provider supplies essential infrastructure services such as power and connectivity.

Finally, *Reseller hosting* is a unique model in which a hosting provider allocates space and resources to a customer, who then independently resells these services, often bundled with additional features, to end-users.

2.2.2 Abuse of Web Hosting Services

Web hosting services are not immune to misuse. Web attacks are frequently delivered through the same platforms and websites enabled by hosting services, creating a direct link between the abuse of hosting and the propagation of attacks. Attackers exploit hosting platforms for malicious purposes, either by registering websites with harmful intent or compromising legitimate sites [43, 142, 155]. Maliciously-registered websites are created with the intent to distribute malware [77], launch phishing campaigns, or host illegal content [155, 82]. Alternatively, attackers compromise existing websites, for example by exploiting unpatched vulnerabilities in Content Management Systems (CMS) [152, 125, 209], and repurpose them for malicious activities [132, 28].

Recent works [195, 133, 152] point out the existence of additional scenarios, where the attacker neither registers a new domain nor compromises a third party's domain, but rather get assigned a domain from a hosting provider (e.g., a free subdomain). This possibility became feasible with the availability of inexpensive (if not free) services offered by hosting providers, possibly without thorough registration checks. These services may include free object storage, free subdomains for E-commerce websites, or online marketplaces.

Researchers developed various techniques to study and detect hosts supporting attackers' operations, for example by matching server configurations with that of known malicious servers [155], by constructing fingerprints of the network responses captured in Internet-wide probes [156], or by exploiting search engines to find indicators of vulnerable web servers [152, 96, 263, 93]. Hosting infrastructure is also often studied for the complex redirection chains attacker build, also called "Traffic Distribution Systems" [131], to monetize clicks [221], deliver the right attack payload to a visitor [130], or evade detection [130, 132].

An orthogonal line of work explores and measures the security posture of hosting services [162, 38] and websites, observing the presence at large scale of multiple vulnerabilities [124, 104, 13], improper security headers, and outdated software (e.g., [72, 100]), although without linking it to an ongoing exploitation. Further, Canali et [23]. al, Stock et al. [216, 215], and Ethembabaoglu et al. [60] observed the presence of vulnerable web application software at hosting providers, prompting them via vulnerability notifications

to observe any remediation action, however with limited results.

2.3 Vulnerability Notifications

Vulnerability notifications play a crucial role in mitigating the abuse of web hosting services, as they alert stakeholders of security weaknesses before they are exploited. Notifying stakeholders about vulnerabilities or compromises is now a well-established practice, supported by guidance from organizations such as US-CERT (CISA) [41], ENISA [57, 56], and OWASP [175]. A vulnerability notification refers to the process of informing website owners, hosting providers, or other relevant parties about security flaws that put their systems at risk [175]. These vulnerabilities can range from outdated software and misconfigurations to more critical issues like memory leaks or vulnerable IoT devices [48, 216, 127, 29]. By alerting website operators to security risks, these efforts help prevent breaches, reduce financial losses, and limit the spread of malware or phishing campaigns, resulting in takedowns or patches.

Ideally, notifications should be issued as soon as a vulnerability is identified, allowing affected parties to take corrective action before attackers can exploit the weakness. The timing of these notifications is particularly important when dealing with actively exploited vulnerabilities or those affecting a large number of websites [48, 29]. The responsibility of issuing notifications typically falls on cybersecurity researchers, threat intelligence agencies, hosting providers, and sometimes even governmental cybersecurity agencies [127, 31].

Previous work has examined the content and structure of notification messages to enhance their effectiveness in prompting remediation. For instance, Dietrich et al. highlight that "the body of the email must be comprehensive, specific and comprehensible", noting the challenge of crafting a message that is persuasive to technically skilled recipients while remaining accessible to less experienced users [45]. Further, Stock et al. have explored the impact of various formatting features, such as the presence of a digital signature, HTML formatting, or characteristics of the sender's email address, on the perceived trustworthiness of the message [215].

The notification process itself varies depending on the severity of the issue and the availability of contact information. Direct emails to website administrators or hosting providers are common [23, 216, 132, 141], but in cases where direct contact is not feasible, notifications may be published on public forums, CERT (Computer Emergency Response Team) advisories, or through security mailing lists [177, 127, 117]. Several studies have assessed the availability and responsiveness of various contact channels, including WHOIS emails [215, 216], standardized RFC-defined addresses [39, 30], social media profiles, and even physical mail [140]. While WHOIS-based contact points are generally the most accessible and frequently used, they still yield low remediation rates. Physical mail has shown comparatively high effectiveness, although this may be influenced by the legal or formal nature of the notifications involved.

2.3.1 Responses to Vulnerability Notifications

Vulnerability notifications succeed in lowering internet abuse when they are acted upon by recipients or responsible parties. Unfortunately, research studies report vulnerability notification campaigns with low remediation rates [127, 216]. Thus, recent research has extensively explored ways to improve remediation rates for vulnerability notifications by modifying specific properties of VNs, such as email features [215, 127] and sender's attributes [31, 215], the nature and severity of the vulnerability [215, 237, 141], the communication channel [30, 215, 140, 141, 216], or the role of the recipient [217, 30, 216, 23]. Despite these efforts, the response and remediation rates for notified web vulnerabilities still average between 20 to 30% [30, 215, 155], and no effective alternatives have yet been identified (e.g., still no better channel than WHOIS [216, 30, 182]).

While studies with website owners offered some understanding of their VN evaluation criteria [84, 141], website owners are usually not the recipients of such notifications, as their contacts are not easily available at a large scale. Moreover, as hosting providers have access to the attacker-controlled hosts, they can make an impact in lowering the concentration of abuse in their hosting spaces [223]. Other studies have explored the causes of non-remediation through interviews with website and system operators [16, 215], revealing a complex landscape. Operators report of multiple undiscovered security misconfigurations that could lead to incidents [45], which underscores the importance of VNs for raising awareness. However, awareness alone does not guarantee remediation [215], suggesting that factors beyond VN characteristics influence these decisions. Further research examined whether technical or organizational factors impacted remediation but found no single factor to be more significant than others [16]. As a result, we still lack a clear understanding of what happens after a vulnerability notification is received, as none of the previous works has investigated the internal processes of the receiving entities.

2.3.2 Organizational Challenges in Implementing Security

Multiple studies have analyzed the behavior of IT practictioners when implementing security practices, focusing for example on update practices [231, 128], incident management [212], and vulnerability management [6, 207]. Operators' struggles in applying updates derive not only from the high workload, but also from the difficulty in predicting the update's outcome while ensuring the reliability and stability of the service [231, 128]. A similar overwhelming situation is reported by De Smale et al. [207], who describes the prioritization and filtering applied at organizations in vulnerability management processes. Alomar et al. [6] reports instead how vulnerability management is mostly hindered by organizational and management practices internal to the organization, which include operators' compliance-oriented approaches together with the de-prioritization of security when conflicting with business practices. These studies highlighted many challenges in vulnerability remediation when the security of the company itself is on the line. It remains thus unclear whether the same struggles apply to vulnerability management in hosting provider companies, where the notified vulnerabilities concern customer spaces rather than the IT infrastructure essential to the functioning of the company.

3

Deceptive Previews in Social Platforms

This chapter introduces and analyzes *deceptive previews*, focusing on *whether new integrations, technological developments, or communication platforms can be exploited to mount deception-based attacks (RQ1)* in the context of social platforms. Deceptive previews, which are benign-looking previews for malicious links, are the first of two cases studied that involve new technology-enabled deception attacks.

Social platforms, which include media as social networks and instant messaging applications, have become a popular mean of sharing and accessing information [9]. Their popularity however attracted malicious actors, too, who use them to distribute malicious links [10]. As link previews can influence users' decision to click [188], we focus our investigation on link previews as the mean used by attackers to draw the attention of their victims.

We comprehensively study the link preview creation process and design in 20 social platforms, observe existing countermeasures against the distribution of malicious links, and test the link preview creation process in an adversarial setting. Our results demonstrate the lack of a standardized preview design and creation process, as well as the widespread lack of countermeasures against the distribution of malicious links, observed in all but two platforms. These observations lead us to distill seven recommendations towards more robust and trustworthy link previews.

3.1 Framing of the Study

3.1.1 Sharing External Content on Social Media Platforms

Sharing text messages on social platforms, such as social networks, is usually a straightforward process: a user logs into the platform, types the message, and posts it. The message is then stored and delivered to all friends when they update their timeline. When the message contains a URL, the platform retrieves the resources in the shared page to build a link preview. In theory, link previews can be created either by the client-side program (e.g., Javascript) or the server-side programs. However, as URLs often originate from third-party domains, most platforms cannot rely on the client-side programs because the same-origin policy for cross-origin requests (SOP for CORs) prevents the client-side programs from fetching resources from other origins by default. Accordingly, platforms tend to use server-side requests [178] (SSRs).

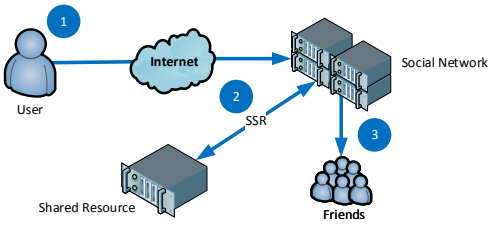


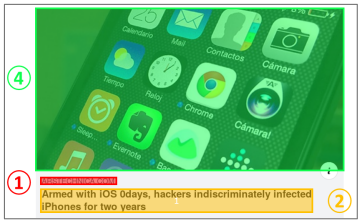
Figure 3.1: Sequence of steps when sharing pages on social networks.

Figure 3.1 shows the sequence of steps when sharing URLs on social platforms. The user accesses the social media platform through their browser, or through a mobile app, and then types the URL in the input box to share the URL content with friends or contacts (Step 1 Figure 3.1). Then, the platform performs a number of SSRs to retrieve the URL and the linked resources, e.g., images (Step 2 Figure 3.1). Then, the platform processes the collected

```
<meta name="twitter:site" content="①">
<meta name="twitter:title" content="②">
<meta name="twitter:description" content="
    ③">
<meta name="twitter:image" content="④">

<meta property="og:site_name" content="①"
     />
<meta property="og:title" content="②" />
<meta property="og:description" content="
    ③" />
<meta property="og:image" content="④" />
```

Listing (3.1) Open Graph and Twitter Cards tags for both Previews.



(a) Preview on Facebook

(b) Preview on Twitter

Figure 3.2: Example of real world use of meta tags.

Open Graph	Twitter Cards	Description
`og:title`	`twitter:title`	The title of the article without any branding
`og:description`	`twitter:description`	A brief description of the content.
`og:image`	`twitter:image`	The URL of the image that appears in the preview.
`og:url`	-	The canonical URL for the page, without session variables or user identifying parameters. This URL is used to aggregate likes and shares.

Table 3.1: Description of meta tags used to create the link preview of HTML content

resources to create a preview for the webpage. The construction of the preview can be aided with a set of additional HTML meta tags specifying suggested content for each field of the preview, such as the page title and page description. Two popular meta tags languages are Open Graph [62] by Facebook and Twitter Cards [233] by Twitter. Table 3.1 shows the list of meta tag types that can be used to create previews for HTML content. Listing 3.1 shows an example of meta tag use to create two previews of the same article. Figure 3.2 shows two screenshots for the resulting previews.

3.1.2 Case Studies

We conduct the study of this paper on 20 popular social media platforms, ten of which are social networks, and ten are instant messaging apps. In this section, we present the selection criteria we used.

3.1.2.1 Social Networks

We created an initial list of social networks by combining two sources. First, we manually inspected the Alexa Top 1M domains, retrieved in May 2019, and removed all

the websites which do not fall under the *Social Network* category (e.g. `amazon.com`); then, we manually visited the remaining ones until we collected 30 social networks, with no pre-established cutoff on the domain rank value. Then, we merged the 30 social network domains from the Alexa Top 1M domains with additional 30 domains of social networks ranked by the number of users. For this ranking, we used the list maintained by Wikipedia[1], retrieved on July 2019. Then, from these 60 social networks, we removed duplicates obtaining a list of 47 social networks.

We inspected each of the 47 social networks manually, and removed 37 of them for one of the following reasons: (i) social networks that no longer exist, (ii) we were unable to create user accounts[2], (iii) the social network is ranked too low in the Alexa Top 1M, (iv) platforms that do not support link sharing (e.g., Soundcloud), (v) platforms that require Premium subscriptions, (vi) social networks that merged with already discarded ones, and (vii) posting prevented due to bot detection. Table 3.2 lists the 10 social networks that we used for the study of this paper.

3.1.2.2 Instant Messaging Apps

We created the list of candidate instant messaging apps by crawling the first 32 apps in order of appearance from the category "Communication" of the Google Play store. To these samples, we added six more apps (i.e., Instagram, Discord, Slack, Kik, Signal, and Snapchat), that we considered popular but not part of the initial list. From these 38 apps, we removed duplicates obtaining a list of 28 instant messaging apps. Then, we inspected each app manually and removed 18 of them for the following reasons: (i) not available in the Apple Store[3], (ii) no instant messaging function, (iii) link previews not supported, and (iv) a low number of downloads. Table 3.2 lists the 10 apps we used for the study of this paper.

3.1.3 Threat Model

We now present the threat model of this paper. In this paper, we assume the best scenario possible for both the attacker and the victim, i.e., a strong attacker and a tech-savvy user.

Attacker — The attacker of this paper intends to lure their victims into visiting a malicious web page. The specific final attack delivered with the malicious page can vary, based on the motivations of the attacker. For example, an attacker with economic interest may want to steal credit card numbers with a phishing page. In this paper, we also consider highly-motivated powerful attackers such as state-sponsored attackers that can use malicious pages to deliver 0-day exploits to compromise users' device.

[1] See, `https://en.wikipedia.org/wiki/List_of_social_networking_websites`

[2] The main reason was the language barrier. Then, even when using automated translation and help from a native speaker (Chinese), we were deemed to be a robot or a non-trusted user, and denied access to the platform. We would speculate this occurred because our mobile phone numbers were not Chinese or because of the geo-location of our IPs.

[3] We ignored apps that are not in the Apple Store because of our testing setting (See Section 3.2). We used one iPhone device and one Android device: one for the user sharing a link, and the other for the user clicking on the link preview.

Social Network	**Alexa**
Facebook	3
Twitter	11
VK	15
LinkedIn	23
Pinterest	67
Tumblr	75
Medium	113
Xing	1.294
Plurk	1.341
MeWe	5.142
App	**Downloads**
Instagram	1.000.000.000+
Messenger	1.000.000.000+
Skype	1.000.000.000+
Snapchat	1.000.000.000+
WhatsApp	1.000.000.000+
Line	500.000.000+
Viber	500.000.000+
KakaoTalk	100.000.000+
Telegram	100.000.000+
Slack	10.000.000+

Table 3.2: List of platforms.

The attacker uses social media platforms to distribute the link to the malicious page. For example, in the case of social networks, the attacker can register one or more accounts to direct the campaign. The attacker can also use stolen credentials to spread malicious links over a platform, including instant messaging systems. Their goal is to post malicious links while, on the one hand, being undetected by possible active or passive detection systems put in place by the hosting platform and, on the other hand, misleading the users, who make use of the link preview to decide whether to click. To this end, the attacker creates a mismatch between the malicious content in the page and its benign-looking link preview, by including in the attacker's code specific meta tags.
Victim — The victim of these attacks can be a specific individual or small group of individuals (i.e., targeted attack), or as many users as possible, indiscriminately. For the analysis of this paper, we consider skilled and experienced social network users—a category of users who is less prone to click on malicious URLs [186, 46, 88, 246].

3.2 Characterizing Link Preview Creation

In essence, link previews synthesize a web page, creating the expectation on what the user would see when clicking on the preview. The analysis of this section intends to shed some light on the ways social media platforms create link previews. This analysis reviews the content of previews of a set of test web pages, and identifies precisely the fields that are displayed and under which circumstances. After presenting a comprehensive overview of link preview creation, our analysis studies the network traffic to retrieve the resources to build the link preview, looking for distinctive features that can be used to discover social media platforms' requests. Finally, our analysis investigates the extent to which the coherence between previews and web pages content holds.

Experimental Setup For the analysis of this section, we prepared a set of controlled experiments. Our experiments involve a user submitting links of test web pages we control, and another user observing the created link preview. Accordingly, we registered two user accounts for each platform. Facebook is the only platform offering test accounts, which are users separated from regular users.

We conducted our experiments on social networks using Firefox (version 69.0 for Ubuntu 18.04), Chrome (77.0.3865.75 for Ubuntu 18.04) and Brave Software (0.68.132 based on Chromium 76.0.3809.132 for Ubuntu 18.04) browsers. For IMs, we purchased two mobile phone SIM cards and used two different mobile phones for our experiments, i.e., an iPhone 5S (OS version 12.4.1) and an Android Pixel device (OS version Android 9).

To serve our test pages, we set up an Internet-facing web server serving resources over different subdomains. We used one subdomain for each social media platform and each experiment, achieving a high degree of isolation among the experiments on one platform and across all platforms. Also, we configured our web server to deliver test pages only when accessed via one of the unique subdomains and not through our public IP address, reducing the noise caused by bots of search engines and rogue web scans. All web pages of our experiments contain a unique page title, text paragraphs, and one image. Depending on the specific test, web pages can contain Open Graph and Twitter Cards meta tags in different combinations. We detail the content of meta tags in the corresponding subsection below. Finally, we logged the main fields of the HTTP requests incoming to the server, for further analysis.

3.2.1 Displayed Information

Link previews intend to summarize the content of the embedded links, by showing a site name, an image, and a brief description of the web page's content, typically. These fields originate from the web page's HTML code, either from the standard HTML tags or from ad-hoc meta tags such as Open Graph or Twitter Cards markups. The goal of this section is learning the exact information shown to a user across different social media platforms, and tracing back the content of each preview field to the web page.

To that end, we defined a set of controlled experiments by posting links to resources hosted on our web server, and observing the resulting link preview. As the link preview

	Visible Features					User Actions		
Name	Site title	Site descr.	Image	Host	Shared URL	Mouse over	Add. Info	**Priority**
Facebook	●	●	●	●	○	DP	●	O, H, T
Twitter	◐	◐	◐	◐	○	SU	○	T, O, H
VK	●	○	●	●	○	DP	○	O||T, H
LinkedIn	●	○	●	●	○	URL	○	O, H, T
Pinterest	○	○	●	●	○	URL	○	O||H, T
Tumblr	●	◐	◐	●	○	DP	○	O, H, T
Medium	●	●	●	●	○	URL	○	O, H, T
Xing	●	◐	◐	●	○	DP	○	O, H, T
Plurk	●	○	◐	○	○	URL	○	O, T, H
MeWe	●	●	◐	●	○	URL	○	O, H, T
Instagram	●	◐	◐	○	●	-	○	O||T||H
Messenger	●	●	●	●	●	-	●	O, H, T
Snapchat	●	○	●	●	○	-	○	O, T||H
WhatsApp	●	◐	◐	●	●	-	○	O, H, T
Skype	●	◐	◐	●	●	-	○	T, O, H
Line	●	●	●	○	●	-	○	O||T, H
Viber	●	○	●	●	○	-	○	T, O, H
KakaoTalk	●	◐	●	●	●	-	○	O, H, T
Telegram	◐	◐	◐	◐	●	-	○	O||T, H
Slack	◐	◐	◐	◐	●	-	○	O, T, H

Table 3.3: Characterization of the link preview creation. For the visible features, we use "●" when we observed a field in all of our experiments, "○" when we never observed a field, and "◐" when the presence of the field depends on the context, e.g., meta tags or user edits. We use "DP" for dereferal page, "SU" for shortened URL, and URL for the shared URL. For the priority, we use "O" for Open Graph, "T" for Twitter Cards, and "H" for standard HTML tags.

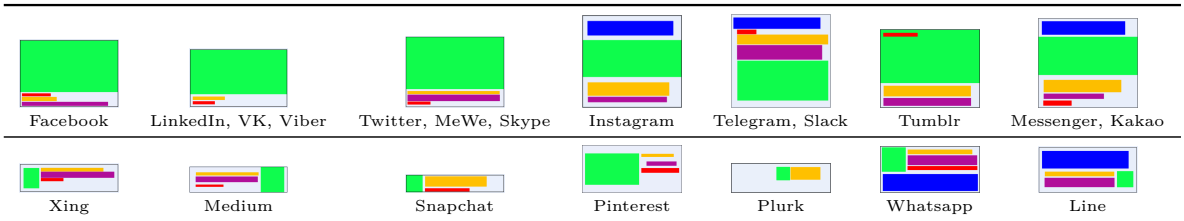


Table 3.4: Color-coded link preview layouts grouped by visual similarity, i.e., same field order and position. Color coding: Red is for the domain name, green for the image, yellow for the site title, purple for the site description, and blue for the URL.

could show data originating from both standard HTML tags and meta tags, we created web pages with Open Graph or Twitter meta tags, both meta tags at the same time, and no meta tags. When creating our test pages, we used unique values (i.e., titles, descriptions, and images) for each of the meta tags to allow us to identify the exact source of the data values used by the preview creation. Also, we intend to study the ways the link preview may change for pages delivered with redirections. Accordingly, we repeated our experiments using server-side and client-side redirections. Table 3.3 summarizes the result of our analysis.

3.2.1.1 Visible Features

We start our analysis by pinpointing the exact fields that social media platforms include in link previews and their location. Table 3.3, columns "Visible Features", lists the displayed fields that we observed. We say that a link preview field is visible ("●") when the field is present in all previews created during our experiments. We say that a link preview field is not visible ("○") when the field is not present in any link preview of our experiments. Finally, we say that a field *may* be visible (symbol "◐") when at least one link preview shows that field. Table 3.4 shows the position of each field per platform.

Inconsistent Use and Position of Fields All platforms include a different combination of the following fields: title of the web page, description of the web page, an image, the domain name, and the shared URL. We observed that there is no regular usage of these fields and that there is no field that is always displayed. The ones that are presented by most of the platforms are the site title (16 over 20 platforms) and the hostname (14 platforms). Then, interestingly, the image field is not shown all the times, and 11 platforms out of 20 may fail in showing an image when, for example, the linked web page does not include the meta tag for images.

When looking at the shared URL field, we observed a noticeable difference between social networks and instant messaging platforms. As opposed to IMs, none of the social networks shows the shared URL in the preview. However, we need to clarify that IMs do not have a dedicated field for the URL. Instead, by default, IMs show the URL in the textbox of the user's message.

Finally, the content of link previews varies with the presence of meta tags. Across all platforms, a total of 25 fields are not present in the link preview when the linked web page does not include any meta tag, i.e., the web page contains only standard HTML tags. Such behavior may be caused by shortcomings of HTML parser, or more probably, by intentional decisions of the developers due to the cost of processing a large number of web pages.

Heterogeneous Link Preview Templates When visiting Table 3.3 per platform, one can observe that only nine platforms out of 20 (Facebook, VK, LinkedIn, Pinterest, Medium, Messenger, Snapchat, Line and Viber) create link previews with a consistent number and types of fields, regardless the presence of meta tags. However, when looking at the variety of fields shown across these nine platforms, we observe four different sets of fields indicating that there may not be an accepted consensus on which fields

constitute a link preview. For example, the previews created by Facebook and Medium include all fields except for the shared URL, which is instead present in Messenger. VK, LinkedIn, Snapchat and Viber show only site title, image, and hostname, whereas both Pinterest and Line show a different subset of fields each (Pinterest's title and description have to be user-provided at posting time).

Then, the preview created by the remaining eleven platforms varies with the presence of meta tags. Interestingly, the absence of these fields is not consistent within the same platform. Only three platforms (Twitter, Telegram, and Slack) fail to build a preview of pages containing only standard HTML tags. The previews of the other eight platforms incoherently display fields. For example, Instagram shows only the title and the shared URL of pages with only HTML tags.

Finally, when looking at the visual position of field in the preview, we identified 14 distinct template layouts. Table 3.4 lists the layouts we observed, grouping layouts by same order of fields and position.

3.2.1.2 Priority

The second part of our analysis studies the behavior of the platforms when processing web pages with multiple meta tags and without meta tags. The goal of this analysis is to learn the importance assigned to each field. Table 3.3, columns "Priority", summarizes our findings. We use the letter `H` for standard HTML tags, the letter `O` for the Open Graph meta tags, and the letter `T` for Twitter Cards. The three letters are ordered from left to right by priority. When we cannot establish a clear priority, e.g., the preview contains a mix of tags, we use the symbol "`||`". We cross a letter when the type of tag is never used for the preview.

Our analysis reveals that, with few exceptions, the content of link previews originates predominantly from the meta tags, even when they differ from the content of the page. For example, concerning the hostname field, Facebook, Messenger and WhatsApp show the domain name of the URL of the `og:url` meta tag even when it differs from the URL hosting the resource. We observed similar behavior with Xing, Telegram, and Slack, that show the content of the `og:site_name` meta tag in the host field. A few platforms, i.e., Pinterest and VK, directly prompt the user for text for the preview when the platforms fail at rendering the link preview.

Finally, we observe that Open Graph is, by far, the most used markup language for link previews. Open Graph is also the first one displayed by all platforms except for three, i.e., Twitter, Skype and Viber. While Twitter Cards seems to be rarely used by social networks, it has a bigger userbase among IMs, where only two platforms (Messenger and KakaoTalk) do not seem to support it.

3.2.1.3 User Actions

The third analysis of this section involves fields that a user can inspect only upon an action. We identified two of such fields (see, Table 3.3, columns "User Actions").

The first field is the URL shown when the user moves the mouse over the link preview. Typically, when moving the mouse over an anchor tag, the browser shows in the status bar the hyperlink. Social networks respect such an expected behavior;

however, 50% of the social networks do not show the original URL in the status bar but prefer showing either a shortened URL ("SU") or a dereferer page ("DP"). By *dereferer page* we indicate a social network-specific proxy interposed between the user and the shared web page, e.g. as a click aggregator.

The second field is specific only to Facebook and Messenger. Within the link preview, both platforms show an additional UI button—called *"Context Button"*[4]—to display a dialog box with additional information about the domain name of the `og:url` tag. Such additional information, when available, includes (i) content from Wikipedia, (ii) domain name registration date from the WHOIS database, (iii) a link to the Facebook page associated to the domain name, (iv) the number of times that link was shared, and (v) a map showing the locations on earth of users who shared the link.

3.2.1.4 Page redirections

The final analysis of this section studies the link preview generation when pages are delivered with redirections. For that, we repeated the previous experiments by concealing the URL of the final page with a redirections. We implemented both server-side redirections with 303 and 307 status codes, and client-side redirections either via HTML tags or via JavaScript code. The results of our analysis are not in Table 3.3, and we report them in this section briefly. All platforms correctly handle server-side redirections. Facebook is the only platform supporting client-side redirections (both HTML and JavaScript ones). Overall, the link preview does not differ significantly from the previews created when posting direct links.

3.2.2 Network Signatures

After analyzing the displayed information, we look for unique signatures in the incoming HTTP requests. Our goal is to identify distinguishing features that can be used by the owner of a web page to determine when the incoming request originates from a social media platform. For this analysis, we process the entries in our server log files to identify such signatures.

In general, when sharing URLs to our pages on social networks, we should expect that other users may click on the link previews, introducing spurious entries in our logs. To avoid the presence of user activities, we limited the visibility of our posts whenever a platform supports such a feature. Only two platforms do not support access restrictions, i.e., Medium and Plurk; however, upon manual inspection, we verified our logs did not contain any user activity but only requests from both platforms. Finally, we point out that the same concern does not apply for IMs as messages are visible only to the recipient, that, in our setting, is another user under our control.

From our log files, we parsed all entries and extracted the user-agent strings and the IPs. We compared user-agent strings against known strings for browsers, and we looked for substrings that can be used to identify a platform uniquely. An example of these substrings is "`facebookexternalhit`" for Facebook or "`vkShare; +http://vk.com/dev/Share`" for VK. When the user agent contains such unique

[4]See, `https://www.facebook.com/help/publisher/1004556093058199`

	User Agents			IPs		
Name	# UAs	Org.	Bot	# ASN	Res.	Prov.
Facebook	2	1	1	1	0	1
VK	1	0	1	1	0	1
Twitter	1	0	1	1	0	1
LinkedIn	1	0	1	1	0	1
Tumblr	1	0	1	1	0	1
Pinterest	2	1	1	1	0	1
Xing	3	0	3	2	0	2
MeWe	1	0	1	1	0	1
Plurk	1	0	1	1	0	1
Medium	5	0	5	2	0	2
Instagram	12	9	3	1	0	1
Messenger	6	3	3	1	0	1
Skype	2	1	1	1	0	1
Snapchat	3	0	3	2	1	1
WhatsApp	2	0	2	1	1	0
Line	3	2	1	2	0	2
Viber	1	0	1	1	1	0
KakaoTalk	2	1	1	2	0	2
Telegram	1	0	1	1	0	1
Slack	3	0	3	2	0	2

Table 3.5: Analysis of access logs considering IP and User-Agent for each social media platform

strings, we classify the entry as bot. When the user-agent string matches one of the known user-agent strings of browsers, we classify the entry as organic. Then, starting from the collected IPs, we resolved the autonomous system numbers (ASNs) and searched the AS name strings for unique substrings. For example, Facebook's request originate from AS 32934, whose name is "`Facebook, Inc.`". However, not all platforms manage an autonomous system, but they may be relying on third-party providers. For example, Pinterest's requests originate from AS 14618, whose name is "`Amazon AES`". When the autonomous system name matches the name of a platform or a known network provider, we classify the entry as a service provider.

Table 3.5 summarizes the results of our analysis. All the 20 social media platforms under test use at least one user agent string linked to the name of the company or the service itself, allowing for immediate traffic filtering. Of these, 13 platforms use only one user-agent header, and seven platforms (Xing, Medium, Instagram, Messenger, Snapchat, Whatsapp and Slack) use multiple ones. Seven platforms (Facebook, Pinterest, Instagram, Messenger, Skype, Line, and KakaoTalk) request web pages using user-agent strings that are indistinguishable from browsers, posing a potential problem for the identification. However, the analysis of the IPs and the ASes provides a stronger signal than user-agents. As a matter of fact, all platforms perform HTTP requests from IPs of either one or two autonomous systems that can be linked to the platforms. Three instant messaging apps (Whatsapp, Snapchat, Viber) request resources directly from the user's phone, slightly increasing the difficulty in distinguishing if the visitor is organic or not, as the AS usually is from a residential area; nonetheless, all three of them include the

app name in the user-agent string, so we can categorize the respective entries as bots.

3.2.3 Link Preview Coherence

The final analysis of this section investigates the coherence between the link preview and the web page. In particular, we are interested in studying the ways social media platforms keep up to date the link previews in which a page changes over time. To this end, we generated new, unique URLs, one for each platform, and posted them. Then, we developed a bot controlling a pool of web browsers which is visiting periodically (every 30m) the platforms' pages showing the preview, over a period of 14 days. As IMs messages are expected to be short lived, we did not consider them for these experiments.

The analysis of our logs revealed that eight out of 10 social networks request the page at least once on the submission date, and never again. Twitter and Pinterest are two exceptions, requesting the web page multiple times across a period of 14 days. For what concerns the associated resources, seven social networks requested them only once at submission time, and never again. The remaining three platforms, i.e., Facebook, Twitter and LinkedIn, request the link preview images more regularly.

3.2.4 Takeaway

The analysis of this section shed some light on three key aspects of social media platforms when creating a link preview. To summarize, this section makes the following findings:

- Social media platforms rely unconditionally on meta tags for rendering previews, especially on the Open Graph markup language. When meta tags are not present, link previews display fields in an inconsistent manner, exposing users to a great variety of heterogeneous link preview templates. As a result of all this, we speculate that users are misled into taking the wrong security decision. Also, the heterogeneity of templates and inconsistent use of fields may fail in building a secure mental model of link preview outlooks.
- Platforms' requests contain distinguishable signatures that can be used by web sites owners to determine when a request originates from social media platforms. This is a required feature to enable cloaking attacks.
- The temporal analysis reveals that platforms tend to fetch the resources for the link preview very rarely over a period of 14 days. A longer time window may show a different behavior, however, it should be noted that 14 days is sufficient for a successful malicious campaign.

3.3 Malicious Content and User Awareness

Section 3.2 studied the behavior of social media platforms when sharing links to benign web content. However, as observed by prior work, adversaries can also share malicious content on social media platforms such as phishing pages (see, e.g., [123, 219]). Anecdotal evidence suggests that social media platforms, social networks in particular, may have deployed defenses to counter the spread of malicious content in their systems. For example, Twitter claims to match shared links against a database of potentially harmful URLs [232] and to additionally use their shortening service to interpose informative safeguarding pages in between `https://t.co` links and their malicious targets. Facebook reports the employment of dedicated teams and tools against spam on the platform [63], as well as anti-virus measures in the file upload and download processes [61].

The second analysis of this paper studies the presence and effectiveness of possible deployed countermeasures when sharing malicious URLs. Also, our analysis reviews the created link previews to evaluate to what extent users may be aware of the risk of clicking on previews of malicious links. In this section, we leverage on the knowledge acquired during the observations of Section 3.2, which we will use as a behavioral baseline to compare social media platforms behavior when dealing with malicious content. Our focus is not built on the attacker's perspective, rather on the observation of existing active or passive countermeasures preventing the distribution of malicious content; the most fitting scenario is the one of malware and phishing spread prevention.

Experimental Setup The experiments of this section involve sharing links to two types of malicious content to check for the presence of different countermeasures. First, we want to test platforms against the presence of URL filtering mechanisms. For example, a social network may check whether the shared URL is flagged as malicious by existing URL blacklists, e.g., Google SafeBrowsing [74]. Accordingly, we searched for URLs on PhishTank [172] and verified that the URLs are also blacklisted by Google SafeBrowsing [74]. We used a total of three different blacklisted URLs across platforms, all with the same characteristics, due to their short uptime before being deactivated. Second, we want to check whether platforms proactively scan the content of web pages for malicious content. To this end, we created unique links to our server to download the trojan Win32.Virut. For IMs, we did not perform such an experiment as downloading mobile apps through a browser is not a major attack vector.

When running our tests, we also monitored the exact point where we can observe the effects of any countermeasures. In our analysis, we considered two points: when posting the URL, and when creating the link preview. Table 3.6 shows the result of our analysis.

3.3.1 URL Posting

The first aspect that we monitored during the execution of our experiments is whether the platform accepts malicious URLs. Only Twitter detected the blacklisted URL as malicious and prevented posting altogether. Also, Twitter showed a warning message:

Sharing Type			Social Networks										Instant Messengers									
Test	Resource	Observ.	Facebook	Twitter	Vk	LinkedIn	Pinterest	Tumblr	Medium	Xing	Plurk	MeWe	Instagram	Messenger	Snapchat	WhatsApp	Skype	Line	Viber	KakaoTalk	Telegram	Slack
Direct	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-
		Preview	●	○	◑	○	◑	◑	○	●	○	◑	-	-	-	-	-	-	-	-	-	-
	Block. URL	Posted	●	×	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	○	●	×	●	◑	-	●	-	●	●	●	●	●	●	●	●	●	○	○
Client Red.	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	
		Preview	●	●	●	○	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	
	Block. URL	Posted	●	●	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	●	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
Server Red.	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-
		Preview	●	○	◑	○	◑	◑	○	●	○	◑	-	-	-	-	-	-	-	-	-	-
	Block. URL	Posted	●	×	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	○	●	○	◑	◑	-	●	-	◑	●	●	●	●	●	●	●	●	○	○

Table 3.6: Test results when sharing a malware and a blocklisted URL.

This request looks like it might be automated. To protect our users from spam and other malicious activity, we can't complete this action right now. Please try again later. All other platforms did not show any error or warning messages and created a URL preview instead.

3.3.2 Preview Creation

Social media platforms can detect malicious URLs in later stages of the URL processing pipeline, e.g., when fetching the resources. However, our analysis revealed that the vast majority of platforms do not seem to implement any security check.

Malware When sharing the malware program, all platforms correctly retrieved the binary from our server. However, as the binary program does not contain HTML code, platforms tend to render a bare-minimum link preview (i.e., Facebook, Xing), possibly prompting the user to provide more information (i.e., VK, Pinterest, Tumblr, and MeWe) or render no preview at all (i.e., Twitter, LinkedIn, Medium, and Plurk). Also, all platforms did not show any error message or warning, and, clicking on the link preview results in downloading the malware program.

Blacklisted URL When sharing a blacklisted URL, only one platform, i.e., LinkedIn, detected the malicious URL after posting. Here, LinkedIn modified the text of the link to point to a redirector page (`linkedin.com/redir/phishing-page?url=`URL). When a user clicks on the preview, Linkedin shows an informative page explaining that

the site was blacklisted according to Google Safe Browsing, thus blocking access to the target URL. In spite of repeated attempts, the user account was not deactivated.

Sixteen social media platforms over 18 treated the blacklisted links as regular links: their bots visit the page and render a preview based on the specified meta tags (if any) or fall back to parsing HTML, when possible. Eight social media platforms (Facebook, VK, MeWe for SNs and Messenger, Snapchat, Line, Viber, KakaoTalk for IMs) created a rich preview with no distinguishable difference from a regular innocuous link. The remaining eight platforms either showed partial information (page title and host, but no image and no description) or did not render a preview at all, due to their implementation.

3.3.3 Takeaway

The analysis of this section intends to investigate the presence of possible mechanisms to prevent the distribution of malicious URLs on social media platforms. To summarize, our analysis makes the following findings:

- In general, our experiments could not find evidence of widespread use of countermeasures to prevent the distribution of malicious content at submission time.
- All platforms—except for Twitter and LinkedIn—do not show specific warnings or error messages to the users, indicating potential danger when clicking on the previews. Also, link previews for blacklisted URLs can contain the same semantic elements that are typical of previews of benign web pages, i.e., title, description, a picture, and the domain name.
- Two out of 20 social media platforms perform security checks on the posted URL. For example, LinkedIn uses the Google Safe Browsing API to detect malicious URLs. While Twitter forbids posting blacklisted URLs, LinkedIn accepts the URLs, but it replaces the URL in the preview with a link to an own warning page.
- Twitter and LinkedIn are the only two platforms implementing a form of defense. However, we could bypass these defenses by using server- and client-side redirections.

3.4 Attacks

So far, we studied the behaviors of social media platforms when processing both benign and malicious webpages, and we learned the various ways platforms could create link previews and validate URLs. This section will take a look at the link preview creation from an adversarial point of view. Here, we consider an attacker who intends to lure one or more users to visit a malicious webpage that is distributed over social media platforms. To do so, the attacker needs to hide their malicious intent by using, ideally, a benign-looking link preview. At the same time, as platforms may be validating URLs against blacklists, the attacker needs to avoid the detection of malicious URLs. In this section, we consider both problems. First, in Section 3.4.1, we present a set of shortcomings of social media platforms that allow attackers with different capabilities to craft arbitrary link previews, regardless of the actual content or purpose of the shared page. Then, in Section 3.4.2, we show how an attacker can bypass URL validation countermeasures.

We summarize our attacks in Table 3.7. Overall, our results show that all platforms are vulnerable to our attacks—except for two (Plurk and Medium) that we did not test with malicious URLs as they cannot limit the visibility of posts. Four platforms, i.e., Facebook, Xing, Plurk, and Slack, can be attacked by attackers who control the content of a webpage only. The remaining platforms are vulnerable to attackers who can also register domain names for the server distributing malicious pages.

3.4.1 Adversarial Analysis of the Link Previews Creation

The goal consists in creating a malicious web page whose preview, when shared on social media platforms, is similar to the preview of a benign webpage, requiring an attacker to be able to replace the content of each field with ones of their choice. In this section, we study the extent to which an attacker can arbitrarily influence the link preview creation considering two attackers with different capabilities, i.e., a first one that controls the content of a web page and an another one that can also register domain names. Table 3.7 shows the results of our analysis.

3.4.1.1 Crafting Fields

We evaluate the replacement of the preview fields considering two types of attacker models. The first one is a person that can create malicious web pages and upload them on a web server. This setting intends to model the common scenario where the attacker exploits vulnerabilities in existing servers or web applications to upload malicious content such as phishing pages. Since this attacker controls the web page content, they can modify the title, the description, and the images with ones of their choice. Here, the attacker can store the selected values in the meta tags or the standard HTML tags. In Table 3.7, we mark these field with "◆". However, such an attacker may not be able to alter the content of the domain name and the shared URL.

The second type of attacker possesses the capabilities of the previous attacker and extends them with the ability to register domain names. This scenario intends to model the typical attacker that registers fraudulent domain names to support their malicious

	Crafted Fields					Bypass		Attacker	
Name	Site title	Site descr.	Image	Host	Shared URL	Client Red	Server Red.	Blacklisted URL	Capability
Facebook	◆	◆	◆	◆	-	-	-	✓	Page cnt.
Twitter	◆	◆	◆	◇	-	✓	-	✓	Domain
VK	◆	-	◆	◇	-	-	-	✓	Domain
LinkedIn	◆	-	◆	◇	-	-	✓	✓	Domain
Pinterest	-	-	◆	◇	-	-	-	✓	Domain
Tumblr	◆	◆	◆	◇	-	-	-	✓	Domain
Medium	◆	◆	◆	◇	-	-	-	-	Domain
Xing	◆	◆	◆	◆	-	-	-	✓	Page cnt.
Plurk	◆	-	◆	-	-	-	-	-	Page cnt.
MeWe	◆	◆	◆	◇	-	-	-	✓	Domain
Instagram	◆	◆	◆	-	◇	-	-	✓	Domain
Messenger	◆	◆	◆	◆	◇	-	-	✓	Domain
Snapchat	◆	-	◆	◇	-	-	-	✓	Domain
WhatsApp	◆	◆	◆	◆	◇	-	-	✓	Domain
Skype	◆	◆	◆	◇	-	-	-	✓	Domain
Line	◆	◆	◆	-	◇	-	-	✓	Domain
Viber	◆	-	◆	◇	-	-	-	✓	Domain
KakaoTalk	◆	◆	◆	◇	◇	-	-	✓	Domain
Telegram	◆	◆	◆	◆	◇	-	-	✓	Domain
Slack	◆	◆	◆	◆	◆	-	-	✓	Page cnt.

Table 3.7: Summary of the evaluation of our attacks. We use "◆" when the attacker can change a field via HTML tags. We use "◇" when the attacker can replace the value of a field via the domain name of the malicious URL. We use "✓" when a bypass technique and attack succeeded. Finally, we use "-" when the field is not present or when we did not test the platform.

activities. Being able to register domain names extends the abilities of the previous attacker as it allows for crafting the domain name and shared URL too.

In the remainder, we present our analysis, discussing in detail what an attacker could do to change the content of these two fields. We grouped our results in five distinct classes based on the observed behaviors:

Link Previews without Domain Name One platform, i.e., Plurk, does not include any information regarding the landing page URL, i.e., neither the domain name nor the original URL. In this case, the creation of a crafted link preview is straightforward. An example of preview for Plurk is Figure 3.3b.

Instagram and Line do not show the domain name either. However, we point out that they show the original URL. In our experiments, we could not find a way to remove or replace the string of the shared URL from the preview. Accordingly, it can be changed only by an attacker who has full control of the URL string.

Replacing Domain Name using `og:url` In Facebook, we observed that when the URL of the shared webpage mismatches the `og:url` meta tag, the preview fields title, image, description and host are retrieved from the webpage hosted at the URL specified in the `og:url` meta tag rather than in the shared one. Nonetheless, the final landing page remains the URL of the shared web page. In this case, the attacker can assign to the tag `og:url` a URL of a benign resource, resulting in a preview that is entirely indistinguishable from a benign preview. Figure 3.3c shows such a benign-looking preview. The Messenger app shows the same behavior, but the attacker cannot remove the shared URL from the message text; due to the mismatch between the shared URL and the preview, we say that this attack is possible only for an attacker that can register domain names.

WhatsApp replaces the content of the host field only, showing the URL specified in the `og:url` meta tag. Also in this case, the shared URL cannot be removed from the message text, requiring the attacker to register a new domain name for this purpose.

Removing Shared URLs in IMs One IM platform, i.e., Slack, permits the editing of the content of sent messages. We verified that a user could edit the URL string of a message too, after the creation of the preview, effectively eliminating this field from the rendered preview. The platforms Snapchat, Skype and Viber remove the URL from the message text after posting, although we observe that they include the domain name in the preview, which is extracted directly from the shared URL. We could not find a way to replace the domain name with an arbitrary string. Therefore, this attack may not be successful for an attacker controlling the webpage content only.

Replacing Domain using `og:site_name` During our experiments, we discovered that three platforms, i.e., Xing, Telegram and Slack, replace the domain name with the content of the `og:site_name` meta tag.

As mentioned before, Slack allows removing the shared URL from the message text after posting the link. Accordingly, an attacker can generate a preview that looks like

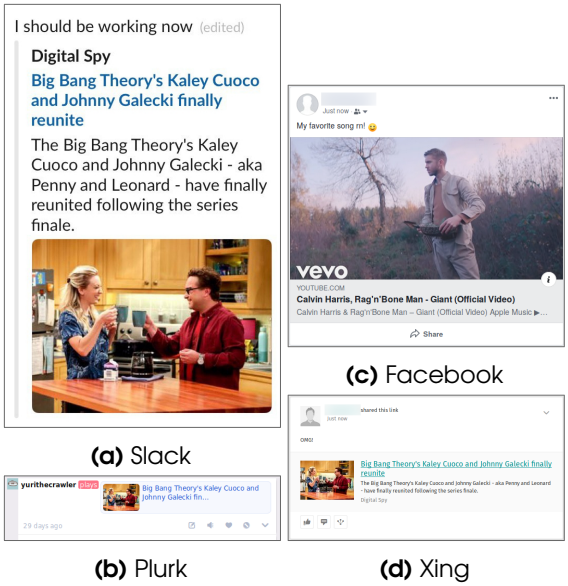


Figure 3.3: Examples of previews that can be crafted by an attacker who controls the content of a webpage.

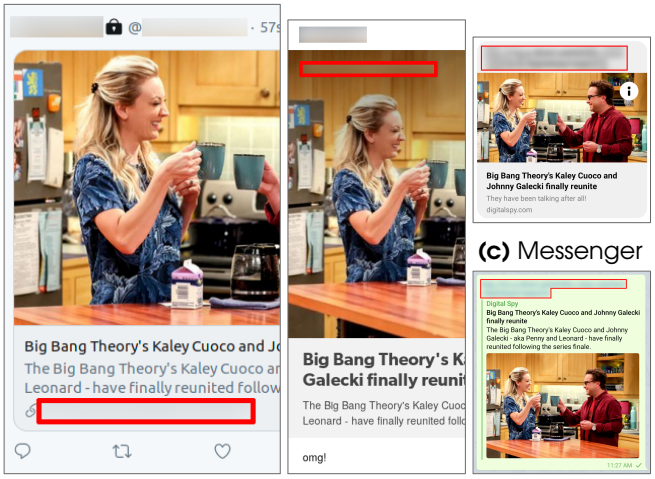
a benign one only by controlling the HTML code of the page. Figure 3.3a shows an example of such a link preview.

Xing does not include the original URL; therefore, controlling the web page content is sufficient to craft a URL preview where the domain name is replaced with the site name. Figure 3.3d shows such a link preview.

Then, replacing the domain name of Telegram's preview with the `og:site_name` meta tag may not be sufficient as Telegram includes the shared URL that we could not remove. Accordingly, the creation of a Telegram's preview is more suitable for an attacker that can register domain names.

3.4.1.2 Attacks

To summarize, our analysis shows that it is possible to create an attack against each platform. Our attacks can create entirely indistinguishable link previews against four platforms, i.e., Facebook, Xing, Plurk, and Slack, by changing only the content of the malicious web page. In three cases, the attacker needs to exploit seemingly innocuous behaviors of the platforms to achieve their goal. For example, on Facebook, the attacker can replace the domain name with the domain of `og:url` meta tag, whereas for Xing and Slack, the attacker can replace the domain name by using the `og:site_name` tag. As Slack includes the shared URL too, the attacker can also remove the original URL from the preview after its creation. We point out that, in all these four cases, even when the attacker replaces or hides the domain names and the shared URLs, the landing pages, i.e., the malicious pages, of the link preview remain unchanged. When the attacker controls the domain name, then the remaining platforms can be targeted as well. Figure 3.4 show examples of partially crafted link previews. The areas in red



(a) Twitter **(b)** Tumblr **(c)** Messenger **(d)** Telegram

Figure 3.4: Examples of crafted previews that always show the domain name. The red box shows the position of the domain name. We blurred domain names and user names as they contain strings that can reveal the affiliations of the authors of this paper.

contain either the domain name or the original URL. Finally, the evaluation for two platforms, i.e., Medium and Plurk, was limited to the generation of the previews. On these two platforms, we did not share any malicious URLs as they cannot restrict the visibility of the shared content.

3.4.2 Bypassing Countermeasures

When sharing malicious content, social media platforms may detect the maliciousness of the shared web page. As shown in Section 3.3, only two platforms can detect when a URL is known to be distributing malware by using, e.g., Google Safe Browsing [74]. In this section, we focus on these two platforms and show that, despite the efforts of validating URLs, it is possible to bypass these controls by creating ad-hoc web pages. In this page, we consider two approaches that are based on the findings of Sections 3.2.1.4 and 3.2.2.

3.4.2.1 Redirections

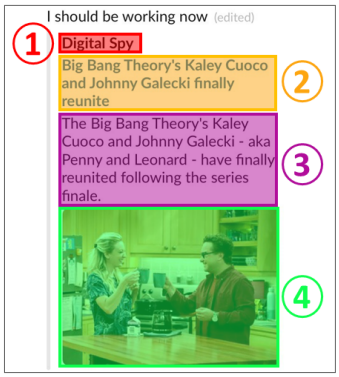
During our experiments of Section 3.2.1.4, we observed that all platforms except for one (Facebook) do not support HTTP redirections. As a result, those platforms may not be able to determine the next URL in the redirection chain, and accordingly, they should fail in verifying whether the URL is malicious. We tested our hypothesis and confirmed that client-side redirections could effectively bypass both Twitter and LinkedIn URL validation. The evaluation with redirections is summarized in Table 3.6.

However, interestingly, we also found out that it is possible to bypass the URL filtering of LinkedIn with a server-side redirection, i.e., 30x response. Here, we suspect that LinkedIn does not validate the `Location` header of the HTTP response sent by the redirector.

```
<head>
<title>HTML title</title>
<meta property="og:site_name" content="①">
<meta property="og:title" content="②" />
<meta property="og:description" content="③">
<meta property="og:image" content="④"/>

<meta name="twitter:title" content="②">
<meta name="twitter:description" content="③">
<meta name="twitter:image" content="④">
</head>
<body>
<!-- Malicious content -->
</body>
```

Listing (3.2) Example of Malicious Page Shared on Slack



(a) Rendered Preview

Figure 3.5: Example of Malicious Link Preview.

3.4.2.2 Link Cloaking

As a final step, an attacker may resort to cloaking attacks. The analysis Section 3.2.2 showed that the source IP and the user agent strings of the social media platforms are unique, and an attacker can leverage on these features to change the behavior of the servers selectively. For example, when the incoming request matches one of the known signatures, the server will deliver the benign web page for link preview creation. Otherwise, the server delivers the malicious web page.

3.5 Discussion and Recommendations

In this section, we discuss our results and distill a set of recommendations for social media platforms towards the creation of more reliable link previews.

3.5.1 Variety of Layouts and Processing Rules Can Lead to Underestimate the Risk

Our results show a great variety of layouts used by the platforms under evaluation. We distinguished 14 distinct templates for link previews. Also, we observed that the same platform could create many variants of the same template, for example, by removing or replacing fields.

The variety that we observed suggests that there is no general consensus on (i) which fields constitute a link preview, (ii) under which circumstances fields are displayed, and (iii) the processing rules and priority. The lack of consensus can have a dramatic impact on the way users evaluate the trustworthiness of a preview. As users can be exposed to different layouts, they may neglect the importance of a field, underestimating the overall security risks of a link.

(R1) Standardize Content and Construction Rules of Link Previews Our first recommendation is to define and agree on the content of link previews, and the exact rules to construct them.

3.5.2 Distrustful Scenario

The scenario in which social platforms operate is characterized by distrust. On the one hand, social platforms cannot verify the truthfulness of webpages content. For example, they cannot decide whether an image or a title is appropriate for a given page. Accordingly, social platforms cannot trust webpages. On the other hand, users can leverage on their own experiences and skills to navigate the web and inspect both URLs and the circumstances that led them to see those URLs looking for warning signals, indicating that pages may be dangerous. Experienced users may be trusting webpages they are familiar with, e.g., their web email provider; however, in the general case, they will not trust any page.

In a scenario with these trust relationships, social media platforms act as intermediaries between web pages and users, providing to the latter syntheses of the former. In playing such a role, social platforms should avoid introducing interpretations of the content of the webpages or using processing rules that can hide or distort the preview of the page. Also, social platforms should enforce the presence of security-relevant fields that users can use to decide whether to click, i.e., domain names, and original URLs. While most of the social platforms under test include a domain name or the original URL, four of them, i.e., Facebook, Xing, Plurk, and Slack do not satisfy such a requirement. From the analysis of these four platforms, we derive the following recommendations:

(R2) Show Domain or URL As reported in Table 3.3 and further detailed in Section 3.4.1.1, the link preview created by the social network Plurk does not include any

host field, and there is no URL in the post text. As this information is significant in assessing the trustworthiness of the link preview, we include as part of our recommendations that link previews must include either the domain name or the shared URL. Among the platforms under evaluation, only Plurk does not comply with our recommendation.

(R3) Limit Edits of Posts or Refresh Previews Platforms may want to allow users to edit previous posts. In these cases, they should forbid changing the shared URLs. Alternatively, when changing the URL is admitted, platforms should re-build the link preview and replace the old preview with the new one. In our experiments, and in particular in Section 3.4.1.1, we observed that Slack allows users to remove URLs from previous messages without updating the link preview. This feature can be misused as shown in Figure 3.3a, especially if the domain name field contains an arbitrary string rather than the actual domain or URL.

(R4) Create Preview Without Retrieving Referred Pages Platforms should create link previews using data items contained in the code of the landing page. When the landing page contains external links such as `og:url`, platforms could consider such resources as long as they are in the same domain as the landing page. Furthermore, platforms should not use such URLs to build the entire preview. In Section 3.4.1.1, we observed that Facebook creates the entire preview by using the content of the URL in the `og:url` tag, and an attacker can hide a malicious webpage by creating a link preview with a YouTube video using only the `og:url` meta tag (see, Figure 3.5a).

(R5) Type Fields In Section 3.2.1.2 we observed that, in a few social platforms, it is possible to override the content of the domain name field by adding the `og:site_name` meta tag. When the platform additionally does not include the shared URL in the text field of the post, as observed in Section 3.4.1.1 for the social network Xing, the final link preview contains no trusted information on the URL, as the domain field can contain an arbitrary string. Therefore, we recommend that each field of a link preview should have a well-defined type, e.g., image, description, title, domain, and URL. Then, when creating a preview, platforms should not use the content of a field of type t_1 to fill a field of a different type t_2.

3.5.3 Upstream vs Downstream URL Validation

During the lifetime of a link preview, there are different points in time when malicious links can be detected, e.g., when platforms accept the URL and when users click on the preview. In the remaining, we discuss where and how such a check should be enforced.

(R6) Do Upstream URL Validation When testing social media platforms against phishing URLs, we observed that not all browsers could show Google Safe Browsing warning messages before loading malicious URLs. In particular, we verified that the in-app browsers as used per default configuration by Messenger, Slack, Telegram,

Line, Instagram, and WhatsApp (both on Android and iOS) do not show any warning when loading our phishing URLs. Also, we verified that external browser apps might not reliably show Safe Browsing warnings. We reproduced such behavior on Chrome Browser 76.0.3809.123 for iOS 12.4.1, Chrome for Android (Android 9,Pixel Build/PQ3A.190801.002 and Pixel 2 Build/PQ3A.190801.002), Safari (12.1.2 Mobile), Brave Browser for Android (1.3.2 based on Chromium 76.0.3809.132), and Firefox Focus for Android (8.0.16). Only one mobile browser, i.e., Firefox for Android (68.1), showed the warning correctly. We point out that we used the default configuration of both all tested apps and the operating systems. Finally, desktop browsers were more consistent than the mobile ones in showing the warning. Here, we tested Chrome Browser (77.0.3865.75 for Ubuntu 18.04), Brave Software Browser (0.68.132 based on Chromium 76.0.3809.132 for Ubuntu 18.04), and Firefox (69.0 for Ubuntu 18.04). Independent non-academic research confirmed the presence of a discrepancy between Google Safe Browsing mobile and desktop. See, for example, [181, 99].

The reasons for such a discrepancy are not fully understood, and further research is required. Nevertheless, such results indicate that browsers may fail to or will not detect malicious URLs, and, accordingly, browser-side countermeasures should not be considered as a bulletproof last line of defense. Based on that, we recommend developers to implement upstream URL validation during the generation of link previews. Among the 20 platforms we verified, only two implement such a mechanism.

(R7) Do Proper URL Validation An HTTP agent can reach web resources by following chains of redirections. While in the past redirections were only implemented via HTTP response codes and the refresh HTML meta tag, nowadays redirections are also implemented via JavaScript code. When validating URLs, it is fundamental that all URLs of a redirection chain are validated as well. Unfortunately, the only two platforms implementing a form of URL validation (Twitter and LinkedIn) did not validate URLs during redirections, allowing attackers to bypass their countermeasures. Table 3.6 sums up the results of our experiments with these two social networks.

3.5.4 Ethical Considerations

Our experiments raise the valid ethical concern of sharing malicious content on social media platforms. For example, users not aware of our experiments may click on our previews and become victim of an attack. To avoid attacking users, we limited the visibility of the shared malicious links of the platform accounts we control. When the platform did not support limiting the post visibility, i.e. for the social networks Medium and Plurk, we did not share the phishing link and, instead of distributing the Win32.Virut malware, we used the innocuous EICAR test file, used to test antivirus software.

The second concern of our experiments is sharing malware from our servers. The main risk of these experiments is that both the network and the domain name of our institute may be blacklisted, affecting the work of the research and administration staff. To avoid such a risk, we registered a first-level domain name and moved our servers on Amazon Web Service EC2.

Summary

In this chapter we presented the threat of deceptive link previews on social platforms. Our study focuses on link preview creation processes and how attackers can exploit them in an adversarial way, leading to the creation of benign-looking preview for malicious web pages.

Our characterization of link previews shows that most platforms have a different rendering format for the same meta tags. This variability might confuse users about which preview fields are security critical, leading them to uninformed security decisions. We further observed that, in four platforms, attackers can craft benign-looking link previews leading for malicious webpages, fooling even tech-savvy users. Crafting a benign-looking preview for the remaining 16 social media platform requires only the ability to register a new domain. Finally, we observed that only two platforms implement checks against the distribution of malicious links, and that these defences are easily bypassable. We analyzed the impact of misleading previews on users' behavior, evaluating the resulting security risks, and suggested seven recommendations for possible improvements.

In the next chapters, we will present and dissect a second case study of technology-enabled deception attack using the browser technology.

4

Deception when Browsing: Clickbait PDFs

This chapter presents the second case study of technology-enabled deception attacks, exploring *whether new integrations, technological developments, or communication platforms can be exploited to mount deception-based attacks (RQ1).*

We introduce and present *clickbait PDFs*, PDF files which leverage seamless browser rendering to disguise themselves as regular web pages and attract user clicks. Clickbait PDFs include deceptive web user interface elements and, in addition to visual and textual deception, they rely on contextual elements such as the web environment in which they are distributed, most often through poisoned search results. Our study is the first to comprehensively characterize this threat. It shows that PDF files, and especially clickbait PDFs, are no longer merely ancillary tools used in phishing email campaigns. Instead, they now serve as entry points for a variety of web-based attacks, enabled by the links embedded under their visual baits.

We analyzed a real-world dataset of 176 208 samples and observed that 89% of them are distributed via search engine poisoning in large volumes and over prolonged periods. These features, combined with the embedded malicious links, distinguish clickbait PDFs from traditional malicious PDFs, which typically contained malware [121, 120] and were distributed through short-lived email campaigns [205]. To support future work in this area, we release the code and metadata related to this study [55, 53].

4.1 Background and Methodology

Before presenting our study, we define clickbait PDF attacks (§ 4.1.1) and outline our methodology (§ 4.1.2).

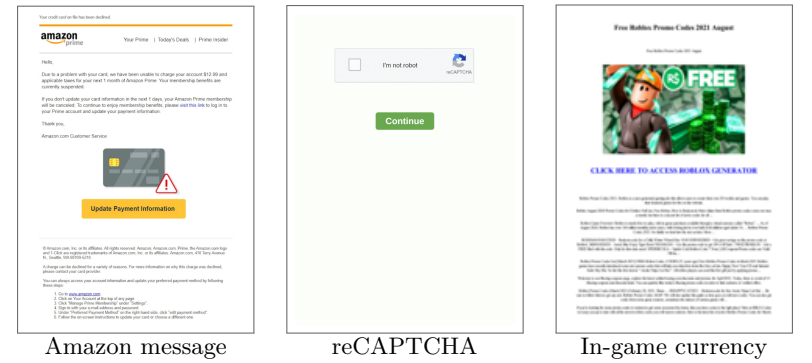
4.1.1 Background

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) [120, 121, 229, 205].

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 [176, 147]. Clickbait PDFs rely on a wide variety of visual deceits to lure users into clicking on specific areas of the documents. Figure 4.1 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.

4.1.2 Problem Statement and Methodology

The threat posed by clickbait PDFs has been object of concern by leading security teams in industry [147, 176]. 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.



Amazon message | reCAPTCHA | In-game currency

Figure 4.1: Examples of clickbait PDF files.

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., [205, 240, 94]) 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 [165], cross-linking resources [258], and use of benign websites for linked resources [97]. 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 [214], for 30 days.

4.2 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 (§ 4.2.1). Then, we report the procedure we followed to extract clusters of visually similar documents (§ 4.2.2).

4.2.1 Dataset

Data Sources. The sources of our dataset are two industrial partners, i.e., Cisco and InQuest Labs[1], 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 4.2 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%.

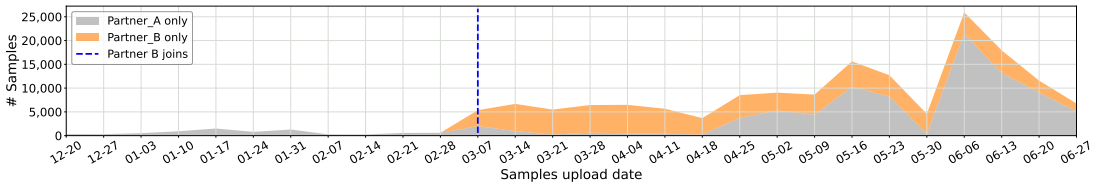


Figure 4.2: Weekly sum of daily uploads of the two datasets, stacked.

Data Collection. Cisco retrieves data from VirusTotal [244] (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 [183], 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

[1]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.

by InQuest Labs is publicly available [91]. 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).

4.2.2 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., [205, 240, 94]). 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., [2]). 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 [80, 79]. Thus, we utilize the DeepCluster framework [27], 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 [115]) 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` [163] 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 [59], using the learnt embeddings as distance metric (as in, e.g., [27]): 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 8.1 (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 8.1.1.

4.3 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.

4.3.1 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` [58], 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 4.3. 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.

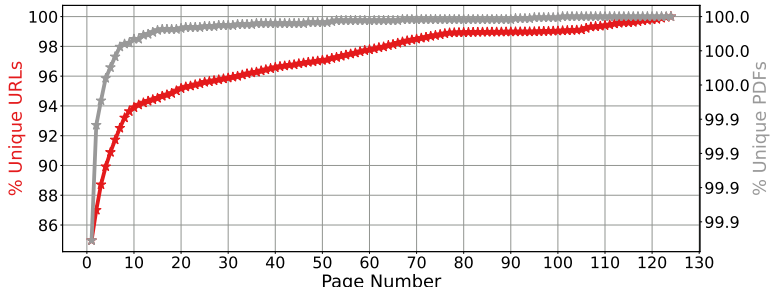


Figure 4.3: Distribution of *bait URLs* per PDF page (red) and number of unique PDFs embedding them (grey). The graph shows the .95 quantile of PDF pages (max: 524) for visibility reasons.

4.3.2 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) [74]. 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 [171]. 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`[236]. 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 8.2 (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.

4.3.3 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

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

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 § 4.6.2. 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 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 Section 8.1.2. 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.

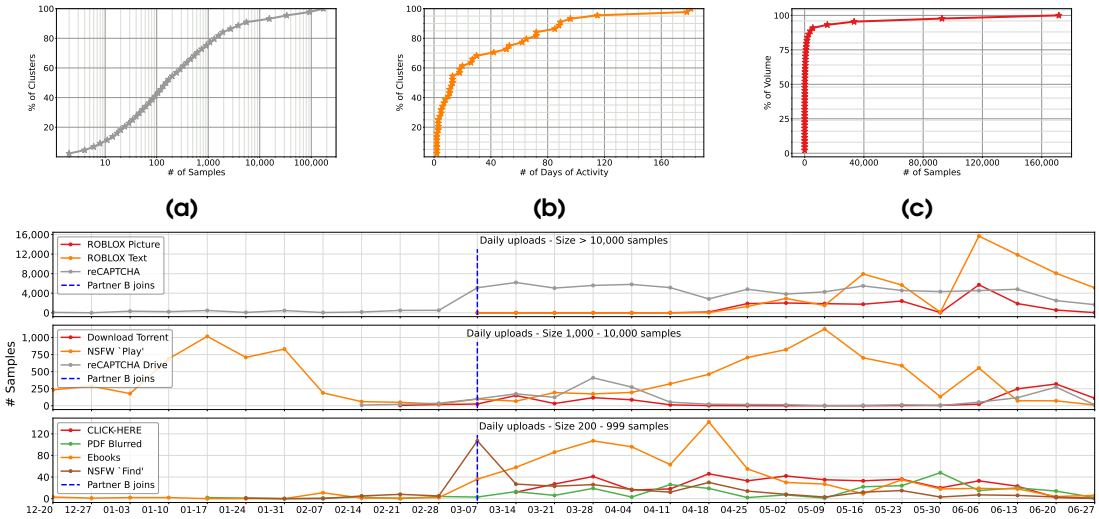
Overall, the URL analysis returned eight different outcomes, reported in Table 4.1 and detailed in the following. *Malicious advertisement and Data harvesting (16 clusters, symbol: ○)*: in this attack, the user is redirected to a personalized advertisement page or is prompted to provide personal data to receive a reward (similarly to, e.g., [103]). *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.

4.3.4 Summary of Findings

The goal of this section was to analyze representative URL samples for all the clusters obtained in § 4.2.2, 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.

4.4 Clusters Characterization

We now characterize each of the 44 clusters identified in § 4.3.3. First, we look at volumetric and temporal properties of each cluster (§ 4.4.1). Second, we analyze the visual deceits of each cluster (§ 4.4.2), providing a categorization of the type of fraudulent activities and their visual elements. Then, we explore the effectiveness of the VirusTotal maliciousness score (§ 4.4.3). Finally, we study the geographical reach of each cluster by observing the languages used in their text (§ 4.4.4).



(d) Clusters volumes grouped by cluster over time.

Figure 4.4: 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.

4.4.1 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 4.4a and 4.4c).

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 4.4d 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 4.4b). While few clusters operate for 60 days or more (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 [205]. Table 4.1 shows size, prevalence, duration and temporal location for all the 44 malicious clusters.

4.4.2 Visual Deceits

Attackers use visual deceits to lure victims into clicking [15, 44]. 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

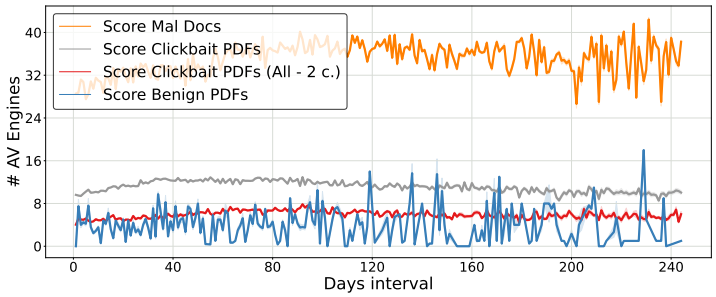


Figure 4.5: VirusTotal score comparison between MalDocs and clickbait PDFs. Data collection until Aug, 18th.

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.

4.4.3 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 [266] 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 4.5 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 8.2), 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.

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

4.4.4 Languages

We further investigate whether clusters target specific geographical areas by using language information obtained via the Google Vision API [73] 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 4.2 and 8.3.

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., [98]), 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.

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

4.5 Distribution Vectors

In this section, we present two experiments to confirm the use of two distribution vectors. In § 4.5.1, 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 § 4.5.2, we go through search engine results looking for clickbait PDFs distributed via Search Engine Optimization (SEO) attacks.

4.5.1 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. [205], which is hard to get in practice, or subscribing to services specialized in malicious email feeds, e.g., MX Mail Data [42], which costs tens of thousands of dollars.

As email phishing campaigns target a large number of addresses at once [205], 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 [243]. Accordingly, we use VT tags as an additional data source in this analysis.

Results. Table 4.3 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 § 4.4.3, 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, 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.

4.5.2 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* [165], 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* [258], 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 [96], as search engines tend to rank them more quickly than newly registered domains. We verified via GSB [74] 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., [121, 120]) 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) [126]. 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

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 4.4: Search engines results.

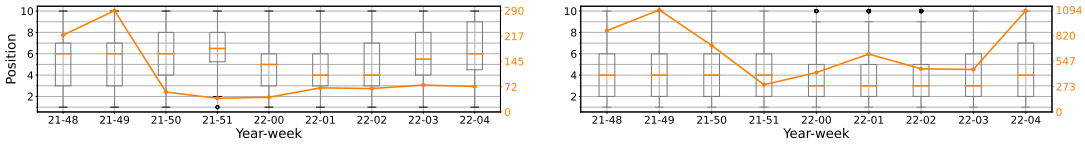


Figure 4.6: Number and position of PDFs found on Google (on the left) and Bing (on the right) over time.

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 8.1.3 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 [214]. 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 4.4 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 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 4.6 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.

4.6 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 8.1.4 further discusses possible limitations of our study.

4.6.1 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 [176], 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.

4.6.2 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 [74] and the rule-based ad-blocking provided by EasyList and EasyPrivacy [64] 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 § 4.3.2, 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 § 4.3.2. Then, we retrieved EasyList and EasyPrivacy blocklists via the Wayback Machine [92], 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 [211, 33] in our context. Established techniques [211, 208] 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 [33] offer flexibility in this similarity metric, allowing variations such as N differing PDF objects.

We evaluated Hidost's [211] 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 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 § 4.2.1). 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 [33] did not lead to better results, as they concern the similarity metric and not the feature selection. In conclusion, although existing methods such as [211, 33] 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 § 4.4.3 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 § 4.5.2). 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 § 4.5.2 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 § 4.4.1, 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 [176], 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.

4.6.3 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 [53] and [55].

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 § 4.5.2) and raised awareness of the ongoing threat following the state of the art for vulnerability notifications [215, 127].

Summary

In this chapter we introduced clickbait PDFs, PDF files which do not contain any malware but embed a malicious link under a visual bait shown in the first page. 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, identifying six different types of attacks in total. As part of our findings, we observed that several clusters include visual elements typical of web pages, e.g., fake reCAPTCHA buttons, positioning clickbait PDFs as a threat leveraging contextual deception to trick

users into clicking on embedded malicious links. Our experiments on clickbait PDF distribution vectors prove that clickbait PDFs mainly spread through SEO attacks (89% of our dataset), a delivery context contributing to the deception of this threat. Moreover, 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 [205].

After dissecting and characterizing clickbait PDF files, the next chapter investigates the operational side of this phenomenon, focusing on the websites and web services attackers leverage to distribute malicious files at large scale.

5

Web Infrastructure in Clickbait PDF Campaigns

In the previous chapter, we comprehensively studied the threat posed by clickbait PDFs, PDF files leading to Web attacks and distributed via the web context. In this chapter we shift the focus to the websites and web services hosting clickbait PDFs, which are key for the success of the attack, tackling our RQ2 on *how attackers leverage web infrastructure to support deception-based attacks*.

Studying the supporting infrastructure has been a critical aspect when analyzing other similar threats, such as drive-by download [155], phishing pages [170], spam [133], or when looking at server compromise [28, 97] and their role in the attacks [170]. Despite previous research efforts, these findings do not directly apply to clickbait PDFs, as they focus on threats with different core features, as volume of the campaign or its temporal duration [171, 170, 155, 132], or examine web infrastructure with limited scope [133, 132].

To investigate this, we performed a large-scale, real-time analysis of hosting infrastructure, examining 4 648 939 clickbait PDFs delivered by 177 835 distinct hosts over a 17-month period. Our findings highlight a heterogeneous ecosystem, with hosts categorized into three primary types of hosting setups. Additionally, we detected eight commonly used software components that enable file uploads and appear to be systematically exploited to distribute clickbait PDFs. We engaged in responsible disclosure by notifying affected parties at scale through a coordinated vulnerability notification effort.

To support reproducibility and future works, we released the code for identifying and monitoring the abused websites and web services [54].

5.1 Scope and Contributions

In this chapter, we shed light and provide a comprehensive description of the infrastructure behind clickbait PDF attacks. We use the broader term *abused infrastructure* to indicate a large amount of websites, managed by one or more providers, or part of the same hosting service, whose usage is inappropriate, often illicit, resulting in significant harm to the owner and its users. The term can include hosts or domains that are maliciously registered or compromised [43, 142, 155] as well as hosts running on free subdomains of known providers [195]. In the context of clickbait PDFs, the supporting infrastructure is the ensemble of websites, services and providers whose resources are being misused by attackers to host clickbait PDFs. Figure 5.1 illustrates the different domains involved in clickbait PDF hosting and delivery as part of the malicious search engine optimization (SEO) campaign (`domain1.com`, `subdomain.domain2.com`, `domain3.org` and `domain4.net`). In the context of these SEO campaigns, all domains involved both host clickbait PDFs and serve as backlink sources to one another. The PDFs are interlinked, each containing hyperlinks to other PDFs hosted on different domains, forming a dense backlink graph across the infrastructure. Since SEO attacks are the main distribution vector for clickbait PDFs [P2], the rest of this paper focuses on clickbait PDFs distributed through SEO.

Below, we present our research questions and outline the contributions and framing of this study with respect to a recent work in this field.

The overarching goal of this study is to observe the web infrastructure abused for the distribution of clickbait PDFs, investigating specific properties concerning its volume

and evolution in time. The first challenge we undertake (**Research Question 1**) is to understand its composition in terms of hosts or services, for example by identifying Autonomous Systems or any specific hosting services involved, and to which extent. We tackle this research question in § 5.3.1. Next, we ask ourselves how attackers acquire upload capabilities to these domains (**Research Question 2**). Specifically, we look for security-related properties, as the presence of outdated, vulnerable or misconfigured software components which might have been exploited by attackers to gain the ability of uploading clickbait PDFs. We investigate multiple security properties and report our findings in § 5.3.2. Following, we focus on the duration and volume of the abuse (**Research Question 3**). We define the duration of abuse by monitoring the online status of all clickbait PDFs in our dataset with the granularity of a single day (§ 5.4.1), and its volume by observing the distribution of clickbait PDFs over the types of hosting we previously identified (§ 5.4.2). Lastly, we focus on measures that could be taken to help mitigating the spread against clickbait PDFs, ultimately protecting users and improving the security of the abused hosts. Existing protection methods, as blocklists, provide limited protection for users (§ 5.5.1), thus, we evaluate the effectiveness of responsibly disclosing the issue to affected parties (**Research Question 4**) (§ 5.5.2), observing as impact indicators both the number of PDFs that were cleaned up and the domains that did (or did not) see any further upload.

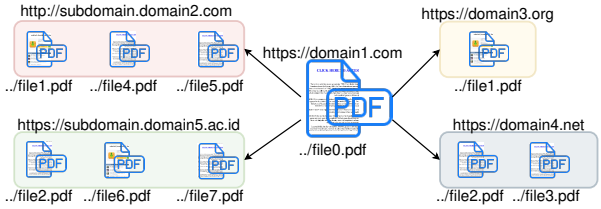


Figure 5.1: The interconnections between clickbait PDFs.

5.2 Dataset and Pipeline

5.2.1 Main and Seed Datasets

Answering our research questions requires knowledge of the hosts serving clickbait PDFs, for example in the form of a list of URLs leading to these PDFs. A source of URLs is given by clickbait PDFs themselves, as clickbait PDFs include URLs to other clickbait PDFs as backlinks (see § 2.1.1.3, 5.1, and Figure 5.1). We leverage this property to construct a first dataset of clickbait PDFs, the *Seed DS*, acting as source of URLs to other clickbait PDFs. By visiting these URLs and downloading the corresponding PDFs we build the dataset for this study, *Main DS*. The inclusion of a downloaded PDF to the *Main DS* (as well as to the *Seed DS* in the previous step) is subject to the evaluation of SEO-specific properties (detailed in § 5.2.1.2 below), ensuring that no benign or non-clickbait PDF is included.

5.2.1.1 Data Collection

Our starting dataset, *Seed DS*, counts 609 576 PDFs with unique SHA-256 signatures, covering a period of 17 months (from March 14th, 2022 to June 26th, 2023). The nine-month gap between the start of our study and the end of Stivala et al.'s raises questions about whether those clickbait PDFs are still online and part of an attack campaign, which we address by collecting up-to-date clickbait PDFs provided by two industrial partners, who retrieve them from VirusTotal. The two partners contribute unevenly, accounting for 69% and 29% of the entire dataset, respectively.

We start downloading PDFs to construct the *Main DS* after a three-month setup phase. This second data collection lasted 13 months, during which we monitored 4 648 939 `.pdf` links. URLs that are unreachable, do not serve PDFs, or serve non-clickbait PDFs are discarded, resulting in 2 710 959 URLs that returned a clickbait PDF at least once during the Main phase of the study. Table 5.1 reports the number of clickbait PDFs in the *Seed DS* and the number links extracted from them, as well as the number of clickbait PDF observed online. § 5.2.2.1 reports the implementation steps behind our data collection.

5.2.1.2 Filtering Criteria

We implement two filtering criteria to limit the inclusion of benign or non-clickbait PDFs in our datasets, in line with prior works [P2]. Identifying clickbait PDF involves verifying the presence of SEO characteristics, which are not visible from the `.pdf` URLs but can be observed by inspecting the PDF structure and content (see § 2.1.1.3 and Figure 5.1). We thus download and parse PDFs, ensuring the presence of SEO characteristics in two ways before adding them to the *Seed DS* and *Main DS*. These criteria (hereinafter *SEO metric*) ensure the presence of at least five `.pdf` links in total, relaxed from the original ten, and a mean number of at least one `.pdf` link per page, consistent with [P2]. This change was due to our different data sources (VirusTotal and backlinks in clickbait PDFs) where the distribution of benign documents is much lower than that of search engines like Google and Bing. The lower threshold is designed to

	DS	Setup Phase	Main Study
Start		2022-03-14	2022-06-22
End		2022-06-21	2023-07-26
PDFs	□	105 598	503 978
of which SEO	□	66 614	384 601
Extracted `.pdf` links	-	1 350 201	4 648 939
of which online & SEO	■	-	2 710 959

Table 5.1: Volume of unique PDFs in *Seed DS* (□) and *Main DS* (■), and unique `.pdf` links extracted from them.

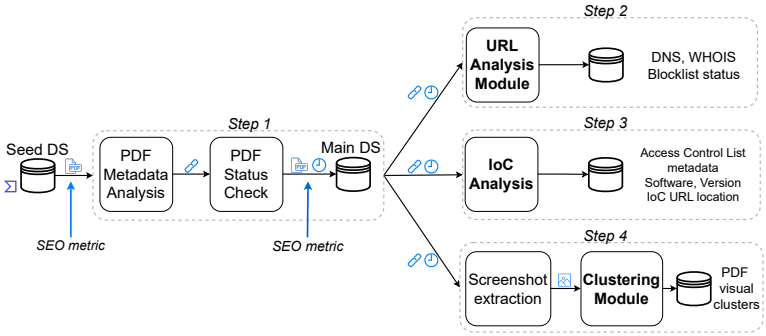


Figure 5.2: *Grape* modules and I/O data connections.

include a large number of clickbait PDF documents while minimizing false positives. § 5.6 reports on the accuracy of this metric.

5.2.2 The *Grape* Pipeline

The initial three-month setup phase are necessary to build *Grape*, shown in Figure 5.2, a modular pipeline running daily in real-time. *Grape* ingests and processes millions of tiny PDF-related pieces of information from various sources every day. When "mashed" together, these pieces reveal valuable insights into the clickbait PDF threat. We release the code of *Grape* at https://github.com/emerald1010/hosts-supporting-clickbait-PDFs.

The first module (*Step 1*) processes the PDF binaries received from our industry partners, extracting useful metadata such as the embedded URLs. These are fed into the *PDF Status Check*, which visits them and defines their online or offline status. These pairs `(URL, datetime_information)` constitute the basis of the *Main DS* and are the input of all following modules. Specifically, we fetch DNS and WHOIS of all URLs in the *Main DS* (*Step 2*), visit the websites hosting online PDFs looking for indicators of compromise (*Step 3*) and, finally, download online PDFs and extract the screenshot of the first page (*Step 4*) to determine the groups of visual baits. The modules are orchestrated and monitored via an instance of Apache Airflow. Following, we detail the behavior of each module.

5.2.2.1 PDF Analysis Module

We begin by choosing PDFs from the *Seed DS* that meet the SEO metric. Next, we extract their URLs and metadata (*PDF Metadata Analysis*), and subsequently verify the online status of those URLs leading to a PDF (*PDF Status Check*). In the *PDF Metadata Analysis*, the URLs are obtained by reconstructing the PDF tree with a modified version of the open-source library `peepdf` [58], by navigating the tree breadth-first looking for nodes encoding URLs (e.g., `URI`) or whose parent node's attributes include `Subtype Link`, `Rect` and either `Type Annot` or `Type A`. This approach was preferred to a simple string matching (e.g., looking for `http://`-like strings) as it allows extracting URLs in compressed streams. Lastly, we collect the document title by inspecting its `Document Information Dictionary` and obtain the screenshot of the first page via the `Poppler` [163] utility using 150 dots per inch.

PDF Status Check consists of a module performing daily HTTP requests to the extracted `.pdf` links, de-facto recording the uptime of each linked PDF. We monitor each link on a daily basis starting from the day of its initial observation, and continue until it remains offline for three consecutive days. A URL is considered offline when its `Content-Type` header is different from `application/pdf`, or if it returns a status code >= 300. To reduce the load on the target domains, we initially perform `HEAD` requests, and proceed with a `GET` only if the above criteria are met. Moreover, we store the linked clickbait PDF on the first visit. We also included the use of numerous VPN endpoints to check that a given domain is not blocklisting us before marking its URLs as offline. *PDF Status Check* became operative on June 22nd, 2022, marking the start of the Main phase of our study (no PDF was downloaded prior to this date). § 5.6 discusses possible limitations of this approach and § 5.7 discusses the measures we took to reduce the load of our analyses on target websites. Before adding new PDFs into the *Main DS*, we ensure they meet the *SEO metric*, and then we reapply the *PDF Metadata Analysis*.

5.2.2.2 URL Analysis Module

In this step we perform analyses on the extracted URLs. We collect DNS records of each fully-qualified domain name (FQDN) actively serving clickbait PDFs, extract its IP and fetch the corresponding WHOIS record, including Autonomous System numbers. Next, we collect the blocklist status of each extracted `.pdf` link, using Google SafeBrowsing (pre-installed on more than 84% of users' browsers [239]) and VirusTotal, popular both in research and in industry (see, e.g., [168, 266]) as reference.

5.2.2.3 Indicators of Compromise Collection Module

The collection of indicators of compromise is a multi-faceted procedure which comprises different analyses depending on the target host. It is performed by two sub-modules collecting evidence of vulnerable or misconfigured software components.

The first module collects indicators linked to the presence of software components and plugins running on the server-side by visiting with a full-fledged Chrome browser the homepage of a domain actively serving clickbait PDFs. When loading the page,

the browser waits up to 15 seconds, intercepting all network requests happening in the background. This functionality is similar to that realized by [256], which we incorporate for easier interaction with the Linux Traffic Interface. We then process the network traces applying a rule-based approach (we integrate that of [251] for simplicity, similarly to [43]) to obtain information on the web server (e.g. Apache), programming languages (e.g., PHP), hosting panels (e.g., Plesk), web application framework (e.g., Wordpress) and add-ons (as WordPress Themes and Plugins).

Our second module is a custom vulnerability scanner developed to verify the presence of misconfigured or vulnerable components which may lead to file upload. The scanner visits pre-selected URL paths which we observe are indicators signalling the presence of a component allowing file upload. § 5.3.2.5 details the inner workings of this component. In case no evidence could be collected we trigger additional analyses for this FQDN, where the Chrome browser visits $n \leq 20$ random pages extracted from the homepage of the domain to possibly observe additional software components.

5.2.2.4 Clustering Module

Clickbait PDFs can be clustered with respect to the visual deceit (e.g., position and aspect of their bait elements) shown on the first page [P2]. Previous work identified 44 clusters using a Deep Learning approach based on Convolutional Neural Networks (CNNs).

We develop our own CNN model to perform feature extraction, creating a feature space where visually-similar samples are mapped close to each other. The model takes a screenshot of the first page of each document and returns a 32-dimensional vector denoting its position in the new feature space. We create a training set starting from the one provided by [P2]. We performed data cleaning when necessary, removing outliers and filtering or remapping elements to new groups based on their similarity. Finally, we augment it with more recent data from our data feeds, obtaining a total of 23 098 training samples divided into 47 groups. Next, we use a semi-hard triplet selection process and the triplet-loss function to train the model weights (see [201]). With this model, we extract a feature vector for each PDF and then apply DBSCAN [59] for clustering. To reduce manual intervention, we incorporate pre-labeled samples, or "anchors", into the pool of unseen documents. This way, we can automatically label the clusters based on the group of anchors they contain. If multiple anchors are associated with the same computed group, we re-cluster its samples using a smaller ϵ with DBSCAN until the conflict is resolved. Human intervention is only required when our *Clustering module* identifies a new cluster. Section 8.2.1.1 provides further details on the model and clustering procedure.

Autonomous System	# FQDNs	Autonomous System	# PDFs
WEEBLY, US	41 483	WEEBLY, US	241 851
AMAZON-02, US	9 222	*AMAZON-02, US*	142 200
WILDCARD-AS	5 351	CDN77 ^_^, GB	59 213
GOOGLE-2, US	4 301	*CLOUDFLARENET*	57 156
ZETTA-AS, BG	4 091	*GOOGLE-2, US*	46 264
AUTOMATTIC, US	1 556	UNIFIEDLAYER	37 504
CLOUDFLARENET	1 363	OVH, FR	34 974
OVH, FR	1 141	*GO-DADDY-CO*	31 080
IWEB-AS, CA	1 097	*ARUBA-ASN, IT*	25 731
UNIFIEDLAYER	1 086	FASTLY, US	25 703

Table 5.2: Top ten Autonomous Systems sorted by number of FQDNs (on the left) and by the number of PDFs (on the right). The two lists report different AS names depending on their rank determined by the sorting criterion.

5.3 Characterizing Support Infrastructure

The goal of this section is twofold. Firstly, we examine the host and service composition, seeking similarities among hosts. Addressing this early on in the setup phase enables us to conduct specific analyses later on for these host types, which we study during the main phase. To tackle **RQ 1**, we investigate the network properties (Autonomous System, DNS lookup, URL) of the 1 350 201 URLs extracted from *Seed DS* (backlinks leading to to clickbait PDFs, see § 5.1). Our analysis of these properties (§ 5.3.1) reveals the presence of large groups of hosts with similar traits. Specifically, we observe three different types of hosting, covering 54 eTLD+1s. Next, in § 5.3.2 we run ad-hoc analyses on the websites serving 2 710 959 live clickbait PDFs during the Main phase of the study. We identify six plugins and two web frameworks facilitating file upload, and 12 927 origins hosting outdated software components, answering **RQ 2**.

5.3.1 Analysis of Network Properties

The goal of this section is to identify whether certain hosts within the supporting infrastructure share similar features, which we define in terms of network properties.

To find out if and which components make up the supporting infrastructure, we conduct an exploratory analysis of the *Seed DS* backlinks. Since attackers target large amounts of websites having the same security flaw (see, e.g., [242, 263, 152]) we analyze our data to find large groups of hosts sharing similar network properties. Our approach does not aim at identifying hosting provider *organizations* [210] but groups of similar Web hosts targeted by attackers.

5.3.1.1 Methodology

We focus on those indicators that can either be observed directly (e.g., domain name) or obtained via well-established channels (e.g., DNS queries). For example, given a URL `http://babemozigu.weebly.com/dir/file.pdf` we extract its FQDN (`babemozigu.weebly.com`) and its eTLD+1 (`weebly.com`), or "domain root". We

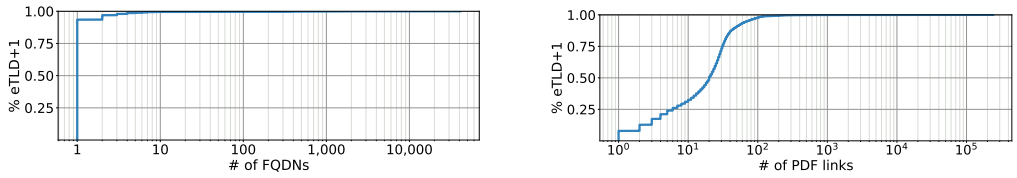


Figure 5.3: (a) Distribution of FQDN per eTLD+1. (b) Distribution of `.pdf` links per eTLD+1. Data from the *Setup phase*.

obtain the IP address and the Autonomous System (AS) for each FQDN from the respective DNS and WHOIS records. For readability purposes, we aggregate different AS names belonging to the same company (e.g., `CLOUDFLARENET, US` and `CLOUDFLARESPECTRUM Cloudflare, Inc., US`, in italics) and report in Table 5.2 two distinct lists of the ten most affected ASes, independently sorted by number of unique FQDNs and by number of observed clickbait PDFs.

We noticed a significant difference in the order of ASes between the two lists. For instance, the ASes for `Weebly`, `Wilcard`, and `Zetta-AS` (first, third, and fifth ASes) were found to be the most frequently abused in terms of FQDN, but their overall rank differs considerably when sorting them by number of clickbait PDFs. Figure 5.3a shows the distribution of FQDNs per domain root. The graph shows a sharp increase, indicating that the majority of domain roots (96%) have either no subdomain or just one subdomain. However, a small percentage ($< 0.01\%$) of domain roots have ten or more subdomains. To further analyze this, we set an empirical threshold of 100 FQDNs per domain root and manually investigate the resulting 20 domain roots. These eTLD+1s represent 97% of the domain roots with at least one subdomain. For example, `babemozigu.weebly.com`, `babewepuk.weebly.com`, and `babexunerasosib.weebly.com` are among them.

In the right column of Table 5.2, we present different ASes based on the number of served clickbait PDFs. The distribution of PDF files across domain roots (Figure 5.3b) shows that most eTLD+1s host a maximum of 100 clickbait PDFs, while only 2% of the domains serve more than that. We adopted a conservative approach to identify candidate domain roots by using the number of `.pdf` links as a criterion. As uploading a large amount of PDFs to a compromised website is easier than obtaining free subdomains, we set an empirical threshold of 5 000 PDF links per eTLD+1. We manually investigated the domain roots that exceeded this threshold in terms of PDF volume.

5.3.1.2 Results

This procedure identified 26 unique eTLD+1s. We confirm the existence of specific hosting services running on that domains by conducting a separate market research for services exhibiting similar characteristics. As a result, we identified three services running on these eTLD+1s, namely *Object storage*, *CDN* and *Website hosting*, which we explain below.

Object storage is a hosting service that manages unstructured data, such as PDFs, as individual units, or *objects*, stored in a single location [75]. The URLs of these objects

include strings resembling unique identifiers, either as subdomains or in the URL path. Although these origins cannot be browsed, files can be retrieved using known URLs. We found one domain root belonging to this category, whose service includes a free tier accessible after thorough checks (e.g., providing a valid credit card number).

CDN origins exhibit a filesystem structure that resembles that of *Object storage* services, where PDFs (and other static resources) reside on a separate origin from where the main website operates, as depicted in Figure 5.4. Through our manual analysis, we were able to link all but two of them (`sqhk.co` and `f-static.net`) to a specific hosting service, such as E-commerce marketplaces or Shared hosting. During our market research on online hosting services, we discovered one entity using multiple eTLD+1s, as `s123-cdn-static-a.com` and `s123-cdn-static-b.com`. In our dataset, we identified five such instances and included them in this category.

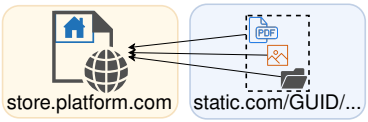


Figure 5.4: Example showing static resources residing on a different domain (PDFs in the *CDN* category).

In *Website hosting* services, multiple websites run on the same server. The services running on those domains observed in our data offer affordable options, including free subdomains or automated website building, an online service that enables users to create websites without coding skills by combining pre-designed modules. We verified that these services allow users to publish a website without requiring a credit card or a valid email address.

We perform two extra checks to ensure that no other hosting service with a lower volume of abuse went undetected. First, we investigate the remaining FQDNs via a third-party web analytics service [204], observing 23 additional domain roots classified as *Web Hosting and Domain Names*. We verify the correctness of this label before adding them to our *Website hosting* group. Additionally, we checked our URLs against a manually curated list of hosting services, finding two URL matches for `digitaloceanspaces.com` (DigitalOcean) and four URL matches for `storage.googleapis.com` (Google). However, we do not include them in our further analyses as the volume of URLs for these two providers is negligible with respect to that of other *Object storage* providers identified by our methodology (e.g. Amazon, 49 065). Table 5.3 reports the volume of clickbait PDFs per hosting type and category, while exhaustive details on the identified hosting services (as eTLD+1, volume of clickbait PDFs and FQDNs) are reported in Table 8.4 (Appendix). We observe that the coverage of our websites provided by [204] is limited (10% of all domain roots), which might be explained by the low rank of some websites or by their offline status. In the remaining, we refer to websites in none of the groups *Object storage*, *CDN*, or *Website hosting* as *Undetermined hosting type*.

5.3.1.3 Takeaways

In this section we addressed **RQ 1** by scrutinizing observable properties of URLs hosting clickbait PDFs. Our methodology identified a total of 54 domain roots (26 via analysis of network properties, five via manual analysis, and 23 via a third-party service [204]), which we have verified correspond to existing hosting types and services. For the scope of

Hosting Type	# URLs
Object storage	166 356
CDN	595 385
Website hosting	853 514
Remaining URLs (*Undetermined hosting type*)	4 126 172
Education	204 679
Graphics Multimedia and Web Design	129 954
Computers Electronics and Technology	116 090
Web Hosting and Domain Names	99 165
Sports	96 612
Remaining categories	289 680
No category found by [204]	2 095 555

Table 5.3: Number of URLs to clickbait PDFs over hosting types or website categories. Data from the *Main phase*.

this paper, we organized them in three broad groups, *Object storage*, *CDN*, and *Website hosting*. Note that these names might not cover all the extensive services provided by major providers. For instance, *Website hosting* might involve Website Builder services, along with managed and unmanaged shared hosting.

5.3.2 Indicators of Compromise

In this section, we investigate factors which may have facilitated the upload of clickbait PDFs on the abused hosts, answering **RQ 2**. Our analyses are tailored to the characteristics of each hosting type, investigating Access Control Lists, presence and up-to-date status of software components related to website abuse, and plugins which we observed to lead to file upload. We observe the strong presence of outdated and vulnerable components on *Undetermined hosting type* websites, while *Website hosting* domains present a bare software stack which is rarely outdated. Finally, we summarize our main findings.

5.3.2.1 Experimental Setup

Different hosting types expose distinct properties, requiring the development of custom analyses modules for each type.

Firstly, when their URL is requested (e.g., via `HTTP GET`), *Object storage* hosts return "data units", and authorized users can upload new data via protocols specified by the service provider.

Next, we consider *CDN* providers and observe that domains in this category return an HTTP status code `403` when requesting the base path ("`/`") or any path segment preceding a PDF file. In fact, their filesystem structure cannot be inspected via simple `HTTP` requests, similarly to *Object storage* origins. Collecting data on the respective "storefront" of *CDN* origins is impossible because systematically linking *CDN* origins to their respective homepage domains is infeasible (see Figure 5.4). Consequently, we removed all domains belonging to this category from the analysis.

Conversely, websites belonging to the *Website hosting* or *Undetermined hosting type* categories can be inspected via regular crawling. We determine the presence of outdated or vulnerable components in two ways. First, we compile a list of server-side software components that previous works found to be connected to Internet abuse. These are: *(i)* type of web application, specifically CMSes and E-commerce software; *(ii)* their version (see, e.g., [72, 225]); *(iii)* a list of plugins and themes, as the ones for WordPress, when applicable (see, e.g. [100, 242]). *(iv)* the presence of Unrestricted File Upload vulnerabilities, as highlighted to be used in conjunction with SEO attacks [97]. Second, we performed a manual analysis of selected URLs, which led to the identification of eight additional components linked to file upload, for which we develop a custom scanner.

We follow best practices and disclosure guidelines in these analyses. Due to ethical concerns, we develop non-intrusive analyses looking for indicators of compromise (hereinafter IoCs), refraining from sending `POST` requests to verify vulnerabilities when this would trigger a state change on the target website.

5.3.2.2 Misconfigured S3 Buckets

The only *Object storage* service in our dataset corresponds to Amazon's Simple Storage Service. Thus, our analysis of *Object storage* websites is based on the collection of metadata on S3 buckets permissions. We develop our S3 scanner module relying on a popular library [193] on top of the AWS SDK. Similarly to [38], we proceed with the

SW Category	SW Name	# versions	# FQDNs
CMS	WordPress	188	4,041
CMS	Joomla	3	209
CMS	Drupal	3	112
Ecommerce	WooCommerce	150	1,310
Ecommerce	EasyDigitalDownloads	11	24
Ecommerce	Magento	1	4
Prog. language	PHP	280	8,206
Web servers	Apache	40	1,884
Web servers	Nginx	68	192
Web servers	IIS	7	438
WP plugins	Yoast SEO	193	1,463
WP plugins	WooCommerce	150	1,310
WP plugins	Revslider	115	623
WP themes	Astra	56	170
WP themes	Hello Elementor	8	71
WP themes	OceanWP	29	66

Table 5.4: Three most popular outdated software components per category.

inspection of each bucket, collecting Access Control Lists (ACLs) and bucket contents when possible. For ethical reasons, we do not try to write any file to the buckets. We observed that a bucket may still exist even if one or more referenced PDFs are not online, thus, we feed the S3 scanner module all *Object storage* links, regardless of their online status. We probed 1 776 unique buckets in total, obtained from 159 403 links, where 243 were reachable at the time of scanning, while the remaining ones raised an error (e.g., `NoSuchBucket` or permission denied). We find that 67 of them have a readable Access Control List, where 21% of the buckets leave `Full Control` permissions, 28% of the buckets leave `Write` permissions, and 51% of the buckets allow to read a bucket's ACL (`READ_ACP` permission) to unauthenticated users.

5.3.2.3 Outdated Software Components

Next, we consider *Website hosting* and *Undetermined hosting type* websites. We proceed with a two-way approach: first, we collect data on the software components running at all *Website hosting* and *Undetermined hosting type* websites actively serving PDFs. When no data point has been collected for a domain, we randomly select $n \leq 20$ additional links from its home page and visit them, to increase the probability of triggering and detecting a vulnerable component.

We focus on software components of the following categories: Content Management Systems (CMSs), Ecommerce software, Hosting panels, Web servers, plugins and themes (as those of WordPress), and software components using the PHP programming language. We visited all FQDNs that served at least one clickbait PDF, i.e., 85 582 websites, and observed indicators relative to the above categories for 29% of them, identifying a total of 299 software components. Next, we determine outdated software components by comparing their observed version on a target domain to their latest version at the time. We observe that most of the domains where this information is available are *Undetermined hosting type* domains (96% of the total observations), where more than

SW Component	# FQDNs		
	Path IoC	Scanned	% vulnerable
KCFinder	799	262	100
CKfinder	2 436	4 396	100
FCKEditor	232	4 933	0
CKEditor	88	4 840	91
Webform	482	-	-
Formcraft	621	-	-
SLiMS	1 018	396	73
E-Learning Madrasah	396	396	38

Table 5.5: Number of FQDNs running software facilitating file upload, with IoCs found in the URL path or via crawling.

half of these websites run outdated components. Conversely, only 26% of the software components observed on *Website hosting* domains are outdated. Table 5.4 reports the most popular outdated components per category.

As a last step, we inspected the network traces of our scanners to determine why no information was collected for a large amount of FQDNs. This inspection revealed that 90% of the websites that did not return any information are `weebly.com` subdomains, where the crawling was unsuccessful for `Timeout` errors as the IP was blocked. All the other domain roots were regularly visited by our scanner[1].

5.3.2.4 Vulnerable Software Components

We construct Common Platform Enumeration identifiers [157] using the retrieved software and version information (115 software components with version), and query the National Vulnerabiliy Database (NVD) [158] to obtain corresponding CVE information. We enrich this data with vulnerability information from the WPScan Wordpress Vulnerability Database [257].

Among these, we identified 26 software components whose version, at the time of our inspection, was vulnerable. We filtered out vulnerabilities less likely to be linked with clickbait PDFs (e.g., buffer overflow) and focused on "Unrestricted File Upload" vulnerabilities. In total, we observed ten vulnerabilities of this type affecting five software components among those we inspected. Among those domains with software and version information, 11 815 ran a component listed in either the NVD or the WP vulnerability database, and 225 of them had a UFU vulnerability, all of them belonging to the *Undetermined hosting type* group.

5.3.2.5 Software Facilitating File Upload

An exploratory manual analysis of *Website hosting* and *Undetermined hosting type* websites revealed the massive presence of specific vulnerable or misconfigured plugins which could be abused to upload files. In particular, we analyzed the URLs looking for recurring URL path elements on a large scale, with a volume sufficiently large

[1] We strived to reduce the load on target websites performing analyses only once per FQDN.

for them to be considered as a deliberate target. Our intuition comes from the observation that large numbers of URLs can be grouped together by path segments, e.g., 119 662 URLs residing on 1 016 different domains share the path segment `wp-content/plugins/formcraft/`. A manual analysis of the most common URL path groups (we could confirm 19 unique URL path patterns inspecting 194 websites) led to the identification of eight CMS add-ons and two Web frameworks[2], all having associated CVEs or a public exploit in popular repositories (Section 8.2.2.1 reports details and vulnerabilities for each component, while Section 8.2.2.2 lists path segment indicators).

The presence of IoCs in the path of a URL may be an early indicator of the presence of vulnerable software, which however does not exclude the presence of the same vulnerable components on websites whose URL paths do not have such indicators. We determine that a website runs a vulnerable component by matching the source code and version string of the component against a regular expresssion[3]. We found specific `.txt`, `.js`, or `.html` files exposing plugin versions through exploit repositories, manual inspection of compromised websites, or by inspecting the source code of the eight components. We compiled a list 107 possible locations for these files, which our crawler visits. We ran this analysis for four plugins, i.e., CKFinder, KCFinder, CKEditor and FCKEditor (verifying the vulnerability for the other two plugins was not allowed, as it required sending `POST` requests.) Visiting all 107 potential IoC locations for the unseen *Website hosting* and *Undetermined hosting type* websites daily is an expensive operation, not to mention the traffic load imposed on the target websites. To reduce the dimension of the data in our daily analyses we *(i)* group domains by URL path (i.e., all path segments excluding the file name), as an identical server-side directory structure is a clear indicator of the presence of a shared server-side component, and *(ii)* visit ten randomly-sampled websites per path group. After two weeks, we inspect the results and remove all potential IoC locations that did not produce any match, lowering their number to 59.

We observed 9 800 websites mounting one or more of the four "CK" plugins, 55% of which were vulnerable. It is remarkable that these domains, all marked *Undetermined hosting type*, actively served a total of 190 258 PDFs. We adopted a similar approach to verify the presence of vulnerable components in the SLiMS and E-learning websites. Table 5.5 shows the amount of domains whose URL path contains an IoC on the left and the amount of domains scanned looking for a vulnerable software component on the right, where its vulnerability was confirmed by observing its software version.

5.3.2.6 Takeaways

The goal of this section was to identify features of the infrastructure hosting clickbait PDFs which may facilitate the upload of clickbait PDFs.

Firstly, upon collecting ACL information for 27% of all active S3 buckets, we observed that all of them allowed unauthenticated users to perform operations, e.g., via the `FullControl` or the `Write` permission. In the remaining cases, we found that

[2]The plugins CKEditor [36], CKFinder [37], FCKEditor [110], KCFinder [235], Formcraft [159], Webform [252], and the Web frameworks E-Learning Madrasah [222, 260] (shipped with CKFinder) and Senayan Library Management System [202].

[3]For example, `FCKeditorAPI={ Version:'2.3.2', VersionBuild: '1082'}`.

most of the PDFs were offline or the buckets were non-existent by the time we visited them, which suggests the possibility of a prior cleanup action. Consistently with these observations, the buckets with observable IoCs counted 4 191 unique URLs leading to clickbait PDFs.

We crawled 31 724 *Website hosting* and *Undetermined hosting type* FQDNs successfully (e.g., no `Timeout` errors) and observed that 51% of them run outdated software components. Among them, the amount of domain suffering from Unrestricted File Upload is low (2%), hinting at the fact that this might not be the primary mean used by attackers to upload clickbait PDFs. In total, these domains served 1 075 835 clickbait PDFs.

Additionally, we confirmed that 16.4% of the 31 724 websites were running at least one component of the "CK" family, facilitating file upload, serving 190 258 clickbait PDFs. We underline that this is a lower bound of the possible websites running these components, as we reduced the amount of website scanned due to the large daily amount of scans otherwise necessary. The number of IoCs observed on URL path hints at a higher number of websites, i.e., 21.3%. Overall, our analyses observed indicators of compromise for 46% (1 251 059) of the URLs analyzed in § 5.3.2.

5.4 Use of Support Infrastructure

Having identified the types of hosting most abused by cybercriminals and the solutions to upload clickbait PDFs on them, we proceed to measure the duration of this activity via the *PDF Status Check* module, answering **RQ 3**. These analyses are conducted on clickbait PDF links in the *Main DS*, having discarded those with an offline status. Next, we group these PDFs by visual similarity using our *Clustering Module* (see § 5.2.2.4) and observe how these clusters distribute over the hosting types.

5.4.1 Duration of Abuse

We calculate the duration of the abuse as the mean uptime of each clickbait PDF hosted on a specific origin (with the granularity of a single day), as shown in Figure 5.5. Among the 54 domain roots identified as hosting services, we observed the live abuse of 38 of them (the PDFs hosted on the remaining 16 eTLD+1s were not online at the time we observed their URLs).

The average uptime for a single clickbait PDF is quite long, i.e., approximately five months. However, due to the continuous upload of new PDFs on the same hosts, the overall abuse of hosting services extends even further, averaging around nine months. It seems as if attackers persistently exploited these hosting services throughout our 13 months of observations, with 1 818 domain roots receiving new uploads for this entire period. The type of hosting providing the longest average PDF uptime is *Object storage*, where this value reaches six months.

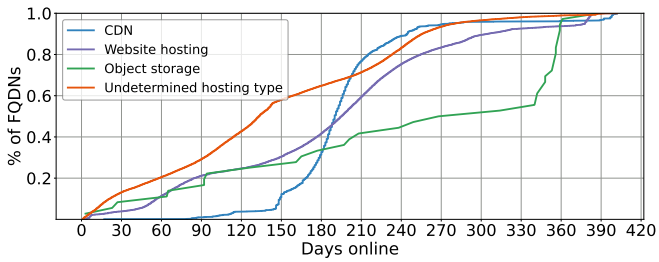


Figure 5.5: Distribution of clickbait PDF uptimes per hosting type, across our 13-month study.

5.4.2 Distribution of PDF Clusters on Hosts

In the *Main DS*, clickbait PDFs can be categorized, by visual bait similarity, into ten groups, seven of which align with those previously reported in [P2], and four are newly identified. We observed fewer campaigns than [P2], which could be attributed to attackers changing the visual baits used (since our data collection began 11 months after their experiments) or due to filtering out non-strictly-SEO campaigns. We gave the new clusters arbitrary names, i.e., *Click here*, *Doc. column*, *Green line* and *White*.

We use the group information to measure the distribution of live PDF clusters on different types of hosts, shown in Figure 5.6. When reading the graph by focusing on

hosting types, we observe that all groups of clickbait PDFs make use of *Undetermined hosting type* spaces, although to different extents.

Two groups (*Doc. column* and *ROBLOX Picture*) upload PDFs solely on this category of hosts. Two more (*ROBLOX Text* and *White*) rely almost uniquely on these domains, hosting there more than 98% of their samples. Conversely, PDFs belonging to the *Download Torrent* group are uploaded almost exclusively *Website hosting* hosts (91,7% of the samples belonging to this campaign). Finally, we observe that Amazon's S3 storage is the type of hosting targeted by the highest number of clusters, as we could observe six different ones[4].

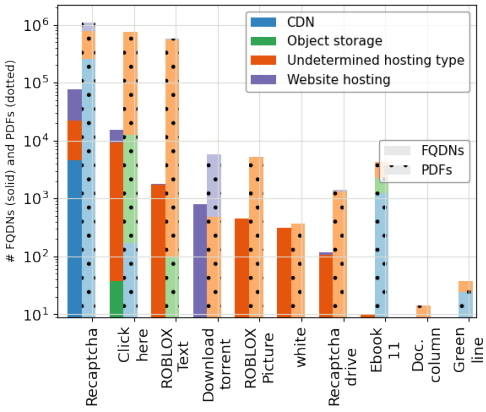


Figure 5.6: Stacked histogram showing clusters distribution across hosting types. Solid blocks represent the volume of FQDNs per cluster, while dotted blocks represent clickbait PDF volume.

Conversely, when focusing on the PDF visual clusters, we observe that they differ in how they use hosting types. For example, the *Ebook 11* cluster tends to perform large uploads of PDF on few hosts, as there large imbalance between the number of FQDNs where PDFs are uploaded and the number of uploaded PDFs (approximately 150 PDFs per eTLD+1, see Figure 5.6). Differently, the *Click-here* and *Recaptcha* clusters distribute on average a smaller amount of PDFs per origin (approximately 2 per eTLD+1). A third example is that of *Download Torrent*, where there are large uploads of approximately 200 clickbait PDFs on two *Undetermined hosting type* domains alongside smaller batches of uploads on many *Website hosting* FQDNs.

5.4.3 Connection with IoCs

We also observed that 43% of clickbait PDFs belonging to the *reCAPTCHA* campaign and 52% of clickbait PDFs belonging to the *Roblox Text* campaign are hosted on websites running one of the targeted plugins (regardless of their observed version). These numbers represent a conservative estimate of the actual impact, as we chose to limit the number of IoCs tested to avoid excessive stress on the target websites.

[4] *Click here*, *Ebook 11*, *Recaptcha*, *Recaptcha drive*, *ROBLOX Text*, *white*.

5.5 Fighting Clickbait PDFs

We now evaluate solutions to counter the distribution of clickbait PDFs, tackling **RQ 4**. We first consider existing solutions, in the form of blocklists, evaluating the protection they offer to users. Our observations indicate that blocklists provide limited user protections, motivating the need to take action against the spread of clickbait PDFs. Our proposed solution involves the notification of affected parties, where we report our observations on the presence of clickbait PDFs and on the status of the components running on the websites hosting the PDFs.

5.5.1 Blocklists

In this section, we investigate whether common blocklists, as VirusTotal (VT) and Google SafeBrowsing (GSB), take action against clickbait PDFs by blocklisting their URL. This would offer a viable protection to users, which would then be protected when accidentally visiting the page of the PDF. We base our observations on 17 months of Google SafeBrowsing and VirusTotal daily lookups (i.e., since the beginning of this study).

We request scan results for 4 thousand clickbait PDF URLs daily to VirusTotal (approximately 50% of the daily amount) and receive a response in only 14% of the cases, where URLs are mostly flagged as malicious. This confirms the uncanny observation in [P2] that URLs in clickbait PDFs are only partially scanned by VT and that this happens on the day the PDF is uploaded to the platform. When considering the type of hosting, we observe that VirusTotal flags domains belonging to all four of them, with *Website hosting* having the highest average rank (five AV engines) and *Object storage* having the lowest average rank (one AV engine).

Next, we observe that the number of URLs blocklisted by GSB is low, i.e., 0,4%. These URLs belong to 451 domains, with a mean ratio of URLs per domain of 41 (min 1, max 1377), which suggests that GSB is taking actions against clickbait PDFs and their hosts, blocklisting entire directories, but on a very small scale. Additionally, 99.7% of the blocklisted URLs belong to *Undetermined hosting type* URLs, suggesting that GSB does not take any action against clickbait PDFs hosted on well-known, reputable domains. When considering the overall lifetime of a clickbait PDF, as measured by the *PDF Status Check* module, we observe that a significant amount (29%) of the blocklisted PDFs is still online, which leads to think that blocklisting does not always correspond to a cleanup action.

5.5.2 Vulnerability Notification

Our next goal is to evaluate solutions beyond blocklisting to help reduce the spread of clickbait PDFs. One way to protect victims from the attack and, at the same time, to reduce the effectiveness of the SEO attack is taking down the PDFs by removing them from their location at the host. We thus undertake a large-scale notification of the threat posed by clickbait PDFs to the affected parties. Our primary goal is to observe the responsiveness of the hosting providers, measuring the amount of PDFs taken down as an effect of our reports.

5.5.2.1 Setup of the Study

We designed the notification procedure following best practices in this field [216, 215, 127, 210, 45, 102].

Selection of Contacts On Dec 1st, 2022 we select 799 930 `.pdf` links found online by our *URL Analysis* module on the previous day and divide their FQDNs equally in Treatment and Control group (8 843 and 8 842 respectively). Then, we look up their IP addresses and proceed to collect WHOIS records, obtaining 32 302 email contacts for 12 043 IPs. If necessary, we prioritize contacts from the same record, selecting `abuse@` contacts when present, `hostmaster@` contacts otherwise (following RFC 2142). If none of them are available, we choose one randomly. We obtained no WHOIS record for 153 domains, thus, we generate "synthetic" contacts by combining the aliases `abuse@`, `info@`, `security@`, `hostmaster@` with the domain name.

Content and Timeline of Notification The notification e-mail briefly explains the threat posed by clickbait PDFs, then lists up to three clickbait PDF links among those hosted on up to three FQDNs belonging to the addressee. As a possible mitigation, we suggest the removal of the reported files and recommend a revision of the software components running at those domains. A CSV attachment reports all clickbait PDFs links for all the domains belonging to the addressee. Finally, recipients are given the possibility to opt out of the study or reach back for any feedback. The full text of our notification message is reported in Section 8.2.3.

Finally, we set a time window of 30 days, from Dec 1st to Dec 31st, 2022. We notified domains in the Treatment group once every ten days for a total of three times and notified the domains in the Control group at the end of the study. The choice for a ten-day time interval is motivated by the observations reported in [215] where, in spite of the 14-day interval between each reminder, the number of fixes does not increase after ten days.

Ethics We did not seek IRB involvement for this procedure, addressing ethics concerns as follows.Contact points (participants) were chosen depending on the presence of clickbait PDFs on their domains. Participants were informed of the study and given the option to opt out immediately if in the Treatment group, or at the end of the monitoring period otherwise (Control group). Although vulnerability notifications might represent an additional overhead for security operators at hosting providers, the benefit gained from clickbait PDF takedown and a security review of the software stack outweigh this cost. To reduce recipient overhead, we grouped domains per abuse contact. Finally, we did not collect any user data and sought to increase privacy of operators and providers by processing answers per anonymous ID rather than email address.

5.5.2.2 Process

A final amount of 1 545 contact emails was selected as recipient for the notification. The discrepancy between the number of contacts and FQDNs stems from them sharing the same eTLD+1 or a provider managing multiple FQDNs. Due to a technical problem,

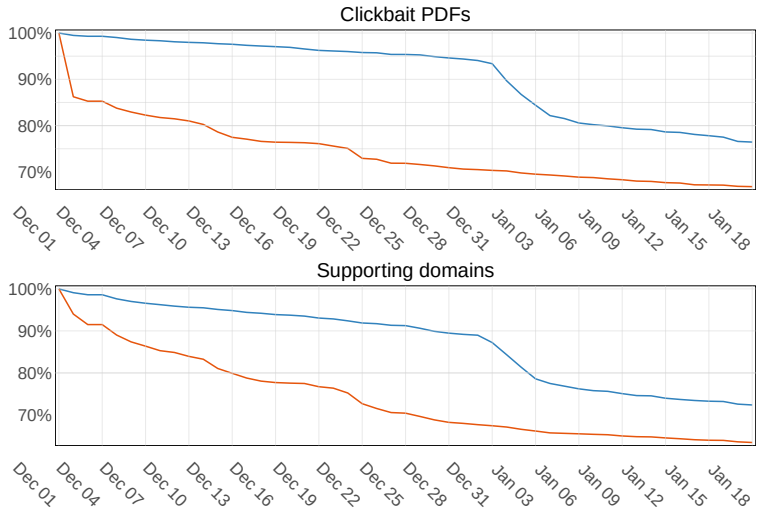


Figure 5.7: Takedown of clickbait PDFs and domains over time. The Treatment group is depicted in red and the Control group in blue.

19 domains were not included in the reports or not reported at all, resulting in 1 522 emails being sent successfully. These contacts were notified together with those in the Control group, but removed from the reports.

As part of the notification process, we excluded one contact, who asked to stop the analyses of the PDFs residing on their domain. Moreover, we adopted a "cooperation policy" whenever explicitly asked, e.g., we re-sent the attachment or provided clarification on the threat (124 replies), acknowledged false positives (9 PDFs, $< 0.01\%$), or submitted a copy of the report via a Web form (25 submissions). Moreover, we estimated a lower bound of 257 contacts we never reached by inspecting the headers of bounced emails. As these providers could not be reached in the first round, we removed them from the Control group and did not notify them again.

5.5.2.3 Effectiveness

Figure 5.7 shows the effectiveness of our notification by comparing the number of online clickbait PDFs in the Treatment and in the Control group. The remediation rates are 29.567% for clickbait PDFs in the Treatment group and 6.055% for those in the Control group, where their difference is statistically significant with $\rho < .001$ (estimated by using a Generalized Linear Model [160, 146]). The number of online PDFs decreases sharply on the first days, while a less steep decrease is visible for the domains (Figure 5.7). One explanation for that may be that a few affected parties hosting a large number of clickbait PDFs took action immediately, while a larger number of entities, hosting less clickbait PDFs each, took longer to react. The low-but-existent remediation rate for the Control group suggests the presence of some form of "natural decay", where a small fraction of clickbait PDFs go offline for causes not related to our notification. Nonetheless, the significantly higher remediation rate in the Treatment group shows an increased number of cleanup actions with respect to this phenomenon.

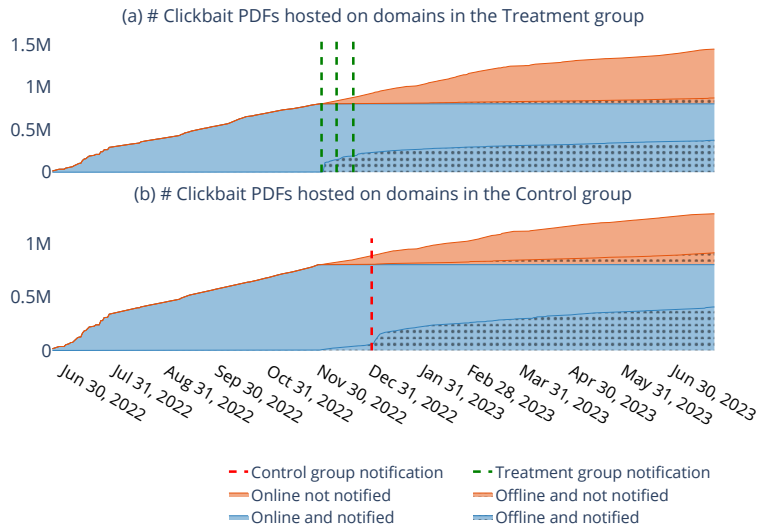


Figure 5.8: (a) Volume of PDFs in the Treatment group over time, online (solid color) and offline (dotted), versus new, unreported PDFs hosted by the same affected entities. (b) as for (a) but for PDFs in the Control group. (Control group volume is rescaled).

We observed that no affected party could remediate with respect to all reported domains (nor all PDFs, if on a single domain), and that 17% of the affected parties only partially remediated the notified issue. In particular, *(i)* 104 entities (7%) only removed those PDFs listed in the email body, ignoring the attachment. *(ii)* 154 entities (10%) cleaned all or some of the reported PDFs. However, after the notification, we gained visibility into unseen (and unreported) clickbait PDFs hosted by them, which we observed stayed online. *(iii)* 319 entities (21%) performed a full cleanup and also removed any PDF observed after the notification.

We also monitored the presence of the reported PDFs on VirusTotal to observe if any of the affected parties submitted the PDFs as a result of our notification. Given the large amount of clickbait PDFs involved and the limited API quota available to us, we opted to randomly sample unique PDFs, in equal amounts from the Treatment and the Control group. We fetched 111 787 reports relative to notified clickbait PDFs. 57 042 of these returned a record, where a negligible amount of them (1%) was either first submitted or last seen after the start of the notification. The number of domains hosting these PDFs belong almost equally to our Treatment and Control group. Thus, it does not seem that submissions to VT were triggered by the notification.

5.5.2.4 Long-Term Effectiveness

We observed a moderate but positive response to the vulnerability notification in terms of PDFs that were cleaned up. Our notification message clarified the possible presence of additional, unreported PDFs and recommended security audits on the software running on the affected domains. We further investigated the long-time effects of our notification of the affected hosts, measuring how many of them still served clickbait PDFs, albeit

unseen ones. The observation of online unseen clickbait PDFs on notified hosts can be attributed to either new uploads from attackers or a partial cleanup by the responsible entity. Figure 5.8 shows the online status of PDFs served by the domains involved in the notification. Starting from Dec. 1st, 2022 and Dec. 30th, 2022, we notice an increase in PDFs going offline, which mostly remains constant after the notification period concludes. Simultaneously, we continue to register unseen PDFs on the same origins, and their volume keeps growing over time. This disheartening finding shows that attackers have, and will continue to have, a relatively stable pool of hosts to upload PDFs in support of their attack.

Our findings also suggest that disclosing the presence of clickbait PDFs is a moderately effective means of reducing the volume of online PDFs at a specific point in time. However, it proves ineffective in enhancing the overall security level of the affected hosts.

5.5.2.5 Feedback from Affected Parties

We observed two main types of reactions to the notification, i.e., appreciation and interest versus an uncooperative attitude.

Security Issues A few affected parties confirmed our report and provided additional details or engaged in a conversation, allowing us to gain some invaluable insight on the issues they observed. Five of them confirmed that their clients were running the plugins we identified in § 5.3.2.5, specifically plugins of the "CKFinder" family or Formcraft for Wordpress. Eight of them only generically replied that their client's CMS software was outdated (e.g., Joomla, Drupal) adding that they observed one or more PHP shells most likely used by the attackers to upload PDFs. One entity mentioned that their customer was running a custom web application. Finally, in three cases, the answers reported that the website seemed to be abandoned by the customer, who was also unreachable.

Not Phishing Interestingly, one addressee answered all three notifications arguing that our report was unsubstantiated. They insisted that the reported PDFs did not pose any threat. Although we clarified the attack, they stated that they would not remove these legit files, as "*an interactive PDF with an attached hyperlink protected by recaptcha does not fall within the scope of phishing*", referring to PDFs reproducing the reCAPTCHA service to trigger a click.

5.6 Discussion

SEO metric. Our *SEO metric* was designed with the goal of filtering out benign or not-clickbait PDFs to avoid processing personal data or poison our SEO-focused dataset. We perform a manual inspection to confirm that it only selects clickbait PDFs by inspecting up to 500 PDFs, fitting the SEO metric and randomly sampled from each cluster, for a total of 3 000 PDFs. The manual analysis confirmed the null number of false positives. Conversely, some clickbait PDFs with too few backlinks may be excluded. In the worst-case scenario, where all PDFs failing the *SEO metric* are clickbait PDFs, the false negatives would amount to 4.6% of 4.6 million links. We inspected 1 000 PDFs failing the *SEO metric*, randomly sampled from the *Seed DS*, and observed a much lower amount of false negatives due to the presence of benign or non-clickbait PDFs.

Development of *Grape*. The *PDF Status Check* module is a core component implementing the daily monitoring of online clickbait PDFs and enabling further analyses. We ensured the reliability of its results by repeating requests to endpoints leading to an error three times, or by using a VPN service. An interesting observation emerged where, in rare cases, an origin returned a different `HTTP` response for the same `.pdf` link. Specifically, we found that the `Content-Type` header differed between the `HEAD` request (not `application/pdf`) and the `GET` request (`application/pdf`). We examined a sample of 359 `.pdf` links marked as offline over the course of a week and did not observe any inconsistencies in the reported status. Moreover, we observed one origin cloaking the content of the `HTTP` response, i.e., serving clickbait PDFs only when visited by a browser instance with enabled JavaScript, and two origins protected by the CloudFlare Bot Management service. *PDF Status Check* does not intend to bypass bot protections, and interestingly, we observed that such mechanisms are notably scarce in prevalence.

Identification of Hosting Types. Our procedure for the identification of hosting types is based on observable metrics and indicators. All domain roots identified by our procedure correspond to an existing hosting service, confirming the validity of our methodology. We enriched this finding with the domain roots obtained from [204], which we verified belong to regional hosting providers. We cannot rule out the possibility that attackers might also abuse other types of services to a lesser extent. For instance, `documentcloud.org`, a document sharing platform, served 121 clickbait PDFs at one point. However, we did not come across any further instances of such activity.

Indicators of Compromise. Our findings show a grim picture of the landscape of software components running on the hosts part of the supporting infrastructure. Outdated and vulnerable components are especially present in *Undetermined hosting type* origins, whose software stack is likely not managed by the service provider. Ethical concerns on the traffic generated by our analyses on these origins limited the amount of scanning we performed to determine the component likely exploited by attackers to upload clickbait PDFs. Therefore, we believe the measurements we presented to be a lower bound of the amount of outdated or misconfigured software.

Vulnerability Notification. Our vulnerability notification procedure effectively reduced the number of clickbait PDFs supporting the SEO attack and provided valuable insights into the software components running on a few notified websites, corroborating our automated analyses.

One methodological choice in this procedure may have influenced its outcome. Specifically, we formed the Treatment and Control groups based on domains instead of contact points. This decision aimed to achieve granularity in measuring remediation, focusing on individual PDF files rather than affected organizations or entities. However, we acknowledge that this approach might have increased the likelihood of cleanup for other websites in the Control group falling under the same entity's responsibility. Moreover, our "cooperation policy", driven by a commitment to a safer Web, could have potentially influenced our results in a positive manner. We believe this impact to be limited, as we only engaged with 6% of the contact points.

External Threats to Validity. Our measurements might paint a less severe picture of the supporting infrastructure due to our partial visibility of the clickbait PDF threat. We mitigate this issue by collecting data from multiple sources: we build the *Main DS* starting from the *Seed DS* and observe that 93% of the total samples are not shared. We believe that a complete picture of this ecosystem might be visible only to entities whose crawling and processing resources are far above ours.

Looking Forward. Section 5.5.2 investigates the effectiveness of large-scale vulnerability notifications to address clickbait PDFs' abuse of hosting resources and protect users. While this approach proves effective in reducing online clickbait PDFs in the short term, there may be alternative methods to combat their distribution at various stages. For instance, making it more difficult for clickbait PDFs to rank high in search results could increase the attack cost and reduce the overall phenomenon. An implementation of this strategy could involve adding a module to a search engine crawler. The vast information available to search engines could serve as a crucial vantage point in preventing clickbait PDFs from achieving high rankings. Future research on clickbait PDFs could investigate which aspects of these documents are useful for detection.

5.7 Ethical Considerations

We designed the experiments for this paper keeping a series of ethical concerns in mind. The daily scanning of online PDFs and indicators of compromise may raise ethical concerns. We followed established guidelines [49], which included minimizing the frequency and load of experiments whenever possible (e.g., using `HEAD` requests instead of `GET`) and indicating the study's purpose, contact information, and opt-out option in the `User-Agent` header. Additionally, we conduct a manual analysis to focus on high-probability IoC endpoints, minimizing unnecessary scanning, and follow best practices in vulnerability disclosure, refraining from testing endpoints where this is not allowed (avoid verifying vulnerable endpoints when this requires sending state-changing `POST` requests). Finally, we reported all observed clickbait PDFs available with our large-scale vulnerability notification, started on Dec 1st, 2022. Our notification text

explained about the threat posed by clickbait PDFs and included our contact points; we further gave participants the possibility to opt out of the study at any time. We plan to conduct another notification campaign reporting the PDFs that are still online at submission time.

Summary

This chapter tackles RQ2, investigating *how attackers leverage web infrastructure to support deception-based attacks*. Our investigation is centered around the websites and web services (the "support infrastructure") which host and deliver hundreds of thousands of clickbait PDFs for up to nine months on average. The support infrastructure has a key role in clickbait PDF attacks as it not only hosts the PDF files, but also implements the SEO poisoning attack via backlinking, ensuring that clickbait PDFs can be found in search engine results.

We carried out a 17-month study counting 177 835 hosts and 4 648 939 links to clickbait PDF, observing that the websites supporting clickbait PDF attacks belong to different types of hosting, such as *Object storage*, *CDN*, and *Website hosting* observed in our dataset, and that their continued abuse lasts nine months on average. Given the diversity of hosting types and services, we observe that the flaws exploited by attackers to upload clickbait PDFs are also diverse, and that addressing them is complex. While notifying affected entities can temporarily reduce the volume of clickbait PDFs, it often fails as a long-term strategy, since many of these parties continue to experience repeated re-uploads.

To better understand this persistent issue, the next chapter explores the specific challenges that IT operators face when responding to vulnerability notifications.

6

Vulnerability Remediation at Hosting Providers

The previous chapters focused on clickbait PDFs, a deception-based web threat leveraging functionalities of modern web browsers as deception mechanism (Chapter 4), and analyzed the web hosts and services that enable their distribution (Chapter 5).

This chapter addresses the problem of low remediation rates for web abuse and vulnerabilities by investigating the underlying causes. Our exploratory study focuses on understanding why recipients of vulnerability notifications (VNs) often fail to act. Specifically, we aim to answer RQ3 by examining *how vulnerability notifications are processed within hosting provider organizations*. We thus perform a qualitative study through semi-structured interviews with 24 IT experts working at hosting provider organizations (HPOs) and dealing with issues concerning web hosting and web applications on a daily basis.

Our findings show that hosting providers have strict boundaries defining what they address. Providers may respond by taking down illegal content, but remediation of the underlying causes of the abuse is often considered a responsibility of customers, or web agencies in charge. While reachability challenges persist, these are not the primary cause for lack of remediation. Finally, providers report overwhelming amounts of abuse reports, which, together with the low value attributed to customer instances, contribute to the disconnect they feel towards VNs, highlighting a misalignment between reporters' expectations, business priorities, and website owners' behaviors and beliefs.

6.1 Methods

Our investigation was driven by the observed low remediation rates to web vulnerability notifications and our desire to uncover the roadblocks that prevent addressing these problems. Against this background, we investigate:

RQ1: *How are vulnerability notifications received and processed within HPOs?*

RQ2: *What are the characteristics of HPOs (their internal factors) that influence their VN-handling and the respective remediation processes?*

To answer these research questions, we first conducted a content analysis of HPO websites. This analysis provided a map of the research field and informed the development of our sampling strategy. Using this strategy, we then carried out a qualitative study based on semi-structured interviews with individuals owning or working for small- to large-sized hosting provider companies. The interviews focused on VN remediation both in detail and from a broader perspective, while allowing participants to share their own viewpoints.

Data analysis and interpretation of the interview study were carried out using a combination of Qualitative Content Analysis [145, 199, 200], a top-down approach, with some components of Grounded Theory [70, 32, 218] which operates in an inductive, bottom-up manner [196]. We used this two-pronged approach to systematically identify patterns and themes in our interviews. On the one hand, we aimed at superficially known but understudied aspects of hosting activities as reflected in our interview-guide questions and corresponding codes. On the other hand, the bottom-up component allowed us to identify phenomena which have not been noticed by previous research.

6.1.1 Sampling Outline

Our sampling strategy followed the general principle of maximal structural variation [107], with the aim of capturing diverse behavior across providers offering different combinations of services. The goal was to expand the surface of investigation with respect to our main research focus: interventions following vulnerability notifications. We approached this from two angles. First, we considered the potential for hosting provider intervention. Different types of hosting services imply varying degrees of provider control [225]. For instance, shared hosting typically grants customers fewer privileges while leaving a larger portion of the technical infrastructure under the HPO's control, compared to dedicated hosting. Second, we scoped our study to services known for suffering high levels of abuse, as these are more likely to receive vulnerability notifications. This supported our focus on the broader category of shared hosting services, which consistently show particularly high concentrations of abuse [225, 7, 224]. From this framing, we derived two key HPO characteristics likely to influence remediation behavior: the type of shared hosting service (e.g., web application, VPS) and the level of service management (managed vs. unmanaged). These two dimensions formed the basis of a sampling matrix, which we used to map the hosting landscape and identify specific providers for interview recruitment.

6.1.1.1 Mapping the Landscape of Hosting Services

We manually compiled a list of hosting provider organizations. Using corporate datasets [173, 40, 47] was prohibitively expensive, and identifying organizations via IPs or Autonomous Systems from ranked website lists (e.g.,[192, 122]) was discarded due to bias toward global providers. Instead, we identified companies by analyzing discussions on relevant Reddit communities (r/VPS, r/agency, r/webhosting, r/Hosting). Posts were selected if they discussed hosting services, excluded if tagged as advertisements or focused on out-of-scope topics. This process yielded 150 posts and a preliminary list of 197 companies.

Next, we manually reviewed the websites of these companies, discarding those offering only out-of-scope services or that no longer existed. Using a top-down content analysis of advertised services, we mapped offerings into the sampling matrix (service types and management). This map served then as the basis for our sampling process, as we anticipated that perspectives on VN remediation would vary across different segments. To capture this variation, we aimed to interview representatives from organizations operating in diverse areas of the hosting landscape. After this review, we finalized a list of 175 companies with a total of 716 distinct offerings (which we will release upon publication), spanning six high-level categories of hosting services. These are Dedicated Server, Virtual Private Server, Reseller Hosting, Shared Hosting, Website Builder, Web Agency.

Hosting Services and VNs in Scope In this work, we consider VNs reporting issues as web application code vulnerabilities [23, 216, 215], misconfigurations [180, 189, 127], or internet services abuse, such as malware, phishing, or SEO infections [31, 241, 84,

P3]. Consequently, our focus is on hosting services and providers operating closer to the application layer, as VPS providers, shared hosting providers, and web agencies.

6.1.1.2 Web Hosting Services

Hosting services in scope can be categorized in three high-level types, here ordered by required technical expertise.

Virtual Private Servers (VPS) offer dedicated portions of shared server resources, ensuring better performance and scalability but requiring higher technical expertise for setup and maintenance. *Shared hosting* involves instead multiple users sharing a single server's resources, such as bandwidth, CPU, and memory, ensuring affordability and simplicity for website owners. Website setup is facilitated through graphical interfaces, making the service accessible even to users with low technical skills.

Web agencies (or "resellers") act as intermediaries between end-users and hosting infrastructure. Their revenue comes from developing and managing websites, with customization being central to their business model. The term has a dual meaning: "reseller hosting" refers to a service offered by HPOs, while "web agencies" are the businesses using this service to deliver final products to customers.

Hosting services may be either managed or unmanaged. *Managed hosting* services include provider assistance, for example with configuration or maintenance, as defined in service agreements, while *unmanaged hosting* leaves these responsibilities to the client. The level of management is closely tied to the underlying hosting type and can also vary across providers, as per their terms of service. For a VPS, management may focus on low-level tasks, such as operating system updates. In contrast, managed shared hosting often extends to application-layer services, such as website setup and configuration, catering to less technically proficient users.

6.1.1.3 Recruitment

We used the service data to group companies by similar offerings. We represented each company's services as a one-hot-encoded vector and applied the k-modes clustering algorithm to group them, using an empirically-chosen value of $k = 13$ for an optimal balance of interpretability and granularity. Our clustering results mirrored the distribution of services in our dataset, where the majority (78%) of companies provide more than one service in various combinations. We obtained five clusters of HPOs offering primarily one service (e.g., website builder, web agency), and eight clusters where HPOs either offered combinations of two or three services (e.g., shared hosting and VPS) or were primarily centered on one service with additional, less significant offerings.

We chose to start from companies offering a single service to reduce the complexity of HPO characteristics that could influence VN remediation. This decision was based on our categorization of service offerings and management. We recruited participants using a mix of cold-calling via available contacts (phone, email, ticketing portal, or chat, resulting in nine participants), snowballing and personal contact networks (still guided by our sampling map, resulting in nine participants), and professional social networks (as LinkedIn, resulting in six participants). This resulted in a total of 24

hosting providers and web agencies from eleven countries. We describe hosting providers and individual participants in Section 6.1.5 and Section 6.1.5.

6.1.2 Interviewing Procedure

Following established methodological recommendations for semi-structured interviews [119, 255], we developed an interview guide as a flexible framework to steer the data collection process. The primary aim was to explore if and how hosting provider organizations handle vulnerability notifications. Thus, a large portion of the guide addressed VNs, investigating employee awareness, communication channels, and decision-making processes.

Recognizing that VN handling may be shaped by broader factors, we included sections examining technical factors (e.g., infrastructure and tools), business factors (e.g., provided services and management), and organizational factors (e.g., internal procedures and management priorities) [128, 17, 45]. To familiarize ourselves with industry procedures for security management, we relied on the NIST Computer Security Incident Handling Guide [35], which helped shaping questions on identified assets, perceived risks, and operational aspects, such as the use of playbooks (step-by-step procedures for employees) and the involvement of third-party entities in incident response. We report the complete interview guide in Appendix 8.3.2.

The interview guide was tested in two pilot interviews with hosting-acquainted colleagues at our research institution. The guide was iteratively refined throughout the study by incorporating insights from earlier interviews to partly re-focus on some newly emergent, relevant issues. For instance, some questions were tailored based on whether the interviewee was a classical hosting provider or part of a web agency. When participants spontaneously introduced some of the topics in the course of the interview, we followed their lead allowing for a change in the order of discussed topics. However, the interview guide always contained a stable core of topics we discussed with all our participants throughout our entire study. This approach aligns with best practices in qualitative research, which emphasize adaptability to enrich data quality.

Nearly all interviews were conducted via Zoom[1] between Aug '24 and June '25, lasting between 60 and 90 minutes. Before the interview, all participants completed a questionnaire providing consent, demographic details, and information about their HPO, reported in Table 6.2. All interviews were audio-recorded and transcribed using our in-house AI-based transcription tool. Most interviews were conducted in English, with some in Italian later translated for data processing (four via DeepL and three via our in-house AI-based transcription tool). All interview transcripts were anonymized by removing personal and HPO-specific information.

6.1.3 Data Analysis

The analysis of interview data combined a top-down approach of Qualitative Content Analysis [145, 199, 200] with elements of Grouded Theory's bottom-up methodology [70, 32, 218]. An initial set of codes was derived from the topical areas and specific questions

[1] One interview was conducted in person due to the proximity of HPO's premises to the location of our research institution.

in our interview guide. It was clear from the very beginning that our interviewees talked about these issues, because we explicitly asked them about it. To avoid fully subsuming our rich data under predetermined categories, we also allowed for the creation of new codes in a bottom-up manner inspired by Grounded Theory's open-coding procedure. This approach enabled us to capture specificity and novelty emerging from the data while anchoring it within the broader topical areas represented by the initial top-down codes [196].

Initially, three researchers (a computer scientist, a psychologist and a sociologist) independently coded four transcripts to identify discrete text units relevant to the research questions. To ensure consistency and resolve interpretive differences, we collaboratively reviewed and refined the initial codes through detailed discussions, fostering shared understanding and adapting codes as necessary. We did not calculate inter-coder agreement rate as our final agreement on codings approached 100% [109, 108, 90, 198].

Following this, the researchers conducted several iterations of descriptive coding, grouping open codes into broader thematic categories. This iterative process led to the development of increasingly abstract categories that captured key themes and patterns in the data, resulting in a preliminary codebook. The codebook was then applied to five additional interviews, during which it was tested, refined, and expanded to include omitted codes, ensuring comprehensive data coverage. Finally, we used the finalized codebook (Appendix 8.3.3) to code the remaining fifteen interviews, with all coding performed using ATLAS.ti [71].

We conducted interviews until all HPO clusters identified via the mapping procedure were covered, and our analysis indicated saturation in identifying and describing phenomena relevant to the research questions. By the end of this process, no new themes or problems emerged, confirming that the studied phenomenon and its diversity were comprehensively described. The final code system was then used for a category-driven re-examination of the primary data. A comparative matrix was created to extract RQ-relevant information from each interview, enabling the identification of similarities and differences in how individual HPOs addressed the problems focused by our research questions.

6.1.4 Ethical considerations

During the recruitment phase, we ensured that each company was contacted only once. If no interest was expressed, the company was excluded from further follow-up. Additionally, we sent a maximum of two reminders to potential interviewees who showed interest.

We conducted the interviews via videotelephony, but only recorded audio tracks. These audio files were used exclusively for transcription and were deleted after the transcription was completed. To protect participant confidentiality, we kept demographic data and personally identifiable information (e.g., details collected during recruitment) strictly separate from the study data. Personally identifiable information was deleted upon completion of the data-gathering phase. Anonymity was preserved through the assignment of unique participant ID numbers and the use of password-protected

spreadsheets accessible only to the principal investigators. Additionally, transcripts were fully anonymized to remove references to individuals, company names, specific locations, associated organizations, and any identifiable tools or products developed by the company.

All participants provided informed consent prior to their involvement. As compensation, we offered each participant 50 USD. Further, the responsible ERB for our institution reviewed and approved our study procedure.

6.1.5 Participant Data and HPO Information

Our interview study included 24 participants from eleven different countries, reported in Section 6.1.5. Participants represented a wide variety of positions and represented companies varying significantly in size, ranging from small (less than 10 employees, nine companies) to mid-sized (11–50, 51–200, 201–1,000, twelve companies) to large (over 1,000 employees, three companies). Participants demonstrated very diverse technical skillsets tied to their roles. In a nutshell, their tasks can be divided in three categories: leaders (ten participants), customer-facing (seven participants), and backend roles (seven participants).

Regarding the services offered, the study captured a wide range of market coverage, reported in Table 6.2. Web application hosting services ("shared hosting") were the most common, provided by 12 companies. Eight companies offered VPS solutions, while four operated as web agencies. When examining the nature of hosting management, the split was almost even: eleven companies provided managed hosting, while ten operated on an unmanaged basis, and three offered both options.

While most participants focused on a single company during the interview, four offered insights into more than one. Two of them (*WA-M-1*, *VPS-MU-1*, marked with $\star$ in Section 6.1.5) discussed previous employers, which were more relevant to our study than their current roles. The remaining two reported experiences from multiple companies: *SH-U-3* referenced three organizations, and *SH-MU-1* discussed two. We noted partial overlap between *VPS-U-1* and *SH-U-2*, as *SH-U-2* is a spin-off from *VPS-U-1*, established to diversify their product offerings. Moreover, we remark that the service categorizations reported in Table 6.2 follow the companies' self-descriptions, though their actual offerings and management levels vary and are not always consistent across companies. Lastly, *VPS-U-3* is a non-profit organization in the education sector, whose services align closely with those of typical hosting provider organizations, while *SH-U-3* and *SH-MU-1* both provide registrar services alongside of hosting.

6.1.6 Structure of Findings

The following sections 6.2 through 6.5 address our research questions with case-specific examples and conclude with key takeaways. Section 6.6 discusses main findings, offers actionable insights for future notifications, and suggests directions for further research. We use « angular parentheses » to report verbatim quotes from interviews.

HPO ID	HPO size	CySec dept. or person	Primary education	Current role	Age	YoE	Years in HPO	Multiple HPOs	Country
WA-M-1	11-50	●	Computer Science	Front-end dev., Proj. Manager	29	7	1-3	⋆	IT
WA-M-2	51-200	●	Computer Science	System Operator	24	2.5	4-10		DE
WA-M-3	1-10	□	Computer Science	CEO	41	10	11-20		IT
WA-U-1	1-10	●	Industrial electronics	Responsible Executive	59	24	11-20		IT
SH-M-1	1-10	◇	Computer Science	Owner	23	7	4-10		NL
SH-M-2	201-1000	▲	Media and Film	Product Owner	37	7	4-10		BG
SH-M-3	11-50	▲	Computer Science	Customer Success Engineer	32	4	4-10		IN
SH-M-4	51-200	●	Electronics and Telecomm.	Technical support executive	25	3	1-3		IN
SH-M-5	> 1000	▲	Computer Animation	Digital Fraud Analyst	42	8	11-20		USA
SH-M-6	11-50	▲	Computer Science	CTO	42	11	> 20		BG
SH-M-7	201-1000	▲	History	CEO	37	20	1-3		UK
SH-U-1	1-10	□	Telecommunications	CEO	45	27	> 20		DE
SH-U-2	11-50	□	Computer Science	Developer	29	10	4-10		DE
SH-U-3	51-200	▲	Computer Science	Technical Support	27	7	1-3	3	PH
SH-U-4	1-10	□	Computer Science	CEO & CTO	38	22	> 20		AT
SH-MU-1	> 1000	▲	Computer Science	Linux System Administrator	26	5	< 1	2	IT
VPS-M-1	11-50	▲	Computer Science	Cloud Administrator	37	5	4-10		IT
VPS-U-1	1-10	□	Did not attend university	Owner	45	22	> 20		DE
VPS-U-2	11-50	▲	Business and economics	Operations and maintenance		10	4-10		IT
VPS-U-3	51-200	◇	Computer Science	System Administrator	31	5	1-3		IT
VPS-U-4	1-10	□	Computer Science	Customer Success Manager	19	2	1-3		PL
VPS-U-5	1-10	-	High School Diploma	CEO	19	2	1-3		IN
VPS-MU-1	> 1000	▲	Computer Science	Senior Network Engineer	28	10	< 1	⋆	DE
VPS-MU-2	1-10	●	Communication	CEO	42	20	4-10		DK

Table 6.1: Participant details collected via survey. Legend for *HPO ID*: (Type)-(Mgmt)-(ID), where: Type: WA = Web Agency, SH = Shared (Web Application) Hosting, VPS = VPS. Mgmt: M = Managed, U = Unmanaged, MU = Both. ID: incremental number. Legend for *CySec dept.*: ● participant; ◇ another employee; □ collective decision; ▲ HPO security/abuse department; - no department/employee. Legend for *Multiple HPOs*: ⋆ for previous employer only, or total number N of employers discussed. Note: participant's country may differ from the HPO's, e.g., for global enterprises.

Managed services	VPS	Reseller Hosting	Shared Hosting	Web Agency
Managed	*VPS-M-1*, *VPS-MU-1*, *VPS-MU-2*, *SH-MU-1*, *SH-M-4*, *SH-M-6*	*SH-M-2*, *SH-M-3*, *SH-M-4*, *SH-M-5*, *SH-M-6*	*SH-M-1*, *SH-M-2*, *SH-M-3*, *SH-M-4*, *SH-M-5*, *SH-M-6*, *VPS-M-1*	*WA-M-1*, *WA-M-2*, *WA-M-3*
Unmanaged	*VPS-U-1*, *VPS-U-2*, *VPS-U-3*, *VPS-U-4*, *VPS-MU-1*, *SH-U-4*, *VPS-U-5*	*SH-U-2*, *SH-U-4*, *VPS-U-1*, *VPS-U-2*, *VPS-MU-2*	*SH-U-1*, *SH-U-2*, *SH-U-3*, *VPS-U-3*	*WA-U-1*
Management outside contract	*VPS-MU-1*, *VPS-MU-2*		*SH-U-1*, *SH-U-2*, *SH-U-4*, *SH-MU-1*	

Table 6.2: Distribution of HPOs' services by support levels (rows) and service types (cols). The row *management outside of contract* is a new finding, absent from initial market mapping.

6.2 Notification Channel and Message

This section presents our findings on the unprompted communications received by HPOs from external actors regarding the security and privacy of their infrastructure or hosted customer instances, addressing the first part of **RQ1** on *How are vulnerability notifications received*. Our questions focused on factors explored in prior work, such as reachability and email characteristics, while allowing participants to share their experiences with all forms of unprompted communication.

6.2.1 Receiving VNs

6.2.1.1 Awareness of VNs

All but three participants (*WA-M-1*, *VPS-M-1*, *SH-M-3*) were familiar with the concept of vulnerability notification. Most participants regularly engaged with them (e.g., *SH-M-1*, *SH-U-1*, *WA-M-2*, *VPS-U-1*, *VPS-MU-1*, *SH-MU-1*) and had a positive or neutral view of this mean of communication. Conversely, a subset of participants expressed skepticism regarding the intent of VN senders (*WA-M-3*), or doubted that external actors could identify vulnerabilities they were not already aware of (*VPS-U-3*, *SH-M-2*). For others, the idea of receiving notifications from *unrelated* third parties was unfamiliar, expecting to receive vulnerability reports only from established business relationships or trusted communication channels (*VPS-M-1*, *SH-M-3*, *VPS-U-5*).

6.2.1.2 Reachability

Participants were asked about the channels available for external actors to report VNs. Responses highlighted seven distinct communication endpoints and six different channels, with some participants identifying multiple methods.

WHOIS was the most frequently mentioned channel (seven HPOs), primarily among organizations managing their own infrastructure. One large company reported primarily receiving machine-readable abuse reports as a paid service, and two more reported preferring submissions via the company portal. Interestingly, one VPS provider reported only receiving abuse notifications through their provider, as they rent all infrastructure and don't own any IPs.

Web agencies provided contact details either in website credits or on a separate page, such as an "Impressum" or a `security.txt` file (one agency). None relied on WHOIS for communication. Small web agencies focusing on website development and management relied heavily on their hosting providers to inform them of issues, often due to a lack of in-house expertise or lack of alignment with their business model and setup (*WA-M-1*, *WA-M-3*).

6.2.2 Content and Sender Characteristics

6.2.2.1 Senders of VNs

HPOs receive security-related communications from a diverse range of senders. Eight participants reported receiving notifications regularly from various entities, which can

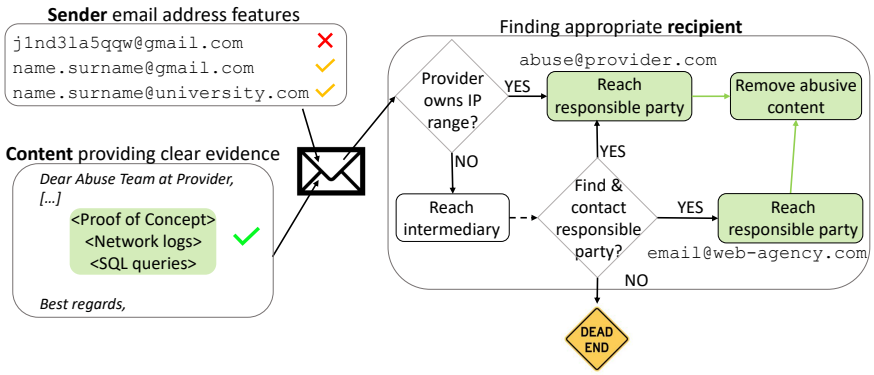


Figure 6.1: VN creation process based on provider feedback. The sender's email is only relevant if it raises red flags; the body's evidence is most important. On the left, the diagram shows how the email's routing depends on the recipient: if the HPO owns the IP range (per WHOIS), they can be contacted directly. Otherwise, the VN is sent to the infrastructure provider (i.e., the IP range owner, e.g., a data center or cloud provider), who acts as an intermediary and must forward the message to the affected HPO.

be grouped into three main categories.

First, four participants (*SH-U-2*, *VPS-U-1*, *VPS-MU-1*, *SH-U-3*) mentioned receiving security advisories or investigation requests from government agencies, CERTs, or police departments. Then, four participants mentioned established commercial companies focused on spam, malware, phishing (e.g., Netcraft), or legal issues as brand protection and DMCA reports. These senders were seen as reliable and their notifications were generally addressed promptly due to their established reputations.

Three participants reported receiving reports from private individuals, referred to as « white hat hackers » (*WA-M-2*) or « technician[s] » (derogatory, *WA-M-3*). However, opinions on these reports were divided. While some participants appreciated these notifications, others found it challenging to distinguish genuine reports from unsubstantiated claims or requests for monetary compensation. One participant summarized the skepticism: « ["Security researcher"] is not an official protected term, a lot of people call themselves security researcher, which might be the 14 year old from Bangalore who thinks revealing the web service version number is dangerous » (*VPS-U-1*). Several participants, including both small and large HPOs, reported experiencing high volumes of spam in their VN inboxes. The cause remained unclear, as both affected and unaffected participants used spam filters and public `abuse@` addresses. Abuse reports from private individuals were uncommon: only three participants recalled interactions with academics, while others mentioned cases involving individuals directly affected by abuse on their platforms, such as victims of DDoS attacks or impersonation.

6.2.2.2 Email Content and Metadata

Among participants familiar with VNs, the criteria for evaluating reports were a key discussion point. While prior work highlighted email features (style, tone, metadata), most participants downplayed their importance. Only a few preferred formal email addresses; most based decisions on technical content, prioritizing reports with evidence

(e.g., logs or Proof-of-Concept), and directly examined of the reported website. *WA-M-3*, *WA-M-2*, and *VPS-MU-2* report receiving emails from private addresses, raising suspicion: « [T]his is a cryptic email address [...]. So everything looks really, really phishy. And afterwards you say, okay, thanks for your report » (*WA-M-2*). *VPS-MU-2* also addressed the case of random email addresses, which prompted extra scrutiny: « Some people [...] send it from some random streaming of characters @gmail.com. In which case, we take a look at it a little bit closer ». In summary, most providers accept personal email addresses, but emails from random-looking senders are examined more critically, with the final assessment based on the report's content.

Takeaways Our findings do not indicate significant roadblocks concerning awareness and posture towards VNs. While we observed that reachability can be an issue, given by hosting setups involving middle layers as registrars, resellers, and outsourced services, most providers still indicated WHOIS as the right source for their contacts. Figure 6.1 illustrates the creation of a VN after the feedback from our providers. Providers reported being contacted regularly by security companies, which creates established communication channels and patterns. Rarely receiving notifications from individuals, such as researchers, might impact HPOs' perceptions of these communications, potentially making them more suspicious. However, participants noted that reports including strong evidence of issues were always considered, regardless of sender reputation and email metadata.

6.3 VN Handling Procedures

This section examines the internal processes triggered when a security issue is detected or reported to an HPO, addressing the second part of **RQ1** on *How are vulnerability notifications processed*. Participants described how security event management was handled in their organizations, including the roles and responsibilities of security teams, remediation procedures, internal communication protocols, and interactions with customers or external stakeholders.

6.3.1 Internal Handling Procedures

HPOs can adopt structured methods to ensure consistent handling of common problems, sometimes referred to as "playbooks", though the level of formalization varies significantly. We investigated the presence of formalized procedures regulating vulnerability notification handling.

We observed that most providers do not follow a strict or formalized process for handling vulnerability notifications, though some have more structured approaches than others. For example, operators at *VPS-MU-1* and *SH-M-5* create playbooks for managing security events as part of their regular tasks, while *SH-MU-1* reported having a playbook to address VN reports at their previous employer. Other HPOs standardized interventions in customer spaces via internal « wikis » (e.g., *SH-U-1*, *SH-M-2*, *SH-M-3*, *SH-U-3*).

Most participants described having minimal or no formal guidelines for handling security events. For example, at *WA-M-2*, security decisions are made based on log interpretations, while at *SH-M-2* and *SH-MU-1*, operators are encouraged to address issues leveraging their own skills. *VPS-U-1*, also owner of a spin-off, *SH-U-2*, relies on a habitualized [18] cybersecurity awareness grounded in his employees' professional socialization within tech-savvy hacktivist backgrounds or open-source programmer communities. Thus, they are assumed to be highly sensitive to S&P topics: « The majority of my team members are coming from like hacker spaces and generally the open source community. And they already have a deeply ingrained approach, they discover security problems in open source software on a daily basis and handle them in appropriate ways. And I think they do this in the best and most responsible way ».

6.3.1.1 Impact of Internal Organizational Structures

We interviewed IT professionals across a wide range of roles, including executives, abuse analysts, and support staff, from both small and large hosting providers, ensuring a diverse and balanced perspective across organizational scales. This diversity allowed us to examine who handles VNs, the steps involved, and the interactions among professionals within hosting organizations. This was particularly evident in larger companies, where specialization and compartmentalization are more common. In such cases, VNs are typically managed by a dedicated abuse department with access to the `abuse@` mailbox. These operators evaluate VNs in a self-contained process that rarely involves other departments, and may also develop internal playbooks for handling specific types of application-level abuse. When VNs involve legal concerns, as fraud or DMCA complaints,

they may be forwarded to the legal department. In rare instances, when VNs relate to infrastructural issues, such as those at the IP level, the abuse team may collaborate with infrastructure-focused employees, such as DevOps or Site Reliability Engineers.

Customer support also plays a role in abuse handling as the first point of contact for clients. While most interactions involve routine setup or configuration, support agents may occasionally assist clients affected by exploits. In complex cases, and where company structure permits, support staff can escalate issues to more experienced personnel, such as "Level 2 support" or the abuse team. However, in practice, the two departments usually operate independently. Participants reported no interdepartmental friction or procedural obstacles in addressing VNs. In smaller companies, roles are often consolidated, with a single employee managing the entire process.

6.3.1.2 Procedures Established by Certifications

Certifications are formal attestations granted by recognized standardization bodies, indicating that an organization's processes or products meet specific industry requirements. When a company certifies its procedures or software, it means these have been evaluated and shown to comply with the defined criteria of the relevant standard (e.g., ISO/IEC 27001 for information security management). Certification can apply to all internal procedures or only selected ones, depending on organizational priorities. In this context, we sought to determine whether holding certifications related to security or incident response had any impact on how HPOs manage VN handling. Few participants (*VPS-MU-1*, *VPS-M-1*, *SH-MU-1*) reported getting one or more certifications to access specific market segments, such as public administration (*VPS-MU-1*, *SH-MU-1*) or Limited Liability Companies (*VPS-M-1*). « We are about to take the ISO 27001 certification, [...] Because, you know, it is fundamental for this type of thing. [...] Because, above all, customers ask for it » (*VPS-M-1*). Certifications were however not mentioned as a factor playing in procedures for remediating issues or vulnerabilities in customer spaces, situations for which participants reported following other legal frameworks (especially nation state's or GDPR). These participants reported certifying procedures such as « datacenter encryption » (*VPS-MU-1*) or « SLA requirements and disaster recovery » (*VPS-M-1*). Notably, *WA-U-1*, a web agency developing software for the public administration, reported having no constraint on the application software.

6.3.2 Procedures Involving External Stakeholders

We examined potential remediation challenges arising from a multi-stakeholder setup. Participants did not report dependencies on external entities involved in remediation, such as CERTs or Managed Security Service Providers.

6.3.2.1 Interactions between Hosting Providers, Web Agencies, and Final Customers

One notable multi-stakeholder dynamic is the "reseller hosting" model (introduced in Section 6.1.1.2). We investigated the responsibility and authorizational boundaries between hosting providers and their clients ("resellers", or web agencies). *SH-M-2*

and *SH-M-3*, both managed hosting providers, described their admin panel as having a fine-grained access control system that enables both intermediaries (web agencies) and end-users (web agency clients) to act within customer spaces, effectively removing authorization barriers. While this paints a positive picture, none of the web agencies outsourcing hosting (*WA-M-1*, *WA-U-1*, *WA-M-3*) reported using managed shared hosting. Instead, they preferred service packages without customer support, shifting the burden of remediation to themselves. A few participants described a strict authorization boundary in both directions (e.g., *SH-M-1*, *SH-M-6*, *WA-M-2*, *WA-U-1*). Hosting providers noted that some resellers enforce clear separation of responsibilities (« We do not interfere there, and actually many of them do not allow us to touch [their] products », *SH-M-6*). Conversely, web agencies reported losing contact with clients after the initial website development, as ongoing management was transferred to the client's internal IT department.

Finally, web agencies (e.g., *WA-M-2*, *WA-M-3*) might experience issues when onboarding of customers with externally developed websites. In such cases, *WA-M-3* notes that their typical solution is to freeze the codebase until the customer agrees to pay for a complete rewrite, even if the website runs outdated or vulnerable code.

6.3.2.2 Procedures at Registrars

Domain registrars are companies authorized to manage the reservation of internet domain names and typically operate separately from hosting providers. In cases of abuse, registrars may receive complaints related to domain ownership or misuse, where they can react by, e.g., disabling domain resolution by web clients. However, enforcement actions like content removal fall under the responsibility of hosting providers.

Two participants, *SH-U-3* and *SH-MU-1*, also acted as domain registrars, providing unique insights absent from accounts by other interviewees. We report their valuable insights, acknowledging the limited generalizability. The participants reported checking all incoming emails, regardless of sender or email features, considering as only criterion whether the domain or content was under their control. Providers reported taking immediate action in case the domain hosted illegal material (e.g., phishing, copyright violations, child pornography, or other breaches of local law), suspending the domain and notifying both customer and hosting provider. For all other types of issues, HPO behaviors differed depending on their business model: *SH-U-3*, offering unmanaged services, left the website untouched, even if obviously compromised. *SH-MU-1*, offering both managed and unmanaged services, evaluated whether the reported flaw posed risks to infrastructure stability, other customers' instances, or sensitive data security. If none of these applied and the customer was not a premium client, they sent an email notification to the client without further action.

Takeaways Few HPOs reported having a clearly defined process outlining specific steps and criteria for handling vulnerability notifications, while most of them follow semi-formalized procedures which participants generally describe as flexible and not limiting to their operations. Handling of incoming reports is typically left to the discretion of individual operators, where most rely on unwritten but commonly followed rules. Internal HPO structure and division of responsibilities was not reported as a roadblock by participants working in medium to large HPOs. Few hosting providers reported having clear authorization boundaries when dealing with web agencies as intermediaries, in which cases they cannot take action directly and simply forward the received VN. Most web agencies confirmed that they rely on hosting providers to be informed about website issues.

6.4 Deciding Whether to Intervene

After establishing that some VNs are received and that handling procedures are often informal, we asked participants what actions they would take and why, addressing **RQ2**. Their decision-making criteria largely fell into three categories, outlined below. Participants also reported challenges in determining whether to intervene, as well as exceptional circumstances that prompt remediation.

6.4.1 Type of Vulnerability or Issue

We investigated whether the type of reported vulnerability influenced providers' remediation decisions. In general, action was taken when there was evidence of ongoing malicious activity, such as phishing, spam, malware delivery, or distribution of illegal content. Almost all participants reported they would respond by removing the malicious content, and potentially blocking parts of the website or taking the website offline. Additionally, they notified affected customers, using methods ranging from simple alerts to more engaged outreach prompting customer-side remediation.

Alerts concerning *potential* exploits, such as vulnerabilities in web applications, were treated differently. VPS providers, whether managed or unmanaged, stated they were not contractually obligated to oversee application software installed by customers, even when deployed using tools provided by the HPO. As a result, they did not address security and privacy issues at the application layer. This approach was described by *SH-U-2* as « the infrastructure-customer divide », defining the boundary of responsibility between provider and customer. These providers focused on maintaining infrastructure stability by managing elements such as operating systems, databases, and programming environments, and responded to reports involving Denial of Service, port scanning, or email spam, but drew a firm line at application-level issues.

In the case of shared hosting, unmanaged providers described a hands-off approach to all web application concerns, stating « it's basically out of scope for us to take care of that [...]. Then you've got a shell there. Have fun. » (*SH-U-2*). Managed service providers expressed a similar stance: « You get a place where you do not think of operating system upgrades, control panel upgrades, compatibility issues, [...] everything is covered. But it's that's where we our job ends and it's your responsibility to upload the software and after that to update it and keep it to date and safe » (*SH-M-6*). This underscores a strict division of responsibility between hosting infrastructure and code management, further echoed by another provider: « [But if] the issue with the application [is] like, in the coding, so we do not provide any development-related support » (*SH-M-4*).

Operators reported two reasons behind this approach: first, the low cost of their offerings, which does not compensate for the time spent remediating and, additionally, the « bajillion » tickets they need to deal with daily (*SH-M-5*), requiring them to minimize the time spent for each of them.

6.4.2 Legal Constraints

Legal obligations play a significant role in motivating hosting providers to take action. For example, there are cases where the operator must apply a patch or update to

avoid legal liability, or instead, they refrain from intervening, even when capable, because taking action could expose them to legal liability. Two examples illustrate these dynamics. Participants generally reported no awareness of legal requirements concerning the maintenance of hosting infrastructure or software. *SH-MU-1*, for instance, confirmed the absence of such regulations but emphasized that existing laws do penalize service providers in the event of a data breach. To reduce this risk, the company has adopted an internal update policy aimed at preventing incidents that could lead to legal consequences: « [T]here is no law that tells you that all systems must be in the last stable version, but clearly it tells you that if, because of the system, [because of] a vulnerability... personal data, sensitive data, and so on, get stolen, you are responsible » (*SH-MU-1*).

The second example concerns unmanaged contracts. In these cases, the legal obligation typically extends only to notifying the client about the issue (e.g., *SH-U-1*, *WA-M-2*, *SH-U-3*, *SH-M-6*). Once the report is delivered, responsibility shifts to the client, and the operator is no longer involved. Applying code patches or modifying incompatible modules is not only unnecessarily costly when the issue does not affect others, but also risky, especially for highly customized setups. An unfamiliar operator might unintentionally break the site: « [I]n the [BUSINESS UNIT] where I am, there are about 50 people more or less, and the clients are.. hundreds of thousands... so... Understanding and taking action on every single personalization that can be done on the various websites, CMS websites [...] would be madness. » (*SH-MU-1*). As a result, proactive remediation is not an option, since making changes would make the operator legally responsible. In managed contracts, when customers request a fix, some operators may choose to intervene but require the customer to sign a waiver before making any changes to the hosting environment.

6.4.3 Customer-Specific Factors and Interactions

6.4.3.1 Interactions with Customers

In some cases, customers react to vulnerability notifications and reach back to their hosting providers. Participants expressed mixed feelings about customers raising such issues (e.g., *SH-U-1*, *WA-M-2*, *WA-M-3*). They frequently reported that customers lacked the necessary understanding, demanded significant time for explanations, and created an unnecessary burden on their operations. Customer service is particularly costly for providers, creating challenges for both unmanaged and managed hosting services. Many HPOs face risks of financial losses due to the high cost of support agents relative to their low prices.

Many HPOs (e.g., *SH-U-1*, *WA-M-2*, *VPS-U-2*, *VPS-MU-1*) report that customers are often unwilling to pay security services (e.g., website cleanup). To mitigate this, providers like *SH-U-2* and *VPS-U-1* focus heavily on creating detailed wiki articles to assist customers, without direct support, while *SH-M-2* requires users to consult their knowledge base before contacting support: « [I]f everybody comes to tickets, our tech support team will be simply overwhelmed » (*SH-M-2*).

6.4.3.2 Management Defined by Contract

Companies offering unmanaged services limit customer support actions to written suggestions, such as troubleshooting steps or links to online resources (e.g., *SH-U-2*, *SH-U-3*). Few of them reported expectations mismatches with customers on the level of management happen,usually solved via polite explanations (*SH-U-2*, *VPS-U-1*) or by proposing additional paid services (e.g., *WA-M-2*, *SH-MU-1*, *SH-M-7*), with some, like *SH-U-3*, upselling services with sales-based bonuses. Managed service providers offered broader support, though intervention typically required a customer request and varied broadly by provider. Support agents often demonstrated goodwill in helping customers by updating software (e.g., *WA-M-1*, *SH-M-1*), restoring backups, or removing affected content upon request (e.g., *VPS-M-1*, *SH-M-2*, *SH-M-3*). No provider mentioned actively inspecting for vulnerabilities, and only one (*WA-M-1*) reported reviewing plugin code to ensure functionality.

Code-level remediation was rare and generally offered when contractual services included custom web application development (*SH-M-1*, *WA-M-3*, *VPS-MU-1*, *WA-M-2*). For example, *WA-M-3* described a complex, manual remediation of a hacked WordPress site redirecting visitors to malicious content. This intervention followed a customer complaint triggered by a Google SafeBrowsing warning. However, the same provider admitted to ignoring outdated PHP versions with known vulnerabilities and websites lacking HTTPS. This suggests that customer involvement may have strongly influenced their decision to act.

Conversely, business concerns might motivate providers not to address security issues. For instance, *WA-M-2* and *WA-M-3* develop custom software, generating revenue by offering improvements as new product features. *WA-M-3* notes that while their custom CMS is regularly updated and patched, updates are withheld from customers due to their unique customizations and to encourage them to purchase updated installations.

6.4.3.3 Exceptional Proactive Remediation

Occasionally, hosting providers take proactive measures to address compromised customer instances without waiting for them to reach back. These actions, such as disabling compromised websites or sections of them, are taken to safeguard the overall infrastructure's security and ensure fair resource allocation among all customers. This approach is often necessary because customers are slow to respond to notifications (e.g., *SH-U-1*, *SH-U-2*, *VPS-U-1*, *SH-MU-1*).

The motivations for such proactive efforts are often rooted in business strategy. For example, *SH-U-1* emphasizes high-quality customer service as a key differentiator but seeks to reduce time-intensive customer calls by addressing issues proactively. *VPS-MU-1*, which blends managed and unmanaged services, may exceed their contractual obligations to maintain goodwill and strengthen customer relationships–a decision driven by their administrative department. *SH-U-2*, *VPS-U-1*, and *VPS-MU-2* on the other hand, may proactively address compromised customer instances out of a commitment to prevent internet abuse, a stance shaped by their organizational and ethical principles: « We consider this as part of our responsibility that we not only have for our customers, but also for the rest of the internet. We don't want our machines to attack other

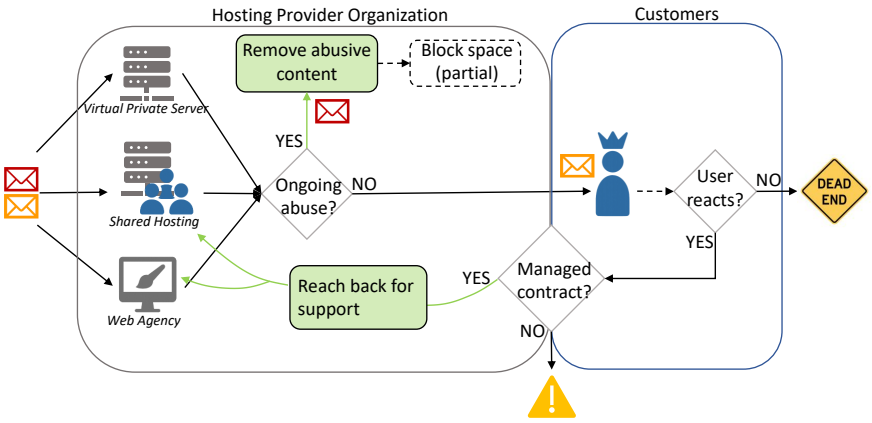


Figure 6.2: Notification handling workflow. Incoming reports are analyzed by abuse teams and abusive content taken down, with further actions depending on the HPO. Reports are otherwise forwarded to customers, and escalation to customer support depend on user action and contract terms.

machines » (*VPS-U-1*).

Takeaways Figure 6.2 illustrates the VN handling workflow emerged as emerged from our interviews. Most providers indicated their abuse team promptly addresses abuse such as phishing, malware delivery, or distribution of illegal content, regardless of the type of hosting and of the degree of service management. Reports concerning other types of web-application issues are forwarded to customers. In fact, our findings reveal a common understanding of service boundaries: customers, or web agencies in their stance, are responsible for code maintenance, while hosting providers focus on infrastructure and service availability. Moreover, the type of service provided affects the level of customer support, with some providers supporting users in remediations to a larger extent. Finally, providers are also influenced by factors that extend beyond the content of the VN, as the company's ethical position on addressing internet abuse, the perceived value of long-term customer satisfaction, and the motivations of individual support agents.

6.5 The Role of VNs in Infrastructure Security

While most hosting providers do receive vulnerability notifications, our earlier findings indicate that these are often not prioritized in practice, highlighting a misalignment between provider security goals and those of researchers. To better understand this disconnect, we examined providers' awareness of abuse within their infrastructure, the security measures they employ at a technical level, and their perceptions of risk related to compromised customer instances.

6.5.1 Provider Awareness of Abuse and Security

We began by examining the extent to which hosting providers are aware of abuse and of the security posture of their customers operating on their infrastructure. Our goal was to assess whether visibility given by VNs could offer providers insights they currently lack, demonstrating practical value in enhancing their situational awareness.

Most hosting providers focus on maintaining visibility and control over their infrastructure to ensure reliable service and uphold service-level agreements. Providers who own their hardware (e.g., *SH-M-6*, *SH-MU-1*, *WA-M-2*, *VPS-U-3*, *VPS-MU-2*) typically implement routines that monitor network activity, I/O, RAM, and other resource usage. For some, these monitoring systems also serve as early warnings for potential security incidents, but it usually falls to a skilled operator to determine if unusual activity reflects legitimate use or malicious behavior (*WA-M-2*). HPOs that do not manage their own hardware (e.g., *WA-M-1*, *WA-U-1*, *WA-M-3*, *VPS-U-5*) often depend on tools supplied by their upstream hosting partners, using these to gain some operational insight. In some cases, structured monitoring is minimal or absent, with providers relying mainly on customer reports to identify problems (*WA-M-3*, *VPS-U-4*).

Overall, the primary goal of monitoring is to maintain infrastructure performance, not to detect abuse or assess the security of customer spaces. As a result, if customer instances are compromised but do not disrupt service or exceed expected resource usage, these incidents frequently go unnoticed and unaddressed. Only a minority of providers report actively assessing the security posture of their customers' environments (*SH-U-1*, *SH-M-2*).

6.5.1.1 Securing Hosting Infrastructure

Most providers secure their infrastructure's by employing protections such as DDoS mitigation, firewalls, and isolated virtual networks (*VPS-U-2*, *VPS-M-1*). Many use Linux permissions and least privilege principles to prevent customer infections from escalating (*SH-U-1*, *SH-U-2*, *VPS-U-1*), though implementation varies widely based on provider size and whether hosting is in-house or outsourced. Only few providers monitor for abuse higher up the stack, with some reducing risk by by preventing web app code edits: « the WordPress code itself is locked. Like no one can edit it. [...] So that's something that is totally secure and locked down » (*SH-M-3*). Alternatively, hosting providers use web application firewalls, or run signature-based file scans (*SH-M-1*, *SH-U-1*). Web agencies may also deploy code-level protections (*WA-U-1*, *WA-M-3*).

6.5.2 Perception of Risk from Customer Spaces

HPOs described a range of technical measures used to secure their infrastructure, with a primary focus on ensuring resilience and service continuity. Because their responses centered on infrastructure security, we examined their perceptions of risks associated with customer spaces. Our goal was to understand the reasons behind their limited engagement, including whether this results from a perceived low risk or deliberate trade-offs.

6.5.2.1 Perceived Risk from Customer Instances

Customer instance compromise is common, « we see WordPress exploits the whole time », as *SH-M-7* reports. However, providers expressed confidence in their infrastructure's ability to contain such incidents, keeping systems stable and functional. As *VPS-U-1* noted, « our infrastructure does not care about phishing ». Only one participant, *WA-M-1*, recounted a full compromise of their hosting space after they transitioned to a less managed setup to meet the demands of a larger client.

VPS instances are typically compromised due to customer misconfigurations or are acquired directly for illicit use (e.g., *VPS-MU-1*, *VPS-U-2*, *VPS-MU-2*). In shared hosting, abuse often stems from outdated websites and plugins, malicious uploads, cracked plugins, or weak or leaked admin passwords. Although attacks on popular open-source web applications are well known, providers continue to support them and deal with the resulting abuse. For example, *WA-M-3* accommodates customers who request WordPress sites, despite acknowledging they are frequent targets: « [E]very now and then we get that client who insists on having WordPress no matter what, so, to avoid losing the client or the project during the quote phase... ».

Web agencies expressed a different perception of risk compared to shared hosting and VPS providers. For them, the main concern is protecting data, as they see little incentive for attackers to target custom code given the high cost and low potential gain. *WA-M-3* believes that custom-built web applications are inherently secure due to their unique nature, leveraging a form of "security by obscurity". *WA-M-2* similarly claims they have never had a hacked website, although they experienced a sophisticated data exfiltration attack. While custom code can reduce exposure to automated attacks, *WA-M-3* and *WA-M-2* also acknowledged that it introduces risks from implementation flaws and overlooked interactions. Nonetheless, they believe the resources required to test for vulnerabilities across many customized deployments are not economically justifiable.

Finally, risk perception also depends on what providers consider worth protecting. *WA-M-3* for instance, disregards HTTPS implementation on sites he believes contain nothing valuable. Similarly, *VPS-U-2* suggests that security should reflect business value, stating that loose measures are fine if all you are protecting is « three fishbones and four sandwiches ».

Ultimately, we found that most providers in our study shared a similar philosophy, particularly those offering low-cost, standardized hosting services through public sign-up processes. In contrast, two providers primarily serving large businesses or public administration (*VPS-MU-1*, *VPS-M-1*), as well as one with a mixed clientele (*SH-MU-*

1), demonstrated significantly greater attention to security. These providers described remediation processes that addressed more complex threats than typical WordPress compromises.

Our findings suggest that most hosting providers view customer instance compromise as posing minimal risk, considering proactive application-layer security systems as an optional enhancement rather than a necessity. Providers frequently targeted by low-sophistication attackers, such as "script kiddies", may be particularly likely to hold this view. In contrast, providers who implement comprehensive security measures tend to face more sophisticated threats and do not rely on standardized setups commonly targeted by automated attacks. This, in turn, may reinforce the disconnect observed with vulnerability notifications among the former group, as security in these hosting environments is regarded as an added benefit rather than a core requirement.

Takeaways The security posture of hosting providers is shaped primarily by operational priorities and the need to protect their own infrastructure, rather than by concerns for individual customer environments. Resource monitoring is primarily intended to maintain service level agreements and manage legal risks, and any detection of abuse within customer spaces tends to be incidental rather than the result of targeted efforts. Providers generally express confidence in their technical safeguards to isolate issues and protect core systems. This aligns with and reinforces their passive stance towards application-level vulnerability notifications.

6.6 Discussion

We conducted and analyzed semi-structured interviews to explore why hosting provider organizations often fail to remediate reported vulnerabilities. Our analysis focused on how vulnerability notifications are received and handled (**RQ1**), identifying the underlying criteria behind the limited remediation responses commonly noted by researchers (**RQ2**). This section summarizes our findings and future directions based on insights from our 24 participants.

6.6.1 Comparison with Prior Works

We investigated impactful factors highlighted by prior works, such as contact channels, email content, and features of the sender, as well as HPOs' internal structures, analyzing whether and to what extent they contribute to the lack of VN remediation.

First, our study revisits the reachability issues observed in previous work [216, 30, 182]. In fact, most hosting providers confirmed that their contact is available via WHOIS, for example through ARIN [8] or RIPE [187] databases. However, the issue of reachability persists when expanding the scope to actors not considered in prior studies, such as web agencies, who may have an actual impact on remediating reported issues. Additionally, trust plays a different role than in earlier research, which emphasized general distrust [84, 215, 141]. Our participants noted that clear, technical emails with credible explanations are usually considered, though not always acted upon. Finally, while prior research emphasized inner-organizational technical and structural barriers to secure practices [89, 45, 212, 6], and challenges with keeping systems up-to-date [128, 231, 95, 143], participants in our study reported no major roadblocks in managing their own infrastructure, suggesting that these factors are less critical for VN remediation [16]. The observed lack of remediation was explained by hosting providers as a result of the nature of the vulnerability, often viewed as outside the provider's responsibility. This new framing also helps explain why operator awareness did not always lead to remediation [127, 215, 48, 16].

Notably, the reasons reported by operators differ from those described in [60], which investigates the lack of patching following vulnerability reporting in Dutch municipalities. While the concept of responsibility is central in both works, Ethembabaoglu et al. attribute non-patching to a lack of awareness leading to neglect [60], whereas our study shows that remediation in commercial hosting providers is often refused based on contractual terms. In web agencies, remediation is only partially implemented, depending on business-related factors.

6.6.2 Implications and Future Directions

6.6.2.1 VNs as a Source of Awareness

We thoroughly investigated the technical infrastructure and security measures at hosting providers. Most providers believe their infrastructure is resilient to malicious activity occurring in customer environments. As a result, and sometimes also due to their large customer base, many providers do not closely monitor ongoing abuse or enforce

remediation. Previous works also observed the lack of proactive remediation by HPOs [23] which, complemented with website owners' neglect or forgetfulness concerning the security posture of their websites [83, 217], underscores the relevance of vulnerability notification campaigns, which help fight internet abuse at scale, raising HPOs' awareness.

6.6.2.2 Improving on Reachability

The most straightforward, though not necessarily more effective, approach to increase remediation rates may be to notify those actually responsible. Identifying website owners at scale remains a well-known challenge [215, 141]. To overcome this, researchers could explore more advanced automated methods for identifying contact points, selecting appropriate targets such as website owners or web agencies based on the issue type. Future work could involve AI agents capable of navigating websites [213], locating contact information, and extracting relevant text. However, researchers should consider that many end users do not hire professionals to maintain their websites. The content of vulnerability notifications should therefore be tailored to the recipient, as prior studies have shown that end users interpret such messages very differently [85, 84].

At the same time, successful large-scale vulnerability notification has been shown to be feasible through paid services [153]. This raises a broader question about whether the challenge lies in the lack of a standardized infrastructure available to researchers. Future work could address this by developing a shared contact database, similarly to the development of Tranco [122] as an alternative to Alexa 1 Million, in order to reduce the impact of limited reachability on remediation outcomes.

6.6.2.3 Lowering Cost of Remediation for Providers

Many providers expressed a cynical view of the current state of abuse in shared hosting, where hackers compromise hundreds of sites with a single click and website owners are described as unskilled and careless, both in managing websites and in valuing website data. Especially among website hosting providers, the volume of daily abuse tickets was reported to be overwhelming (similarly to [207]), leaving little room for careful attention to individual cases. A few providers mentioned occasionally resolving issues proactively, not out of policy but to avoid more costly customer service calls. Still, increasing staff to handle abuse reports more thoroughly appears impractical, as it would require raising prices in a highly cost-sensitive market.

Although providers set pretty strict responsibility boundaries, researchers could focus on making remediation easier and less resource-intensive for them. Many providers value technical detail, suggesting that notifications including clear evidence of the issue and detailed remediation steps could lead to higher response rates.

Moreover, while simply updating the web application software is sometimes the most effective and straightforward remediation, this is often not feasible, as updates can cause incompatibilities with plugins. Our study shows that in such cases, providers often choose not to take action. Previous studies have shown that end users may be unaware of the need to update or may be unwilling to do so when updates impair website functionality [83]. This issue is further complicated by the open-source nature of many free web application tools, which are often untested or no longer maintained. In

such situations, researchers who discover the problem could help identify stable versions of the software to which updates are possible, or recommend alternative software that maintains overall functionality. End users, who often prioritize functionality over security, may otherwise never seek such alternatives, possibly due to a lack of awareness of the associated security risks.

6.6.3 Limitations

The field of research on vulnerability notification is broad, and our study does not aim to cover all possible scenarios. We focused specifically on web application vulnerabilities, whereas responses may differ for issues affecting HPO infrastructure, lower layers of the Internet stack, or internet-connected devices. Additionally, we did not examine hosting services such as Dedicated Servers, Colocation, and Website Builders. These were excluded due to the nature of provider involvement in each case. Dedicated Servers and Colocation offer an extremely high degree of user control, with minimal provider intervention, while Website Builders represent the opposite extreme, offering limited customization through provider-specific, closed-source platforms.

Third, we interviewed only one participant per HPO, potentially missing other perspectives on security in larger organizations. Despite our efforts, recruiting proved challenging. While interviewees provided valuable insights into company practices, future studies should conduct multiple interviews within the same HPOs to better capture innerorganizational dynamics of dealing with cy-sec issues and VN handling.

Fourth, some providers may not have been fully transparent about their practices, potentially withholding information for reputational reasons. While we observed no signs of this during interviews, it remains a possible limitation. Finally, our study may be influenced by opt-in as well as social desirability biases, and demand effects. To minimize these, we avoided disclosing the study's exact focus on security and privacy topic during recruitment, encouraging open discussion and reducing bias related to prior VN experiences.

Summary

In this chapter, we investigated the roadblocks that IT professionals experience in remediating web vulnerability notifications. This study shifts the lens of vulnerability notification research by examining the recipients–hosting provider organizations–rather than optimizing the reporting process. To our best knowledge, this is the first study to deeply investigate how HPOs internally process and respond to vulnerability notifications. Through interviews with 24 HPOs, we find that while most providers are reachable and familiar with handling VNs, remediation remains limited due to structural and economic factors rather than technical ones.

Low remediation rates stem from clear service boundaries, where responsibility for code vulnerabilities lies outside the scope of hosting providers' obligations. Cost considerations, the commoditized nature of hosting services, and the perceived insignificance of individual customer environments further reduce the incentive to act. Moreover, remediation decisions are shaped by business priorities, ethical stances, and the discretion

of individual operators rather than systematic security policies. These insights challenge existing assumptions in VN literature and suggest that improving remediation rates requires addressing the underlying misalignment between the security goals of external reporters and the service logic of HPOs.

This chapter concludes the research work of this thesis, which analyzed how attackers leverage web-based technological innovations to develop novel deception-based attacks, as well as the operational aspects of these attacks and the shortcomings in the vulnerability remediation process. We conclude this thesis by highlighting the main contributions of the works presented and discussing future research directions.

7
Conclusion

In the last chapter of this thesis we summarize our contributions and provide pointers for future research directions.

7.1 Summary of Contributions

In this thesis, we looked at novel deception-based attacks enabled by technological innovations in two scenarios, providing a comprehensive analysis and systematization of these phenomena. Moreover, we looked at the operational aspects of these attacks, investigating which other enabling factors allow attackers to acquire and consistently keep access to the necessary web infrastructure.

Our first research question focused on *whether new integrations, technological developments, or communication platforms, can be exploited to mount deception-based attacks (RQ1)*, which we contextualized and conducted in the context of social platforms and browser technologies. Deceptive previews and clickbait PDFs, respectively, are web-based threats that misuse technology features and integrations to create deceitful content, with elements of textual, visual, and contextual deception.

Chapter 3 presented and systematized *deceptive previews*, UI components that social platforms automatically generate upon link sharing. We observed the link preview creation process in ten social networks and ten instant messaging applications, observing how most platforms adopt different layouts for the same semantic elements and meta tags. This heterogeneity of templates and inconsistent use of fields may prevent users from forming a secure mental model of link preview appearances. Moreover, four of these platforms allow a malicious user to create a benign-looking preview for a malicious link, possibly fooling even a tech-savvy user, while 18 fail to implement any countermeasure against malicious link distribution. We concluded our study with seven recommendations towards the creation of more reliable link previews.

Chapter 4 shifted focus to technological integrations of web browsers, focusing on how the seamless integration of PDF rendering into web pages can be misused. Clickbait PDFs have specific visual baits and lead to a variety of web attacks, as demonstrated by our identification of 44 clusters and eight distinct types of attacks, each with own volume and temporal patterns. Moreover, most clickbait PDFs are distributed via search engine optimization attacks, which places them out of the email delivery ecosystem that was typical of malware-bearing PDF files.

In Chapter 5 we study the operationalization of clickbait PDF campaigns, answering RQ2 on *how attackers leverage web infrastructure to support deception-based attacks*, by shifting our focus to the web infrastructure attackers use to deliver these PDFs. We conducted real-time analyses on hosts, collecting data on 4 648 939 clickbait PDFs served by 177 835 hosts over 17 months. Our results revealed a diverse infrastructure, with hosts falling into three main hosting types, a variety of eight exploited server-side components leveraged to upload the PDFs, and an average duration of abuse per affected entity of nine months.

These observations led us to investigate ways to stop the malicious activity and to understand why operators took little to no action in response to disclosure notifications, an outcome also observed in previous related work. Chapter 6 answers RQ3 on *how vulnerability notifications are processed within hosting provider organizations* through

our investigation on the causes of failures to remediate notifications of web vulnerabilities and abuse. We conducted a qualitative study with IT operators working at hosting provider organizations, investigating how are notifications received and processed, and which factors internal to the company influence the remediation outcome. Our findings indicate that low remediation rates are not primarily caused by unreachability, distrust, or internal procedural barriers. Instead, they stem from clear service boundaries, where responsibility for code vulnerabilities lies outside the scope of hosting providers' obligations. Cost considerations, the commoditized nature of hosting services, and the perceived insignificance of individual customer environments further reduce the incentive to act. Moreover, remediation decisions are shaped by business priorities, ethical stances, and the discretion of individual operators rather than systematic security policies.

Overall, our findings show that deception-based attacks remain a persistent threat, continually evolving as attackers adapt emerging technologies to serve deceptive purposes. The web infrastructure remains susceptible to exploitation and abuse, even through long-known vulnerabilities. This situation is reinforced by a sustained lack of responsiveness from hosting providers and website owners, which often allows malicious activity to go unchecked for extended periods. Moving forward, research should investigate whether alternative notification strategies can lead to more effective remediation, or whether shifting focus toward more cohesive and less fragmented web technologies could offer a more reliable path forward.

7.2 Future Research Directions

This thesis examined two forms of deception-based attacks: clickbait PDFs and deceptive previews. In both cases, we explored how attackers exploit trust by presenting users with seemingly benign information that conceals malicious intent.

Clickbait PDFs exemplify how attackers can manipulate search engine rankings (widely considered trusted entry points to information) to lure users toward malicious destinations. Similarly, the misuse of link previews on social platforms shows how attackers adapt to evolving content-sharing ecosystems, crafting visual summaries that appear trustworthy but ultimately redirect to harmful content.

Over time, we observed a shift in how users access information [9], moving from traditional search engines to social platforms. A further transition is now underway: the increasing reliance on large language models (LLMs) and automated agents for information retrieval and synthesis. As these systems become primary intermediaries between users and data, they introduce new opportunities for abuse. In particular, adversaries may attempt to influence the content presented by LLMs through various forms of manipulation. Echoing the techniques studied in this thesis, an attacker might craft responses that appear legitimate while embedding harmful links, or distort the factual correctness of a response by polluting upstream data sources. Understanding whether these forms of deception can be extended to LLM-mediated interactions represents a promising direction for future research.

Current research already highlights several vectors through which LLMs may be compromised. One is training-time poisoning, where malicious or misleading content is included in the training corpus–especially relevant for models trained on large-scale web

snapshots or dynamic sources [26]. Another avenue involves systems that use external resources through Retrieval-Augmented Generation (RAG). Even if the core model is trained safely, the external knowledge bases it queries can be poisoned, allowing attackers to inject crafted content that influences the model's responses. Recent work demonstrates that introducing just a few targeted documents can alter outputs for specific queries, effectively hijacking the model's behavior [268]. A further risk lies in prompt injection attacks. Here, malicious instructions are embedded in the content the model processes at inference time. Studies and informal experiments have shown that LLMs can be tricked into executing unsafe actions or disclosing sensitive data when exposed to carefully crafted input [174, 262].

Of particular relevance to our future research direction is the possibility that LLM-generated responses may include phishing links [227, 185]. This scenario mirrors the dilemmas explored in our case studies: users must decide whether to trust a system-generated context or treat embedded links with suspicion. As LLMs become more deeply integrated into daily workflows, understanding the implications of such trust dynamics becomes critical. Going forward, a promising line of investigation will be to systematically explore how LLMs access, interpret, and incorporate external information, and whether deception techniques similar to those studied in this thesis can persist or evolve in these new environments. In parallel, we must design and evaluate defenses that ensure the trustworthiness of LLM-based systems, both by detecting and mitigating deceptive content and by reinforcing the integrity of the models' inputs and outputs.

8
Appendix

8.1 Appendix to Chapter 4

8.1.1 PDF Clustering

In this section, we expand on the procedure used to cluster visually similar documents, described in § 4.2.2. First, we report the evaluation on the embeddings returned by DeepCluster [27]. Then, we explain which parameters were used to run DBSCAN [59] 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 [206], and perceptual hashing.

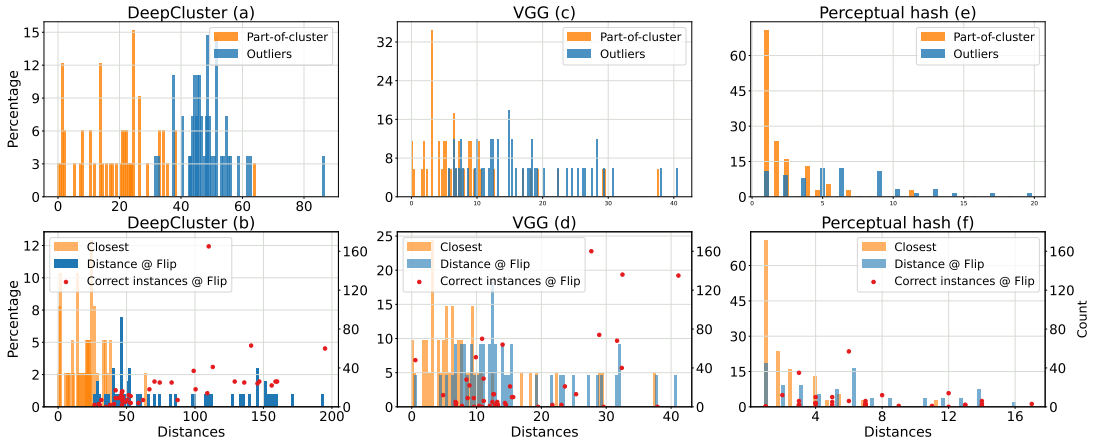


Figure 8.1: Comparing: (above) closest distances for outliers and 'part-of-cluster' samples, (below) closest distances of 'intra-cluster' samples and distances at cluster-flipping points. Scatter plots display correct instances (same cluster) at flipping points.

The first row in Figure 8.1 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

Step	Input Size	Clusters Clust. No.	Clusters Docs. No.	Incr.	New clusters Largest Campaign Name	Rank	New clusters Smallest Campaign 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	6	Download File	47
					Drive			
Full-manual	1 504	-	-	+36	AS PDF / File #12	10	Shared Excel	48
Outliers	389	-	-	0	-	-	-	-

Table 8.1: PDF cluster identification: overview of the input/output properties of each step.

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[1], collecting 1 071 samples, and

[1]Clusters can contain fewer than 20 files.

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[164].

8.1.2 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, 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.

8.1.3 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).

	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 8.2: 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

(a)

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

(b)

Table 8.3: (a) Clusters targeting one language (Vol. > 10 docs). (b) Multi-regional clusters.

8.1.4 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).

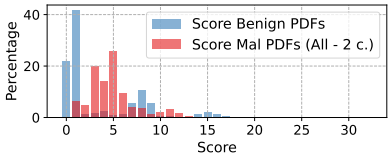


Figure 8.2: Histograms of the VT scores of benign and clickbait PDFs, without the two largest clusters, for the first 30 days.

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. § 4.3.3 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 [264], by Zhang et al., tackles both challenges. Unfortunately, this tool is not available in practice[2], making it challenging to analyze URLs at scale for our purpose.

8.2 Appendix to Chapter 5

8.2.1 The *Grape* Pipeline

8.2.1.1 Clustering Module

Creating a clustering algorithm for the number and nature of samples presented a challenge, due to the necessity of handling shifting visual baits appearance and identifying

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

ID	eTLD+1	# FQDN	# URLs	Host. Type
•	amazonaws.com	9	49 065	*Object storage*
•	strikinglycdn.com	1	54 052	*CDN*
•	f-static.net	1	47 931	*CDN*
•	sqhk.co	1	15 484	*CDN*
⊗	squarespace.com	1	13 000	*CDN*
•	shopify.com	1	11 994	*CDN*
•	s123-cdn-static.com	1	10 200	*CDN*
•	filesusr.com	3 241	9 829	*CDN*
•	mozfiles.com	1 741	2 366	*CDN*
M	s123-cdn-static-d.com	1	531	*CDN*
M	s123-cdn.com	1	139	*CDN*
M	s123-cdn-static-a.com	1	138	*CDN*
M	s123-cdn-static-c.com	1	134	*CDN*
M	s123-cdn-static-b.com	1	124	*CDN*
•	weebly.com	40 803	241 092	*Website hosting*
•	epizy.com	4 242	5 722	*Website hosting*
•	pbworks.com	1 005	4 255	*Website hosting*
•	wordpress.com	1 547	4 039	*Website hosting*
•	rf.gd	3 071	3 765	*Website hosting*
•	iblogger.org	1 617	1 975	*Website hosting*
•	22web.org	1 506	1 829	*Website hosting*
⊗	myhome.cx	20	1 220	*Website hosting*
•	getenjoyment.net	459	1 101	*Website hosting*
•	mywebcommunity.org	419	1 059	*Website hosting*
•	myartsonline.com	423	1 041	*Website hosting*
•	mypressonline.com	432	1 040	*Website hosting*
•	onlinewebshop.net	388	968	*Website hosting*
•	mygamesonline.org	408	958	*Website hosting*
•	sportsontheweb.net	406	951	*Website hosting*
•	scienceontheweb.net	376	951	*Website hosting*
•	medianewsonline.com	391	950	*Website hosting*
•	atwebpages.com	390	940	*Website hosting*
⊗	linkpc.net	4	896	*Website hosting*
•	66ghz.com	171	260	*Website hosting*
⊗	esy.es	1	208	*Website hosting*
⊗	wpengine.com	3	197	*Website hosting*
⊗	webhostmurah.com	1	113	*Website hosting*
⊗	gridserver.com	2	64	*Website hosting*
⊗	ovh.net	2	45	*Website hosting*
⊗	yolasite.com	44	44	*Website hosting*
⊗	hekko24.pl	1	42	*Website hosting*
⊗	webbazaar.com	2	35	*Website hosting*
⊗	000webhostapp.com	1	34	*Website hosting*
⊗	pokladnicka.cz	1	21	*Website hosting*
⊗	altervista.org	1	21	*Website hosting*
⊗	jpn.ph	1	20	*Website hosting*
⊗	leszno.eu	1	19	*Website hosting*
⊗	cafe24.com	1	18	*Website hosting*
⊗	tenten.vn	1	18	*Website hosting*
⊗	hostsolutions.ro	1	15	*Website hosting*
⊗	belonnanotservice.ga	1	8	*Website hosting*
⊗	home.pl	1	1	*Website hosting*
⊗	webd.pl	1	1	*Website hosting*
⊗	micron21.com	1	1	*Website hosting*

Table 8.4: Second-level domains and providers. Identification method (ID): • by threshold, M by manual analysis, ⊗ via Web analytics service (204).

Layer Name	Size In	Size Out	# Kernel
CNN-1	$128 \times 128 \times 3$	$128 \times 128 \times 8$	$(3, 3)$
CNN-2	$128 \times 128 \times 8$	$128 \times 128 \times 16$	$(3, 3)$
CNN-3	$128 \times 128 \times 16$	$128 \times 128 \times 32$	$(3, 3)$
MAXPOOL-1	$128 \times 128 \times 32$	$32 \times 32 \times 32$	$(4, 4)$
CNN-4	$32 \times 32 \times 32$	$32 \times 32 \times 64$	$(3, 3)$
MAXPOOL-2	$32 \times 32 \times 64$	$8 \times 8 \times 64$	$(4, 4)$
CNN-5	$8 \times 8 \times 64$	$8 \times 8 \times 128$	$(3, 3)$
MAXPOOL-3	$8 \times 8 \times 128$	$2 \times 2 \times 128$	$(4, 4)$
FLATTEN	$2 \times 2 \times 128$	512	
FC	512	128	
FC	128	32	
L2	32	32	

Table 8.5: Details of the model architecture.

new clusters without an available GPU. The pipeline operates in two steps: first, a CNN extracts a 32-dimensional feature vector from each sample, then, we use multiple iterations of DBSCAN to obtain document clusters. The embeddings are extracted daily, while the clustering procedure is manually triggered by a human operator.

The model. The model takes the screenshot of the first page of the PDF as a $128 \times 128 \times 3$ matrix and returns a 32-dimensional vector. It consists of five convolutional blocks (a sequence of Convolution, BatchNormalization, PreLu, and dropout functions), three downsampling operations (MaxPooling), and two final FC layers. Additional details about the model are shown in Table 8.5. For training, we used a contrastive triplet loss with a margin of 0.2, implementing a semihard online triplet generation approach, as described in [201].

Clustering. We use DBSCAN to group the PDF embeddings based on their appearance. To reduce the need for human intervention, we include a list of 20 pre-labeled items per group in the set of samples to be clustered. This list aids in automatically associating the clusters created by DBSCAN with existing known groups of visually-similar clickbait PDFs. DBSCAN starts clustering samples with a default $\epsilon_0 = 0.25$. If a computed cluster contains anchor samples from different campaigns, we reprocess its elements with $\epsilon_{i+1} = \epsilon_i - 0.01$ until the conflict is resolved. The clustering procedure is initiated manually by a human operator, who regularly inspects newly discovered clusters to verify the quality of the results.

Validation. We manually inspected 3 840 samples by selecting at most 500 random elements for each cluster. In total, we found 105 misclassified samples, resulting in an error rate under 3%.

8.2.2 IoCs

8.2.2.1 Software Components Facilitating File Upload

This section presents the eight software components whose poor security status may have facilitated the upload of clickbait PDFs on a website.

FCKEditor, CKFinder, CKEditor, KCFinder. FCKEditor was a rich text

SW Category	SW Name	# versions	# FQDNs
CMS	WordPress	189	5912
CMS	Joomla	3	879
CMS	Drupal	3	347
Ecommerce	Cart Functionality	0	1828
Ecommerce	WooCommerce	159	1491
Ecommerce	EasyDigitalDownloads	11	24
Hosting panels	Plesk	0	1029
Prog. language	PHP	280	18279
Web servers	Apache	73	15065
Web servers	Nginx	68	5592
Web servers	LiteSpeed	0	2013
WP plugins	Contact Form 7	53	2105
WP plugins	Yoast SEO	196	1776
WP plugins	WooCommerce	159	1491
WP themes	Astra	57	203
WP themes	Hello Elementor	9	87
WP themes	OceanWP	31	83

Table 8.6: Three most popular software components per category.

editor first developed and released open-source by Frederico Caldeira Knabben in 2003 [110]. In January 2008, he released the first version of CKFinder [37], "the advanced file manager for FCKEditor" [111]. FCKEditor has been assigned eight CVEs, among which `CVE-2006-2529`, affecting all versions until 2.3 Beta, allows an attacker to upload files of any type. CKFinder has been assigned two CVEs, among which `CVE-2019-15862`, affecting all versions until 2.6.2.1, allows an attacker to upload files of any type. In 2009, the author renames FCKEditor to CKEditor, releasing for the first time CKEditor 3 [112] and founding CKSource Holding LTD. The development of FCKEditor was discontinued. Later, in 2015, right before the release of CKEditor 4.5, the plugin allegedly counted 15 million total downloads [113]. CKEditor has been assigned `CVE-2015-9349` for a Cross-Site Scripting (XSS) vulnerability affecting all versions before 4.5.3.1. A popular exploit repository has shared the code to open a reverse shell in websites running CKEditor 4.4.7 or earlier [101].

Finally, KCFinder was developed independently by Pavel Tzonkov [235] as a replacement to CKFinder, and to be compatible with FCKEditor and CKEditor. Its source code is still available [234], although archived in 2021. KCFinder has been assigned three CVEs, two of which are due to an XSS vulnerability and allow an attacker to inject and execute scripts. Affected are versions 3.20 and earlier, i.e., all versions. Multiple exploit repositories shared the code to exploit multiple vulnerabilities, e.g., Arbitrary File Upload in version 2.2 [197], Shell Upload in version 2.53 [14].

E-Learning Madrasah. This Web application was developed by the Indonesian Government as a response against the stop of all educational activities during the Covid-19 pandemic [222, 260]. Educational institutions (e.g., high schools) were equipped with an online platform ("E-learning Madrasah") allowing all remote teaching activities. This platform comes with the vulnerable component CKFinder installed, whose exploit code is publicly available [69].

Senayan Library Management System (SLiMS). This is an open-source web

framework for library management developed in Jakarta. Its popularity might be higher in Indonesia, as all websites mounting this framework have a `.id` country code. Moreover, the manual analysis showed that most of these websites were websites of educational institutions. SLiMS 7 and SLiMS 9 have been found vulnerable of multiple XSS, receiving two and three CVEs respectively, whose exploits are published in popular exploit databases [105, 166].

FormCraft, Webform. FormCraft is a WordPress plugin offering form building functionalities [159]. Webform is a form builder plugin built for Drupal [252]. FormCraft versions below 1.2.6 and below 3.6 have been assigned two CVEs for two XSS vulnerabilities, and a popular exploit repository published the code targeting FormCraft version 2.0 leading to Shell Upload [106]. Conversely, Webform was found vulnerable to multiple vulnerabilities, including an XSS introduced by the inclusion of the vulnerable CKEditor library [190].

8.2.2.2 URL Path Indicators

Below is the list of indicators of compromise, where the URL path segments give out the presence of a possibly vulnerable component.

- SLiMS: keywords `__statics`, `gudangsoal` or `repository` in the URL path.
- CkFinder: URL path, param, query or fragment contain the keywords `ckfinder` or `ckimage` or `kcfinder` or `ckeditor` or `fckeditor` in the URL path.
- Formcraft: keyword `formcraft` in the URL path.
- WebForm: keyword `webform` in the URL path.
- SuperForms: keyword `super-forms` in the URL path.
- Formidable: keyword `formidable` in the URL path.

8.2.3 Notification email

I am a security researcher at `Institution Name` in `Country`. As part of an academic research project, we discovered that `N` of your domains (`domain1.com`, `domain2.com`, `domain3.com` among them) are used to host and distribute `M` clickbait PDF files. These files embed links leading visitors to malicious web pages delivering phishing attacks, malware, or online scams. Victims discover these clickbait PDFs with search engines such as Google and Bing, leveraging the reputation of your domains.

We do not know how exactly the attackers manage to upload these files in your domains and we believe that your domains may have a vulnerable or misconfigured component that enables unrestricted file uploads. Here is an example of three relative URLs to clickbait PDFs hosted by the above domains:

```
domain1:
/path/to/file/1.pdf
/path/to/file/2.pdf
```

```
/path/to/file/3.pdf
domain2.com:
/another/path/to/file/4.pdf
/another/path/to/file/5.pdf
/another/path/to/file/6.pdf
domain3.com:
/yet/another/path/7.pdf
/yet/another/path/8.pdf
/yet/another/path/9.pdf
```

We attach a CSV listing the clickbait PDFs relative paths per domain. Please note that the list we provided might not be exhaustive as attackers may have uploaded new files after this notification.

MITIGATIONS: As a first step, we encourage you to immediately remove these PDFs from your domains to hamper the effectiveness of the phishing campaign. However, we recommend a security review of your websites, looking for outdated, unpatched, vulnerable, or misconfigured software components to prevent attackers from uploading new files.

As part of our study, we will monitor the `N` domains to verify if they still serve such PDFs. You can opt out of this study by contacting us at `author email`. The details in this email should be sufficient for you to mitigate the problem, nonetheless, feel free to contact us at the same address should you have any question or feedback.

DISCLAIMER: This message is part of an academic research project. Researchers did not (and will not) attempt to reproduce the attack. We are not trying to sell any product or service, and we are not trying to obtain any bounty.

8.3 Appendix to Chapter 6

8.3.1 Detailed Sampling Procedure

Compiling a List of Companies We can create a list of hosting provider organizations following different strategies. First, we could use corporate datasets (see [267]), such as [173, 40, 47], which provide list of companies following user-specified criteria (e.g., business sector, geographical location, size). However, we discarded this option, as they are prohibitively expensive. Alternatively, we considered identifying organizations via their IPs or Autonomous Systems using website lists (e.g., [192, 122]). However, as these lists rank sites by popularity, they are biased towards global hosting providers, leading us to discard this approach.

Accordingly, we manually compile a list of hosting providers by searching for names in a popular online community (Reddit) discussing topics such as web hosting, development, and hosting providers. We identified four relevant and active subreddits: two specific to hosting services (r/VPS, r/agency) and two discussing web hosting more broadly (r/webhosting, r/Hosting), identified through keyword matching[3]. Then, we collected 50 posts per subreddit using the following criteria. First, we applied inclusion criteria, e.g.,

[3]We inspected but discarded r/HostingHostel and r/webhosting services as inactive or irrelevant.

posts *(i)* asking for hosting suggestions or discussing past experiences, or *(ii)* mentions of hosting companies, either positive or negative. Second, we applied exclusion criteria, such as posts *(iii)* tagged as "rants", "bad experiences", or containing reviews; *(iv)* solely discussing out-of-scope services (e.g., colocation services, companies offering *only* domain registration, email or game servers, or web hosting software without hosting); *(v)* involving technical troubleshooting (e.g., with domain records or DNS); and finally, *(vi)* containing referral links or advertisements. We collected the text strings of qualifying posts and their associated comments mentioning hosting companies, resulting in 150 posts and a preliminary list of 197 companies.

Collecting Service Data We manually visited each company's website, removing those offering out-of-scope services, when this was not evident from the post, or if they were defunct, unreachable, or had been acquired, and collected information on all their hosting services in a top-down content analysis of the advertisements. Next, we organized these services into a structured map, noting the type of service and whether it was managed or unmanaged. This map serves as the basis for our sampling process, as we anticipate that perspectives on VN remediation will vary across different segments. To capture this variation, we aim to interview representatives from organizations operating in diverse areas of the hosting landscape. After this review, we finalized a list of 175 companies offering a total of 733 distinct services, spanning ten categories of hosting services: shared hosting, dedicated hosting, VPS, reseller hosting, website builder, web agency, cloud hosting, cloud server, cloud VPS, Virtual Dedicated Server (VDS).

Clustering for Sampling Using the compiled service data, we aimed to understand the distribution of hosting services across the market, obtaining groups of companies with similar offerings. We encoded each company's service offerings as a one-hot-encoded vector, representing the availability of specific hosting services and their management details using binary true/false values. Missing information was treated as false. Next, we applied the k-modes clustering algorithm to group companies with similar offerings. Given the sparsity of the data, clustering results were initially suboptimal. To identify an appropriate number of clusters, we tested values of k between 10 (the number of observed services) and 30 (selected empirically). Larger values of k produced overly fine-grained clusters with minimal differences between them. We ultimately selected $k = 13$, balancing interpretability with cluster granularity.

8.3.2 Interview Guide

In the following, we present the interview questions used to conduct the semi-structured interviews.

Introductory Questions

The company and interviewee's role in it.

- Please describe in as much detail as possible your role within your company.

- How is the provision of hosting services embedded into the general business strategy of your company?

Technical Infrastructure Questions

General security posture of the company and identified risks.

- What are general security considerations that you deem relevant for the operation of your company's hosting services?
  - What are possible types of attacks ('threat models') affecting hosting services provided by your company?
  - What security-related risks are there for hosting services?
- What does your company exactly do to prevent this specific kind of attack / threat?
  - Are there specific departments in the company in charge of mitigating these risks?
  - Are there in-house developed tools, or external tools?

Legal obligations for maintenance.

- If possible, can you describe the legal regulations that the company has to abide to, concerning security maintenance of the hosting services?

Organizational Aspects Questions

Security management.

- How is management/handling of security events organized in your company?
- Does your company manage security events (when they happen) internally (by its own staff), or are they outsourced to an external contractor?
  - *If handled externally:* Can you please describe which type of company is this and what services they offer?
  - How are responsibilities distributed between your company (organization) and this external service provider?
- Is there any set of (internal) instructions on what to do in such situations (a "playbook" of some kind)?
- Are there specific people (organizational units) who are in charge of responding to security events?
  - Does addressing the report involve multiple teams?
  - Does the company distinguish between those who decide on remediation and those who implement it?

- Does addressing the report involve external stakeholders?

Personal security even handling experience.

- Did you ever experience any of these cybersecurity events at your company?
  - How did your company learn about the (security) event?
  - What was done in your company once the event was recognized?
  - What was your specific role in this process?
  - How did the whole situation end?
  - What would you describe as "lessons" that your company learned from this event?
- The course of action you described: Is it part of a procedure formalized by the company, or is it something informal?
- Is this procedure regularly followed?

Vulnerability Notifications Questions

Being informed about risks and events.

- How are vulnerabilities and other S&P problems detected in the infrastructure of your company's hosting services?
- Does your company have a dedicated channel for reporting vulnerabilities and other S&P problems?
- Does your organization receive vulnerability notifications from third parties?
- How does your company become aware of security events happening / happened?
  - *If no external source of info was mentioned:* Is there the possibility that the company would be informed of a security event from an external entity (person or organization) who were not contractors?

Vulnerability notification handling.

- *If they have received VN:* What did you do with the external report?
- *If they have never received a VN:* Imagine you've just received a report from an external describing a security issue on your servers.
- Please describe which factors played/would play into your evaluation when deciding on how to act on this message.
- Which challenges do you face when evaluating external security reports on security events?
  - How did/would the medium in which you received the report affect your decisions (e.g., email vs. contact form)?

  - How did/would the identity of the sender of the report affect your decisions?
  - What were/would be measures you applied/y to verify information in the report?
  - How did/would legal requirements influence your decisions?
  - How did/would your technical skills influence your decisions?
  - How did/would authority-related questions influence your decisions?
  - How did/would trust in the message in general affect your decisions?

- Can you please walk us through the steps you take/took to remediate a security report?

- Have you ever received security reports about systems that were not on your perimeter?

Vignettes describing specific VN issues.

- Malicious or suspicious behavior: a malware (e.g. `.exe`, or MS document with macro), a malicious file (e.g., a clickbait PDF), a file containing malicious code/functionality accessible from the web (e.g., a phishing page, a script redirecting to malicious/illegal content).

- Misconfiguration: a misconfiguration in HTTP headers enforcing security mechanisms (e.g. for transferring cookies), a misconfiguration in access control for a restricted area of a web app (e.g., default password, no access control).

- Code vulnerability: an error in the implementation of a software functionality, which can be observed via active testing or by passively observing indirect indicators (e.g., direct exploit of XSS, SQLi, or observation of software version indicators).

  - How would you say your company considers these vulnerabilities?
  - *If they are considered:* Which tools does your company use to keep track of these vulnerabilities?
  - *If they are considered:* Is there any regular maintenance task to prevent them?
  - *If they are considered:* When deciding whether to address a vulnerability, would it influence your decision to know that it is a malicious file / misconfiguration / code vulnerability?

8.3.3 Codebook

HPO

- Business Aspects
  - Feel responsible for internet security

  - Ideologically driven hosting policies
- Business Considerations
- Services Offered
  - Maintenance
- Organizational Aspects
  - Company Structure
    - Communication Between Teams
    - Departments
  - The infrastructure-customer divide
- Functioning of Services
  - Take Action to Ensure SLA
  - Take Action to Safeguard Infrastructure Safety
  - Manuals
- CySec Aspects
  - Abuse of HPO Infrastructure
    - Our infrastructure does not care about phishing
  - Security Posture
    - General Proactive Measures
    - General Security Mindset
      - Keep updated
    - Incident Response
      - Proactive-Security Practices
      - Proactive-Security Software
      - Detection
      - Detection Software
      - Reactive-Security Practices
      - Reactive-Security Software
      - Incident Response
    - Organization Policies
    - Weaknesses
      - Customer Created
      - HPO Created
      - HPO-Customer Co-responsible
- External Relations

  - Externals Contacting HPO
    * Brand/ Phishing Protection Companies
    * Companies Reporting Phishing or Malware
  - Service Provider Relationship

Customer

- Business model of your customer
- Characteristics
  - Customer Structure
  - Customer Tech-Savvyness
  - Customer Types
- Choice of Product / Service
- Customer Resources
  - Own IT Team
  - Own Software
- Mindset of HPO Customers
- Reaction to Security Event
- Relationship & Communication with HPO
  - Customer Complains to HPO
  - Customer Reports VN received
- Security Posture of Customer

Events

- CySec Events
  - How
  - Detection Stage
  - Post-Incident Stage
  - Response Stage

Participant

- Active Role
- Background / Professional Biography

Regulations

- Certification
- Documents with Legal Implication
- Must-Dos
- Regulators
- Won't-Dos
- Legal Regulations for Security

Security Third Parties

- CERT
- Google SafeBrowsing
- VirusTotal

Tech Used

- Server-Side Software
- Tech for Hosting

Vulnerability Notifications

- Assessment
  - Criteria
    - Business Criteria
    - Interaction-Based Verification
    - Validate Email Style
    - Validate Email Metadata
    - Validate Informative Content
  - Outcome
- Challenges
  - Customers Don't Understand
  - Unclear Motivations of Senders
  - Receive Spam
- Communication Channels
- Handling Method

  - Evaluation by Human
  - Gut Feeling or Experience
- Handling Practices
  - Validation via Sender Communication
- HPO's VN Knowledge
- Improvements
- Receiving Entity
  - Company Receives Notification
  - Customer Receives Notification
- Sender
- VN Exposure
- VN reachability

Others

- Attack Types
- Information Sources

Bibliography

Author's Papers for this Thesis

[P1] Stivala, G. and Pellegrino, G. Deceptive previews: A study of the link preview trustworthiness in social platforms. In: *Network and Distributed System Security Symposium (NDSS)*. 2020.

[P2] Stivala, G., Abdelnabi, S., Mengascini, A., Graziano, M., Fritz, M., and Pellegrino, G. From Attachments to SEO: Click Here to Learn More about Clickbait PDFs. In: *Proceedings of the 39th Annual Computer Security Applications Conference*. 2023.

[P3] Stivala, G., De Stefano, G., Mengascini, A., Graziano, M., and Pellegrino, G. Uncovering the Role of Support Infrastructure in Clickbait PDF Campaigns. In: *IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. 2024.

[P4] Stivala, G., Mrowczynski, R., Hellenthal, M., and Pellegrino, G. Behind the Curtain: How Shared Hosting Providers Respond to Vulnerability Notifications. In: *Under submission.* 2026.

Other Papers of the Author

[S1] Beluri, M., Acharya, B., Khodayari, S., Stivala, G., Pellegrino, G., and Holz, T. Exploration of the Dynamics of Buy and Sale of Social Media Accounts. In: *Proceedings of the 2025 ACM on Internet Measurement Conference (IMC)*. 2025.

[S2] Stivala, G., Meyer, M., and Pellegrino, G. An Analysis of Malicious File Distribution in Free Hosting Providers. In: *Under submission.* 2025.

Other references

[1] @MsftSecIntel. *Operators of the malware known as SolarMarker, Jupyter …* 2021. URL: `https://x.com/MsftSecIntel/status/1403461397283950597` (visited on 02/28/2025).

[2] Abdelnabi, S., Krombholz, K., and Fritz, M. Visualphishnet: Zero-day phishing website detection by visual similarity. In: *ACM Conference on Computer and Communications Security*. 2020.

[3] Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. A comparison of machine learning techniques for phishing detection. In: *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. 2007, 60–69.

[4] Acharya, B., Lazzaro, D., López-Morales, E., Oest, A., Saad, M., Cinà, A. E., Schönherr, L., and Holz, T. The imitation game: Exploring brand impersonation attacks on social media platforms. In: *33rd USENIX Security Symposium (USENIX Security 24)*. 2024.

[5] Alcantara, J. M. *Fake CAPTCHAs, Malicious PDFs, SEO Traps Leveraged for User Manual Searches*. 2025. URL: https://www.netskope.com/blog/fake-captchas-malicious-pdfs-seo-traps-leveraged-for-user-manual-searches (visited on 04/09/2025).

[6] Alomar, N., Wijesekera, P., Qiu, E., and Egelman, S. " You've got your nice list of bugs, now what?" vulnerability discovery and management processes in the wild. In: *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. 2020.

[7] APWG. *The Global Phishing Survey: Trends and Domain Name Use in 2016*. https://docs.apwg.org//reports/APWG_Global_Phishing_Report_2015-2016.pdf?_ga=2.220159775.701668710.1639446174-1616809074.1639170127. 2016. (Visited on 04/16/2025).

[8] ARIN. *American Registry for Internet Numbers*. https://www.arin.net/. (Visited on 06/03/2025).

[9] Armstrong, M. *Referral Traffic - Google or Facebook?* 2017. URL: https://www.statista.com/chart/9555/referral-traffic---google-or-facebook/ (visited on 09/10/2019).

[10] Ars Technica. *Armed with iOS 0days, hackers indiscriminately infected iPhones for two years*. 2019. URL: https://arstechnica.com/information-technology/2019/08/armed-with-ios-0days-hackers-indiscriminately-infected-iphones-for-two-years/ (visited on 09/06/2019).

[11] ATT&CK, M. *Phishing*. https://attack.mitre.org/versions/v16/techniques/T1566/. (Visited on 03/10/2025).

[12] ATT&CK, M. *Phishing: Spearphishing Attachment*. https://attack.mitre.org/techniques/T1566/001/. (Visited on 04/29/2025).

[13] Backes, M., Rieck, K., Skoruppa, M., Stock, B., and Yamaguchi, F. Efficient and flexible discovery of php application vulnerabilities. In: *2017 IEEE european symposium on security and privacy (EuroS&P)*. IEEE. 2017.

[14] Black.Hack3r. *KCFinder 2.53 Shell Upload*. https://packetstormsecurity.com/files/125836/KCFinder-2.53-Shell-Upload.html. 2014. (Visited on 01/31/2023).

[15] Blythe, M., Petrie, H., and Clark, J. A. F for fake: four studies on how we fall for phish. In: *SIGCHI Conference on Human Factors in Computing Systems*. 2011.

[16] Bondar, T., Assal, H., and Abdou, A. Why do internet devices remain vulnerable? a survey with system administrators. In: *Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb 2023). NDSS*. 2023.

[17] Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S., and Fisher, B. Towards understanding IT security professionals and their tools. In: *Proceedings of the 3rd symposium on Usable privacy and security*. 2007.

[18] Bourdieu, P. *Outline of a Theory of Practice.* Cambridge University Press, 1977.

[19] Breuer, A., Eilat, R., and Weinsberg, U. Friend or faux: Graph-based early detection of fake accounts on social networks. In: *Proceedings of the web conference 2020*. 2020, 1287–1297.

[20] Brown, G., Howe, T., Ihbe, M., Prakash, A., and Borders, K. Social networks and context-aware spam. In: *Proceedings of the 2008 ACM conference on Computer supported cooperative work*. 2008, 403–412.

[21] Burda, P. *Let the weakest link fail, but gracefully: understanding tailored phishing and measures against it.* Eindhoven University of Technology, 2024.

[22] Burda, P., Allodi, L., and Zannone, N. Cognition in social engineering empirical research: a systematic literature review. *ACM Transactions on Computer-Human Interaction* (2024).

[23] Canali, D., Balzarotti, D., and Francillon, A. The role of web hosting providers in detecting compromised websites. In: *Proceedings of the 22nd international conference on World Wide Web*. 2013.

[24] Canali, D., Cova, M., Vigna, G., and Kruegel, C. Prophiler: a fast filter for the large-scale detection of malicious web pages. In: *Proceedings of the 20th International Conference on World Wide Web*. WWW '11. Association for Computing Machinery, 2011.

[25] Cao, Q., Yang, X., Yu, J., and Palow, C. Uncovering large groups of active malicious accounts in online social networks. In: *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 2014, 477–488.

[26] Carlini, N., Jagielski, M., Choquette-Choo, C. A., Paleka, D., Pearce, W., Anderson, H., Terzis, A., Thomas, K., and Tramèr, F. Poisoning web-scale training datasets is practical. In: *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2024, 407–425.

[27] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[28] Catakoglu, O., Balduzzi, M., and Balzarotti, D. Automatic extraction of indicators of compromise for web applications. In: *Proceedings of the 25th international conference on world wide web*. 2016.

[29] Çetin, O., Ganán, C., Altena, L., Kasama, T., Inoue, D., Tamiya, K., Tie, Y., Yoshioka, K., and Van Eeten, M. Cleaning Up the Internet of Evil Things: Real-World Evidence on ISP and Consumer Efforts to Remove Mirai. In: *NDSS*. 2019.

[30] Cetin, O., Ganan, C., Korczynski, M., and Van Eeten, M. Make notifications great again: learning how to notify in the age of large-scale vulnerability scanning. In: *Workshop on the Economics of Information Security (WEIS)*. Vol. 23. 2017.

[31] Cetin, O., Hanif Jhaveri, M., Gañán, C., Eeten, M. van, and Moore, T. Understanding the role of sender reputation in abuse reporting and cleanup. *Journal of Cybersecurity* (2016).

[32] Charmaz, K. C. *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. Sage, Thousand Oaks, Calif., 2006.

[33] Chen, Y., Wang, S., She, D., and Jana, S. On training robust PDF malware classifiers. In: *29th USENIX Security Symposium*. 2020.

[34] Cialdini, R. B. *Influence: The psychology of persuasion*. Collins New York, 2007.

[35] Cichonski, P., Millar, T., Grance, T., Scarfone, K., et al. Computer security incident handling guide. *NIST Special Publication* (2012).

[36] CKSource Holding LTD. *CKEditor 5*. `https://ckeditor.com/`. 2015. (Visited on 01/31/2023).

[37] CKSource Holding LTD. *CKFinder*. `https://ckeditor.com/ckfinder/`. 2023. (Visited on 01/31/2023).

[38] Continella, A., Polino, M., Pogliani, M., and Zanero, S. There's a hole in that bucket! a large-scale analysis of misconfigured S3 buckets. In: *Proceedings of the 34th Annual Computer Security Applications Conference*.

[39] Crocker, D. *Mailbox Names for Common Services, Roles and Functions*. 1997. URL: `https://www.rfc-editor.org/rfc/rfc2142` (visited on 07/07/2025).

[40] Crunchbase. *Crunchbase: Discover innovative companies and the people behind them*. `http://crunchbase.com`. (Visited on 12/26/2024).

[41] Cybersecurity & Infrastructure Security Agency. *Federal Incident Notification Guidelines: Submitting Incident Notifications*. 2017. URL: `https://www.cisa.gov/federal-incident-notification-guidelines#submitting-incident-notifcations` (visited on 07/07/2025).

[42] Data, M. M. *MXMAILDATA: Email Threat Data*.

[43] De Silva, R., Nabeel, M., Elvitigala, C., Khalil, I., Yu, T., and Keppitiyagama, C. Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs. In: *30th USENIX security symposium (USENIX security 21)*. 2021.

[44] Dhamija, R., Tygar, J. D., and Hearst, M. Why phishing works. In: *SIGCHI Conference on Human Factors in Computing Systems*. 2006.

[45] Dietrich, C., Krombholz, K., Borgolte, K., and Fiebig, T. Investigating system operators' perspective on security misconfigurations. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018.

[46] Downs, J. S., Holbrook, M., and Cranor, L. F. Behavioral response to phishing risk. In: *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM. 2007.

[47] Dun & Bradstreet. *Dun & Bradstreet: Intelligent Data for Business Performance*. https://www.dnb.com/. (Visited on 12/26/2024).

[48] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., et al. The matter of heartbleed. In: *Proceedings of the 2014 conference on internet measurement conference*. 2014.

[49] Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: fast internet-wide scanning and its security applications. In: *22nd USENIX Security Symposium (USENIX Security 13)*. 2013.

[50] ECMA Intenational. *ECMA-376 Office Open XML file formats 5th edition, December 2021*. https://ecma-international.org/publications-and-standards/standards/ecma-376/. (Visited on 04/29/2025).

[51] Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. Compa: Detecting compromised accounts on social networks. In: *NDSS*. 2013.

[52] Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. Towards detecting compromised accounts on social networks. *IEEE Transactions on Dependable and Secure Computing* (2015).

[53] emerald101. *Clickbait PDFs Dataset*. https://www.kaggle.com/datasets/emerald101/from-attachments-to-seo. (Visited on 02/21/2025).

[54] emerald101 and De Stefano, G. *Support Infrastructure Code*. https://github.com/emerald1010/hosts-supporting-clickbait-PDFs. (Visited on 02/21/2025).

[55] emerald101 and S-Abdelnabi. *Clickbait PDFs Code*. https://github.com/emerald1010/from_attachments_to_seo. (Visited on 02/21/2025).

[56] ENISA. *Coordinated Vulnerability Disclosure Policies in the EU*. 2022. URL: https://www.enisa.europa.eu/publications/coordinated-vulnerability-disclosure-policies-in-the-eu (visited on 07/07/2025).

[57] ENISA, CIRAS. *Cybersecurity Incident Reporting and Analysis System*. URL: https://ciras.enisa.europa.eu/ (visited on 07/07/2025).

[58] Esparza, J. M. *peepdf*. https://github.com/jesparza/peepdf. 2016. (Visited on 06/01/2023).

[59] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. 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)*. 1996.

[60] Ethembabaoglu, A., Wegberg, R. van, Zhauniarovich, Y., and Eeten, M. van. The Unpatchables: Why Municipalities Persist in Running Vulnerable Hosts. In: *33rd USENIX Security Symposium (USENIX Security 24)*. 2024.

[61] Facebook Inc. *I got a message from Facebook saying a file I tried to share has a virus.* URL: https://www.facebook.com/help/223268604538225 (visited on 09/04/2019).

[62] Facebook Inc. *The Open Graph protocol*. URL: https://ogp.me/ (visited on 09/01/2019).

[63] Facebook Inc. *What is Facebook doing to protect me from spam?* URL: https://www.facebook.com/help/637109102992723 (visited on 09/04/2019).

[64] fanboy, MonztA, Famlam, Khrin. *EasyList*. (01/22/2022).

[65] Federal Bureau of Investigation. *Internet Crime Report 2023*. 2023. (Visited on 02/26/2025).

[66] Federal Bureau of Investigation. *Internet Crime Report 2023*. https://www.ic3.gov/AnnualReport/Reports/2023_IC3Report.pdf. 2023. (Visited on 01/28/2025).

[67] Fette, I., Sadeh, N., and Tomasic, A. Learning to detect phishing emails. In: *16th international conference on World Wide Web*. 2007.

[68] Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., and Zhao, B. Y. Detecting and characterizing social spam campaigns. In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 2010.

[69] Gh05t666nero. *E-Learning Madrasah 2.0 - Arbitary File Upload*. https://cxsecurity.com/issue/WLB-2020080121. (Visited on 06/01/2023).

[70] Glaser, B. G. and Strauss, A. L. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, 1967.

[71] GmbH, A. *ATLAS.ti | The #1 Software for Qualitative Data Analysis. Online*. https://atlasti.com/. (Visited on 12/21/2024).

[72] Goethem, T. v., Chen, P., Nikiforakis, N., Desmet, L., and Joosen, W. Large-scale security analysis of the web: Challenges and findings. In: *International Conference on Trust and Trustworthy Computing*. 2014.

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

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

[75] Google Cloud. *What is Object Storage?* https://cloud.google.com/learn/what-is-object-storage. (Visited on 06/02/2023).

[76] Greene, K. K., Steves, M., Theofanos, M., Kostick, J., et al. User context: an explanatory variable in phishing susceptibility. In: *in Proc. 2018 Workshop Usable Security*. 2018.

[77] Grier, C., Ballard, L., Caballero, J., Chachra, N., Dietrich, C. J., Levchenko, K., Mavrommatis, P., McCoy, D., Nappa, A., Pitsillidis, A., et al. Manufacturing compromise: the emergence of exploit-as-a-service. In: *Proceedings of the 2012 ACM conference on Computer and communications security*.

[78] Grier, C., Thomas, K., Paxson, V., and Zhang, M. @spam: the underground on 140 characters or less. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security*. 2010.

[79] Guérin, J. and Boots, B. Improving Image Clustering With Multiple Pretrained CNN Feature Extractors. In: *British Machine Vision Conference BMVC*. 2018.

[80] Guérin, J., Gibaru, O., Thiery, S., and Nyiri, E. CNN features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700* (2017).

[81] Gupta, S., Khattar, A., Gogia, A., Kumaraguru, P., and Chakraborty, T. Collective classification of spam campaigners on Twitter: A hierarchical meta-path based approach. In: *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee. 2018.

[82] Hao, S., Kantchelian, A., Miller, B., Paxson, V., and Feamster, N. PREDATOR: proactive recognition and elimination of domain abuse at time-of-registration. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016.

[83] Hellenthal, M., Gotsche, L., Mrowczynski, R., Kugel, S., Schilling, M., and Stock, B. The (Un) usual Suspects–Studying Reasons for Lacking Updates in WordPress (2025).

[84] Hennig, A., Neusser, F., Pawelek, A. A., Herrmann, D., and Mayer, P. Standing out among the daily spam: How to catch website owners' attention by means of vulnerability notifications. In: *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 2022.

[85] Hennig, A., Vuong, N. T. T., and Mayer, P. Vision: What the hack is going on? A first look at how website owners became aware that their website was hacked. In: *Proceedings of the 2023 European Symposium on Usable Security*. 2023.

[86] Ho, G., Cidon, A., Gavish, L., Schweighauser, M., Paxson, V., Savage, S., Voelker, G. M., and Wagner, D. Detecting and characterizing lateral phishing at scale. In: *28th {USENIX} Security Symposium*. 2019.

[87] Ho, G., Sharma, A., Javed, M., Paxson, V., and Wagner, D. Detecting credential spearphishing in enterprise settings. In: *26th USENIX security symposium (USENIX security 17)*. 2017, 469–485.

[88] Hong, J. The current state of phishing attacks (2012).

[89] Huaman, N., Skarczinski, B. von, Stransky, C., Wermke, D., Acar, Y., Dreißigacker, A., and Fahl, S. A {Large-Scale} interview study on information security in and attacks against small and medium-sized enterprises. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

[90] Huaman, N., Suray, J., Klemmer, J. H., Fourné, M., Klivan, S., Trummová, I., Acar, Y., and Fahl, S. "You have to read 50 different {RFCs} that contradict each other": An Interview Study on the Experiences of Implementing Cryptographic Standards. In: *33rd USENIX Security Symposium (USENIX Security 24)*. 2024.

[91] InQuest. *yara-rules*.

[92] Internet Archive. *The Internet Archive*. `https://archive.org`. (Visited on 06/07/2022).

[93] Invernizzi, L., Comparetti, P. M., Benvenuti, S., Kruegel, C., Cova, M., and Vigna, G. Evilseed: A guided approach to finding malicious web pages. In: *2012 IEEE symposium on Security and Privacy*.

[94] Irani, D., Webb, S., Giffin, J., and Pu, C. Evolutionary study of phishing. In: *2008 eCrime Researchers Summit*. 2008.

[95] Jenkins, A. D., Liu, L., Wolters, M. K., and Vaniea, K. Not as easy as just update: Survey of System Administrators and Patching Behaviours. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2024.

[96] John, J. P., Yu, F., Xie, Y., Abadi, M., and Krishnamurthy, A. Searching the Searchers with SearchAudit. In: *USENIX Security Symposium*. 2010.

[97] John, J. P., Yu, F., Xie, Y., Krishnamurthy, A., and Abadi, M. deSEO: Combating Search-Result Poisoning. In: *20th USENIX Security Symposium*. 2011.

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

[99] Johnson, K. *Google Safe Browsing can differ between desktop and mobile. Why?* 2019. URL: https://www.wandera.com/mobile-security/google-safe-browsing/ (visited on 09/10/2019).

[100] Kasturi, R. P., Fuller, J., Sun, Y., Chabklo, O., Rodriguez, A., Park, J., and Saltaformaggio, B. Mistrust Plugins You Must: A Large-Scale Study Of Malicious Plugins In WordPress Marketplaces. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

[101] KedAns-Dz. *Ckeditor 4.4.7 Shell Upload / Cross Site Scripting*. https://packetstormsecurity.com/files/130807/Ckeditor-4.4.7-Shell-Upload-Cross-Site-Scripting.html. 2015. (Visited on 01/25/2023).

[102] Kenneally, E. and Dittrich, D. The menlo report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102* (2012).

[103] Kharraz, A., Robertson, W., and Kirda, E. Surveylance: Automatically detecting online survey scams. In: *2018 IEEE Symposium on Security and Privacy (SP)*.

[104] Khodayari, S. and Pellegrino, G. It's (dom) clobbering time: Attack techniques, prevalence, and defenses. In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2023.

[105] KingSkrupellos. *Slims CMS Senayan Library Management System 7.0 Shell Upload*. https://packetstormsecurity.com/files/151676/Slims-CMS-Senayan-Library-Management-System-7.0-Shell-Upload.html. 2019. (Visited on 01/25/2023).

[106] KingSkrupellos. *WordPress FormCraft 2.0 CSRF / Shell Upload*. https://packetstormsecurity.com/files/152122/WordPress-FormCraft-2.0-CSRF-Shell-Upload.html. 2019. (Visited on 01/25/2023).

[107] Kleining, G. Umriss zu einer Methodologie qualitativer Sozialforschung. *Kölner Zeitschrift für Soziologie und Sozialpsychologie* 34 (1982), 224–253.

[108] Klivan, S., Höltervennhoff, S., Panskus, R., Marky, K., and Fahl, S. Everyone for Themselves? A Qualitative Study about Individual Security Setups of Open Source Software Contributors. In: *2024 IEEE Symposium on Security and Privacy (SP)*. 2024.

[109] Klostermeyer, P., Amft, S., Höltervennhoff, S., Krause, A., Busch, N., and Fahl, S. Skipping the Security Side Quests: A Qualitative Study on Security Practices and Challenges in Game Development. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 2024.

[110] Knabben, F. *FCKEditor (retired)*. `https://sourceforge.net/projects/fckeditor/`. 2003. (Visited on 01/31/2023).

[111] Knabben, F. *CKFinder 1.1 released*. `https://ckeditor.com/blog/CKFinder_1.1_released/`. 2008. (Visited on 01/31/2023).

[112] Knabben, F. *CKEditor 3.0 is here!* `https://ckeditor.com/blog/CKEditor-3.0-here/`. 2009. (Visited on 01/31/2023).

[113] Knabben, F. *WOW! Over 15 Million Downloads!* `https://ckeditor.com/blog/WOW-Over-15-Million-Downloads/`. 2015. (Visited on 01/31/2023).

[114] Kotzias, P., Bilge, L., and Caballero, J. Measuring {PUP} Prevalence and {PUP} Distribution through {Pay-Per-Install} Services. In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, 739–756.

[115] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (2012).

[116] Krombholz, K., Hobel, H., Huber, M., and Weippl, E. Advanced social engineering attacks. *Journal of Information Security and applications* (2015).

[117] Kührer, M., Hupperich, T., Rossow, C., and Holz, T. Exit from Hell? Reducing the Impact of {Amplification}{DDoS} Attacks. In: *23rd USENIX security symposium (USENIX security 14)*. 2014.

[118] Laskov, P. and Šrndić, N. Static detection of malicious JavaScript-bearing PDF documents. In: *Proceedings of the 27th annual computer security applications conference*. 2011, 373–382.

[119] Lazar, J., Feng, J. H., and Hochheiser, H. *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

[120] Le Blond, S., Gilbert, C., Upadhyay, U., Gomez-Rodriguez, M., and Choffnes, D. R. A Broad View of the Ecosystem of Socially Engineered Exploit Documents. In: *NDSS*. 2017.

[121] Le Blond, S., Uritesc, A., Gilbert, C., Chua, Z. L., Saxena, P., and Kirda, E. A look at targeted attacks through the lense of an NGO. In: *23rd USENIX Security Symposium*. 2014.

[122] Le Pochat, V., Van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., and Joosen, W. TRANCO: A Research-Oriented Top Sites Ranking Hardened Against Manipulation (2018).

[123] Lee, S. and Kim, J. Warningbird: Detecting suspicious urls in twitter stream. In: *NDSS*. 2012.

[124] Lekies, S., Stock, B., and Johns, M. 25 million flows later: large-scale detection of DOM-based XSS. In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013.

[125] Leontiadis, N., Moore, T., and Christin, N. Measuring and analyzing Search-Redirection attacks in the illicit online prescription drug trade. In: *20th USENIX Security Symposium (USENIX Security 11)*. 2011.

[126] Leontiadis, N., Moore, T., and Christin, N. A nearly four-year longitudinal study of search-engine poisoning. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014.

[127] Li, F., Durumeric, Z., Czyz, J., Karami, M., Bailey, M., McCoy, D., Savage, S., and Paxson, V. You've got vulnerability: Exploring effective vulnerability notifications. In: *25th USENIX Security Symposium*. 2016.

[128] Li, F., Rogers, L., Mathur, A., Malkin, N., and Chetty, M. Keepers of the machines: Examining how system administrators manage software updates for multiple machines. In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019.

[129] Li, Y., Huang, C., Deng, S., Lock, M. L., Cao, T., Oo, N., Lim, H. W., and Hooi, B. {KnowPhish}: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing {Reference-Based} Phishing Detection. In: *33rd USENIX Security Symposium (USENIX Security 24)*. 2024.

[130] Li, Z., Alrwais, S., Wang, X., and Alowaisheq, E. Hunting the red fox online: Understanding and detection of mass redirect-script injections. In: *2014 IEEE Symposium on Security and Privacy*. 2014.

[131] Li, Z., Alrwais, S., Xie, Y., Yu, F., and Wang, X. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In: *2013 IEEE Symposium on Security and Privacy*.

[132] Liao, X., Alrwais, S., Yuan, K., Xing, L., Wang, X., Hao, S., and Beyah, R. Lurking malice in the cloud: Understanding and detecting cloud repository as a malicious service. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016.

[133] Liao, X., Liu, C., McCoy, D., Shi, E., Hao, S., and Beyah, R. Characterizing long-tail SEO spam on cloud web hosting services. In: *Proceedings of the 25th International Conference on World Wide Web*. 2016.

[134] Lin, Y., Liu, R., Divakaran, D. M., Ng, J. Y., Chan, Q. Z., Lu, Y., Si, Y., Zhang, F., and Dong, J. S. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In: *30th USENIX Security Symposium*. 2021.

[135] Liu, J., Pun, P., Vadrevu, P., and Perdisci, R. Understanding, measuring, and detecting modern technical support scams. In: *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. 2023.

[136] Liu, R., Lin, Y., Yang, X., Ng, S. H., Divakaran, D. M., and Dong, J. S. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

[137] Liu, R., Lin, Y., Zhang, Y., Lee, P. H., and Dong, J. S. Knowledge expansion and counterfactual interaction for {Reference-Based} phishing detection. In: *32nd USENIX Security Symposium (USENIX Security 23)*. 2023.

[138] Lu, L., Perdisci, R., and Lee, W. Surf: detecting and measuring search poisoning. In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011, 467–476.

[139] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009.

[140] Maaß, M., Clement, M.-P., and Hollick, M. Snail Mail Beats Email Any Day:On Effective Operator Security Notifications in the Internet. In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. ARES '21. Association for Computing Machinery, 2021. URL: https://doi.org/10.1145/3465481.3465743.

[141] Maass, M., Stöver, A., Pridöhl, H., Bretthauer, S., Herrmann, D., Hollick, M., and Spiecker, I. Effective notification campaigns on the web: A matter of trust, framing, and support. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

[142] Maroofi, S., Korczyński, M., Hesselman, C., Ampeau, B., and Duda, A. COMAR: classification of compromised versus maliciously registered domains. In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2020.

[143] Martius, F. and Tiefenau, C. What does this update do to my systems?–an analysis of the importance of update-related information to system administrators. In: *Workshop on Security Information Workers, WSIW*. 2020.

[144] Mavrommatis, N. P. P. and Monrose, M. All your iframes point to us. In: *USENIX security symposium*. 2008.

[145] Mayring, P. Qualitative Content Analysis. *Forum: Qualitative Sozialforschung / Forum: Qualitative Social Research* 1, 2 (2000), Art. 20.

[146] McCullagh, P. and Nelder, J. A. Generalized linear models 2nd edition chapman and hall (1989).

[147] Microsoft Defender Security Research Team. *Phishers unleash simple but effective social engineering techniques using PDF attachments*. https://www.microsoft.com/security/blog/2017/01/26/phishers-unleash-simple-but-effective-social-engineering-techniques-using-pdf-attachments/. 2017.

[148] Milletary, J. and Center, C. C. Technical trends in phishing attacks. *Retrieved December* 2007 (2005).

[149] Mink, J., Luo, L., Barbosa, N. M., Figueira, O., Wang, Y., and Wang, G. {DeepPhish}: Understanding user trust towards artificially generated profiles in online social networks. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022, 1669–1686.

[150] Mislove, A., Post, A., Druschel, P., and Gummadi, P. K. Ostra: Leveraging Trust to Thwart Unwanted Communication. In: *Nsdi*. 2008.

[151] MITRE. *CWE-451: User Interface (UI) Misrepresentation of Critical Information*. `https://cwe.mitre.org/data/definitions/451.html`. (Visited on 03/14/2025).

[152] Moore, T. and Clayton, R. Evil searching: Compromise and recompromise of internet hosts for phishing. In: *International Conference on Financial Cryptography and Data Security*. 2009.

[153] Moura, G. C. M., Daniels, T., Bosteels, M., Castro, S., Müller, M., Wabeke, T., Hout, T. van den, Korczyński, M., and Smaragdakis, G. Characterizing and Mitigating Phishing Attacks at ccTLD Scale. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. CCS '24. Association for Computing Machinery, 2024.

[154] Nahapetyan, A., Prasad, S., Childs, K., Oest, A., Ladwig, Y., Kapravelos, A., and Reaves, B. On sms phishing tactics and infrastructure. In: *2024 IEEE Symposium on Security and Privacy (SP)*. 2024.

[155] Nappa, A., Rafique, M. Z., and Caballero, J. Driving in the cloud: An analysis of drive-by download operations and abuse reporting. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. 2013.

[156] Nappa, A., Xu, Z., Rafique, M. Z., Caballero, J., and Gu, G. Cyberprobe: Towards internet-scale active detection of malicious servers. In: *NDSS 2014*.

[157] National Institute of Standards and Technology. *Common Platform Enumeration (CPE)*. `https://csrc.nist.gov/projects/security-content-automation-protocol/specifications/cpe`. 2022. (Visited on 06/01/2023).

[158] National Institute of Standards and Technology. *National Vulnerability Database*. `https://nvd.nist.gov/developers`. 2022. (Visited on 08/20/2022).

[159] nCrafts. *FormCraft – Contact Form Builder for WordPress*. `https://wordpress.org/plugins/formcraft-form-builder/`. 2022. (Visited on 01/31/2023).

[160] Nelder, J. A. and Wedderburn, R. W. Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society* (1972).

[161] Nelms, T., Perdisci, R., Antonakakis, M., and Ahamad, M. Towards Measuring and Mitigating Social Engineering Software Download Attacks. In: *USENIX Security Symposium*. 2016.

[162] Nikiforakis, N., Joosen, W., and Johns, M. Abusing locality in shared web hosting. In: *Proceedings of the Fourth European Workshop on System Security*. 2011.

[163] Noonburg, D. and Astals, A. *Poppler, a PDF rendering library*. `https://gitlab.freedesktop.org/poppler/poppler`. 2021. (Visited on 01/17/2022).

[164] Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749* (2021).

[165] Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D. Detecting spam web pages through content analysis. In: *Proceedings of the 15th international conference on World Wide Web*. 2006.

[166] nu11secur1ty. *SLIMS 9.5.2 Cross Site Scripting Vulnerability*. https://vulners.com/zdt/1337DAY-ID-38158. 2023. (Visited on 01/25/2023).

[167] o365devx, AlexJerabek, Linda-Editor, kbrandl. *Office VBA Reference*. https://learn.microsoft.com/en-us/office/vba/api/overview/. (Visited on 04/29/2025).

[168] Oest, A., Safaei, Y., Doupé, A., Ahn, G.-J., Wardman, B., and Tyers, K. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In: *2019 IEEE Symposium on Security and Privacy (SP)*.

[169] Oest, A., Safaei, Y., Zhang, P., Wardman, B., Tyers, K., Shoshitaishvili, Y., and Doupé, A. PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists. In: *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, 379–396. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/oest-phishtime.

[170] Oest, A., Safei, Y., Doupé, A., Ahn, G.-J., Wardman, B., and Warner, G. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In: *2018 APWG Symposium on Electronic Crime Research (eCrime)*.

[171] Oest, A., Zhang, P., Wardman, B., Nunes, E., Burgis, J., Zand, A., Thomas, K., Doupé, A., and Ahn, G.-J. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In: *29th USENIX Security Symposium (USENIX Security 20)*.

[172] Open DNS. *PhishTank*. URL: https://www.phishtank.com/ (visited on 09/09/2019).

[173] OpenCorporates. *Legal-entity data you can trust*. https://opencorporates.com/. (Visited on 12/26/2024).

[174] OWASP. *2025 Top 10 Risk & Mitigations for LLMs and Gen AI Apps*. 2025. URL: https://genai.owasp.org/llm-top-10/ (visited on 07/11/2025).

[175] OWASP. *Vulnerability Disclosure Cheat Sheet*. https://cheatsheetseries.owasp.org/cheatsheets/Vulnerability_Disclosure_Cheat_Sheet.html. (Visited on 01/29/2025).

[176] Palo Alto Networks Unit 42. *2020 Phishing Trends With PDF Files*. https://unit42.paloaltonetworks.com/phishing-trends-with-pdf-files/. 2020. (Visited on 08/12/2021).

[177] Pan, Y., Ascheman, A., and Rossow, C. Loopy Hell(ow): Infinite Traffic Loops at the Application Layer. In: *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, 2024. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/pan-yepeng.

[178] Pellegrino, G., Catakoglu, O., Balzarotti, D., and Rossow, C. Uses and Abuses of Server-Side Requests. In: *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses*. 2016.

[179] Planning for Library of Congress Collections. *Microsoft Office Excel 97-2003 Binary File Format (.xls, BIFF8)*. https://www.loc.gov/preservation/digital/formats/fdd/fdd000510.shtml. (Visited on 04/29/2025).

[180] Pletinckx, S., Borgolte, K., and Fiebig, T. Out of Sight, Out of Mind: Detecting Orphaned Web Pages at Internet-Scale. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021.

[181] Porta, L. L. *Google's security efforts are falling short on mobile*. 2019. URL: https://www.brianmadden.com/opinion/Google-Safe-Browsing-differs-between-desktop-and-mobile (visited on 09/10/2019).

[182] Poteat, T. and Li, F. Who you gonna call? an empirical evaluation of website security. txt deployment. In: *Proceedings of the 21st ACM Internet Measurement Conference*. 2021.

[183] Project, Y. *YARA: The pattern matching swiss knife for malware researchers (and everyone else)*.

[184] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N. The Ghost in the Browser: Analysis of Web-based Malware. In: *First Workshop on Hot Topics in Understanding Botnets (HotBots 07)*. USENIX Association, 2007.

[185] Rashid, B. *Large Language Models (LLMs) Are Falling for Phishing Scams: What Happens When AI Gives You the Wrong URL?* 2025. URL: https://www.netcraft.com/blog/large-language-models-are-falling-for-phishing-scams (visited on 07/16/2025).

[186] Redmiles, E. M., Chachra, N., and Waismeyer, B. Examining the demand for spam: Who clicks? In: *SIGCHI Conference on Human Factors in Computing Systems*. 2018.

[187] RIPE. *RIPE NCC*. https://www.ripe.net/. (Visited on 06/03/2025).

[188] Robertson, M. *Modifying Link Previews*. 2017. URL: https://developers.facebook.com/blog/post/2017/06/27/API-Change-Log-Modifying-Link-Previews (visited on 11/12/2019).

[189] Roth, S., Barron, T., Calzavara, S., Nikiforakis, N., and Stock, B. Complex security policy? a longitudinal analysis of deployed content security policies. In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2020.

[190] Rowlands, L., Rockowitz, J., and Xjm, G. K. *Webform - Moderately critical - Cross Site Scripting - SA-CONTRIB-2021-026*. https://www.drupal.org/sa-contrib-2021-026. 2021. (Visited on 01/31/2023).

[191] Ruaro, N., Pagani, F., Ortolani, S., Kruegel, C., and Vigna, G. SYMBEXCEL: Automated analysis and understanding of malicious excel 4.0 macros. In: *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2022, 1066–1081.

[192] Ruth, K., Kumar, D., Wang, B., Valenta, L., and Durumeric, Z. Toppling top lists: Evaluating the accuracy of popular website lists. In: *Proceedings of the 22nd ACM Internet Measurement Conference*. 2022.

[193] sa7mon. *S3Scanner*. `https://github.com/sa7mon/S3Scanner`. 2022. (Visited on 06/01/2023).

[194] Saha, A., Blasco, J., and Lindorfer, M. Exploring the Malicious Document Threat Landscape: Towards a Systematic Approach to Detection and Analysis. In: *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2024, 533–544.

[195] Saha Roy, S., Karanjit, U., and Nilizadeh, S. Phishing in the Free Waters: A Study of Phishing Attacks Created using Free Website Building Services. In: *Proceedings of the 2023 ACM on Internet Measurement Conference*. IMC '23. Association for Computing Machinery, 2023.

[196] Saldaña, J. Coding and Analysis Strategies. In: *The Oxford Handbook of Qualitative Research*. Ed. by Leavy, P. Oxford University Press, 2014, 581–605.

[197] saudi0hacker. *KCFinder 2.2 - Arbitrary File Upload*. `https://www.exploit-db.com/exploits/15254`. 2010. (Visited on 01/31/2023).

[198] Schmüser, J., Sri Ramulu, H., Wöhler, N., Stransky, C., Bensmann, F., Dimitrov, D., Schellhammer, S., Wermke, D., Dietze, S., Acar, Y., et al. Analyzing Security and Privacy Advice During the 2022 Russian Invasion of Ukraine on Twitter. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 2024.

[199] Schreier, M. *Qualitative Content Analysis in Practice*. Sage, Los Angeles et al., 2012.

[200] Schreier, M. Qualitative Content Analysis. In: *The SAGE Handbook of Qualitative Data Analysis*. Ed. by Flick, U. Sage, Los Angeles et al., 2014, 170–183.

[201] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[202] Senayan Developers Community. *Senayan Library Management System*. `https://github.com/slims`. 2022. (Visited on 08/20/2022).

[203] Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., and Downs, J. 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*. 2010.

[204] SimilarWeb LTD. *SimilarWeb*. `https://www.similarweb.com/`. 2023. (Visited on 06/01/2023).

[205] Simoiu, C., Zand, A., Thomas, K., and Bursztein, E. Who is targeted by email-based phishing and malware? measuring factors that differentiate risk. In: *Proceedings of the ACM Internet Measurement Conference*. 2020.

[206] Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations (ICLR)*. 2015.

[207] Smale, S. de, Dijk, R. van, Bouwman, X., Ham, J. van der, and Eeten, M. van. No one drinks from the firehose: How organizations filter and prioritize vulnerability information. In: *2023 IEEE Symposium on Security and Privacy (SP)*. 2023.

[208] Smutz, C. and Stavrou, A. Malicious PDF detection using metadata and structural features. In: *Proceedings of the 28th annual computer security applications conference*. 2012.

[209] Soska, K. and Christin, N. Automatically detecting vulnerable websites before they turn malicious. In: *23rd USENIX Security Symposium (USENIX Security 14)*. 2014.

[210] Soussi, W., Korczynski, M., Maroofi, S., and Duda, A. Feasibility of large-scale vulnerability notifications after gdpr. In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2020.

[211] Šrndic, N. and Laskov, P. Detection of malicious pdf files based on hierarchical document structure. In: *Proceedings of the 20th Annual Network & Distributed System Security Symposium*. 2013.

[212] Staddon, J. and Easterday, N. It's a generally exhausting field: A Large-Scale Study of Security Incident Management Workflows and Pain Points. In: *2019 17th International Conference on Privacy, Security and Trust (PST)*. IEEE. 2019.

[213] Stafeev, A., Recktenwald, T., De Stefano, G., Khodayari, S., and Pellegrino, G. YURASCANNER: Leveraging LLMs for Task-driven Web App Scanning. In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2024.

[214] Statcounter. *Search Engine Market Share Worldwide | Statcounter Global Stats*. `https://gs.statcounter.com/search-engine-market-share/all/worldwide/2020`. (Visited on 06/01/2023).

[215] Stock, B., Pellegrino, G., Li, F., Backes, M., and Rossow, C. Didn't You Hear Me?—Towards More Successful Web Vulnerability Notifications. In: *Proceedings of the 27th Network and Distributed System Security Symposium (NDSS)*. 2018.

[216] Stock, B., Pellegrino, G., Rossow, C., Johns, M., and Backes, M. Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification. In: *25th USENIX Security Symposium*. 2016.

[217] Stöver, A., Gerber, N., Pridöhl, H., Maass, M., Bretthauer, S., Hollick, M., Herrmann, D., et al. How website owners face privacy issues: Thematic analysis of responses from a covert notification study reveals diverse circumstances and challenges. *Proceedings on Privacy Enhancing Technologies* (2023).

[218] Strauss, A. L. and Corbin, J. M. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 3rd ed. Sage Publications, Thousand Oaks, Calif., 2008.

[219] Stringhini, G., Kruegel, C., and Vigna, G. Detecting spammers on social networks. In: *26th annual computer security applications conference*. 2010.

[220] Stringhini, G., Kruegel, C., and Vigna, G. Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. 2013. URL: http://doi.acm.org/10.1145/2508859.2516682.

[221] Subramani, K., Yuan, X., Setayeshfar, O., Vadrevu, P., Lee, K. H., and Perdisci, R. When push comes to ads: Measuring the rise of (malicious) push advertising. In: *Proceedings of the ACM Internet Measurement Conference*. 2020.

[222] Sutiah, S. and Supriyono, S. Software testing on e-learning Madrasahs using Blackbox testing. In: *IOP Conference Series: Materials Science and Engineering*. 2021.

[223] Tajalizadehkhoob, S., Böhme, R., Gañán, C., Korczyński, M., and Eeten, M. V. Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse. *ACM Transactions on Internet Technology (TOIT)* (2018).

[224] Tajalizadehkhoob, S., Korczyński, M., Noroozian, A., Ganán, C., and Van Eeten, M. Apples, oranges and hosting providers: Heterogeneity and security in the hosting market. In: *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*.

[225] Tajalizadehkhoob, S., Van Goethem, T., Korczyński, M., Noroozian, A., Böhme, R., Moore, T., Joosen, W., and Van Eeten, M. Herding vulnerable cats: a statistical approach to disentangle joint responsibility for web security in shared hosting. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, 553–567.

[226] Team, M. D. S. R. *New feature in Office 2016 can block macros and help prevent infection*. 2016. URL: https://www.microsoft.com/en-us/security/blog/2016/03/22/new-feature-in-office-2016-can-block-macros-and-help-prevent-infection/ (visited on 04/29/2025).

[227] The Alan Turing Institute. *Indirect Prompt Injection: Generative AI's Greatest Security Flaw*. 2024. URL: https://cetas.turing.ac.uk/publications/indirect-prompt-injection-generative-ais-greatest-security-flaw (visited on 07/11/2025).

[228] Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D. Design and Evaluation of a Real-Time URL Spam Filtering Service. In: *Proceedings of the 2011 IEEE Symposium on Security and Privacy*. SP '11. 2011. URL: https://doi.org/10.1109/SP.2011.25.

[229] Thomas, K., Huang, D., Wang, D., Bursztein, E., Grier, C., Holt, T. J., Kruegel, C., McCoy, D., Savage, S., and Vigna, G. Framing dependencies introduced by underground commoditization (2015).

[230] Thomas, K., McCoy, D., Grier, C., Kolcz, A., and Paxson, V. Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In: *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*. 2013.

[231] Tiefenau, C., Häring, M., Krombholz, K., and Von Zezschwitz, E. Security, availability, and multiple information sources: Exploring update behavior of system administrators. In: *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. 2020.

[232] Twitter Inc. *About unsafe links*. (Visited on 09/02/2019).

[233] Twitter Inc. *Optimize with Twitter Cards*. URL: https://developer.twitter.com/en/docs/tweets/optimize-with-cards/overview/abouts-cards (visited on 09/01/2019).

[234] Tzonkov, P. *KCFinder web file manager*. https://github.com/sunhater/kcfinder. 2014. (Visited on 01/31/2023).

[235] Tzonkov, P. *KCFinder web file manager [WayBack Machine]*. http://web.archive.org/web/20191103015814/https://kcfinder.sunhater.com/. 2019. (Visited on 01/31/2023).

[236] urlscan. *urlscan.io*. (01/22/2022). 2022.

[237] Utz, C., Michels, M., Degeling, M., Marnau, N., and Stock, B. Comparing large-scale privacy and security notifications. *Proceedings on Privacy Enhancing Technologies* (2023).

[238] Vadrevu, P. and Perdisci, R. What you see is not what you get: Discovering and tracking social engineering attack campaigns. In: *ACM Internet Measurement Conference*. 2019.

[239] Vailshery, L. S. *Global market share held by leading internet browsers from January 2012 to May 2023*. https://www.statista.com/statistics/268254/market-share-of-internet-browsers-worldwide-since-2009/. (Visited on 06/01/2023).

[240] Van Der Heijden, A. and Allodi, L. Cognitive triaging of phishing attacks. In: *28th USENIX Security Symposium*. 2019.

[241] Vasek, M. and Moore, T. Do malware reports expedite cleanup? An experimental study. In: USENIX Association. 2012.

[242] Vasek, M., Wadleigh, J., and Moore, T. Hacking is not random: a case-control study of webserver-compromise risk. *IEEE Transactions on Dependable and Secure Computing* (2015).

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

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

[245] VirusTotal. *VirusTotal - Home*. https://www.virustotal.com/. (Visited on 06/01/2023).

[246] Vishwanath, A., Herath, T., Chen, R., Wang, J., and Rao, H. R. Why do people get phished? Testing individual differences in phishing vulnerability within an integrated, information processing model. *Decision Support Systems* 51, 3 (2011), 576–586.

[247] Viswanath, B., Bashir, M. A., Crovella, M., Guha, S., Gummadi, K. P., Krishnamurthy, B., and Mislove, A. Towards detecting anomalous user behavior in online social networks. In: *23rd usenix security symposium (usenix security 14)*. 2014, 223–238.

[248] Wang, D. Y., Der, M., Karami, M., Saul, L., McCoy, D., Savage, S., and Voelker, G. M. Search+ seizure: The effectiveness of interventions on seo campaigns. In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. 2014, 359–372.

[249] Wang, D. Y., Savage, S., and Voelker, G. M. Cloak and dagger: dynamics of web search cloaking. In: *Proceedings of the 18th ACM conference on Computer and communications security*. 2011.

[250] Wang, Z., Sun, L., and Zhu, H. Defining social engineering in cybersecurity. *IEEe Access* (2020).

[251] Wappalyzer. *Wappalyzer*. `https://github.com/wappalyzer/wappalyzer`. 2022. (Visited on 08/20/2022).

[252] Webform module Open Collective, The Big Blue House. *Webform*. `https://www.drupal.org/project/webform`. 2023. (Visited on 01/31/2023).

[253] Whitty, M. T. and Buchanan, T. The online romance scam: A serious cybercrime. *CyberPsychology, Behavior, and Social Networking* (2012).

[254] Will Dormann and Art Manion, CERT Coordination Center. *Microsoft Internet Explorer window.createPopup() method creates chromeless windows*. `https://www.kb.cert.org/vuls/id/490708`. (Visited on 03/14/2025).

[255] Witzel, A. and Reiter, H. *The Problem-Centred Interview: Principles and Practice*. SAGE Publications Ltd, 2012.

[256] WPO-Foundation. *WPTAgent*. `https://github.com/WPO-Foundation/wptagent.git`. 2022. (Visited on 08/20/2022).

[257] WPScan. *WpScan homepage*. `https://wpscan.com/`. 2022. (Visited on 06/01/2023).

[258] Wu, B. and Davison, B. D. Identifying link farm spam pages. In: *Special interest tracks and posters of the 14th International Conference on World Wide Web*.

[259] Xiang, G., Hong, J., Rose, C. P., and Cranor, L. Cantina+ a feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)* (2011).

[260] XL Axiata. *Long-Distance Learning For Digital Madrasah XL Axiata Collaborates With The Ministry of Religious Affairs Help Madrasah Students In Need*. `https://www.xlaxiata.co.id/en/news/xlaxiata-collaborates-with-kemenag`. 2020. (Visited on 08/20/2022).

[261] Xu, M. and Kim, T. Platpal: Detecting malicious documents with platform diversity. In: *26th USENIX Security Symposium*. 2017.

[262] Zenity Labs - Michael Bargury. *Links and materials for Living off Microsoft Copilot*. 2024. URL: https://labs.zenity.io/p/links-materials-living-off-microsoft-copilot (visited on 07/11/2025).

[263] Zhang, J., Yang, C., Xu, Z., and Gu, G. Poisonamplifier: A guided approach of discovering compromised websites through reversing search poisoning attacks. In: *Research in Attacks, Intrusions, and Defenses*. 2012.

[264] Zhang, P., Oest, A., Cho, H., Sun, Z., Johnson, R., Wardman, B., Sarker, S., Kapravelos, A., Bao, T., Wang, R., et al. CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. In: *IEEE Symposium on Security and Privacy*. 2021.

[265] Zhang, Y., Hong, J. I., and Cranor, L. F. Cantina: a content-based approach to detecting phishing web sites. In: *16th international conference on World Wide Web*. 2007.

[266] Zhu, S., Shi, J., Yang, L., Qin, B., Zhang, Z., Song, L., and Wang, G. Measuring and Modeling the Label Dynamics of Online Anti-Malware Engines. In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020.

[267] Ziv, M., Izhikevich, L., Ruth, K., Izhikevich, K., and Durumeric, Z. ASdb: a system for classifying owners of autonomous systems. In: *Proceedings of the 21st ACM Internet Measurement Conference*. 2021.

[268] Zou, W., Geng, R., Wang, B., and Jia, J. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867* (2024).