Journal of *Cybersecurity and Privacy*

MDPI

Systematic Review

Abuse of Cloud-Based and Public Legitimate Services as Command-and-Control (C&C) Infrastructure: A Systematic Literature Review

Turki Al lelah *, George Theodorakopoulos *, Philipp Reinecke, Amir Javed and Eirini Anthi

School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK; reineckp@cardiff.ac.uk (P.R.); javeda7@cardiff.ac.uk (A.J.); anthies@cardiff.ac.uk (E.A.)
* Correspondence: allelaht@cardiff.ac.uk (T.A.l.); theodorakopoulosg@cardiff.ac.uk (G.T.)

Abstract: The widespread adoption of cloud-based and public legitimate services (CPLS) has inadvertently opened up new avenues for cyber attackers to establish covert and resilient command-and-control (C&C) communication channels. This abuse poses a significant cybersecurity threat, as it allows malicious traffic to blend seamlessly with legitimate network activities. Traditional detection systems are proving inadequate in accurately identifying such abuses, emphasizing the urgent need for more advanced detection techniques. In our study, we conducted an extensive systematic literature review (SLR) encompassing the academic and industrial literature from 2008 to July 2023. Our review provides a comprehensive categorization of the attack techniques employed in CPLS abuses and offers a detailed overview of the currently developed detection strategies. Our findings indicate a substantial increase in cloud-based abuses, facilitated by various attack techniques. Despite this alarming trend, the focus on developing detection strategies remains limited, with only 7 out of 91 studies addressing this concern. Our research serves as a comprehensive review of CPLS abuse for the C&C infrastructure. By examining the emerging techniques used in these attacks, we aim to make a significant contribution to the development of effective botnet defense strategies.

Keywords: botnet; command-and-control C&C; cloud; social network; online service; cyber abuse; systematic literature review

Citation: Al lelah, T.; Theodorakopoulos, G.; Reinecke, P.; Javed, A.; Anthi, E. Abuse of Cloud-Based and Public Legitimate Services as Command-and-Control (C&C) Infrastructure: A Systematic Literature Review. *J. Cybersecur. Priv.* **2023**, *3*, 558–590. https://doi.org/10.3390/jcp3030027

Academic Editors: Martin Gilje Jaatun, Massimiliano Rak and Ferhat Ozgur Catak

Received: 25 July 2023
Revised: 23 August 2023
Accepted: 25 August 2023
Published: 1 September 2023

1. Introduction

Individuals and organizations continuously face threats from different types of malware, including bots, ransomware, trojans, and worms. These malicious entities are predominant tools employed by cybercriminals. Bots, in particular, form a network known as a botnet, which comprises compromised computers controlled remotely by a botmaster or multiple controllers via a command-and-control (C&C) infrastructure. Botnets are among the most dominant threat vectors, posing severe threats to global Internet security.

While the earliest bots were developed for non-malicious purposes to facilitate and coordinate basic automation tasks [1], their misuse by botmasters has led to various malicious operations. These include data exfiltration, system degradation, distributed denial of service (DDoS), and phishing attacks. A botmaster typically employs evasive and reliable techniques, incorporating C&C communication in botnet operations to disseminate commands to an array of bots for achieving malicious aims [2].

Botnets utilize C&C communication channels to accomplish the objectives defined by botmasters and execute various malicious activities. The evolution of these C&C channels has encompassed multiple Internet protocols and botnet architectures, such as IRC, HTTP, and peer-to-peer (P2P). Depending on the communication protocols, C&C channels can be centralized, decentralized, or a combination of both. Typically, when a victim's computer is infected, it is directed to establish a remote connection with the C&C server using IP

addresses, DNS names, or node identifiers within peer-to-peer overlays embedded in the binary code.

The motivation for undertaking this SLR is provided in Section 2.2. Botnet authors have evolved their tactics by abusing the CPLS, as shown in Table 1, which includes services such as Microsoft Outlook, OneDrive, Slack, Dropbox, Pastebin, Twitter, and Google Drive. They utilize these services to establish C&C communication channels. By setting up a serverless C&C infrastructure within these services, botmasters can establish communication with bots that have been planted on victims' systems, thus evading detection. These stealthy communication channels leverage the trust placed in enterprise-authorized services, the reputable names of CPLS vendors, and secure communication protocols like transport layer security (TLS). This combination provides adversaries with an additional layer of protection.

Table 1. Abused CPLS platforms as C&C communication channels.

CPLS Category	CPLS Platforms
Online Cloud Storage Sites	Dropbox, Google Drive, OneDrive, pCloud, Yandex Disk, Mega, Alibaba Cloud, CloudMe
Social Media Platforms	Twitter, Facebook, Instagram
Business Communication Platform	Slack, Teams
Online Developers repository	Pastebin, Github, Microsoft TechNet
Content Sharing Platforms	YouTube, Imgur, ImgBB
Email Service	Outlook, Gmail, Exchange Web Services (EWSs)
Instant Messaging Platforms	Facebook Instant Messenger, Telegram
Miscellaneous	Google Scripts, File.io, Discord, Quora, Google Sites, Google Cloud Messaging (GCM)

Furthermore, users trust CPLS vendors implicitly to protect their data and provide secure access to it. However, malware authors exploit this trust, blending their malicious traffic within the legitimate traffic flow of these services. This subterfuge poses significant challenges for defenders, complicating their efforts to detect and prevent such attacks. These factors make cloud-based services an appealing choice for malware authors.

Despite the range of defense mechanisms proposed in the literature, our analysis revealed a deficiency in detection systems for effectively identifying the abuse of the CPLS as C&C infrastructure. This research gap was identified through this SLR.

Given the absence of a comprehensive overview on the latest attack techniques employed to abuse CPLS as a C&C infrastructure, we investigate these techniques and explore associated detection approaches. Although a related survey [3] addressed abuses against CPLS, it does not comprehensively cover all attack techniques, reporting on only four techniques: encoding, steganography, free accounts, and user generation algorithm (UGA). In contrast, our work encompasses nine techniques. Another study [4] focused on social bots and reported on a single abuse incident, where image steganography was used to employ Facebook and Twitter as C&C mediums. To our knowledge, this is the first systematic review that categorizes attack techniques used to abuse CPLS, along with their detection approaches.

The main contributions are as follows:

- Analyze existing studies relating to the abuse of CPLS as C&C servers.
- Introduce a new taxonomy of attack techniques that abuse the CPLS as a C&C server.
- Introduce a new taxonomy of C&C communication channels.
- Analyze and compare existing studies relating to the detection of the abuse of CPLS as C&C servers.

- Provide insights into the types of CPLS that are commonly targeted for abuse as C&C servers along with the employed attack techniques.
- Identify new challenges and propose directions for future research.

The rest of this paper is structured as follows: The background is presented in Section 2 followed by discussions of related surveys in Section 3. Section 2.3 provides a description of the threat model. The systematic literature review's methodology adopted for this research is explicated in Section 4. Section 5 focuses on the research findings and discussion, while Section 7 delves into works related to abuse detection. Section 8 highlights the directions for future research, and finally, Section 9 concludes the paper.

2. Background

2.1. Botnet Components

To improve the comprehension of botnet operations, we introduce its fundamental elements: bots, the botmaster, and the command and control channel (C&C). These components are depicted in Figure 1.

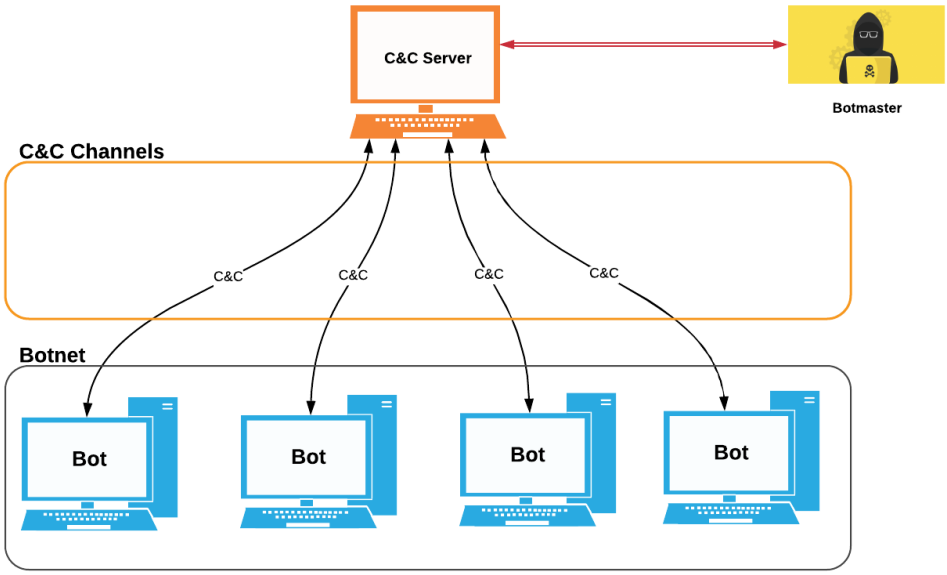


Figure 1. Botnet elements.

2.1.1. Bot

A bot refers to a software program (malware) that is installed on a compromised host, capable of performing a range of activities, often with malicious intent. Bots can be installed on victims' machines through various malware spreading mechanisms, including accessing infected websites or installing trojans. Bots are generally programmed so that they are initialized each time the victim boots up their computer [5,6].

2.1.2. Botnet

A collection of bots is called a botnet, which is connected to a C&C channel, forming a collective unit that awaits commands from the botmaster to carry out malicious activities.

2.1.3. Botmaster

Botmasters are individuals with malicious intent who exercise control over botnets by issuing commands to the bots, enabling them to engage in various malicious operations. These operations can include obtaining financial advantages, exfiltrating confidential data, degrading systems, launching distributed denial of service (DDoS) attacks, or sending spam [5,6].

2.1.4. C&C Communication Channels

It is common for botnets not to develop new network protocols for their own communication. Rather, they typically utilize pre-existing established communication protocols as follows:

- Internet relay chat (IRC)-based C&C channels: These utilize a push-based model, where the botmaster issues new commands to the botnet, which then responds promptly to these commands.
- HTTP-based C&C channels: These adopt a pull-based model, where bots are set up to check in with the C&C server periodically and retrieve any new commands.
- Peer-to-peer (P2P)-based C&C channels: These make use of peer-to-peer communication to either relay commands or to locate a C&C server [7].
- Domain name system (DNS)-based C&C channels: These use DNS tunneling, which is a technique that allows the encapsulation of non-DNS traffic within DNS packets [8].

C&C communication channels are utilized by botnets to facilitate communication between the botmaster and bot clients, as well as between bot clients themselves. In the first generation of botnets, internet relay chat (IRC) was commonly used for this purpose. The bots would connect to the IRC channels created by the botmaster on the C&C server to await commands to perform malicious activity. However, this approach had the significant drawback of relying on a single point of failure. If IRC servers were taken down or identified, the entire army of bot clients would become ineffective. To address this issue, the second generation of botnets transitioned to peer-to-peer (P2P) protocols [7,9]. This mechanism eliminated the vulnerability of a single point of failure. However, managing and controlling P2P-based botnets proved to be a challenge. As a result, botmasters have now shifted to utilizing HTTP for implementing C&C communication [9–11]. Furthermore, a new type of botnet has emerged, combining both P2P and HTTP protocols, known as hybrid botnets [6].

2.1.5. C&C Server

This essential component acts as the communication hub, facilitating the interaction between the botmaster and the bots. It is often referred to as the coordinator server. The botmaster utilizes these C&C servers to issue commands, and maintain and update bot programs. Bots establish connections to the C&C servers to receive commands or download bot binaries [6].

2.2. Motivation

The rising global adoption of CPLS and the increasing complexity of botnet attacks leveraging these as C&C servers are the motivation for this study. Botnet creators find CPLS extremely useful for establishing hidden and robust C&C communication channels. These channels are globally accessible, easy to implement, and offer a cost-effective setup. Moreover, the user-friendly nature of most CPLS, demanding minimal technical skills, lowers the bar for botnet creators.

These factors have turned CPLS into a fertile ground for botnet abuse, presenting substantial challenges for most intrusion detection systems. These systems often struggle to differentiate between legitimate and malicious C&C activities, leading to delayed detection and response. This delay in identifying and countering threats substantially increases the likelihood of successful cyber attacks, potentially leading to catastrophic consequences, such as data breaches, significant financial losses, and damage to organizations' reputations.

2.3. Threat Model

Abusing CPLS as C&C Channel

Understanding the life cycle of a botnet that abuses the CPLS as C&C is crucial for the successful analysis of botnet detection systems. By comprehending each phase of this cycle, it becomes possible to enhance and develop more efficient detection systems. Figure 2 illustrates how threat actors abuse CPLS as centralized C&C servers to control

bots. After compromising a victim's machine, the bot requires a remote control mechanism to communicate with the C&C server for further malicious instructions. To establish this remote C&C channel, bot connection requests are routed through CPLS instead of direct connections [12,13].

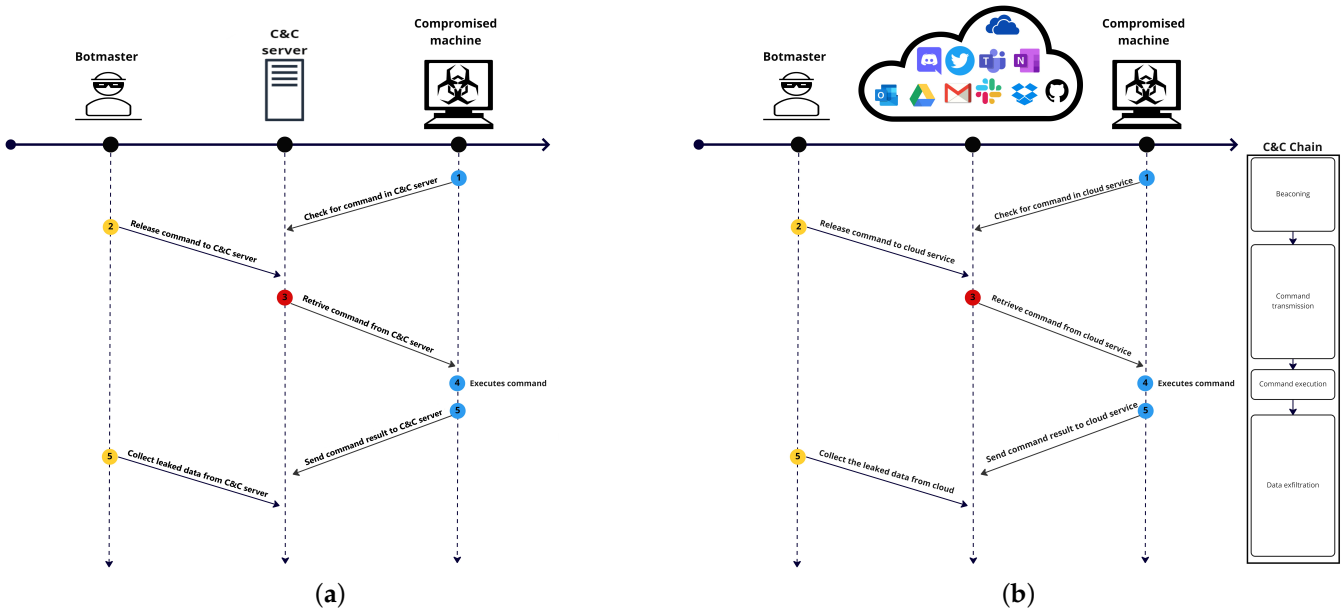


Figure 2. Illustration of (**a**) traditional C&C server communication and (**b**) CPLS platforms being abused as C&C.

The life cycle of abusing the CPLS as a C&C channel involves the following steps:

(a) The botmaster issues commands to the bots in the botnet through the CPLS.
(b) Bots continuously monitor the designated CPLS for new commands from the botmaster.
(c) Bots then execute the malicious commands.
(d) Bots report the results of the commands back to the botmaster through the CPLS.

This structure is stealthy and difficult to detect, as the bot retrieves commands and exfiltrates data through legitimate cloud-based services, making it challenging for network defenders and security solutions to differentiate between malicious and benign network traffic.

3. Related Surveys

Although there is a significant number of surveys focusing on botnet attacks and their detection mechanisms, such as those of Khattak et al. [14], Kuitert [15], Silva et al. [5], and Singh et al. [16], only Radunovic et al. [3] specifically address botnets abusing CPLS as C&C channels. In their survey, Radunovic et al. present a taxonomy of malicious social bots and discuss various attack types at different stages. They also propose four abusive tactics, including encoding communications within text-based social media posts, using steganography, exploiting free accounts, and implementing a username generation algorithm (UGA). Despite recognizing four abusive techniques in this survey, our proposed SLR identifies and categorizes nine different types of abusive attacks.

Table 2 compares our SLR to the survey by Radunovic et al. [3]. Our SLR adopts a unique methodology that diverges from the approach utilized in the related work. The SLR we conduct incorporates a structured and comprehensive search strategy, criteria for inclusion and exclusion, an evaluation of study quality, and data synthesis to formulate a taxonomy of attack techniques employed by botmasters to exploit CPLS C&C channels. Conversely, the related work does not follow a systematic review methodology. It instead provides an overview of current research on botnet C&C channels through social

media, investigating emerging trends, potential defense strategies, and areas warranting further research.

Khattak et al. [14] offer a concise overview of an evasive strategy that abuses social networking websites like Facebook and Twitter for C&C functions and provides a detailed taxonomy of the botnet phenomenon. This review also anticipates a trend, wherein CPLS would either be used to create bots (i.e., botcloud) or to host the C&C on the cloud in the future.

Latah [4] proposes a taxonomy of malicious social bots and characterization of various attack strategies at different stages (initiation, listening, and execution-stage attacks). The main focus is on a social bot, with the discussion being around one attack technique, which involves using steganography to abuse Facebook and Twitter as C&C mediums.

All the aforementioned reviews lack a systematic approach, thereby failing to provide a comprehensive overview of this research area. Despite the differing focuses and methodologies between the related surveys and our SLR, they offer significant insights into the evolving threat landscape of botnets and their use of non-traditional C&C channels. Our SLR, along with two other reviews [3,4], discusses the challenges associated with detecting and preventing the botnet exploitation of these non-traditional C&C channels. These reviews emphasize the necessity for ongoing research and innovative solutions to tackle emerging cyber threats.

Table 2. Comparison of our proposed SLR to that of Radunovic et al. [3].

Aspect	Our Proposed SLR	Radunovic et al. [3]
Focus	Abuse of the CPLS platforms as C&C channels	Abuse of social media platforms as C&C channels
Specific attacks discussed	Covers 10 types of attack techniques that are employed to abuse the CPLS platforms as C&C infrastructure.	Focuses specifically on the use of social media platforms through means such as status updates, comments, direct messages, and the creation of fake accounts.
Taxonomy details	Provides a comprehensive taxonomy of attack techniques used by botmasters to abuse CPLS as C&C channels. These techniques include steganography, encoding, cryptography, fraudulent accounts, use of Botmaster's credentials or hard-coded tokens, compromised victims' accounts, component object model (COM) hijacking, process injection, COMSPEC environment variable exploitation, multiple process exploitation, and AI-powered C&C.	This review discusses the use of text-based social media (SM) posts, hidden communications through image and linguistic steganography, and the utilization of public cloud storage for the unobservable exchange of communications and uploading of stolen files. It also includes the use of domain generation algorithms and the conveying of C&C messages through comments on public SM posts.
Review methodology	Systematic Literature Review.	Not specified.
Time frame	2008–July 2023.	Not specified.
Number of studies	91	Not specified.

4. Systematic Literature Review Methodology

4.1. Research Strategy

Based on the review aims, the methodological framework by Kitchenham and Charters [17] was adopted in this SLR. The review process outlined in this framework summarizes the stages of an SLR into three main phases: planning, conducting, and reporting. The primary motivation for adhering to these stages is to identify, analyze, and interpret all available literature related to the abuse of CPLS as a C&C infrastructure. Adherence to a predefined protocol is crucial for reducing potential research bias in data selection and analysis. It also enhances reliability through the replicability of the process, enabling others to follow the same procedure. Our systematic review commenced with the selection of bibliographic databases for the search, along with the development of a set of inclusion and exclusion criteria and search strings. The search process and the inclusion

and exclusion criteria used are presented below. The search strings were employed to query two primary sources: the academic and industrial literature. Abbreviations and synonyms of search terms were taken into consideration. To retrieve the most relevant literature, the search strings were restricted to titles, abstracts, and keywords.

4.2. Research Questions

Our SLR aimed to explore the state-of-the-art attack and detection techniques employed for abusing CPLS as C&C communication channels. To achieve this objective, we formulated the research categories and questions, which are outlined in Table 3.

Table 3. Research questions.

Category	Research Questions	Aim of Discussion
Abuse Technique	(1) What techniques are utilized to abuse the CPLS as C&C infrastructure?	• Investigate and understand the specific strategies or methods used by attackers to abuse CPLS for their C&C communication channels.
	(2) How frequently are these attack techniques employed, and which types of CPLS are targeted for such abuse?	• Assess the prevalence of these attack techniques, which may help to understand their popularity or effectiveness. • Identify the specific CPLS that are most vulnerable or frequently targeted by these abusive techniques.
Abuse Detection	(3) What countermeasures have been proposed to detect the abusive use of CPLS as C&C infrastructure?	• Explore and evaluate the existing countermeasures or detection methods that have been proposed to identify and combat the abusive use of CPLS as C&C infrastructure.

4.3. Search Process

The SLR process followed is depicted in Figure 3. To extract all literature relevant to the defined research questions, we employed a search strategy using Boolean expressions 'AND' and 'OR' to combine the search terms. The main keywords used in the search for relevant articles were as follows:

- (C2 **OR** C&C **OR** "Command and Control") **AND**.
- (cloud **OR** Legitimate **OR** platform **OR** Service **OR** public **OR** "public service" **OR** OSN **OR** "social network" **OR** blogging **OR** blog) **AND**.
- (bot **OR** botnet **OR** malware) **AND**.
- (abuse **OR** exploit).

These combined queries were then applied to a selection of academic databases that included the IEEE Xplore Digital Library, SpringerLink, ACM Digital Library, ScienceDirect, and Scopus.

Given that the majority of relevant incidents involving the abuse of CPLS as C&C channels were reported by the threat intelligence industry, we expanded our research to include these sources. Examples of these sources include FireEye, TrendMicro, WeLiveSecurity, F-Secure, Unit42 Palo Alto, and SecureList.

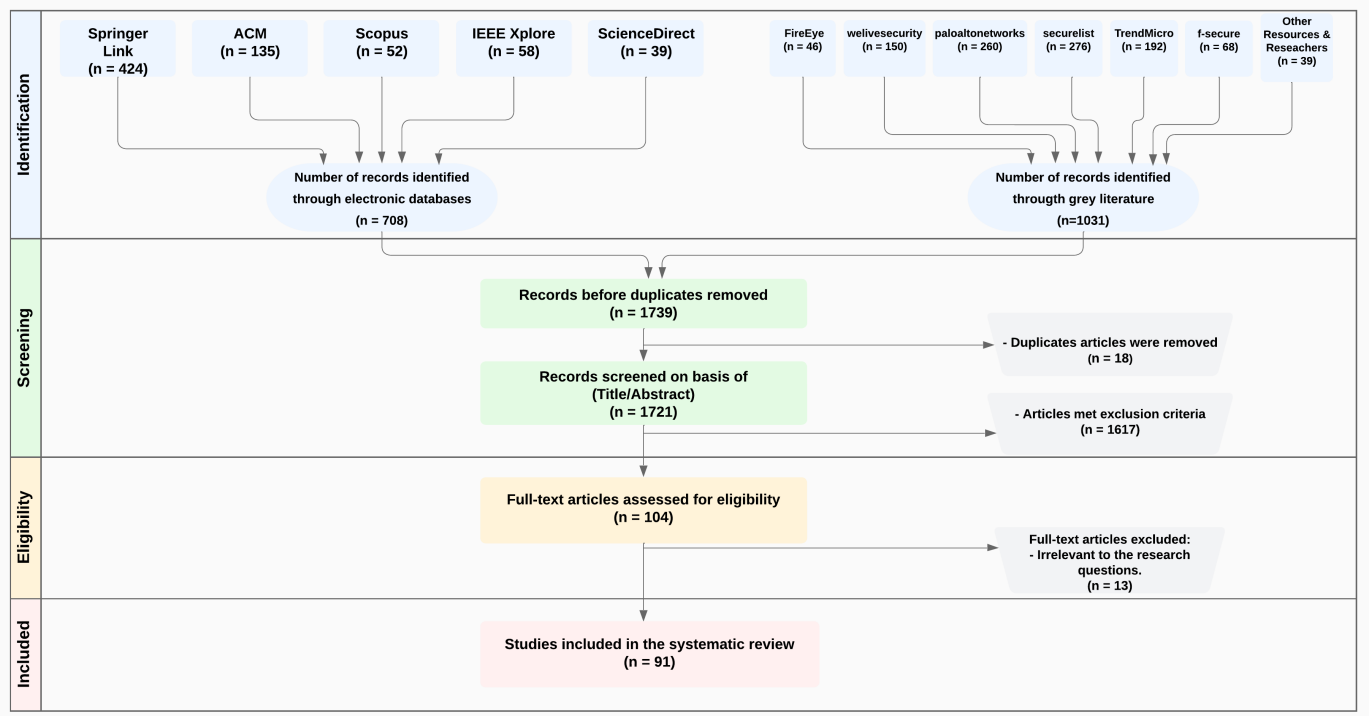


Figure 3. Process for extracting relevant articles.

4.4. Study Eligibility Selection

The search is limited to the literature that focuses on the abuse of CPLS C&C communication channels. Given the large number of articles that could be retrieved using the previously mentioned research strategy, an assessment criterion was necessary to select those that best address our research questions. To retrieve the most relevant articles in the field, we applied the selection criteria outlined in Table 4.

Table 4. Inclusion and exclusion criteria.

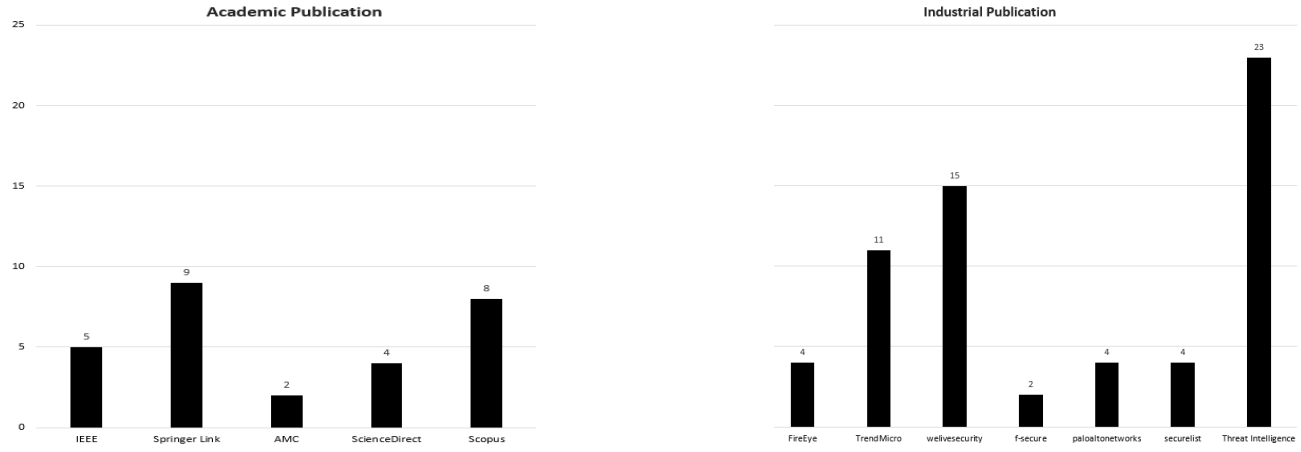
Inclusion Criteria	Exclusion Criteria
• Studies published from 2008, when the first abuse incident was discovered, through July 2023. • Studies focus on the abuse of CPLS as C&C channels. • Studies discuss an approach for identifying and/or preventing the abuse of CPLS as C&C channels.	• Studies written in a language other than English. • Studies focus on the behavior of malicious or automated accounts, bots, using social media to amplify and spread misinformation, increase fake followers, and impersonate genuine (human) accounts. • Studies that do not provide an answer to the research questions.

4.5. Data Extraction and Synthesis

After applying the inclusion and exclusion criteria, we obtained a total of 91 studies. This is illustrated in Figures 3 and 4. In the next phase, our aim is to review each of these studies to distill the following core information:

- The publication date of each study.
- The specific CPLS that was abused for C&C channels.
- The attack techniques that were utilized to abuse the CPLS as C&C channels.
- Any proposed detection mechanisms aiming to mitigate such abuses.

Subsequently, we embarked on the phase of data synthesis, wherein we aggregated all relevant data. This was a crucial step enabling us to comprehensively address our different research questions.



(**a**) (**b**)

Figure 4. The number of included publications from pre-defined (**a**) academic and (**b**) industrial sources.

5. Research Findings

In this SLR, we explored the topic of CPLS abuses as C&C channels, focusing particularly on understanding the attack techniques employed and countermeasures proposed. This abuse is not limited to a particular type of service; instead, it is pervasive across a variety of platforms, including cloud storage, social media, business communication platforms, and more.

Our findings indicate that botnets have evolved significantly over the years, adopting diverse methods to abuse the CPLS for C&C communication. For instance, we observed techniques such as steganography, cryptography, COM hijacking, process injection, and COMSPEC environment variable exploitation. Further discussion and additional attack techniques are presented in Section 6.

Many mainstream platforms were found to be popular targets of abuse: cloud storage sites such as Dropbox and Google Drive, social media sites like Twitter and Facebook, business communication platforms like Slack, developer repositories (GitHub), online clipboard sites (Pastebin), push services for iOS and Android notifications, video and photo sharing sites (YouTube and Instagram), email services (Outlook and Gmail), digital distribution platforms (Discord), and instant messaging software (Telegram).

The countermeasure strategies proposed to date, both in computer and Android environments, involve detecting anomalies in user behavior, CAPTCHA verification, reputation score calculation, and causality measurement between user activity and network traffic. However, each of these strategies come with their own limitations as outlined in Section 7.

5.1. RQ1: What Techniques Are Utilized to Abuse the CPLS as C&C Communication Channels?

Investigating and comprehending the specific strategies or methods employed by attackers to abuse CPLS for their C&C communication channels is essential for understanding the nature and extent of these attacks. The taxonomy of attack techniques presented in Table 5 showcases the wide range of methods utilized by threat actors to exploit CPLS for C&C operations. Each technique corresponds to a specific set of CPLS platforms that are targeted by threat actors for C&C activities. These techniques encompass diverse approaches, including concealing communication within seemingly legitimate files, using encoding to obscure commands or data, employing encryption for secure communication, creating fraudulent accounts, acquiring botmaster credentials or hard-coded tokens, compromising legitimate user accounts, hijacking COM components, AI-powered C&C, injecting malicious code into authorized processes, and modifying system variables. By gaining a comprehensive understanding of the specific techniques employed by attackers, valuable insights can be obtained regarding their tactics. These insights can then inform the development of effective countermeasures and proactive defense strategies to mitigate the risks associated with CPLS abuse in C&C operations.

Table 5. Taxonomy of the abusive techniques.

Technique	Abused CPLS	Description	Reference	Occurrences
Steganography	Dropbox, Google Cloud Messaging (GCM), Discord, Facebook, Twitter, Imgue, ImgBB, Evernote	Hiding communication between bots and C&C servers within legitimate-looking files, such as images or videos, and then transmitting them via cloud storage services	[18–26]	9
Encoding	weibo.com, Twitter, Facebbok, Google Docs, Instagram, YouTube, Yahoo, Quora, GitHub, outdrive, Dropbox, Google Drive, OneDrive, GCM, Microsoft TechNet, Pastebin, Mega Facebook Instance Messenger, Alibaba Cloud	Using encoding to make communications more difficult to detect, such as base64 encoding used to obfuscate C&C commands or data sent to the C&C server	[20,23,27–66]	40
Cryptography	Microsoft Outlook, Gmail, Dropbox, CloudMe, YouTube, Google Drive, OneDrive, Pastebin, Google Docs, Slack, Twitter, Facebbok, Weibo, GCM, pCloud, Yandex Disk, Github, Mega, Alibaba Cloud	Using encryption to secure communications between bots and C&C servers hosted on cloud-based services	[20,21,23,24,28,40,41, 62–65,67–81]	21
Fraudulent account creation	eams, OneNote, Outlook, Discord, Pastebin, Facebook, Twitter	Creating fraudulent accounts on cloud-based services to use as a disguise for C&C servers or to store botnet-related data	[24–26,39,59,82]	6
Botmaster's credentials or hard-coded tokens	Twitter, Telegram, Evernote, Slack, GitHub, Pastebin, Google+, CloudMe, GCM, Google Docs, Dropbox, OneDrive, Google Drive, Gmail, Microsoft Exchange Web Services, pCloud, Yandex Disk, Mega, Alibaba Cloud	Obtaining botmaster credentials or hard-coded tokens to access cloud-based services, which can then be used to host C&C servers or store botnet-related data	[3,12,13,23,30,31,35–38,43,48,48,52–54,63–65,71,72,75,76,78–81,83–85,85,86,86–107]	51
Compromised Accounts	Facebook Instance Messenger, Facebook, Twitter, Outlook, GCM, Dropbox, Microsoft Exchange Web Services Google Drive	Compromising legitimate user accounts on cloud-based services to use as a disguise for C&C servers or to store botnet-related data	[3,18,23,55,56,64,69,83, 84,94,108–110]	13
COM hijacking	Outlook, Gmail, Dropbox	Hijacking COM components on an infected system to communicate with a C&C server hosted on a cloud-based service	[69,70,99,100]	4
AI-powered C&C	Twitter	Employing neural networks for dynamic addressing, identifies attacker accounts via avatars, and embeds command in tweets via hash collisions and data augmentation	[111]	1
Process injection	Evernote	Injecting malicious code into legitimate processes to communicate with a C&C server hosted on a cloud-based service and evade detection	[89]	1
ComSpec environment variable	Dropbox	Modifying the ComSpec environment variable to point to a command shell on a cloud-based service to execute commands and communicate with a C&C server	[98]	1

Further exploration and analysis of these attack techniques can be found in Section 6.

5.2. RQ2: How Frequently Are These Attack Techniques Employed and Which Types of CPLS Are Targeted for Such Abuse?

Evaluating the frequency and prevalence of attack techniques used to exploit CPLS as C&C channels is crucial for understanding their popularity and effectiveness. Based on the information presented in Table 5, these attack techniques have been observed and documented across multiple references, indicating their use to varying degrees. However, obtaining precise quantitative data on the frequency of these techniques can be challenging due to the clandestine nature of cybercriminal operations and the ever-evolving threat landscape. Consequently, we acknowledge that these references may not encompass the entire spectrum of cyber threats associated with abusing CPLS. Despite this, the comprehensive list of references presented in Table 5 and the count of abuses by year for each technique illustrated in Figure 5 emphasize the importance of paying attention to these techniques.

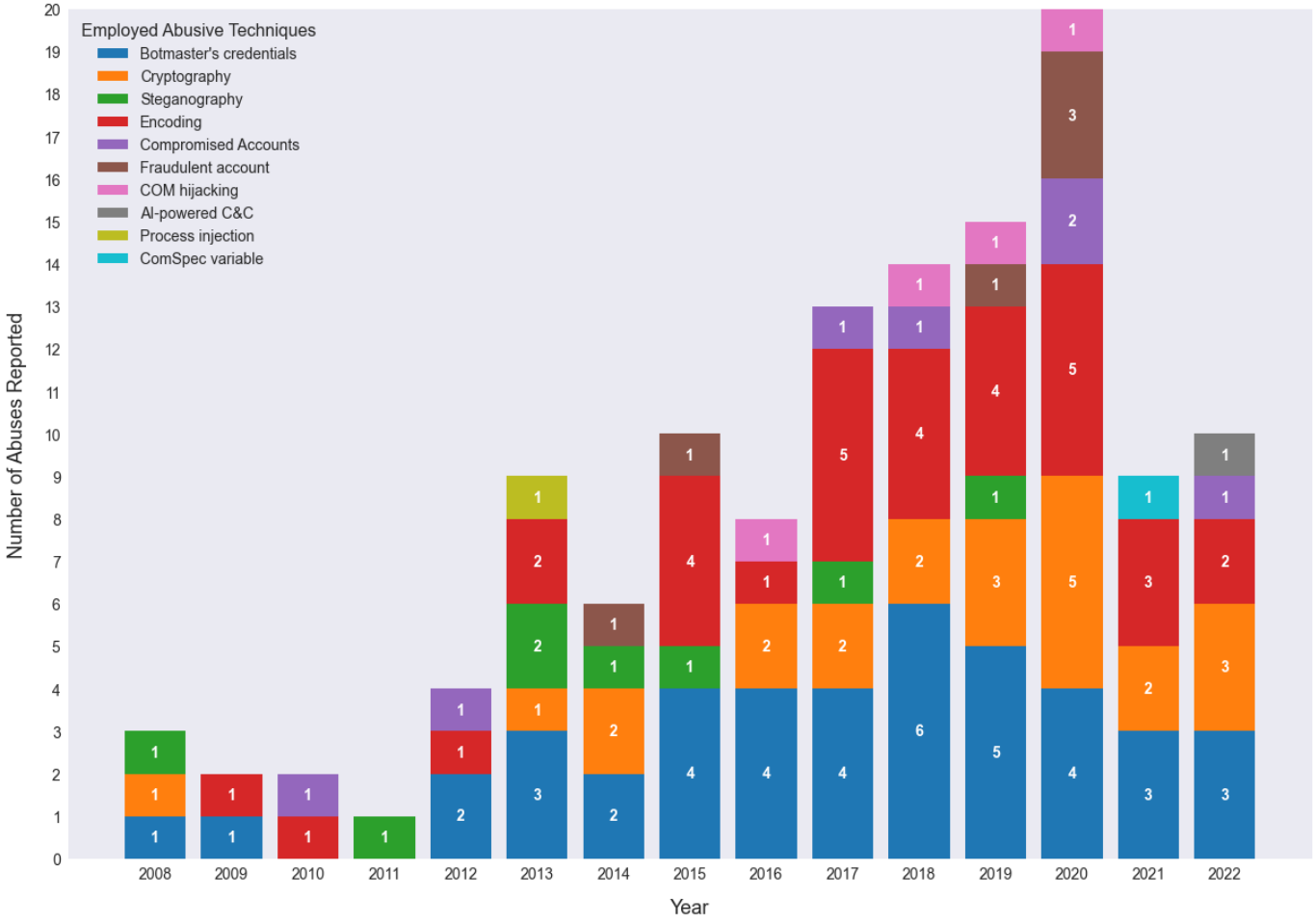


Figure 5. Number of abuses by year for each technique.

5.2.1. Prevalent Techniques

Encoding and the use of botmaster's credentials or hard-coded tokens are the most frequently employed techniques across platforms and over time. Their prevalence could stem from their relatively straightforward implementation and their ability to blend with normal network traffic, which likely make them a preferred choice for many cybercriminals.

5.2.2. Evolution and Trends

The complexity of techniques used by attackers seems to be increasing over the years. While earlier years primarily witnessed encoding and the use of botmaster credentials or

hard-coded tokens, later years show a rise in more advanced techniques, such as encoding, cryptography, compromised accounts, and even AI-powered C&C.

5.2.3. Correlation between Platform User Base and Abuse Occurrence

A positive correlation is observed between the number of users on a platform and the number of malware instances targeting that platform. Platforms with a larger user base, such as Google Docs, Dropbox, Twitter, Google Docs, Google Drive, YouTube, and Facebook, tend to report more abuse instances. The logic behind this correlation is intuitive; a larger user base provides a wider pool of potential victims for cyber attacks. Based on the conducted statistical analysis, there appears to be a positive association between the number of users on a platform and the number of abuse occurrences. Specifically, the slope of the regression line was calculated to be approximately 1.647—see Figure 6. This suggests that for each increase of one unit (one billion users) in a platform's user count, the occurrence of abuse tends to increase by 1.647 units on average.

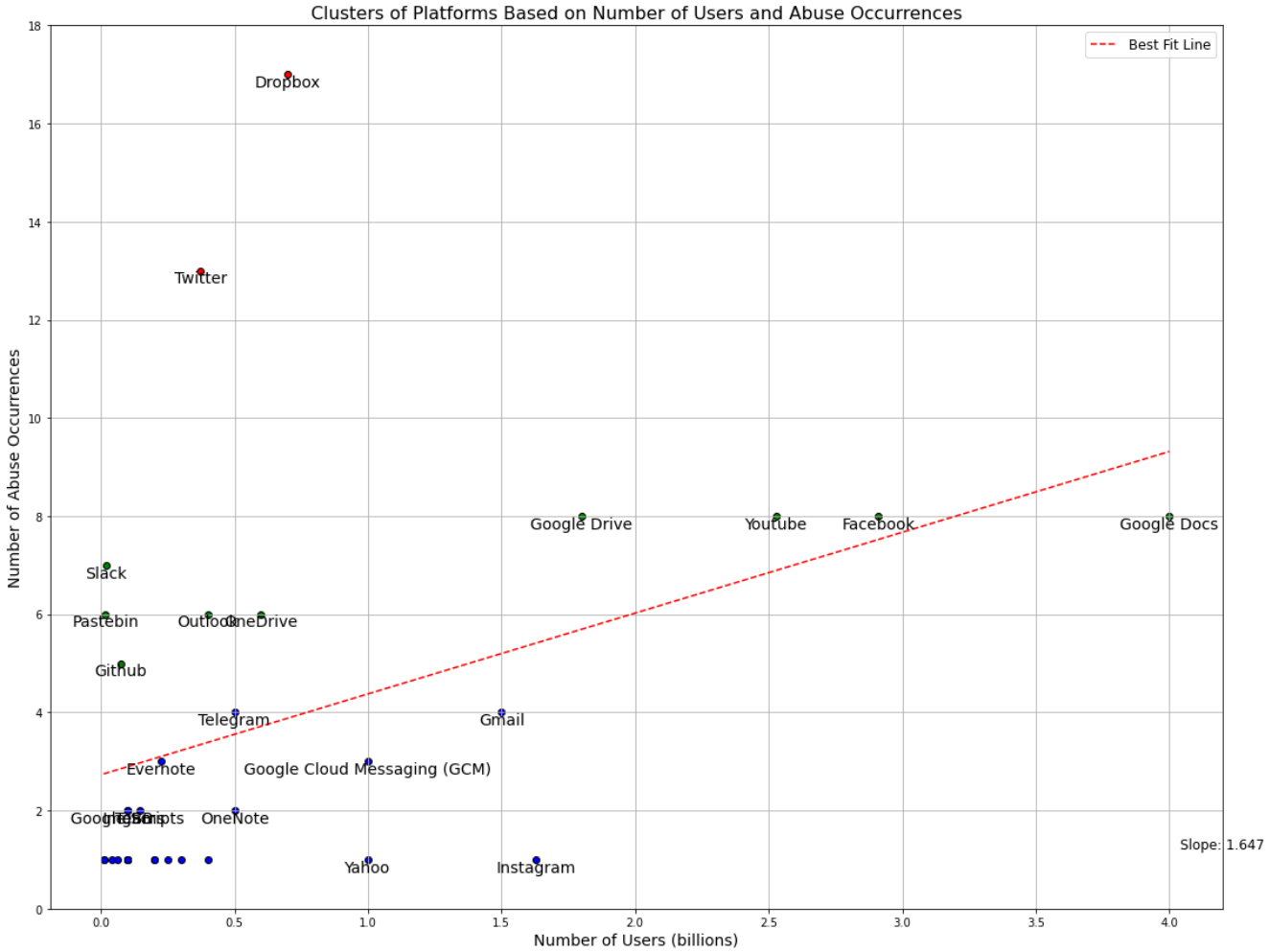


Figure 6. Correlation between number of users and abuse occurrences.

5.2.4. Platform Specific Trends and Corporate Usage

Table 6 presents a summary of malware occurrences grouped by CPLS. An occurrence is either an in-the-wild reported malware incident or a proof-of-concept (PoC) malware developed by security researchers. The data showcase the diverse range of CPLS exploited as C&C communication channels. CPLS like Dropbox, Google Docs, Google Drive, Outlook, and OneDrive, widely utilized in corporate environments, are particularly appealing to threat actors due to their integral roles in data storage, collaboration, and communication.

Our analysis reveals that these platforms have been exploited using a variety of techniques over the years, including botmaster credentials, encoding, cryptography, and compromised accounts. Their popularity in both the public sphere and corporate settings, coupled with their common usage for file storage and sharing, marks them as attractive targets for data theft and ransomware attacks. This emphasizes the need for robust security measures tailored to these commonly targeted platforms.

Table 6. Malware occurrences grouped by CPLS: a reported abuse is for in-the-wild malware incident, whereas a PoC (proof of concept) is malware created by security researchers.

CPLS	Occurrences	Incident Categories	
		Reported Abuse	PoC
Dropbox	17	[20,63,75,76,86,95–98]	[23,81,99–103,110]
Twitter	13	[20,21,30,39,66,85]	[24,28,31,32,48,67,111]
Google Docs	8	[35–41,50]	—
Google Drive	8	[39,51–54,64,77]	[23]
Youtube	8	[41–45,68,73,74]	—
Facebook	8	[33,61,68,107,108]	[18,19,67]
Slack	7	[12,90]	[79,91–93,112]
Outlook	6	[29,55,56,69,109]	[82]
OneDrive	6	[43,63,78,104,105]	[23]
Pastebin	6	[35,36,58–60,74]	—
Github	6	[12,21,47–49]	—
Gmail	4	[70]	[16,71,106]
Telegram	4	[13,87,88,107]	—
Google Cloud Messaging (GCM)	3	—	[22,23,94]
Evernote	2	[20,54,89]	—
Google Scripts	2	[35,36]	—
Discord	2	—	[25,26]
ImgBB	2	[20,39]	—
Microsoft TechNet	1	[57]	—
CloudMe	1	[72]	—
Imgur	1	[20]	—
Google+	1	[58]	—
Facebook Instance Messenger	1	—	[61]
File.io	1	[12]	—
Yahoo	1	[46]	—
Quora	1	[46]	—
Microsoft Teams	1	—	[82]
Microsoft OneNote	1	—	[82]
Google Sites	1	[68]	—
Instagram	1	[29]	—
pCloud	1	[80]	—
Yandex Disk	1	[80]	—
Alibaba Cloud	1	[65]	—
Mega	1	[63]	—
Exchange Web Services (EWS)	1	[3]	—

5.2.5. Anomaly Analysis

There are notable anomalies, wherein some platforms report low abuse instances despite maintaining large user bases. For example, Instagram, with over 1.6 billion users, recorded merely a single instance of abuse as indicated in Figure 6. Conversely, platforms such as Dropbox, Twitter, Slack, Pastebin, and Telegram are outliers with a smaller user base but a high number of abuse occurrences. This can be attributed to the following factors:

- Platform usage patterns: Different platforms cater to different user behaviors and usage patterns. For instance, platforms like Dropbox and Slack, despite having smaller user bases, often attract business and professional users. This can make them more appealing targets for abusers seeking access to sensitive information.

- Security measures: The level of security measures implemented by the platforms plays a crucial role in the number of abuse occurrences. Platforms with robust security features and stringent user verification processes might experience fewer instances of abuse, even with a large user base.
- Platform features and accessibility: platforms that offer a broader range of functionality and ease of use tend to attract more abusers, which provides more opportunities and tools for abusers to exploit.
- Anonymity: platforms that offer a certain level of anonymity seem to attract abusers. This could be because anonymity can make it easier for abusers to avoid identification.

5.2.6. Emerging Threats

The advent of AI-powered C&C attacks in 2022 highlights the continually increasing sophistication of cybercriminal tactics. While currently only observed on Twitter, it is conceivable that threat actors may leverage AI in abusing more platforms like Dropbox, Google Drive, Outlook, and OneDrive in the future. COM hijacking, though less prevalent currently, has been observed in several instances and poses a significant risk due to its ability to persist in a system undetected. This anticipatory threat underscores the need for continuous research and proactive defense strategies in cybersecurity.

5.2.7. Increasing Complexity

In recent years, attackers have increasingly employed a combination of techniques, and often abuse multiple CPLS platforms concurrently to increase the resilience of the C&C infrastructure and complicate tracking and takedown efforts. For instance, we have observed cases where cybercriminals combine encoding and cryptography techniques to enhance the concealment of C&C operations. This trend implies that attackers are continually advancing their tactics in order to evade detection. The combination of techniques increases the difficulty of detection, suggesting that multi-layered security strategies are essential for effective defense.

5.2.8. 2020: A Year of Escalated CPLS Abuse for C&C

As depicted in Figure 5, the year 2020 witnessed a substantial increase in the abuse of CPLS platforms for C&C operations. This escalation could potentially be attributed to the global COVID-19 pandemic, which resulted in a significant increase in online activities. The expanded online presence created a fertile ground for cybercriminals, offering them a larger pool of potential targets.

5.3. RQ3: What Countermeasures Have Been Proposed to Detect the Abusive Use of CPLS as C&C Infrastructure?

Several approaches have been proposed to detect the abusive use of CPLS as a C&C infrastructure. These countermeasures include the following.

5.3.1. Behavior Tree-Based Detection Framework

Kartaltepe et al. [32] propose detection at both the client-side and server-side levels. Host-side detection involves identifying botnets based on self-concealing, dubious network traffic, and unreliable provenance. Server-side detection focuses on identifying suspicious communication with social media platforms by analyzing the content of transmitted messages or posts. Ji et al. [27] present a behavior-tree-based detection framework for identifying social bots by monitoring host activity. This framework analyzes host behavior using a behavior tree constructed from real-world social bot samples. The similarity between suspicious behavior trees and a template library is calculated to determine the presence of social bots.

5.3.2. GCM Flow-Based Detection

Ahmadi et al. [113] propose a detection approach for identifying Android applications that exploit Google Cloud Messaging (GCM) as a C&C channel. This approach uses GCM flows as features in a machine learning model to differentiate between malicious and benign applications.

5.3.3. API Verification with CAPTCHA

Vo et al. [114] present API Verifier, a tool that utilizes CAPTCHA verification to determine if an API call to a social media platform is from a human or a bot. This verification process aims to prevent automated bot actions by requiring human authentication.

5.3.4. Negative Reputation Scores

Ghanadi et al. [115] propose SocialClymene, which calculates negative reputation scores based on user history and identifies suspicious group activity on online social networks. This approach can detect stego-botnets that utilize steganographic images for C&C communication.

5.3.5. Causality Detection

Burghouwt et al. [116] propose a causality detection mechanism for identifying Twitter-based C&C communication. This approach analyzes the causal relationship between user activity and network traffic to distinguish between user-triggered events and bot-originated events.

In conclusion, various countermeasures have been proposed to detect the abusive use of CPLS as C&C infrastructure. These countermeasures include host and server-side detection, behavior tree-based detection frameworks, GCM flow-based detection, API verification with CAPTCHA, negative reputation scores, and causality detection. Each approach offers unique techniques and methodologies to identify and differentiate between legitimate users and malicious bot activities. However, it is important to note that these countermeasures have their limitations, which are presented in detail in Table 7. Continuous research and improvement are necessary to enhance the effectiveness and accuracy of these detection mechanisms in combating the abusive use of CPLS as a C&C infrastructure.

Table 7. CPLS abuse detection mechanisms: a comparative overview of related work.

Reference	Anomaly-Based				Detection Mechanism	Limitation
	Passive		Active			
	Host-Based	Server-Based	Host-Based	Server-Based		
Yuede et al. [27]	✓				By utilizing behavior tree-based methodologies, the bot can be accurately identified through the monitoring of host activity. Upon construction of the behavior tree, the similarity to the designated template will be calculated utilizing the tree edit distance.	• Evade detection by performing random time delays between different behaviors of social bots.
Kartaltepe et al. [32]	✓	✓			The host-based security approach considers connections to social media potentially suspicious if they are not initiated by human interaction. This method utilizes behavioral biometrics, such as the reaction to user input through a keyboard or mouse, as well as the GUI, as indicators to differentiate between legitimate users and bots. On the other hand, the server-based approach assumes that communication with social media platforms is potentially suspicious if the messages or posts sent are textually encoded. To accurately classify incoming messages, the implementation of the J48 decision tree algorithm is utilized.	• Lack of empirical data and evaluation and relies heavily on case studies.
Ahmadi et al. [113]	✓				The Flowdroid tool was modified for use as a flow analysis tool to extract Google Cloud Messaging (GCM) flows. These extracted GCM flows were then utilized as vector space features in a machine learning model, which was implemented to identify malicious Android applications.	• Limited in its applicability, as it only works for Android OS and is not applicable in Windows OS environments. • Challenging to identify if obfuscation or hiding is implemented on GCM flows.
Vo et al. [114]				✓	Authors adopt a CAPTCHA verification technique to authenticate social media accounts by utilizing the MAC address. The purpose of this verification method is to distinguish between API calls originating from a human user or an automated bot.	• Modern botnets can bypass verification code verification by using a technique known as a relay attack. • MAC address can be easily spoofed or changed by attackers.
Ghanadi et al. [115]				✓	The paper proposes a system called SocialClymene which uses stego-images to detect covert botnets in social networks. The system has a negative reputation subsystem that analyzes images shared by social network users and calculates a reputation score for each user based on their history of participating in suspicious activities. The goal is to identify botnets by analyzing the behavior of the users and their association with suspicious activities.	• May not detect new botnets that have no previous history of suspicious behavior. • Challenging to accurately assess the reputation of users, especially in dynamic online environments where reputation can change rapidly.
Burghouwt et al. [116]	✓				Measuring the causal relationship between network traffic and human activity to distinguish between network events triggered by user actions and those bot-originated.	• False positives from legal API used for automated polling Twitter. • Evade detection by mimicking human activity such as mouse clicks or keyboard strokes.
Ji et al. [117]	✓				Incorporates spatial and temporal correlations to identify patterns of behavior that may be indicative of social bot activity.	• Focuses on only six social malicious bots and may not generalize to other botnets.

6. Abuse Attack Techniques

6.1. Steganography

This section examines the advancements in abusing CPLS platforms through image and text steganography. Steganography as a covert channel has garnered substantial attention from academia as demonstrated by proofs of concept, and from threat actors as observed in actual hacking incidents. Our review uncovered multiple instances in which CPLS were abused through the application of steganography techniques as C&C.

6.1.1. CPLS as a Primary C&C Communication Channel

ELISA (Elusive Social Army) [18] is an OSN-based botnet that abuses Facebook as a covert C&C channel, disseminating its messages through victims' social media accounts. ELISA establishes a covert channel using a Unicode steganography technique, inserting non-printable characters and invisible glyphs into user-generated messages posted on OSNs. These hidden elements are not displayed during the rendering process, making them hard to detect.

Stegobot [19], another OSN-based botnet, abuses Facebook as its main C&C communication channel. To establish this covert communication path within the social network, Stegobot employs a JPEG image steganography scheme known as YASS [50]. By doing so, it effectively embeds hidden information within digital images, thereby augmenting its stealth and evasion capabilities.

Punobot [22], a mobile botnet targeting Android systems, cleverly utilizes Google Cloud Messaging (GCM) for its C&C operations. Punobot employs a steganography technique that transforms the original command messages into different ones to evade detection by both users and the push notification service (PNS) server.

HAMMERTOSS [21] is a backdoor developed by the APT29 threat-active community. HAMMERTOSS was well crafted to cover the tracks of APT29 using several techniques, including building an algorithm that produces regular Twitter handles and the steganographic embedding of images with malicious commands. HAMMERTOSS also uses a domain generation algorithm (DGA) to create new Twitter handles. Whenever the malware creates a new handle, the corresponding Twitter page is fetched, and the page is searched for a particular pattern, which represents the encrypted C&C URL. HAMMERTOSS utilizes GitHub and cloud storage services as the primary C&C communication channels to transmit commands and relay stolen data from compromised networks. To obtain the malicious commands, HAMMERTOSS implements steganography techniques through images that contain encrypted malicious commands. Once HAMMERTOSS obtains the GitHub URL from its regular Twitter account, it visits the page, retrieves the steganographic images containing encrypted data, and upon successful connection and downloading, begins the decryption process to extract the actual command and perform the intended malicious operation.

RegDuke [20] malware employs steganography and cryptography techniques to hide data in PNG images. The developer of RegDuke misuses Dropbox by hosting steganography images containing an encrypted malicious command for covert C&C operations. The backdoor lists the Dropbox directory corresponding to a clientId (the compromised machine) and downloads the embedded PNG files. When images from the Dropbox directory are downloaded, the RegDuke code scans through all the image pixels and extracts data from them. The hidden data are specifically extracted from the implant, and the content is decrypted using an advanced encryption standard (AES) key that is hard-coded in the payload, which can be one of the following weapons of attack: Windows executable, Windows DLL, or PowerShell script.

This backdoor only resides in memory and relies on steganography to hide data in images using a technique called "least significant bit", which stores and combines 8 bits of data into a total of 24 bits of data per pixel: 8 for red, 8 for green, and 8 for blue. RegDuke consists of a loader and a payload, both components written in .NET. RegDuke persists by

using a WMI consumer named MicrosoftOfficeUpdates. The threat technical report [20] identifies four different primary variants of the RegDuke loader between August 2017 and June 2019. The first version was not obfuscated and had the encryption key hard-coded in the code. Later versions directly read the encryption key from the Windows registry and employed various obfuscation techniques, such as flattening the control flow or using .NET Reactor, a commercial obfuscator.

In 2012, Shuai et al. [118] proposed a malware called SUbot that leverages blog websites for the creation of covert channel communications to evade detection. The author of SUbot implemented steganography and cryptography strategies, using RC4 to hide a secret message and appending the ciphertext message to the end of a JPG file. The modified JPG file containing an executable command was then uploaded to a blog. When the infected mobile device visits the URL of the blog site, it downloads and retrieves the plaintext commands from the JPG image.

6.1.2. CPLS as a Redirector to C&C Domain

Twitter has been abused by the HAMMERTOSS malware as a mapper for the malicious URL. This is achieved by searching for a tweet with a URL and a hashtag; the URL points to the location of the C&C website with one or more images, while the hashtag allows HAMMERTOSS to extract encrypted instructions from an image file.

PolyglotDuke [20] is malware that is used by APT29 cyber espionage as a downloader for the MiniDuke backdoor. It uses various public websites, such as Twitter, Imgur, ImgBB, or Evernote public notes, to retrieve and decode the C&C URLs. It moreover relies on image steganography for its C&C communication channel.

6.1.3. Insights

Our review of steganography in the context of abusing the CPLS as C&C channels presents several important insights:

- Steganography has become a popular method among threat actors for maintaining a covert communication channel, often serving as the primary C&C channel. Cases like ELISA, Stegobot, Punobot, HAMMERTOSS, and RegDuke demonstrate the use of different steganographic techniques—from Unicode and image steganography to the least significant bit technique—effectively hiding malicious commands and evading detection.
- Online social networks (OSNs), along with other online platforms, have become common targets for misuse. As seen with ELISA and Stegobot using Facebook and HAMMERTOSS exploiting GitHub, these platforms provide a vast, noisy environment in which malicious activities can blend in, thus enhancing the effectiveness of the steganography.
- The sophistication and complexity of steganographic techniques have been increasing. Advanced encryption standard (AES) keys, domain generation algorithms (DGAs), and obfuscation techniques like .NET Reactor have been used in conjunction with steganography to further hide and protect the malicious payloads.
- Mobile platforms are not immune to these types of attacks. As shown by Punobot, steganography can also be employed in attacks targeting mobile devices, in this case using Google Cloud Messaging as a C&C channel.
- Steganography is not only used for direct C&C communications but also serves as a means of redirecting to C&C domains. HAMMERTOSS and PolyglotDuke represent cases where steganography was used to decode the C&C URLs.

6.2. Encoding

Encoding, typically employed to ensure data integrity and confidentiality, is paradoxically being weaponized to aid C&C operations. Our study reveals that all identified malware variants using encoding are designed to obfuscate their activities and evade detection.

6.2.1. CPLS as a Primary C&C Communication Channel

Nazario [30,66] identified a Twitter account being maliciously used as a C&C operations coordinator. The botmaster set up a Twitter account named "upd4t3" to conceal and disseminate malicious C&C messages. The botmaster broadcasts a tweet containing Base64-encoded commands. The bot then retrieves these commands via an RSS feed, decodes them, and executes them.

6.2.2. CPLS as Redirector to C&C Domain

ESET researchers discovered a Korplug variant [50] used by the Advanced Persistent Threat group, the Winnti group. This variant exploited publicly shared Google Docs files to extract its C&C address from a seemingly legitimate block of text using the well-known DZKS and DZJS delimiter strings.

An APT group named Turla used a trojan backdoor housed in a Firefox extension to retrieve the C&C URL. Turla APT is a cyberespionage group with over a decade of activity. In the analyzed sample of this threat report [29], the C&C domain was obscured using an encoding technique in a well-known celebrity's Instagram comment. More specifically, the Firefox extension parsed the photo comments on the official Instagram account and calculated a custom hash value. If the comment hash value equaled 183, a regular expression `(?:\\u200d(?:#|@)(\\w)` was then executed against the matched comment to obtain the C&C shortened URL path, which subsequently led to the actual C&C URL.

Stantinko [42] is a remotely configured cryptomining module that utilizes most of the compromised machine's resources. Stantinko malware interacts with its mining pool indirectly, through an IP address collected from YouTube video descriptions. These C&C IP addresses are concealed in a hexadecimal format in the video description string.

Janicab malware has exploited YouTube [43] to obtain the C&C IP. To retrieve the actual C&C IP, the malware navigates through comments on specific YouTube videos. If a format such as *"our 50380702789658th psy anniversary"* is matched, the obscured number from the satisfactory comments, leading to the C&C IP, is extracted and converted to an actual IP.

Palo Alto [46] documented two similar malware variations, CONFUCIUS_A and CONFUCIUS_B, which abused legitimate websites to retrieve C&C server IP addresses instead of using DNS lookups. To illustrate how the IP for the C&C domain is decoded, CONFUCIUS_A and CONFUCIUS_B use Yahoo and Quora to circumvent traditional mechanisms, parsing for keywords between specific phrases previously posted by the threat actor. The malware then decodes the interim phrase by substituting words for components of an IP address. A basic lookup table is used to decode and derive the C&C IP address.

Palo Alto also analyzed a new variant of the SunOrcal malware family [47], which exploited the GitHub service by employing a de-obfuscation process of Base64 and XOR decoding to extract a C&C server. Strategically, this sample was configured to connect to a specific file hosted on the GitHub repository to extract the data leading to the real C&C server. Text between two particular strings within this target file is parsed and encoded to derive the C&C URL.

The Scote backdoor is malware, discovered by Palo Alto Networks [58], that misuses legitimate third-party web platforms like Pastebin and Google+ as covert C&C communication channels. The Scote payload establishes a connection to these legitimate platforms' URLs to retrieve data and parse for specific commands to be executed by the compromised machine.

6.2.3. Insights

Our study of encoding in the context of abusing the CPLS as C&C channels presents several important insights:

- Encoding serves a dual purpose in these attacks. It not only provides a method of concealing C&C communications within legitimate service traffic, but also, it allows threat actors to hide the actual location of their C&C servers.

- Many of the observed malware variants, including the Korplug variant, Janicab, and CONFUCIUS_A and CONFUCIUS_B, demonstrate creative use of regular expressions and encoding techniques to retrieve C&C server addresses.
- Threat actors exhibit adaptability in their encoding methodologies, leveraging the specific characteristics of each platform to embed encoded data. For instance, using Instagram photo comments or YouTube video descriptions to conceal encoded commands or C&C server addresses.

6.3. Cryptography

6.3.1. CPLS as a Primary C&C Communication Channel

Shuai et al. [118] introduce SUbot, a botnet specifically designed for mobile platforms. SUbot leverages micro-blogging for evasion purposes and implements cryptography using RC4 to conceal a secret message. This message, which contains a malicious command, is appended to the end of a JPG file. Following this, the author of SUbot uploads the modified JPG file, which now contains an executable command, to a blog. When an infected mobile device visits the URL of the site, it downloads the image and retrieves the encrypted commands from the JPG file.

Sebastian et al. [24] designed a botnet to conceal malicious commands within a tweet. In their proposed experiment, the Botmaster covertly injects encrypted malicious commands into tweets to evade security measures. The format of the malware-infected tweet follows the `#keyword command` pattern, where the value of the `command` is encrypted. The bot then retrieves the tweet from the Botmaster's fake Twitter accounts and extracts the provided encrypted command. This command is then decrypted and interpreted, launching an attack on the infected machine.

He et al. [67] constructed prototypes of web test automation rootkit (WTAR) bots: Fbbot, Twbot, and Wbbot. These bots abuse Facebook, Twitter, and Weibo, respectively, as C&C infrastructures. The web test automation (WTA) technique, initially developed for automating browsers and testing websites, can perform actions such as filling in forms, reading data from web pages, and clicking elements on web pages. The WTAR-based technique has been leveraged to mimic typical user behaviors on an OSN. To better conceal the communication, thus making detection more difficult, both the botmaster and the bot use the data encryption standard (DES) to encrypt the commands and their corresponding execution results using a predefined key.

The Turla backdoor is a type of malware that exploits a victim's Outlook mailbox, using it as a transport layer for its C&C operations, receiving commands, and exfiltrating data. To better conceal the malicious commands and execution results, the designer of the Turla backdoor takes advantage of a previously opened session of the victim to gain access to the default mailbox profile [69]. In addition, the backdoor creator employs the MISTY1 symmetric encryption method. This method is used to create specially crafted PDF documents that contain either encrypted malicious instructions or confidential information, which are then attached to the compromised Outlook account's inbox.

ESET researchers [70] investigated a backdoor variant called ComRAT, which abuses Gmail as a covert C&C channel to receive commands and exfiltrate data. The ComRAT botnet authorizes the Gmail account using credentials embedded in the malware payload. ComRAT then connects to the Gmail HTML web interface to search for an email with a specific subject. Once matched, the email attachment is downloaded and decrypted via the AES-256 algorithm to extract the malicious command for execution. The executed command result is then encrypted using RSA-2048 and emailed to the threat actor, often hosted on GMX or VFEmail. To ensure persistence, ComRAT developers rely on a technique known as COM hijacking to tamper with the Windows registry, causing the ComRAT botnet to execute every time the user logs in.

RegDuke malware [20] utilizes cryptographic techniques to secure the data transmitted between the botmaster and the bots. Once an image is downloaded from Dropbox, the bot

loops over the image pixels to extract data. The bot then decrypts the information with an AES key hard-coded in the payload to retrieve malicious commands.

The CloudMe platform was exploited by CloudAtlas malware [72] as a covert communication channel. To make the transmitted messages between the attacker and the victim undetectable, the designer of the CloudAtlas malware employs cryptography with AES and data compression with LZMA techniques. The malware is configured to include encrypted contents such as the C&C's CloudMe URL, a username and password, and two folders on the CloudMe server for storing malicious commands and uploading the victim's data. The malware downloads the encrypted malicious commands uploaded by the threat actor, decrypts them, and then interprets them. It then uses the same mechanism to upload the result back to the server.

6.3.2. CPLS as Redirector to C&C Domain

Dong et al. [28] proposed a botnet design that combines QR codes, Twitter search, domain generation algorithms (DGAs), cryptography (AES and RSA), and Tor. In their design, Twitter is utilized as a mapper to the C&C web server. The Twitter search engine is queried for a particular keyword to locate the botmaster's post, which includes a QR code image. Upon scanning the QR code, the encoded combination of three components—the C&C web server address, the hard-coded token, and the RSA public key—is retrieved. The recovered RSA public key is then used to encrypt the data to be sent to the C&C web server.

The Casbaneiro botnet [73] utilizes YouTube, a legitimate website, to store its C&C server domains. The malware operator of Casbaneiro embeds an encrypted C&C web server address in a false Facebook or Instagram URL within the description of a specific YouTube video. This serves to redirect compromised machines to the threat actor's C&C infrastructure.

6.3.3. Insights

The use of cryptography techniques plays a crucial role in malware and botnet operations. It is primarily used for two purposes: protecting C&C communications and hiding malicious commands or data within legitimate-looking content. Our review of cryptography in the context of abusing the CPLS as C&C channels presents several important insights:

- The combined use of cryptography along with the abuse of popular CPLS platforms for C&C channels illustrates the increasing sophistication of botnet strategies. These platforms provide an additional layer of security and reliability, making it more difficult to detect botnet activities due to the reputable nature of these services.
- Botmasters frequently embed encrypted malicious commands into digital content, such as images or social media posts. These commands are later extracted and decrypted by bots for execution as demonstrated in cases like SUbot, ComRAT, and RegDuke.
- Cryptography is often combined with other techniques to enhance security and evasion capabilities. For instance, the ComRAT botnet merges cryptography with COM hijacking to ensure persistent execution.

6.4. Botmaster Login Credentials or Hard-Coded Token

6.4.1. CPLS as Primary C&C Communication Channel

Nazario [30,66] identified a Twitter account, named "upd4t3", that was being maliciously utilized as a C&C operations hub. The botmaster used this account to distribute Base64-encoded commands via tweets. The bot then fetched these commands using the RSS (really simple syndication) feed, decoded them, and executed them, thus establishing a covert C&C communication route.

Singh et al. [31] developed an OSN-based botnet named SocialNetworkingBot that leverages Twitter for its C&C communications. The malware author used authenticated official Twitter accounts to post tweets containing disguised commands to be interpreted by the bots. The botmaster posted tweets using approximately 300 pre-defined keywords to

facilitate fetching the tweets. Bots queried the Twitter search engine for specific keywords to retrieve these tweets from the botmaster account, which served as a rendezvous point. The bots then fetched and executed the malicious commands embedded in these tweets.

Telecrypt [87,107] is a type of ransomware that abuses Telegram's instant messaging API for its C&C infrastructure. Initially, the botmaster sets up a Telegram bot, which utilizes TeleCrypt ransomware to enable interaction between the threat actors and the compromised machine.

Upon infecting a machine, the ransomware employs Telegram's 'sendMessage' method. This method enables the bot to transmit messages to a chat thread associated with a specific number. This number is hard-coded into the ransomware's body, and is used to deliver a "successful infection" report back to the attackers. This is performed using the following request:

```
https://api.telegram.org/bot<token>/sendmessage?chat_id=<chat>
&text=<computer_name>_<infection_id>_<key_seed>
```

The request includes the following parameters:

- `<chat>`: Represents the chat number with the cybercriminal.
- `<computer_name>`: The name of the infected computer.
- `<infection_id>`: A unique identifier for the infection.
- `<key_seed>`: A number used as the basis for generating the file encryption key.

Following the transmission of this information, the trojan scans the infected machine's hard drives, looking for files with specific extensions. Once such files are identified, they are encrypted bytewise. This encryption is achieved by adding each file byte to the corresponding key bytes, employing a simple encryption algorithm.

ESET researchers have uncovered a unique malicious toolkit, TeleBot.AA [13], developed by the TeleBots APT group. This toolkit was specifically designed to abuse Telegram for C&C operations, utilizing the Telegram Bot API. Each version of the backdoor contains individual hard-coded credentials, indicating the presence of a Telegram Messenger account for each sample. The attacker communicates with the compromised systems through private Telegram chats, enabling the exchange of commands and retrieval of results.

Palo Alto Networks has detailed in an article [88] a malicious Android trojan known as "TeleRAT". This trojan abuses the Bot API of Telegram to carry out C&C activities. The TeleRAT spyware infiltrates the victim's Telegram app by masquerading as a legitimate application that promises to provide a count of profile visitors. The APKs of the malicious app contain hard-coded Telegram bot API keys, enabling them to periodically send beaconing signals at precise intervals of 4.6 s and await specific commands from the attacker.

TrendMicro [54] recently discovered a malware variant named BKDR_VERNOT.A that employs a clever technique to avoid detection. This malware utilizes legitimate services like Evernote, a web-based note-taking application, as a proxy server to establish communication with the botmaster. Upon successfully infecting the victim's machine, BKDR_VERNOT.A drops a .DLL file that injects itself into a genuine process, generating legitimate network traffic in order to evade detection by security solutions. The payload of BKDR_VERNOT.A leverages hard-coded official Evernote account credentials to connect to saved notes, enabling the backdoor to retrieve malicious commands and upload stolen data to a designated drop-off zone.

TrendMicro researchers discovered and analyzed a backdoor called "SLUB", which abuses three legitimate platforms—Slack, GitHub, and File.io—to establish its C&C infrastructure [12]. The threat actor sets up a Slack workspace and a GitHub account to facilitate SLUB backdoor C&C operations. To interact with the Slack API, the designer of SLUB embeds two hard-coded authentication tokens. The operator of SLUB uploads malicious commands to GitHub snippets, which the backdoor then retrieves and executes. The results of these commands are subsequently uploaded to both Slack and File.io.

TrendMicro researchers discovered a variant of the SLUB backdoor four months after the first version was identified [90]. This evolved version discontinued using GitHub for

its C&C operations, instead opting to fully integrate Slack workspaces as the covert C&C communication channel between the malware and its controller. The updated SLUB variant employs the same authentication approach as the previous version between the backdoor and its controller. Once a victim machine is infected by the SLUB backdoor and attempts to join a Slack workspace, a new channel titled `<use_name>-<pc_name>` is created. If the SLUB threat actor wants to execute a malicious command, they post the message to a victim-specific channel in Slack. The SLUB backdoor on the victim's machine then responds by parsing and executing the requested command.

The CloudAtlas malware exploited the CloudMe platform as a covert communication channel [72]. To secure the messages transmitted between the attacker and the victim and to make them undetectable, the designer of CloudAtlas employed a combination of AES cryptography and LZMA data compression techniques. The malware was configured to contain encrypted contents including the C&C's CloudMe URL, a username and password, and two folders on the CloudMe server designated for storing malicious commands and uploading the victim's data. Specifically, the botmaster uploads the encrypted malicious commands to the account, which the malware then downloads, decrypts, and interprets. The malware employs the same mechanism to upload the results back to the server.

Zhao et al. [94] introduced C2DM, an Android botnet architecture that abuses Google's Cloud to Device Messaging (C2DM) service for the dissemination of C&C commands. C2DM, a cloud-based push notification service for Android developers, was exploited in this botnet to eliminate the need for a direct connection between the botmaster and the bots. By blending the malicious bot traffic with the legitimate C2DM traffic from other Android apps, this botnet can covertly transmit its traffic. Furthermore, Zhao et al. [94] highlighted that many existing botnet detection strategies struggle to detect this type of push-like mobile botnet, as both malicious bots and legitimate applications use official push servers to receive updates.

Chen et al. [23] developed CloudBot, an enhanced version of a push-styled botnet [23]. CloudBot is a hybrid structured smartphone botnet (combining hierarchical and P2P structures) that abuses ten cloud-based push services (GCM, JPush, XGPush, ZYPush, GeXinPush, and Airbop) as a C&C downstream channel. Meanwhile, prominent cloud services like Dropbox, OneDrive, and Google Drive are used as a C&C upstream channel. CloudBot's design allows botmasters to send commands to bots in the form of legitimate push traffic via cloud-based push services. The bot then uploads the extracted data using cloud-based storage services. A significant advantage of using push services in a mobile botnet is the ability to avoid direct communication with C&C servers for command retrieval. CloudBot accesses the cloud storage services by embedding the account information and access token into a push message, which is then delivered to bots using the push services. Cloud-based push services can only support text messages. To satisfy this requirement and evade detection during command transmission, CloudBot incorporates three levels of obfuscation: encryption, encoding, and high-order mimic functions.

6.4.2. CPLS as Redirector to C&C Domain

Chen et al. [23] designed and implemented an Android-based push-styled botnet that abuses Google's message push service, GCM. This service is used as a mapper to direct users to the C&C URL domain to carry out malicious activities. For a more detailed discussion, refer to Section 6.4.1.

6.4.3. Insights

Our review of botmaster login credentials or hard-coded tokens in the context of abusing the CPLS as C&C channels presents the following insight:

- Many types of malware, like Telecrypt, TeleBot.AA, TeleRAT, BKDR_VERNOT.A, and CloudAtlas, use hard-coded credentials or tokens to authenticate themselves to these online services. These credentials or tokens are often embedded directly into the malware, enabling it to automatically and seamlessly connect to the service.

6.5. Compromised Accounts

6.5.1. CPLS as a Primary C&C Communication Channel

ELISA [18] is an OSN-based botnet that enables the botmaster to communicate with their bots by leveraging unaware user interactions and concealing its messages within victims' posts. ELISA constructs an overlay network, which interacts with typical users to deliver messages across the entire botnet. To ensure confidentiality, the C&C communication is safeguarded by using encryption and signatures between the botmaster and their bots.

As outlined in Sections 6.3.1, the Turla malware [69] leverages the victim's already opened session to access the default mailbox profile. Consequently, communication between the compromised Outlook email and the botmaster's email takes place through either an encrypted malicious instruction or encrypted PDF attachments.

6.5.2. CPLS as Redirector to C&C Domain

The Koobface botnet [107,108] is a social botnet that primarily relies on popular social networking sites like Facebook and Twitter for propagation. To achieve this, the botnet spams legitimate social network users and takes them through multiple layers of URL redirection to evade blocklist detection. The obfuscation process involves using blogs, RSS feeds, and shortened URLs to resolve and connect to the C&C URL.

6.6. Fraudulent Account

CPLS as a Primary C&C Communication Channel

Sebastian et al. [24] described a method that abuses Twitter as the main C&C communication medium by concealing the malicious commands within tweets. As discussed in Section 6.3.1, the bot retrieves the tweet from the botmaster's fake Twitter accounts, decrypts the embedded command, and executes it.

6.7. Component Object Model (COM) Hijacking

CPLS as a Primary C&C Communication Channel

Researchers at ESET conducted a thorough analysis of the Turla backdoor malware [69]. Turla exploits the victim's Outlook mailbox for C&C communication, receiving instructions, and exfiltrating data. After infecting the host, Turla utilizes the legitimate messaging application programming interface (MAPI) to interact with Outlook, granting complete control over the target mailbox and utilizing additional MAPI functionalities. To ensure persistence and concealment, the operators employ the COM to modify the Windows registry. This Microsoft technology enables developers to manipulate objects in various applications. The communication between the botmaster and the infected bot takes place through email, utilizing specially encrypted PDF attachments to transmit operational commands and exfiltrated information between compromised Outlook emails and the botmaster's email.

6.8. Artificial Intelligence (AI)-Powered C&C

CPLS as a Primary C&C Communication Channel

Wang et al. proposed DeepC2 [111], an innovative AI-powered C&C framework designed to address the challenges of covert communication on OSNs. Their approach leverages a neural network model for dynamic addressing, allowing the malware to identify the attacker's accounts through the extraction of feature vectors from avatars. By utilizing hash collision and easy data augmentation techniques, the attacker embeds commands within normal-looking tweets, ensuring covert communication while avoiding detection by OSN platforms. The framework leverages Twitter Trends as a rendezvous point.

6.9. Process Injection

CPLS as a Primary C&C Communication Channel

As discussed in Section 6.4.1, the BKDR_VERNOT.A malware [54] abuses the Evernote platform for its malicious operations. The threat actor's Evernote account credentials, hard-

coded into the malware binary, enable the bot to fetch malicious codes stored in the notes on the service. Upon execution, BKDR_VERNOT.A drops a specifically formatted .DLL file, or component, into the computer's temporary directory and injects itself into the legitimate process of Windows Explorer. This .DLL file initiates the backdoor's actual operations.

6.10. COMSPEC Environment Variable

CPLS as a Primary C&C Communication Channel

The BoxCaon backdoor, discovered by researchers at Checkpoint [98], abuses Dropbox as its C&C infrastructure. The backdoor employs the COMSPEC environment variable, which typically points to the command-line interpreter (cmd.exe), to execute malicious commands. The procedure works as follows: the attacker uploads files or commands to the Dropbox folder. The malware then fetches this folder and downloads all its contents to a working directory. If the file 'c.txt', which contains the attacker's command, is found in this working directory, the backdoor executes the command using the COMSPEC environment variable. The results of the command execution are then uploaded back to Dropbox, and the command is deleted.

6.11. Exploit Multiple Processes

CPLS as a Primary C&C Communication Channel

Yuede et al. [117] developed Wbbot, a social bot that exploits Twitter for C&C operations. The bot is designed to divide malicious behaviors into multiple processes, aiming to evade behavior detection mechanisms. Each process is dedicated to a specific malicious behavior, allowing them to exhibit benign behavior. Such a decentralized approach can make it challenging for traditional detection methods to accurately identify the overall malicious behavior.

7. Abuse Detection Mechanism

To date, only four countermeasures have been proposed to detect abuses of CPLS platforms as C&C communication channels. Three of these detection strategies have been developed for computer environments [27,32,114], while only one has been created for the Android phone environment [113].

Kartaltepe et al. [32] proposed a two-level abuse detection system: client side and server side. On the client side, they defined three features to identify botnets: self-concealing, dubious network traffic, and unreliable provenance. They presumed that connections to social media platforms could be deemed suspicious if not initiated by human behavior. They used behavioral biometrics, user input responses, and graphical user interface (GUI) interaction as detection attributes to differentiate between legitimate users and bots. For server-side detection, they assumed that any communication with social media platforms is suspicious if the transmitted messages or posts are textually encoded. They employed the J48 decision tree algorithm to classify input messages, distinguishing between Base64 or Hexadecimal-encoded text and plain language content. Nevertheless, these detection approaches have certain limitations: (i) lack of real-time detection, as the mechanisms were simulated in a post-analysis lab environment, and (ii) the potential for bypassing detection if crafty adversaries utilize image-steganography techniques to hide malicious commands within posts.

Yuede et al. [27] proposed a behavior tree-based detection framework aimed at identifying social bots by monitoring host activity. This framework consists of three components: a host behavior monitor, a host behavior analyzer, and a detection approach. To comprehensively understand and construct a suspicious host behavior tree for analysis, they designed a social botnet, named wbbot. This design also employed a sample collection from two sources: real-world social bots [30,31,107,108] and researcher malware samples of social bots [19]. After executing and analyzing this collection of social bots over a certain period, offline processes were used to generate a template library. This library was then employed to calculate the highest similarity value with the suspicious behavior tree. Once

the behavior tree was constructed, the tree edit distance was used to calculate its similarity to the template, resulting in the final detection outcome. However, this detection approach has a significant drawback: a high false positive rate of 29.6%. Moreover, the system can be circumvented if the attacker employs a multi-process mechanism or distributes malicious behaviors across various time spans.

Ahmadi et al. [113] proposed a detection methodology aimed at identifying Android applications that abuse GCM as a C&C channel. This methodology incorporates GCM flows as features within a machine learning model. The authors modified the Flowdroid tool [119] to extract GCM flows by distinguishing GCM callbacks, which were then used to train the model. The resulting derived features of GCM flows encompass information such as the GCM registration ID, sender ID, and the type of GCM message, demonstrating that these GCM flow features can effectively differentiate between malicious and benign applications. However, this approach can be vulnerable to evasion strategies like obfuscation and polymorphism, which attackers can employ to conceal GCM flows or other malicious behavior.

Vo et al. [114] introduced API Verifier, a tool employing CAPTCHA verification to authenticate social media account access via MAC address. The tool discerns whether an API call originates from a human or a bot, providing a layer of defense against automated bot actions. However, there are several limitations to the API Verifier tool presented by Vo et al. Firstly, the CAPTCHA verification system is susceptible to relay attacks, which could potentially allow botnets to bypass the verification process. Secondly, relying on MAC addresses for user identification may not be effective in scenarios where the user utilizes multiple devices, or in cases where MAC addresses can be easily spoofed.

Ghanadi et al. [115] delve into the study of stego-botnets that utilize steganographic images on online social networks for C&C operations. They introduce SocialClymene, a system designed to calculate each user's negative reputation score based on their historical activity. Within a suspicious group activity graph, SocialClymene sums the incoming normalized suspicious values. These values are then weighted by the negative reputation scores of the adjacent nodes, offering further insight into potentially malicious behavior.

Burghouwt et al. [116] proposed a causality detection mechanism that identifies Twitter-based C&C channel communication by measuring the correlation between user activity and network traffic. The authors posit that any network traffic event to the OSN that is not caused by human events based on specific keystrokes or mouse actions can be deemed suspicious. The causality detection approach uses a time frame that begins immediately after a user event to differentiate between network events triggered by user actions and those originated by bots. However, this detection approach has certain limitations. Firstly, legitimate API calls used for periodically automated polling may be falsely flagged as suspicious. Secondly, the primary parameters employed to evaluate the time frame between user activity and network requests may lack accuracy, given that different machines and operating systems exhibit varying delay times and performance characteristics. Lastly, advanced bots can potentially bypass this detection technique by monitoring user events and executing commands based on user-triggered events.

Ji et al. [117] conducted an empirical evaluation of several previously documented abusive social bots. They collected source codes, builders, and execution traces of existing social botnets, such as Twitterbot (Singh [120]), Twebot (Burghouwt et al. [121]), Yazanbot (Boshmaf et al. [122]), Nazbot (Kartaltepe et al. [32]), wbbot (Ji et al. [123]), and fbbot. Their aim was to analyze the mechanisms these bots utilize to evade existing detection approaches. Based on their analysis, the authors introduced a detection strategy that incorporates nine newly identified features with spatial and temporal correlations, along with nine features from existing detection methods. This approach is aimed at enhancing the detection of social bots on a variety of platforms.

Based on the analysis of the proposed detection approaches mentioned earlier, several key observations were discerned:

- Post-detection emphasis: Currently, most methods center on identifying botnet activities post-occurrence, often within a lab environment. This neglect of real-time

detection may prove inadequate, given the rapid evolution of threat landscapes. Therefore, the development of proactive detection techniques that can identify signs of such abuses before they occur is of critical importance.
- Evasion strategies: Modern botnets employ advanced evasion techniques that can bypass current detection mechanisms. More work is needed to understand and anticipate these strategies to improve detection.

8. Challenges and Directions for Future Research

While some progress has been made in detecting CPLS abuses as C&C channels, several challenges remain, and there are directions for future research that can further enhance the detection mechanisms. It is important to note, however, that obtaining precise quantitative data on the frequency of these techniques can be challenging due to the secretive nature of cybercriminal operations and the constantly evolving threat landscape. Therefore, we acknowledge that our references may not capture the entire landscape of cyber threats posed by the abuse of CPLS.

8.1. Quantity and Quality of Datasets

One of the challenges is the lack of dedicated datasets specifically designed for detecting malware that abuses CPLS platforms as C&C channels. Future research should focus on developing comprehensive and representative datasets that cover a wide range of CPLS platforms and abuse techniques, thereby enabling researchers to effectively evaluate and enhance detection mechanisms.

8.2. Emergence of AI-Powered C&C

As discussed in Section 6.8, AI-powered C&C emerged in 2022, indicating that attackers are utilizing AI to abuse Twitter as a C&C infrastructure. Although its current observation is limited to Twitter, it is conceivable that threat actors may employ AI in their abuses across various platforms. This highlights the importance of future research focusing on developing defenses against AI-powered C&C.

8.3. Deep Packet Inspection (DPI)

DPI plays a crucial role in identification the of malicious activities within cloud environments. Its effectiveness, however, is often hindered by tactics employed by attackers such as payload encryption, protocol encapsulation, and obfuscation. These tactics can mask the presence of malicious commands and communications, including those that involve CPLS platforms used as C&C infrastructure. Furthermore, the complexity of CPLS traffic, combined with the large volume of encrypted data and the legitimate use of these platforms, adds another layer of challenge to the application of DPI. This complexity requires more sophisticated analysis and detection techniques, which can discern subtle patterns of abuse amidst the large volume of normal traffic. In light of these challenges, future research should explore innovative approaches to overcome these obstacles and distinguish between legitimate and malicious uses of CPLS platforms without violating user privacy.

8.4. Evasion Tactics

Attackers continuously evolve their tactics to evade detection mechanisms. Future research should investigate advanced evasion techniques, such as obfuscation, polymorphism, and steganography. Robust countermeasures need to be developed to effectively detect and mitigate these tactics.

8.5. Cross-Platform Abuse Detection

Cybercriminals often abuse multiple platforms simultaneously and launch coordinated attacks. Research could focus on the development of cross-platform detection techniques that can identify and correlate malicious activities across different CPLS platforms.

8.6. Collaboration and Information Sharing

CPLS abuses often transcend individuals and require collaboration and information sharing among different stakeholders, including cloud providers, security researchers, and law enforcement agencies. Future research should explore ways to facilitate collaboration and information sharing to enhance the detection and mitigation of CPLS abuses.

9. Conclusions

Due to the massive increase in the use of CPLS solutions, there has been a corresponding rise in their abuse by threat actors as C&C infrastructures.

To address this open problem, it is essential to understand the evolving covert channel strategies as malware developers continuously devise new methods to evade detection.

In light of this, our study systematically reviews the relevant literature and presents a comprehensive overview through a SLR. Our overview describes various offensive techniques employed to abuse CPLS as C&C mechanisms, as well as different types of detection approaches. We evaluated 91 relevant articles from both academic and industry publications, which were published between 2008 and October 2023.

Therefore, our work contributes to the research in this area by introducing a refined taxonomy of abusive strategies. This taxonomy categorizes the methods used by threat actors to abuse CPLS as C&C infrastructures. In addition, our study elaborates on existing detection methods, examines their effectiveness, and highlights some unaddressed gaps and challenges in this field.

The primary objective of this research is to draw the attention of organizations and the research community to these sophisticated threats. Our analysis reveals that the majority of the examined publications primarily investigate abusive strategies, whereas less emphasis is dedicated to detection approaches.

In conclusion, detecting and mitigating CPLS abuses as C&C channels is an ongoing challenge due to the dynamic nature of cloud environments and the evolving tactics employed by attackers. By addressing the challenges and pursuing the directions for future research discussed above, we aim to improve the effectiveness and efficiency of detection mechanisms, enhance the security of CPLS platforms, and mitigate the risks associated with CPLS abuses.

Author Contributions: Conceptualization, T.A.l., G.T. and P.R.; methodology, T.A.l.; validation, T.A.l. and G.T.; formal analysis, T.A.l.; investigation, T.A.l.; resources, T.A.l.; writing—original draft preparation, T.A.l.; writing—review and editing, G.T., A.J. and E.A.; supervision, G.T.; funding acquisition, T.A.l. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jubail Industrial College (JIC) at Royal Commission for Jubail and Yanbu (RCJY), Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The first author thanks the Royal Commission for Jubail and Yanbu (RCJY) and Jubail Industrial College (JIC) for their generous PhD program sponsorship.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grizzard, J.B.; Sharma, V.; Nunnery, C.; Kang, B.B.; Dagon, D. Peer-to-Peer Botnets: Overview and Case Study. *HotBots* **2007**, *7*, pp. 1–7. Available online: https://dl.acm.org/doi/abs/10.5555/1323128.1323129 (accessed on 11 April 2020).
2. Abu Rajab, M.; Zarfoss, J.; Monrose, F.; Terzis, A. A multifaceted approach to understanding the botnet phenomenon. In Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, 25–27 October 2006, Rio de Janeiro Brazil ; pp. 41–52.
3. Radunović, V.; Veinović, M. Malware Command and Control Over Social Media: Towards the Server-less Infrastructure. *Serbian J. Electr. Eng.* **2020**, *17*, 357–375. [CrossRef]

4. Latah, M. Detection of malicious social bots: A survey and a refined taxonomy. *Expert Syst. Appl.* **2020**, *151*, 113383. [CrossRef]
5. Silva, S.S.; Silva, R.M.; Pinto, R.C.; Salles, R.M. Botnets: A survey. *Comput. Netw.* **2013**, *57*, 378–403. [CrossRef]
6. Limarunothai, R.; Munlin, M.A. Trends and challenges of botnet architectures and detection techniques. *J. Inf. Sci. Technol.* **2015**, *5*, 51–57.
7. Fedynyshyn, G.; Chuah, M.C.; Tan, G. Detection and classification of different botnet C&C channels. In Proceedings of the Autonomic and Trusted Computing: 8th International Conference, ATC 2011, Banff, AB, Canada, 2–4 September 2011; pp. 228–242.
8. Dietrich, C.J.; Rossow, C.; Freiling, F.C.; Bos, H.; Van Steen, M.; Pohlmann, N. On Botnets that use DNS for Command and Control. In Proceedings of the 2011 Seventh European Conference on Computer Network Defense, Gothenburg, Sweden, 6–7 September 2011; pp. 9–16.
9. Gu, G.; Perdisci, R.; Zhang, J.; Lee, W. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. In Proceedings of the USENIX Security Symposium. USENIX Association, San Jose, CA, USA, 28 July–1 August 2008; pp. 139–154.
10. Micro, T. Taxonomy of Botnet Threats. Whitepaper, November 2006. Available online: https://sites.cs.ucsb.edu/~kemm/courses/cs595G/TM06.pdf (accessed on 11 April 2020).
11. Liu, L.; Chen, S.; Yan, G.; Zhang, Z. Bottracer: Execution-based bot-like malware detection. In Proceedings of the Information Security: 11th International Conference, ISC 2008, Taipei, Taiwan, 15–18 September 2008; pp. 97–113.
12. Pernet, C.; Cao, E.; Horejsi, J.; Chen, J.C.; Sanchez, W.G. New SLUB Backdoor Uses GitHub, Communicates via Slack. 2019. Available online: https://www.trendmicro.com/en_gb/research/19/c/new-slub-backdoor-uses-github-communicates-via-slack.html (accessed on 11 April 2020).
13. Cherepanov, A. The Rise of TeleBots: Analyzing Disruptive KillDisk Attacks | WeLiveSecurity. 2016. Available online: https://www.welivesecurity.com/2016/12/13/rise-telebots-analyzing-disruptive-killdisk-attacks/ (accessed on 11 April 2020).
14. Khattak, S.; Ramay, N.R.; Khan, K.R.; Syed, A.A.; Khayam, S.A. A Taxonomy of Botnet Behavior, Detection, and Defense. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 898–924. [CrossRef]
15. Kuitert, S. War on Botnets. *Int. J. Inf. Technol. Eng. Res.* **2009**. Available online: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c72b4812cfaf65c88e45e7d8b53fffb355505cd0 (accessed on 11 April 2020).
16. Singh, K.; Srivastava, A.; Giffin, J.; Lee, W. Evaluating email's feasibility for botnet command and control. In Proceedings of the 2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN), Anchorage, AK, USA, 24–27 June 2008; pp. 376–385. [CrossRef]
17. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE-2007-01, EBSE; Durham University: Durham, UK, 2007.
18. Compagno, A.; Conti, M.; Lain, D.; Lovisotto, G.; Mancini, L.V. Boten ELISA: A novel approach for botnet C&C in online social networks. In Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS), Florence, Italy, 28–30 September 2015; pp. 74–82.
19. Nagaraja, S.; Houmansadr, A.; Piyawongwisal, P.; Singh, V.; Agarwal, P.; Borisov, N. *Stegobot: A Covert Social Network Botnet*; International Workshop on Information Hiding; Springer: Berlin/Heidelberg, Germany, 2011; pp. 299–313.
20. Operation Ghost: The Dukes Aren't Back—They Never Left | WeLiveSecurity. 2019. Available online: https://www.welivesecurity.com/2019/10/17/operation-ghost-dukes-never-left/ (accessed on 11 April 2020).
21. HAMMERTOSS: Stealthy Tactics Define a Russian Cyber Threat Group | FireEye. 2017. Available online: https://www.fireeye.com/current-threats/apt-groups/rpt-apt29.html (accessed on 11 April 2020).
22. Lee, H.; Kang, T.; Lee, S.; Kim, J.; Kim, Y. Punobot: Mobile botnet using push notification service in android. In Proceedings of the International Workshop on Information Security Applications, Jeju Island, Republic of Korea, 19–21 August 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 124–137.
23. Chen, W.; Gong, P.; Yu, L.; Yang, G. An adaptive push-styled command and control mechanism in mobile botnets. *Wuhan Univ. J. Nat. Sci.* **2013**, *18*, 427–434. [CrossRef]
24. Sebastian, S.; Ayyappan, S.; Vinod, P. Framework for design of Graybot in social network. In Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 24–27 September 2014; pp. 2331–2336.
25. DaaC2—Using Discord as a C2 | Crawl3r. 2020. Available online: https://crawl3r.github.io/2020-01-25/DaaC2 (accessed on 11 April 2020).
26. GitHub—Crawl3r/DaaC2: Discord as a C2. 2020. Available online: https://github.com/crawl3r/DaaC2 (accessed on 11 April 2020).
27. Ji, Y.; He, Y.; Jiang, X.; Li, Q. Towards social botnet behavior detecting in the end host. In Proceedings of the 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Hsinchu, Taiwan, 16–19 December 2014; pp. 320–327.
28. Dong, Y.; Dai, J.; Sun, X. A Mobile Botnet that Meets up at Twitter. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Singapore, 8–10 August 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–21.
29. Boutin, J.I. Turla's Watering Hole Campaign: An Updated Firefox Extension Abusing Instagram | WeLiveSecurity. 2017. Available online: https://www.welivesecurity.com/2017/06/06/turlas-watering-hole-campaign-updated-firefox-extension-abusing-instagram/ (accessed on 11 April 2020).
30. Singel, R. Hackers Use Twitter to Control Botnet | WIRED. 2009. Available online: https://www.wired.com/2009/08/botnet-tweets/ (accessed on 11 April 2020).

31. Singh, A.; Toderici, A.H.; Ross, K.; Stamp, M. Social Networking for Botnet Command and Control. *Int. J. Comput. Netw. Inf. Secur.* **2013**, *6*, 11–17. [CrossRef]
32. Kartaltepe, E.J.; Morales, J.A.; Xu, S.; Sandhu, R. *Social Network-Based Botnet Command-and-Control: Emerging Threats and Countermeasures*; Applied Cryptography and Network Security; Springer: Berlin/Heidelberg, Germay, 2010; pp. 511–528.
33. Chen, J. Blackgear Cyberespionage Campaign Resurfaces. 2018. Available online: https://www.trendmicro.com/en_us/research/18/g/blackgear-cyberespionage-campaign-resurfaces-abuses-social-media-for-cc-communication.html (accessed on 11 April 2020).
34. Alanazi, N.; Khan, E.; Gutub, A. Inclusion of Unicode standard seamless characters to expand Arabic text steganography for secure individual uses. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 1343–1356. [CrossRef]
35. Carr, N.; Goody, K.; Miller, S.; Vengerik, B. On the Hunt for FIN7: Pursuing an Enigmatic and Evasive Global Criminal Operation | Mandiant. 2018. Available online: https://www.mandiant.com/resources/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-operation (accessed on 11 April 2020).
36. Griffin, N. Carbanak Group Uses Google for Malware Command-and-Control | Forcepoint. 2017. Available online: https://www.forcepoint.com/blog/x-labs/carbanak-group-uses-google-malware-command-and-control (accessed on 11 April 2020).
37. Constantin, L. Malware Uses Google Docs as Proxy to Command and Control Server. 2012. Available online: https://www.pcworld.com/article/455736/malware-uses-google-docs-as-proxy-to-command-and-control-server.html (accessed on 11 April 2020).
38. Brook, C. Windows 8 Malware Using Google Docs to Target Brazilians | Threatpost. 2012. Available online: https://threatpost.com/windows-8-malware-using-google-docs-target-brazilians-111912/77227/ (accessed on 11 April 2020).
39. Mercer, W.; Rascagneres, P.; Ventura, V.; Kuhla, E. Cisco Talos Intelligence Group—Comprehensive Threat Intelligence: JhoneRAT: Cloud Based Python RAT Targeting Middle Eastern Countries. 2020. Available online: https://blog.talosintelligence.com/2020/01/jhonerat.html (accessed on 11 April 2020).
40. Passilly, T.; Tartare, M. The SideWalk May Be as Dangerous as the CROSSWALK | WeLiveSecurity. 2021. Available online: https://www.welivesecurity.com/2021/08/24/sidewalk-may-be-as-dangerous-as-crosswalk/ (accessed on 17 January 2022).
41. Ousaban: Private Photo Collection Hidden in a CABinet | WeLiveSecurity. 2021. Available online: https://www.welivesecurity.com/2021/05/05/ousaban-private-photo-collection-hidden-cabinet/ (accessed on 17 January 2022).
42. Hrčka, V. Stantinko Botnet Adds Cryptomining to Its Pool of Criminal Activities | WeLiveSecurity. 2019. Available online: https://www.welivesecurity.com/2019/11/26/stantinko-botnet-adds-cryptomining-criminal-activities/ (accessed on 11 April 2020).
43. Jarkko, K. News from the Lab Archive: January 2004 to September 2015. 2015. Available online: https://archive.f-secure.com/weblog/archives/00002803.html (accessed on 11 April 2020).
44. Biasini, N.; Brumaghin, E.; Lister, N. Cisco Talos Intelligence Group—Comprehensive Threat Intelligence: Threat Spotlight: Astaroth—Maze of Obfuscation and Evasion Reveals Dark Stealer. 2020. Available online: https://blog.talosintelligence.com/2020/05/astaroth-analysis.html (accessed on 17 January 2022).
45. Cimpanu, C. Astaroth Malware Hides Command Servers in YouTube Channel Descriptions | ZDNet. 2020. Available online: https://www.zdnet.com/article/astaroth-malware-hides-command-servers-in-youtube-channel-descriptions/ (accessed on 17 January 2022).
46. Lancaster, T.; Yates, M. Confucius Says...Malware Families Get Further by Abusing Legitimate Websites. 2016. Available online: https://unit42.paloaltonetworks.com/unit42-confucius-says-malware-families-get-further-by-abusing-legitimate-websites/ (accessed on 11 April 2020).
47. Grunzweig, J.; Miller-Osborn, J. SunOrcal Adds GitHub and Steganography to its Repertoire, Expands to Vietnam and Myanmar. 2017. Available online: https://unit42.paloaltonetworks.com/unit42-sunorcal-adds-github-steganography-repertoire-expands-vietnam-myanmar/ (accessed on 11 April 2020).
48. GitHub—PaulSec/Twittor: A Fully Featured Backdoor That Uses Twitter as a C&C Server. 2015. Available online: https://github.com/PaulSec/twittor (accessed on 11 April 2020).
49. Lunghi, D.; Horejsi, J.; Pernet, C. Untangling the Patchwork Cyberespionage Group. 2017. Available online: https://www.trendmicro.com/en_gb/research/17/l/untangling-the-patchwork-cyberespionage-group.html (accessed on 11 April 2020).
50. ESET_Threat_Report_Q22020.pdf. 2020. Available online: https://www.welivesecurity.com/wp-content/uploads/2020/07/ESET_Threat_Report_Q22020.pdf (accessed on 17 January 2022).
51. Falcone, R.; Lee, B. DarkHydrus Delivers New Trojan That Can Use Google Drive for C2 Communications. 2019. Available online: https://unit42.paloaltonetworks.com/darkhydrus-delivers-new-trojan-that-can-use-google-drive-for-c2-communications/ (accessed on 11 April 2020).
52. Kwiatkowski, I.; Aime, F.; Delcher, P. Holy Water: Ongoing Targeted Water-Holing Attack in Asia | Securelist. 2020. Available online: https://securelist.com/holy-water-ongoing-targeted-water-holing-attack-in-asia/96311/ (accessed on 17 January 2022).
53. Targeted Attacks Using Fake Flash against Tibetans | Volexity. 2020. Available online: https://www.volexity.com/blog/2020/03/31/storm-cloud-unleashed-tibetan-community-focus-of-highly-targeted-fake-flash-campaign/ (accessed on 17 January 2022).
54. Hacquebord, F.; Remorin, L.A. Pawn Storm's Lack of Sophistication as a Strategy. 2020. Available online: https://www.trendmicro.com/en_us/research/20/l/pawn-storm-lack-of-sophistication-as-a-strategy.html (accessed on 17 January 2022).
55. Dahan, A. Operation Cobalt Kitty: A Large-Scale APT in Asia Carried out by the OceanLotus Group. 2017. Available online: https://www.cybereason.com/blog/operation-cobalt-kitty-apt (accessed on 11 April 2020).
56. APT32, SeaLotus, OceanLotus, APT-C-00, Group G0050. 2017. Available online: https://attack.mitre.org/groups/G0050/ (accessed on 11 April 2020).

57. APT17: Hiding in Plain Sight—FireEye and Microsoft Expose Obfuscation Tactic | FireEye. 2015. Available online: https://www.fireeye.com/current-threats/apt-groups/rpt-apt17.html (accessed on 11 April 2020).
58. Grunzweig, J. The TopHat Campaign: Attacks within the Middle East Region Using Popular Third-Party Services. 2018. Available online: https://unit42.paloaltonetworks.com/unit42-the-tophat-campaign-attacks-within-the-middle-east-region-using-popular-third-party-services/ (accessed on 11 April 2020).
59. Rocke Evolves Its Arsenal With a New Malware Family Written in Golang | Anomali Labs. 2019. Available online: https://www.anomali.com/blog/rocke-evolves-its-arsenal-with-a-new-malware-family-written-in-golang (accessed on 11 April 2020).
60. Lambert, T. Threat Hunting in Linux For Rocke Cryptocurrency Mining Malware. 2021. Available online: https://redcanary.com/blog/rocke-cryptominer/ (accessed on 17 January 2022).
61. Chen, W.; Luo, X.; Yin, C.; Xiao, B.; Au, M.H.; Tang, Y. CloudBot: Advanced mobile botnets using ubiquitous cloud technologies. *Pervasive Mob. Comput.* **2017**, *41*, 270–285. [CrossRef]
62. Information on Attacks Involving 3CX Desktop App. 2023. Available online: https://www.trendmicro.com/en_us/research/23/c/information-on-attacks-involving-3cx-desktop-app.html (accessed on 26 May 2023).
63. Porolli, M. POLONIUM Targets Israel with Creepy Malware. 2022. Available online: https://www.welivesecurity.com/2022/10/11/polonium-targets-israel-creepy-malware/ (accessed on 26 May 2023).
64. WeLiveSecurity. 2022. Who's Swimming in South Korean Waters? Meet ScarCruft's Dolphin. Available online: https://www.welivesecurity.com/2022/11/30/whos-swimming-south-korean-waters-meet-scarcrufts-dolphin/ (accessed on 26 May 2023).
65. TrendMicro. BIOPASS RAT New Malware Sniffs Victims via Live Streaming. 2021. Available online: https://www.trendmicro.com/en_us/research/21/g/biopass-rat-new-malware-sniffs-victims-via-live-streaming.html (accessed on 17 January 2022).
66. Nazario, J. Twitter Based Botnet Command and Control. Arbor Networks Security. 2009 Available online: http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command-channel (accessed on 11 April 2020).
67. He, Y.; Zhang, G.; Wu, J.; Li, Q. Understanding a prospective approach to designing malicious social bots. *Secur. Commun. Netw.* **2016**, *9*, 2157–2172. [CrossRef]
68. The Tetrade: Brazilian Banking Malware Goes Global | Securelist. 2020. Available online: https://securelist.com/the-tetrade-brazilian-banking-malware/97779/ (accessed on 17 January 2022).
69. Foltýn, T. Turla: In and out of Its Unique Outlook Backdoor | WeLiveSecurity. 2018. Available online: https://www.welivesecurity.com/2018/08/22/turla-unique-outlook-backdoor/ (accessed on 11 April 2020).
70. Faou, M. From Agent.BTZ to ComRAT v4: A Ten-Year Journey | WeLiveSecurity. 2020. Available online: https://www.welivesecurity.com/2020/05/26/agentbtz-comratv4-ten-year-journey/ (accessed on 17 January 2022).
71. GitHub—Maldevel/Gdog: A Fully Featured Windows Backdoor That Uses Gmail as a C&C Server. 2016. Available online: https://github.com/maldevel/gdog (accessed on 11 April 2020).
72. Cloud Atlas: RedOctober APT Is Back in Style | Securelist. 2014. Available online: https://securelist.com/cloud-atlas-redoctober-apt-is-back-in-style/68083/ (accessed on 11 April 2020).
73. Casbaneiro: Dangerous Cooking with a Secret Ingredient | WeLiveSecurity. 2019. Available online: https://www.welivesecurity.com/2019/10/03/casbaneiro-trojan-dangerous-cooking/ (accessed on 11 April 2020).
74. Numando: Count Once, Code Twice | Welivesecurity. 2021. Available online: https://www.welivesecurity.com/2021/09/17/numando-latam-banking-trojan/ (accessed on 17 January 2022).
75. Faou, M. Turla Crutch: Keeping the "Back Door" Open | WeLiveSecurity. 2020. Available online: https://www.welivesecurity.com/2020/12/02/turla-crutch-keeping-back-door-open/ (accessed on 17 January 2022).
76. APT-31 Leverages COVID-19 Vaccine Theme | Zscaler Blog. 2020. Available online: https://www.zscaler.com/blogs/security-research/apt-31-leverages-covid-19-vaccine-theme-and-abuses-legitimate-online (accessed on 17 January 2022).
77. Raccoon Stealer's Abuse of Google Cloud Services and Multiple Delivery Techniques—TrendLabs Security Intelligence Blog. 2020. Available online: https://blog.trendmicro.com/trendlabs-security-intelligence/raccoon-stealers-abuse-of-google-cloud-services-and-multiple-delivery-techniques/ (accessed on 17 January 2022).
78. Faou, M.; Dumont, R. A Dive into Turla PowerShell Usage | WeLiveSecurity. 2019. Available online: https://www.welivesecurity.com/2019/05/29/turla-powershell-usage/ (accessed on 11 April 2020).
79. GitHub—Coalfire-Research/Slackor: A Golang Implant That Uses Slack as a Command and Control Server. 2019. Available online: https://github.com/Coalfire-Research/Slackor (accessed on 11 April 2020).
80. Léveillé, M.E.M. I See What you Did There: A Look at the CloudMensis macOS Spyware. 2022. Available online: https://www.welivesecurity.com/2022/07/19/i-see-what-you-did-there-look-cloudmensis-macos-spyware/ (accessed on 26 May 2023).
81. Command and Control—DropBox—Penetration Testing Lab. 2017. Available online: https://pentestlab.blog/2017/08/29/command-and-control-dropbox/ (accessed on 11 April 2020).
82. Introduction to Callidus. 2020. Available online: https://3xpl01tc0d3r.blogspot.com/2020/03/introduction-to-callidus.html (accessed on 17 January 2022).
83. Baltazar, R.J.; Costoya, J.; Flores, R. *The Heart of KOOBFACE: C&C and Social Network Propagation*; Trend Micro Threat Research; Trend Micro, Incorporated: Shibuya City, Tokyo, 2009; pp. 25–29.
84. Faghani, M.R.; Nguyen, U.T. Socellbot: A new botnet design to infect smartphones via online social networking. In Proceedings of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Montreal, QC, Canada, 29 April–2 May 2012; pp. 1–5.

85. Threat Analysis: ROKRAT Malware—VMware Security Blog—VMware. 2018. Available online: https://blogs.vmware.com/security/2018/02/threat-analysis-rokrat-malware.html (accessed on 11 April 2020).
86. Mercer, W.; Paul Rascagneres, J.A. Cisco Talos Intelligence Group—Comprehensive Threat Intelligence: ROKRAT Reloaded. 2017. Available online: https://blog.talosintelligence.com/2017/11/ROKRAT-Reloaded.html (accessed on 11 April 2020).
87. TeleCrypt—The Ransomware Abusing Telegram API—Defeated! | Malwarebytes Labs. 2016. Available online: https://blog.malwarebytes.com/threat-analysis/2016/11/telecrypt-the-ransomware-abusing-telegram-api-defeated/ (accessed on 11 April 2020).
88. Nigam, R.; Wilhoit, K. TeleRAT: Another Android Trojan Leveraging Telegram's Bot API to Target Iranian Users. 2018. Available online: https://unit42.paloaltonetworks.com/unit42-telerat-another-android-trojan-leveraging-telegrams-bot-api-to-target-iranian-users/ (accessed on 11 April 2020).
89. Tamaña, N. Backdoor Uses Evernote as Command-and-Control Server—TrendLabs Security Intelligence Blog. 2013. Available online: https://blog.trendmicro.com/trendlabs-security-intelligence/backdoor-uses-evernote-as-command-and-control-server/ (accessed on 11 April 2020).
90. Pernet, C.; Cao, E.; Horejsi, J.; Chen, J.C.; Sanchez, W.G. SLUB Gets Rid of GitHub, Intensifies Slack Use—TrendLabs Security Intelligence Blog. 2019. Available online: https://blog.trendmicro.com/trendlabs-security-intelligence/slub-gets-rid-of-github-intensifies-slack-use/ (accessed on 11 April 2020).
91. GitHub—Bkup/SlackShell: PowerShell to Slack C2. 2018. Available online: https://github.com/bkup/SlackShell (accessed on 11 April 2020).
92. GitHub—Praetorian-Inc/Slack-c2bot: Slack C2bot That Executes Commands and Returns the Output. 2019. Available online: https://github.com/praetorian-inc/slack-c2bot (accessed on 11 April 2020).
93. Using Slack Web Services as a C2 Channel (ATT&CK T1102)—Praetorian. 2019. Available online: https://www.praetorian.com/blog/using-slack-as-c2-channel-mitre-attack-web-service-t1102/ (accessed on 11 April 2020).
94. Zhao, S.; Lee, P.P.; Lui, J.C.; Guan, X.; Ma, X.; Tao, J. Cloud-based push-styled mobile botnets: A case study of exploiting the cloud to device messaging service. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, Fl, USA, 3 December 2012 ; pp. 119–128.
95. Menrige, M. PlugX RAT with "Time Bomb" Abuses Dropbox for Command-and-Control Settings—TrendLabs Security Intelligence Blog. 2014. Available online: https://blog.trendmicro.com/trendlabs-security-intelligence/plugx-rat-with-time-bomb-abuses-dropbox-for-command-and-control-settings/ (accessed on 11 April 2020).
96. China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets | Mandiant. 2015. Available online: https://www.mandiant.com/resources/china-based-threat (accessed on 11 April 2020).
97. Arsene, L. Iranian Chafer APT Targeted Air Transportation and Government in Kuwait and Saudi Arabia. 2020. Available online: https://www.bitdefender.com/blog/labs/iranian-chafer-apt-targeted-air-transportation-and-government-in-kuwait-and-saudi-arabia/ (accessed on 17 January 2022).
98. IndigoZebra APT Continues to Attack Central Asia with Evolving Tools—Check Point Research. 2021. Available online: https://research.checkpoint.com/2021/indigozebra-apt-continues-to-attack-central-asia-with-evolving-tools/ (accessed on 17 January 2022).
99. GitHub—Arno0x/DBC2: DBC2 (DropboxC2) is a Modular Post-Exploitation Tool, Composed of an Agent Running on the Victim's Machine, a Controler, Running on Any Machine, Powershell Modules, and Dropbox Servers as a Means of Communication. 2016. Available online: https://github.com/Arno0x/DBC2 (accessed on 11 April 2020).
100. Chandel, R. Command and Control with DropboxC2. 2019. Available online: https://www.hackingarticles.in/command-and-control-with-dropboxc2/ (accessed on 11 April 2020).
101. GitHub—0x09AL/DropboxC2C: DropboxC2C Is a Post-Exploitation Agent Which Uses Dropbox Infrastructure for Command and Control Operations. 2018. Available online: https://github.com/0x09AL/DropboxC2C (accessed on 11 April 2020).
102. Champion, A. Attack Detection Fundamentals: C2 and Exfiltration—Lab #3. 2020. Available online: https://labs.f-secure.com/blog/attack-detection-fundamentals-c2-and-exfiltration-lab-3/ (accessed on 17 January 2022).
103. GitHub—FSecureLABS/C3: Custom Command and Control (C3). A Framework for Rapid Prototyping of Custom C2 Channels, While Still Providing Integration with Existing Offensive Toolkits. 2020. Available online: https://github.com/FSecureLABS/C3 (accessed on 17 January 2022).
104. Hyvärinen, N. The Dukes: 7 Years of Russian Cyber-Espionage—F-Secure Blog. 2015. Available online: https://blog.f-secure.com/the-dukes-7-years-of-russian-cyber-espionage/ (accessed on 11 April 2020).
105. North Korean APT InkySquid Infects Victims Using Browser Exploits | Volexity. 2021. Available online: https://www.volexity.com/blog/2021/08/17/north-korean-apt-inkysquid-infects-victims-using-browser-exploits/ (accessed on 17 January 2022).
106. GitHub—Byt3bl33d3r/gcat: A PoC Backdoor That Uses Gmail as a C&C Server. 2018. Available online: https://github.com/byt3bl33d3r/gcat (accessed on 11 April 2020).
107. Ivanov, A.; Sinitsyn, F. The First Cryptor to Exploit Telegram | Securelist. 2016. Available online: https://securelist.com/the-first-cryptor-to-exploit-telegram/76558/ (accessed on 11 April 2020).
108. Thomas, K.; Nicol, D.M. The Koobface botnet and the rise of social malware. In Proceedings of the 2010 5th International Conference on Malicious and Unwanted Software, Nancy, France, 19–20 October 2010; pp. 63–70.

109. Ben Koehl, J.H. Microsoft Security—Detecting Empires in the Cloud—Microsoft Security Blog. 2020. Available online: https://www.microsoft.com/security/blog/2020/09/24/gadolinium-detecting-empires-cloud/ (accessed on 17 January 2022).
110. Williams, J. DropSmack: How Cloud Synchronization Services Render Your Corporate Firewall Worthless. 2013. Available online: https://docs.huihoo.com/blackhat/europe-2013/bh-eu-13-dropsmack-jwilliams-wp.pdf (accessed on 11 April 2020).
111. Wang, Z.; Liu, C.; Cui, X.; Yin, J.; Liu, J.; Wu, D.; Liu, Q. DeepC2: Ai-powered covert command and control on OSNs. In *Information and Communications Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 394–414.
112. How New Chat Platforms Can Be Abused by Cybercriminals—Noticias de Seguridad—Trend Micro ES. 2017. Available online: https://www.trendmicro.com/vinfo/es/security/news/cybercrime-and-digital-threats/how-new-chat-platforms-abused-by-cybercriminals (accessed on 11 April 2020).
113. Ahmadi, M.; Biggio, B.; Arzt, S.; Ariu, D.; Giacinto, G. Detecting misuse of google cloud messaging in android badware. In Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, Vienna, Austria, 24 October 2016; pp. 103–112.
114. Vo, N.H.; Pieprzyk, J. Protecting web 2.0 services from botnet exploitations. In Proceedings of the 2010 Second Cybercrime and Trustworthy Computing Workshop, Ballarat, VIC, Australia, 19–20 July 2010; pp. 18–28.
115. Ghanadi, M.; Abadi, M. Socialclymene: A negative reputation system for covert botnet detection in social networks. In Proceedings of the 7'th International Symposium on Telecommunications (IST'2014), Tehran, Iran, 9–11 September 2014; pp. 954–960.
116. Burghouwt, P.; Spruit, M.; Sips, H. Towards detection of botnet communication through social media by monitoring user activity. In *Information Systems Security*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 131–143.
117. Ji, Y.; He, Y.; Jiang, X.; Cao, J.; Li, Q. Combating the evasion mechanisms of social bots. *Comput. Secur.* **2016**, *58*, 230–249. [CrossRef]
118. Shuai, W.; Xiang, C.; Peng, L.; Dan, L. S-URL flux: A novel C&C protocol for mobile botnets. In *International Conference on Trustworthy Computing and Services*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 412–419.
119. Arzt, S. Static Data Flow Analysis for Android Applications. Ph.D. Thesis, Darmstadt University of Technology, Darmstadt, Germany, 2017.
120. Singh, A. Social Networking for Botnet Command and Control. Master's Thesis, San Jose State University, San Jose, CA, USA, 2012.
121. Burghouwt, P.; Spruit, M.; Sips, H. Detection of covert botnet command and control channels by causal analysis of traffic flows. In *International Symposium on Cyberspace Safety and Security*; Springer Berlin/Heidelberg, Germany, 2013; pp. 117–131.
122. Boshmaf, Y.; Muslukhov, I.; Beznosov, K.; Ripeanu, M. Design and analysis of a social botnet. *Comput. Netw.* **2013**, *57*, 556–578. [CrossRef]
123. Ji, Y.; He, Y.; Zhu, D.; Li, Q.; Guo, D. A mulitiprocess mechanism of evading behavior-based bot detection approaches. In *Information Security Practice and Experience*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 75–89.