A Taxonomic Overview of Prevalent Malware Communication Strategies

Steffen Enders, Daniel Plohmann, and Manuel Blatt
Fraunhofer FKIE

This paper was presented at Botconf 2024, Nice, 23-26 April 2024, www.botconf.eu
It is published in the Journal on Cybercrime & Digital Investigations by CECyF, https://journal.cecyf.fr/ojs

Abstract

Malware analysis remains a critical task in cybersecurity, particularly given the prevalent use of network capabilities by many malware samples. Despite the need to discuss and understand the use of networking in malware, there is currently no comprehensive taxonomy to classify the various aspects of malware C&C communication. This lack of taxonomy has resulted in the absence of a categorized overview of the communication strategies utilized by prevalent malware. Furthermore, no structured data set is available that includes representative samples of common malware families along with their respective network traffic captures, which is crucial to develop new malware networking analysis methods or to improve manual analysis skills.

In this paper, we make three main contributions. First, we propose a taxonomy for malware C&C communication strategies, adapted and expanded from the Trend Micro botnet taxonomy, to ensure that it can be systematically applied to categorize and describe C&C communication methods more broadly. Second, we provide an organized summary of the current malware C&C communication landscape, based on the malware families most frequently submitted to MalwareBazaar, to give researchers an overview of common techniques. Third, we release a data set containing samples of these prevalent malware families together with live network traffic captures to facilitate research and the development of new tools for malware networking analysis.

Keywords: malware analysis, C2 communication, network traffic analysis, reverse engineering.

1 Introduction

Malware poses a continuous threat to computer and network security, with its operations largely dependent on effective Command & Control (C&C) communication. Consequently, disrupting this communication channel can significantly disrupt malware operations, as demonstrated in several takedowns, highlighting the importance of C&C as a primary focus for detection and defensive actions. The process of identifying C&C activity within a network is a crucial initial step in the implementation of strategies, such as blocking, to reduce further damage and prevent spread. On a broader scale, actions such as sinkholing or dismantling C&C infrastructure can lead to global disruption of malware activities, causing notable difficulties for attackers. Additionally, the ability to trace the source of C&C traffic opens up the possibility not only of understanding but also of potentially penalizing attackers.

Despite their significance for network defense and malware analysis, a comprehensive taxonomy to classify and characterize malware C&C communication strategies is lacking. Although there are taxonomies that focus on subsets of malware or specific types such as botnets [1], they cannot be broadly applied to the diverse landscape of malware. This absence results in a lack of overviews on common C&C communication strategies and consequently impedes the development of new methods that support analysts and facilitate information exchange or report writing.

A related problem is the lack of readily available curated data sets that comprise representative samples for malware C&C communication strategies of prevalent families. Among the key references are the Malware Capture Facility Project [2] by Stratosphere

Lab, which offers executions of approximately 350 malware binaries, and the Malware Traffic Analysis Blog [3] by Brad Duncan, which is frequently updated with current malware campaigns and identified artifacts. However, there is no structured data set that provides a comprehensive overview or snapshot of prevalent malware C&C communication techniques. Furthermore, analyzing network behavior remains challenging, even with access to malware data sets such as Malpedia [4], because C&C infrastructures evolve rapidly and servers are often shut down.

To address these challenges, we propose a new taxonomy for malware C&C communication methods, apply it to a wide spectrum of common malware families, and release a categorized data set containing the corresponding examples with active network traffic captures. Our taxonomy builds upon the botnet taxonomy presented by Trend Micro in their report [1], but we have adjusted and expanded it to be more broadly applicable to various types of malware. Additionally, we implemented some design changes, such as distinguishing the carrier communication protocols used from the specific malware protocols employed by attackers, enabling a more precise description.

To demonstrate the applicability of our taxonomy, we compile a data set of common malware families based on the frequency of submissions to MalwareBazaar over a period of 1.5 years. The final data set includes samples from 50 families, accounting for more than 83% of all submissions to MalwareBazaar. To facilitate the analysis of real networking traffic, even if servers are shut down or infrastructures are altered, we supplemented each sample with a manually validated sandbox run that includes a network traffic capture with (at the time) live traffic. We subsequently classified the networking strategies and methods of all included malware families according to our taxonomy, providing a snapshot of malware C&C communication behavior at the time of writing. Finally, we publish the data set to facilitate research in malware network analysis [5]. In conclusion, this paper offers the following three main contributions:

- **Contribution A**: Proposing a novel taxonomy for the categorization and classification of malware C&C communication techniques.
- **Contribution B**: Providing a taxonomic summary of the existing landscape of malware C&C communication strategies.
- **Contribution C**: Releasing a data set that includes samples from common malware families along with live network traffic captures.

The remainder of this paper is structured as follows. First, we present a brief summary of the relevant related work in Section 2. Then, the main body of the paper is split into two sections: Section 3 describes a taxonomy to classify malware C&C communication strategies, while Section 4 offers an overview of recently active malware families along with their categorized network behaviors according to the taxonomy. Finally, the paper concludes with Section 5, which covers the summary of the paper.

2 Related Work

In this section, we briefly discuss the most relevant related work, particularly addressing the existing taxonomies to classify botnet behavior and networking on which we base our work.

In 2006, TrendMicro released *Taxonomy of Botnet Threats* [1]. The authors argue that threats to the Internet and botnets pose a growing security risk and require categorization; hence, they propose a botnet taxonomy aimed at improving understanding, detection, and mitigation of botnets. Although this taxonomy includes several botnet-specific elements, such as 'observable botnet activities,' it largely comprises categories relevant to both botnets and malware in general. However, in addition to focusing specifically on botnets, there are a few additional limitations when applying the taxonomy to other types of malware. In particular, their complex categorization of the malware's communication protocol with the carrier protocols utilized can significantly impede the taxonomy's use from a networking perspective.

Hachem et al. proposed another taxonomy for the classification of botnets [6]. The authors' taxonomy goes beyond only covering the C&C behavior and instead covers the entire lifecycle of botnets, including the injection & spreading mechanisms used and the applications of botnets. Concerning C&C behavior, they introduced additional important aspects, such as the type of initiation and the direction of communication. Similarly to the TrendMicro report, the authors decided not to distinguish between the carrier protocol and the protocol for actual communication. Lastly, they also mention several resilience techniques that are frequently employed by botnets.

Another taxonomy for the behavior of botnets was proposed by Khattak et al. [7]. Like earlier research, the authors aim to cover the entire lifecycle and behavior of botnets, including the infection process, botnet objectives, but also rallying and C&C communication. According to the authors, unlike previous studies, their taxonomy sheds light on the entire botnet phenomenon. In addition, the authors also introduce a second taxonomy for the defense mechanisms against botnets, primarily distinguishing preventive methods from remedial techniques.

Although not introduced in an academic publication, another highly relevant resource dates back to 2013, when MITRE first released its comprehensive Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) taxonomy [8] and has continued to maintain and expand it ever since. The taxonomy primarily covers methodologies and techniques commonly employed by Advanced Persistent Threats in their operations. MITRE ATT&CK categorizes these methods into a total of 14 categories, such as reconnaissance, persistence, or exfiltration. Of particular relevance to this work is the C&C component of MITRE ATT&CK that addresses network-based behaviors, offering extensive details on various evasion techniques.

Finally, there are numerous other works that address specific elements relevant to malware C&C behavior and are therefore cited in our taxonomy. These encompass, but are not limited to, additional botnet studies [9, 10, 11], research on techniques used in malware C&C [12], or publications about specific malware families [13]. In the subsequent sections of this paper, we also refer to multiple blog posts or websites detailing particular aspects of various malware families. Due to the vast number of these sources, we do not discuss them further here and encourage readers to consult them for information on the specific family or technique mentioned, when interested.

3 A Taxonomy for Malware Communication Strategies

In this section, we present a taxonomy for communication behaviors associated with malware. Our goal is to achieve a thorough understanding of how communication is used in malware.

3.1 Motivation

Although there have been previous proposals for taxonomies intended to classify malware communication, we found that none of them is sufficiently comprehensive to adequately classify and describe the networking of a wide range of malware families. Instead, most approaches address subsets of malware or specific types, mostly botnets, and are not universally applicable to the diverse malware landscape. This gap leads to insufficient summaries of common C&C communication techniques and hinders the creation of new methods that aid analysts and improve information sharing or documentation.

3.2 Taxonomy

The classification we present in this section is based on the taxonomy introduced by Trend Micro in their 2006 report on malware Command & Control (C&C) and botnet threats [1]. The original report identified six primary areas: attacking behavior, Command & Control, rallying mechanisms, communication protocols, evasion techniques, and observables.

However, since the Trend Micro report focuses mainly on the classification of networking within botnets, it has some limitations regarding broader usage. To address these limitations, we propose the following changes. First, we suggest a more nuanced distinction between the communication protocol used to transmit C&C messages and the internal C&C protocol used by malware. Second, we incorporate the aspect of communication direction [6] to improve the characterization of evasion mechanisms. Since our taxonomy aims to be universally applicable to malware, we omit the botnet-specific components that were part of the original report.

In our taxonomy, we categorize malware communication by the following criteria: The classification of the C&C model (refer to Section 3.2.1), the differentiation of various rallying mechanisms (refer to Section 3.2.2), the general communication behavior (refer to Section 3.2.3), the deployment of carrier communication protocols (refer to Section 3.2.4), and the specific communication protocol itself (refer to Section 3.2.5). Table 1 summarizes the taxonomy and gives a quick view of each category and the possible subcategories and/or method types. In the following sections, we describe each category, potential subcategories, and their detailed functioning.

3.2.1 C&C Models

Initially, we identify various *C&C models*, or topologies, which are frequently used by malware. Basically, this serves as an abstract classification of C&C communication, highlighting the structure of the overlay network that includes bots, botmasters, and C&C servers. Understanding the C&C models is crucial for implementing countermeasures such as disrupting botnet communication channels and reducing damage.

We categorize the C&C models into *Centralized*, *Peer-to-Peer*, *Hybrid*, and *Randomized*, each of which we will describe in more detail now. Additionally, we also discuss the possibility of employing *non-network* based channels for communications and the resulting impact on the remaining taxonomy categories.

3.2.1.1 Centralized C&C Models

In a *Centralized C&C Model*, the adversary controls a centralized infrastructure to manage communication. This infrastructure often takes the form of a C&C server that acts as the central point of contact for all compromised machines. It provides numerous benefits such as ease of use, minimal overhead, concurrent bot communication, and the ability for real-time monitoring and feedback [6, 7, 1, 9].

We further differentiate between a *Star* and a *Hierarchical* model. In a Star model, compromised devices connect directly to a single C&C server for command transmission, providing simplicity and low latency but vulnerable to takedowns as it presents a single point of failure [6, 7]. The hierarchical structure, similar to the Star model but with multiple servers, enhances scalability, reliability, and performance. These hierarchical setups commonly use proxies to ensure anonymity and lower risk of takedowns, though they come with increased latency and complexity compared to the Star model [6, 7, 9, 14].

3.2.1.2 Peer-to-Peer C&C Models

In a *Peer-to-Peer C&C Model*, compromised devices (peers) exchange messages, forward instructions, and distribute updates among a limited set of known peers. Commands from the attacker to a small number of peers are spread throughout the network. This distributed structure ensures there is no single point of failure, increasing robustness against control mecha-

3.2.1 - **C&C Models**	Centralized C&C Models: *Star* or *Hierarchical* Peer-to-Peer C&C Models Hybrid C&C Models Randomized C&C Models *Non-Network Based*
3.2.2 - **Rally Mechanisms**	IP-Address Rallying Domain Name Rallying Further Indirection: e.g. Benign Web-Services, Email-Addresses
3.2.3 - **Communication Behavior**	Command Transmission: *Push* or *Pull* Communication Session: *Unidirectional* or *Bidirectional*
3.2.4 - **Carrier Communication Protocols**	Raw TCP/UDP Socket Raw SSL/TLS Socket Application Layer Protocols: e.g. HTTP(S), IRC Web Services: E.g. Twitter/X or Dropbox
3.2.5 - **C&C Protocols**	Central Communication Protocol, either *Binary* or *Text*-based
3.2.6 - **Evasion Techniques**	e.g. Fallback Channels, Proxy Servers, Stepping Stones

Table 1: This table outlines our taxonomy for classify communication behaviors of malware, consisting of six categories that are discussed in detail in the relevant section.

nisms. However, the use of such decentralized communication can increase latency due to the time taken for messages to pass through multiple peers. Furthermore, this complicates the detection and mitigation of such operations due to their distributed nature [6, 7, 1].

3.2.1.3 Hybrid C&C Models

In addition to centralized and peer-to-peer models, attackers can also employ a hybrid approach. By combining centralized and decentralized structures with distinct roles for compromised machines and multiple protocols for peer list retrieval, internode communication, and C&C communication [7, 15].

3.2.1.4 Randomized C&C Models

Lastly, another option is the use of a *Randomized C&C Model*. In this model, compromised devices wait for incoming connections while the attacker performs network scanning to identify and control these devices. This can create either a chain-like topology, where commands are passed between compromised devices, or a *reverse* star topology, with the attacker directly scanning for devices. Although it has not been observed in the real world to date, it is noted for its stealth and robustness, despite the considerable time required to locate compromised devices [10, 7, 1].

3.2.1.5 Non-Network Based

Apart from the network-based communication channels and models primarily discussed in this paper, there are alternative methods, such as air-gap scenarios, which could be specifically addressed by malware [16]. Generally, communication via offline channels necessitates a different approach from a developer's perspective in comparison to online channels. For example, USB thumb drives could be used to transfer data between compromised machines and/or the threat actor. Another similar technique applicable in online/network-based environments includes dead drops, where a compromised device dispatches marked messages to or retrieves them from content file/hosting sites before being collected or consumed asynchronously. In these cases, the model reverses the strategy we labeled as *randomized*, where the devices need to communicate outward without choice, rather than awaiting contact.

The following taxonomy categories are generally centered on network-based communication methods and may not seem to fit perfectly to such alternate scenarios. Nevertheless, we argue that the above-mentioned situations can also be included: Firstly, we assume rallying functions through structural conventions, naming conventions, or markers within files instead of IPs or domains. Secondly, we consider that the transmission medium in both scenarios is passive (e.g., a file as opposed to a network protocol) or *push* in alignment with our taxonomy (cf. Section 3.2.3). Naturally, the interpretation of the carrier communication protocol depends heavily on the specific context. The remaining two categories for C&C protocols and evasion techniques remain applicable as is.

3.2.2 Rally Mechanisms

Rallying encompasses the actions taken by a compromised machine to establish an initial C&C channel, playing a vital role in its operation. When analyzing a malware sample, this process can reveal crucial details such as IP addresses and domains, helping to block C&C communication and potentially uncovering clues about an adversary's network. We categorize rallying primarily into *IP address Rallying* and *Domain Name Rallying*. Additionally, we explore a few potential strategies to introduce further indirections during the rallying process.

3.2.2.1 IP Address Rallying

During the process of *IP address Rallying*, the compromised machine uses one or more IP addresses to establish a connection. These IP addresses can be embedded directly into the binary or provided during infection and stored elsewhere on the system [7].

Embedding (or *hardcoding*) IP addresses directly into the malicious binary has been a prevalent technique historically, particularly in centralized C&C models. This approach minimizes DNS traffic, providing stealth, but poses the risk of immediate server discovery and blacklisting, which could result in loss of control over compromised machines [7, 1]. Conversely, delivering IP addresses to a compromised machine at the point of infection, hiding them within the compromised machines (such as in the Windows registry), allows for later updates. This strategy is used predominantly in P2P botnets to establish initial peer lists [7].

3.2.2.2 Domain Name Rallying

Another method used by attackers in addition to the IP address Rallying is *Domain Name Rallying*. In this technique, the compromised machine uses one or more domain names that need to be resolved to IP addresses to establish a connection. Attackers often dynamically alter the server's IP address, thus strengthening the resilience of the C&C infrastructure against deactivation. Efforts to mitigate this, such as blocking or taking down malicious domain names, vary in effectiveness based on the DNS service provider's cooperation. These efforts are further complicated when attackers use rogue DNS services that resist takedown requests [7, 1, 9].

Similarly to IP addresses, domain names can be embedded (or *hardcoded*) directly in the malicious binary. Although the C&C infrastructure can remain functional despite blocking IP addresses, domain names are vulnerable to blocking or deregistration [7]. On the other hand, domains can also be produced algorithmically through *Domain Generation Algorithms (DGAs)*. This approach allows for the creation of new domain names at a faster rate than they can be blocked or deregistered, such as on a daily basis, to stay ahead of countermeasures. These algorithms, a shared secret between bots and botmasters, may use indicators such as timestamps or public online data to generate seeds, making it more difficult to block domains before the initial activation. For botmasters, registering a single domain is enough to maintain control, while the challenge for defenders becomes much greater when dealing with a multitude of domains [12, 7, 9, 8].

3.2.2.3 Further Indirection during Rallying

In addition to the previously mentioned rallying methods, attackers can utilize various other strategies for further indirection during rallying. For example, one advanced technique is the use of Dead Drop Resolvers, where C&C IP addresses or domains are concealed within benign web services such as Telegram or Twitter/X. In this scenario, malware extracts this information to connect with the C&C server, often employing obfuscated or encoded formats to avoid detection. This provides significant stealth, exploiting the encrypted traffic of standard web services and making binary analysis more difficult, while also facilitating easy updates to the C&C infrastructure in the event of takedowns [8]. Although we do not cover all indirection techniques here, they are included in this category.

3.2.3 Communication Behavior

Regardless of the topology used, we classify the communication behavior of C&C based on the transmission of commands and the types of communication sessions. Understanding the communication patterns of botnets is essential for analyzing and possibly replicating the communication protocol, which could eventually lead to infiltrating or dismantling a botnet.

3.2.3.1 Command transmission

In terms of how commands are transmitted, there are two main methods: *pushing* and *pulling*. The *Push Method* involves the botmaster directly sending commands to compromised devices via existing channels such as IRC or TCP connections. This approach enables real-time execution and last-minute directives, but is less covert and requires constant channels to all compromised devices [6]. In contrast, the *Pull Method* requires the bot to periodically contact the C&C infrastructure to obtain new commands according to a schedule or interval defined by the botmaster. These solicitations may target an attacker-controlled infrastructure or legitimate web services used by the botmaster to store commands for later retrieval [6].

3.2.3.2 Communication Session

Regarding the type of communication session, we distinguish between uni- and bidirectional communication. As the term implies, in *Unidirectional Communication*, compromised machines receive instructions without sending a reply through the same channel, either not responding at all or using an alternative channel for any feedback [6, 8]. Conversely, *Bidirectional Communication* occurs over the same channel between the bot and the C&C server, allowing the botmaster to receive status updates from the bot and issue commands [6, 8]. Although easy to implement for certain C&C infrastructures, bidirectional communication can increase in complexity when using legitimate services to avoid detection, i.e. bots responding through unconventional channels such as forum comments, tweets, or updates on shared web documents.

3.2.4 Carrier Communcation Protocols

Malware authors often use C&C channels built with multiple layers of standard network protocols, where the deepest layer is usually highly specialized. In this category, we focus on the protocols that handle these specialized data. We make a distinction between raw TCP/UDP or SSL/TLS sockets, application layer protocols, and web services.

3.2.4.1 Raw TCP/UDP Socket

When employing a *Raw TCP/UDP Socket*, there is instant communication via a minimal overhead network channel, often requiring complete design and formulation of the internal C&C protocol from the ground up [11]. Depending on the transmitted data, this type of socket can be efficient while challenging to analyze.

3.2.4.2 Raw SSL/TLS Socket

Employing a *Raw SSL/TLS Socket* operates much like the aforementioned raw sockets; however, it encases the connection in an encryption layer via an SSL/TLS session. This encryption naturally bolsters security against potential attackers and complicates multiple defensive measures.

3.2.4.3 Application Layer Protocols

A common technique in modern malware is the usage of *Application Layer Protocols*. Employing the application layer for C&C communications enables the traffic to blend with regular network activity, thereby avoiding detection and filtering by capitalizing on widely permitted ports and protocols, while profiting from the availability of stable and well-established software. Especially HTTP and HTTPS, being the core components of web traffic, are especially useful for this purpose, aligning smoothly with normal traffic patterns [6, 7, 8].

3.2.4.4 Web Services

To evade blocklisting and other countermeasures while blending in, attackers might use well-known *Web Services* for C&C communication. Using popular platforms such as Telegram, Twitter/X, or Dropbox offers considerable concealment, as traffic to these sites is typically whitelisted, encrypted, and prevalent, thus obscuring the actual source of C&C traffic and improving the robustness of the infrastructure [8]. As a result, this strategy can make it much harder for security teams to detect and counteract these activities.

3.2.5 C&C Protocols

As noted in the previous categories, malware often employs C&C channels built on multiple layers of conventional network protocols. Within these, the central communication protocol, which translates raw bytes or strings into data and commands, functions independently from the underlying transmission carrier protocols. This C&C protocol typically highly specialized, as it has to be tailored to the capabilities and functioning of the specific family. Due to its extensive customization and significant diversity between different malware families, the C&C protocols are not further classified beyond binary versus text.

3.2.6 Evasion Techniques

Modern malware may use different *Evasion Techniques* during the execution of network operations. In the following, we provide a non-exhaustive list of example methods.

3.2.6.1 Fallback Channel

Through the use of backup or alternative forms of C&C, known as *Fallback Channel*, an attacker can potentially (re)establish control if the main communication channel is breached. Furthermore, data transfer restrictions can also be bypassed by fallback channels [6, 8].

3.2.6.2 Proxy Servers

Another evasion tactic that attackers can use is employing *Proxy Servers*. Proxies can be used internally to reduce outgoing connections and externally to conceal the origin of C&C traffic, thus improving anonymity [8].

3.2.6.3 Stepping Stones

In order to safeguard their anonymity, attackers might employ *Stepping Stones* in the network infrastructure of their malware. By placing proxies or SSH servers between themselves and their C&C servers, they avoid direct connections, thus hindering tracking efforts if C&C servers come under surveillance or are breached [7].

3.2.6.4 (Double) Fast-Flux

Much like round-robin DNS, attackers might use *Fast-Flux* by frequently changing the A-record of their domains to redirect to various proxies (which might be compromised machines). Using *Double Fast-Flux*, they additionally alter the NS-record of their domain, adding another layer of obscurity through frequent changes to the nameservers in use (which can also be compromised machines). This strategy enables attackers not only to hide their C&C servers but also to make individual proxy takedowns ineffective. [8, 9].

3.2.6.5 DNS Calculation

Rather than just utilizing the IP addresses provided by the DNS servers in response to queries, attackers might perform a form of *DNS Calculation* on these responses to derive the real IP and port of the C&C server. This approach renders filtering or sinkholing ineffective unless the algorithms used are first reversed and reimplemented [8].

3.2.6.6 Encryption

To bypass content-based detection, malware frequently employs *Encryption* for its C&C communication, requiring reverse engineering to identify the encryption algorithm and keys. Although encryption is advantageous in hiding malicious activities, it also presents a drawback for attackers; the use and reuse of keys or certificates can be monitored, helping researchers and authorities link attacks and attribute samples [8].

3.2.6.7 Data-Obfuscation

To make the detection, decryption, and filtering of C&C traffic more challenging, attackers might employ different *data obfuscation* methods. Such methods include, but are not limited to, junk data, encoding, steganography, and protocol impersonation [8].

3.2.6.8 Non-Standard Ports

Particularly with application layer protocols, adversaries might use *Non-Standard Ports* for their C&C traffic, such as using a port other than 443 for HTTPS. This tactic may allow them to bypass network filtering and make network traffic analysis more challenging [8].

3.2.6.9 Traffic Pattern Manipulation

By employing *Traffic Pattern Manipulation*, such as modifying traffic characteristics, attackers aim to avoid being detected by security systems that rely on statistical analysis of the timing and volume of network traffic. They might achieve this, for example, by reducing the frequency of communication or spreading it over time, synchronizing with office hours or typical traffic patterns, generating additional legitimate-looking traffic, or varying the check-in intervals of infected devices [6, 7].

4 Application of the Taxonomy

In this section, we now examine the applicability of the taxonomy as defined in the previous section by using it on a variety of examples. For this, we conducted an evaluation on a set of 50 malware families prevalent at the time of writing.

4.1 Data Set

To determine prevalent malware families, we use the frequency of occurrence for submissions to MalwareBazaar [111], a malware exchange platform especially popular among practitioners. For our experiment, we chose a 1.5 year period and set the boundaries for consideration to September 1, 2022, and February 29, 2024. During this time, there were 205,437 submissions to MalwareBazaar, of which 177,757 have been assigned one of 862 observed signature values, designating their suspected identified malware family. For our data set, we choose 50 malware families, focusing on the most commonly observed ones, which after deduplication of signatures add up to a representation of 147,652 samples (83.06%).

In order to properly evaluate the taxonomy, we need reliable information about malware C&C communication for these families. To achieve this, we manually reviewed more than 2,500 sandbox runs by Recorded Future Triage [112], with the goal of identifying executions in which a reference malware sample for each family was able to reach their C&C servers. As Triage records network traffic capture files in the PCAPNG format and applies Transport Layer Security (TLS) decryption where possible [113], this serves as a great foundation for our further evaluation.

The data set resulting from the selection procedure is shown in Table 2. In addition to the family and reference samples with the chosen sandbox run, the table lists additional references that describe the respective malware families' C&C protocols.

4.2 Evaluation

For the purpose of evaluation, we apply our taxonomy for all of the collected malware families and discuss the aggregated results per taxonomy dimension. To derive the features, we manually analyzed all PCAP files and to report the observations as found in the reference samples. Because we only cover one sample and instance per malware family, this carries the possibility of other botnet instances having different parameters (e.g. IP addresses instead of domain names for rendezvous). Nevertheless, our results should provide a comprehensive overview of current C&C strategies used in current malware.

Despite our extensive efforts to find adequate reference samples and sandbox runs for each of the considered families, we had to make compromises in seven cases where network streams are considered impure and/or incomplete. Specifically, for the five families `win.bazarbackdoor`, `win.dbatloader`, `win.photoloader`, `win.privateloader`, and finally `win.smokeloader` we were only able to find sandbox runs with live C&C response traffic that also led to the download and execution of a follow-up payload. Consequently, we consider those PCAP files as "impure" because they have network traffic artifacts from another malware family after the target communications. However, these files remain useful for examining live C&C traffic for the specified families, provided that the analyst carefully excludes other traffic.

Unfortunately, for the two families `win.formbook` and `win.lokipws` we were unable to find sandbox runs with responsive C&C servers. Consequently, the respective PCAPs in our data set contain only outward traffic from the bot.

4.2.1 C&C Model

With respect to the C&C model employed, we note that all 50 malware families analyzed use a centralized model. In some cases, it is known due to takedowns or introspection capabilities beyond analysis of PCAPs that their botnet instances have a hierarchical structure, as e.g. was documented for `win.emotet`, `win.qakbot`, and `win.redline_stealer`.

4.2.2 Rally Mechanism

Looking at the rally mechanisms used, we note that almost all families use hardcoded information. We observe 23 cases in which IP addresses have been used and 19 cases in which domains are stored in the configuration. All of these exclusively use only one of the two variants, while the instance of `win.lokipws` has a configuration with an IP address as well as four hardcoded domains. Four of the malware families that exhibit primarily information stealing capabilities are using email as their exfiltration medium of choice, thus having mail account credentials and/or addresses as rally mechanism: `win.404keylogger`, `win.agent_tesla`, `win.hawkeye_keylogger`, and also

ID	Family	References	SHA256	Tria.ge Run	Recorded
1	elf.bashlite	[17, 18]	a7fb819c6a6ded06c2e0393e4aab272ed7ab31a06a71f753e62fc7dbed28bba6	kqz6raghbp	2024-01-23
2	elf.mirai	[19, 20]	acb573afa4ca18a7398ee49e3ccf4d485098d665c890d0818cc04b6f45f52f73	bzn1eaeb9z	2023-01-01
3	win.404keylogger	[21]	5c067b4130c714d5ac0b0acdecb9ac0a69cc5354250a514a5c78c194de8e5f52	shv6tseh85	2023-06-01
4	win.agent_tesla	[22, 23, 24]	21d0345174d67986202fdecdf8e56493628d9e66eafdf4002a8dacb84c46d779	hs7x9sda7z	2023-05-17
5	win.amadey	[25]	919ae827ff59fcbe3dbaea9e62855a4d27690818189f696cfb5916a88c823226	2dyktscg4z	2023-11-26
6	win.asyncrat	[26]	8ced69a3c6796be12a0433300b9935b4c63fe4817b0830e1965b07fffaa360df	anbp9sec37	2023-05-19
7	win.aurora_stealer	[27]	53274ab4f9cebd26058061cd944614586a086d91cd9f36b679e3c8dccae84a7d	t2ylqsaf47	2023-04-14
8	win.ave_maria	[28]	90ebeaa9a68ea0c1bf9aff1f7902d545fd5623af7aba90d8cbc53ece47f43f51	k2qrfsfc57	2023-05-22
9	win.azorult	[29]	fae283e4b242564d01fa9dd0f3014c9742c2529730c988c3774a136e6941f6da	mtds2aeecl	2024-01-22
10	win.bazarbackdoor	[30, 31, 32]	e198fa82dcb7b38b544c5d0c5357c9e22fb905566eda64114200a77fabd9e756	qfg53scb81	2023-11-24
11	win.brute_ratel_c4	[33, 34]	76ef1aab335fd34f9fbaf8c2355a35b3425d52da9eea1e32f450eb9a131eda53	nh5m2sccgl	2022-11-01
12	win.bumblebee	[35, 36]	06215d5a571e4cc6a89bd977f4a7a76a03e4aa99bfe73c2cccf65ff3df747957	ypsbeaba21	2023-09-05
13	win.cloudeye	[37, 38]	3665d429fa491a95b0a14e115e8aff0fa4051830775abbdce9a9b8ce85ba3c59	tnhy5aga8s	2023-10-18
14	win.cobalt_strike	[39]	40ab463703114d972269c34abeecf0f796c88c20cceaaf0e582ed0a132e556fa	xvxyeaba76	2022-12-08
15	win.cryptbot	[40, 41]	d12aaff0ee692d96e72cedfa2c59381d592af36bd70a715527463d66753866c1	x9sl8sfddl	2024-02-01
16	win.darkcomet	[13]	3f0b1837b836c8f882db35bc5b0510b47e6c06a996148371a05b2d2b8b46ee0f	tmxqwsbg33	2022-12-25
17	win.darkgate	[42, 43]	14ca3dc03e22d3794c7f7c133426128ad4f4fdcc907febcf3954e53b1acb1e9f	cb3pbsdebk	2024-02-02
18	win.dbatloader	[44, 45]	718d001bfe38d355ed67211848bb5170ed326cf9d8fee73e451b6d2c7357ed62	hn9lascagr	2024-02-05
19	win.dcrat	[46, 47]	387d5d85440c9e1db680894c297350c26145221e3986ed01bd3b5bddd8cb923e	ftw62shhcm	2024-02-06
20	win.emotet	[48, 49, 50]	fd79e8fa5e3801101a1305b6aba7a5e7fdc852ed9036d6d9a5210be414a5cc5a	kya7lafc68	2023-03-17
21	win.formbook	[51, 52]	b73d4f7b02d7a5900e49757c613a2ecff0c7583201af7d8f9047d64afbbee4e4	whf9fshd84	2024-03-07
22	win.gcleaner	[53]	4636d352e5e5f0bce2ee610076bbe49bff0afefb7ab7b84411ed7ba1dd7df982	ybe4xsag3t	2023-08-02
23	win.glupteba	[54, 55, 56]	4c82c0295e518e75e2375e6bf7dcca3b8eae9a519b6a02a3a12e7a997bf3a861	hkb6mahgh6	2024-02-06
24	win.hawkeye_keylogger	[57, 58]	88b19a54e96629881f26ac18f867e1452d76ea240d1aaf6dd9a31c49a75e1a4c	tgggqabgg7	2024-01-12
25	win.isfb	[59, 60, 61]	858d867cc62c0bf13b16ccdb9f6cd6022d61fc2ab98a7db60806a35c7da9b2e0	rfj54sfe53	2023-10-12
26	win.lokipws	[62, 63]	7b36ba68604194c04691a704eecef41ed011a7c3f36054948a2b3a6d34276e4d	jly78ada91	2023-03-30
27	win.lumma	[64, 65]	ec28b4aef6b5a01a46e533c6123f4e2350dd6b7da1a49d89bca8f5f689986861	emta5sehem	2024-01-26
28	win.masslogger	[66, 67]	9269aa26ade2e9c4bff249a124c2f6c5a4ee0995045e01a15df8e60fcbbe3188	1bjhfseh6v	2023-01-29
29	win.nanocore	[68]	8f76e7c72e17763b750529f776a68ad05e86a6dd3a06374b2acd8d7e4821374b	mq5gksae2y	2023-05-22
30	win.netwire	[69, 70]	e9b94cae938b5549cc9645b0e15337bc0ff894b9413305351937cf7831347d85	m99kmsba88	2023-02-22
31	win.njrat	[71]	604d490b9d5dfff01c9fceb085798c6b42f5778c9f125457be654dc4f436ab04	tqbmtsdcd8	2024-02-03
32	win.panda_stealer	[72]	536a23943567e447c6a279127cc1a05851898fe966f34987228c881a7c8f88ae	l6v3eaccg5	2023-12-31
33	win.phorpiex	[73, 74, 75]	b97040f0f3a2a5158354af553886e7d7737a5640cb32506b50e7bbb3a767e61d	ly5xmsdfg8	2024-02-06
34	win.photoloader	[76, 77, 78]	3c52f7fc9fb2ce7c10f20f0eeee60b093133b497b68c9f87b627ae4664aa56b1	tjfm7shc94	2023-10-18
35	win.pikabot	[79, 80, 81]	01193e9155a79c013ea3ba070142a3ccca0c96cde502a302c37e375f18d38002	l7snnsee2x	2023-11-21
36	win.privateloader	[82, 83]	0baa2d4181de6ef73fefab4d83926e92e20bea4ea2401f79b9bb0014a63d559f	znqa4adb46	2023-09-25
37	win.qakbot	[84, 85, 86]	9f7eb4e5382c19d8e3ecbc71f13aaf467df4f61a835fee78ca199bdd3b7178ca	q3y24she24	2022-11-18
38	win.quasar_rat	[87]	15d27a6eed07fe1747835df49e88fd803cdb0518761a4eafc11c694b5a87ecb9	gp787abef4	2024-01-29
39	win.recordbreaker	[88, 89, 90]	b6c63317d884b7fef58431c1ca61fd85438246a23c3e6920b3584741cc455846	l3eagsbf5w	2023-03-21
40	win.redline_stealer	[91, 92]	4c77f0a6c86f534485f28c73963f05dcfb03818b8b92852b54a56e7a4d70fe8f	lyck4acc45	2023-05-16
41	win.remcos	[93]	7a1bb4fe0f62425fdd2e163ea17d84465323c4f2df8aabb8a50b1433e7d42a9f	xw29hsaf2w	2023-05-05
42	win.rhadamanthys	[94, 95]	f0f70c6ba7dcb338794ee0034250f5f98fc6bddea0922495af863421baf4735f	yrm5gaec6v	2023-07-18
43	win.sectop_rat	[96, 97]	9133b62cf224ab836d86d3aff622629e91730a557ade8fc281261a9f49e7b319	gjkkyabhh5	2024-02-05
44	win.smokeloader	[98, 99]	b32acda2bfe8426594405da486f6d0adc46382b3ba7b3b715e7d3a356d0733c9	fa4f1sgdd3	2023-12-26
45	win.socks5_systemz	[100]	00dc90ea7c9b2541b73cfb3cb8893b8cfc54ba4c578db4975a584c27c924368a	q6dkkshbh7	2024-02-05
46	win.stealc	[101]	ff7167a57007b5368518b03e82ae9110aa55ba9e947719308a416f3aecdc75e1	rsefdsbgc9	2024-02-04
47	win.tofsee	[102, 103, 104]	ef61c941b8ba327d3808655851ef8b13b4c8e3e572a80daa28fb8284c5ce65fe	x4cntsebal	2024-01-26
48	win.vidar	[105, 106]	09fa4342b6a984a007d4ecbb233731aa59c5c8e58b6cb16a6cedb3ae63be0387	rykjlaac2v	2023-05-10
49	win.wikiloader	[107, 108]	34f5759b0e7604dfcc245643aabdac6e24deb01976ee6b0a33b56ac50569c31f	qvgsvsgdel	2024-02-01
50	win.xworm	[109, 110]	e4604c8555fd9ac08b00a5c51795657b9a339fa07ef5e6c64e9f54fe79d28b42	1e49bsagdl	2024-02-05

Table 2: This table presents all data set samples, with one representative per Malpedia family [4]. Each sample includes references that describe the families' C&C communication, its hash, and a Tria.ge run showing a capture of live C&C communication traffic.

win.masslogger. Two instances for the considered malware families, win.dcrat, and win.vidar, offer an additional degree of flexibility by introducing indirect rallying, using the benign web services PasteBin and Telegram to relay the actual C&C information. Although it has been a popular rallying mechanism in the past [114], only one malware family uses a Domain Generation Algorithm (DGA) as its primary rallying mechanism. Starting in 2023, win.bumblebee introduced this feature as its main rally mechanism.

4.2.3 Communication Behavior

Considering communication behavior, we again notice a strong tendency to use a bidirectional pull mechanism, as 41 families have their instances reach out to C&C first and respond to these requests. Another three families invert this relationship and instead use a bidirectional push method (win.aurora_stealer, win.ave_maria, and win.tofsee).

When looking at unidirectional communication patterns, we note that two loaders pull malicious content from benign web services (win.cloudeye, and win.dbatloader), naturally implying that there is no C&C back channel. Otherwise, the four malware families using email for exfiltration (cf. Section 4.2.2) are also considered to use unidirectional push behavior.

4.2.4 Carrier Communication Protocol

Despite the general trend of widely adopting SSL/TLS in recent years [115], we observed 29 (58%) instances in which the transport layer is not secured using these protocols. Another 17 (34%) do make use of SSL/TLS.

Among these 46 malware families are 13 that use a raw TCP connection, implementing their own protocol completely from scratch. In contrast, the other 23 families build their protocol on top of HTTP. By this, they likely try to benefit from the popularity of this application layer protocol in an attempt to blend in.

ID	Family	Model	Rallying	Transm.	Carrier Protocols	Port	Protocol	
1	elf.bashlite	⁎	IPs	⇄ Pull	Raw TCP/UDP	8722	Text	🔓
2	elf.mirai	⁎	IPs	⇄ Pull	Raw TCP/UDP	6666	Binary	🔒
3	win.404keylogger	⁎	Email-Address	→ Push	SMTP	587	Text	🔓
4	win.agent_tesla	⁎	Email-Address	→ Push	SMTP	587	*Carrier-Protected*	
5	win.amadey	⁎	IPs	⇄ Pull	HTTP	80	Unknown	🔒
6	win.asyncrat	⁎	IPs	⇄ Pull	HTTPS	9999	*Carrier-Protected*	
7	win.aurora_stealer	⁎	IPs	⇄ Push	Raw TCP/UDP	8081	Text	🔓
8	win.ave_maria	⁎	IPs	⇄ Push	Raw TCP/UDP	5200	Unknown	🔒
9	win.azorult	⁎	Domains	⇄ Pull	HTTP	443, 80	Unknown	🔒
10	win.bazarbackdoor	⁎	IPs, DGA	⇄ Pull	HTTPS	443, 9001	*Carrier-Protected*	
11	win.brute_ratel_c4	⁎	IPs	⇄ Pull	HTTPS	443	*Carrier-Protected*	
12	win.bumblebee	⁎	DGA	⇄ Pull	HTTPS	443	*Carrier-Protected*	
13	win.cloudeye	⁎	URIs	→ Pull	GoogleDocs	443	*Carrier-Protected*	
14	win.cobalt_strike	⁎	Domains	⇄ Pull	HTTPS	443	*Carrier-Protected*	
15	win.cryptbot	⁎	Domains	⇄ Pull	HTTP	80	Binary	🔓
16	win.darkcomet	⁎	IPs, Domains	⇄ Pull	Raw TCP/UDP	1604	Unknown	🔒
17	win.darkgate	⁎	Domains	⇄ Pull	HTTP	80	*Carrier-Protected*	
18	win.dbatloader	⁎	URIs	→ Pull	OneDrive	443	Unknown	🔒
19	win.dcrat	⁎	URIs	⇄ Pull	HTTP	80	Text	🔓
20	win.emotet	⁎	IPs	⇄ Pull	HTTPS	443, 80	Unknown	🔒
21	win.formbook	⁎	IPs	⇄ Pull	HTTP	80	Unknown	🔒
22	win.gcleaner	⁎	IPs	⇄ Pull	HTTP	80	Unknown	🔒
23	win.glupteba	⁎	IPs	⇄ Pull	HTTPS	443	*Carrier-Protected*	
24	win.hawkeye_keylogger	⁎	Email-Address	→ Push	SMTP	587	*Carrier-Protected*	
25	win.isfb	⁎	Domains	⇄ Pull	HTTP	80	Unknown	🔒
26	win.lokipws	⁎	IPs, Domains	⇄ Pull	HTTP	80	Binary	🔓
27	win.lumma	⁎	Domains	⇄ Pull	HTTPS	443	Text	🔓
28	win.masslogger	⁎	Email-Address	→ Push	SMTP	587	*Carrier-Protected*	
29	win.nanocore	⁎	Domains	⇄ Pull	Raw TCP/UDP	5899	Unknown	🔒
30	win.netwire	⁎	Domains	⇄ Pull	Raw TCP/UDP	3102	Unknown	🔒
31	win.njrat	⁎	Domains	⇄ Pull	Raw TCP/UDP	13538	Unknown	🔒
32	win.panda_stealer	⁎	Domains	⇄ Pull	HTTP	80	Binary	🔓
33	win.phorpiex	⁎	IPs	⇄ Pull	HTTP	80	Unknown	🔒
34	win.photoloader	⁎	Domains	⇄ Pull	HTTP	80	Text	🔓
35	win.pikabot	⁎	IPs	⇄ Pull	Raw SSL/TLS	2967	*Carrier-Protected*	
36	win.privateloader	⁎	IPs	⇄ Pull	HTTP	27323	Binary	🔓
37	win.qakbot	⁎	IPs	⇄ Pull	HTTPS	443, 2222 (+5 more)	Unknown	🔒
38	win.quasar_rat	⁎	IPs	⇄ Pull	Raw SSL/TLS	4412	*Carrier-Protected*	
39	win.recordbreaker	⁎	IPs	⇄ Pull	HTTP	80	Text	🔓
40	win.redline_stealer	⁎	IPs	⇄ Pull	Raw TCP/UDP	4132	Binary	🔓
41	win.remcos	⁎	Domains	⇄ Pull	Raw SSL/TLS	2718	*Carrier-Protected*	
42	win.rhadamanthys	⁎	Domains	⇄ Pull	HTTPS	443	*Carrier-Protected*	
43	win.sectop_rat	⁎	IPs	⇄ Pull	Raw TCP/UDP	15647	Text	🔒
44	win.smokeloader	⁎	Domains	⇄ Pull	HTTP	80	Unknown	🔒
45	win.socks5_systemz	⁎	IPs	⇄ Pull	HTTP	80	Unknown	🔒
46	win.stealc	⁎	IPs	⇄ Pull	HTTP	80	Text	🔓
47	win.tofsee	⁎	Domains	⇄ Push	HTTPS	443	*Carrier-Protected*	
48	win.vidar	⁎	URIs	⇄ Pull	HTTP	9100	Text	🔓
49	win.wikiloader	⁎	Domains	⇄ Pull	HTTPS	443	*Carrier-Protected*	
50	win.xworm	⁎	IPs	⇄ Pull	Raw TCP/UDP	65030	Unknown	🔒

Legend: ⁎ Centralized C&C Model, ⇄ Bidirectional Transmission, → Unidirectional Transmission, 🔒|🔓 (Un)Encrypted

Table 3: This table illustrates the C&C communication strategies that we observed in the network captures corresponding to the family samples outlined in Table 2, according to our taxonomy introduced in Section 3. It is important to note that the listed communication behavior may not fully represent *all* the strategies implemented by each family (see Section 4 for more details).

As already mentioned above, four families use email as their exfiltration method. Consequently, their direct carrier protocol can be considered SMTP. More specifically, we observed three of them using TLS for the establishment of the session and one using a plain session (`win.404keylogger`).

4.2.5 C&C Protocol

Looking at the C&C protocol itself, we notice that most protocols will be custom and a characteristic trait of the malware family. However, we can still categorize them depending on the design choice, specifically if they are a *text-based* or *binary* protocol.

Using this distinction, we identify 10 text-based protocols and 6 binary C&C protocols among the considered families. For another 17 malware families, we are not able to further categorize them, since they appear to be encrypted as part of the protocol. For the remaining 17 families, we cannot make any further assessment because they are already protected by SSL/TLS at the carrier protocol level (cf. Section 4.2.4).

4.2.6 Evasion Techniques

As stated above, the dimension of evasion techniques described in Section 3.2.6 lists examples of such behaviors without claiming completeness. To assess some of the mentioned aspects, insight beyond what can be obtained from network captures is required, similar to certain instances of hierarchical C&C models. With respect to the techniques listed above, we note that several families make use of three techniques, namely Encryption, Data-Obfuscation, and Non-Standard Ports. We will now briefly discuss selected examples as found in the data set.

As shown in Table 3, at least 19 families are using self-defined encrypted channels while otherwise communicating directly over their carrier protocol and not using additional carrier-level protection such as SSL/TLS. They may use this method because it can serve as an extra layer of security to make the malware's communication more difficult to detect and to enhance the channel's resistance to interception.

Furthermore, we also observed several cases where various data-obfuscation methods are used, i.e. when the communication protocol is transmitted over an existing carrier protocol. For example, `win.stealc` bots submit data as form-encoded values, while the server encodes its responses using Base64.

Finally, regarding nonstandard ports, `win.qakbot` is a good example, because it has several configured IP and port combinations that are well known but different from HTTP(S) as used by the malware.

5 Conclusion

Throughout this paper, we presented three major contributions. First, we introduced a detailed taxonomy for the categorization and classification of malware networking techniques. Unlike existing methods, our taxonomy is designed to thoroughly describe the networking behaviors of a wide range of prevalent malware, not just specific types of malware, such as botnets. By separating C&C Models, Rallying Mechanics, Communication Behaviors, and particularly the Carrier Communication Protocol from the actual C&C Protocols and various Evasion techniques, we facilitate accurate categorization and the development of new analysis methods for malware networking. It is important to acknowledge that, despite extensive efforts, some techniques and malware families might still be missing. Although we strongly believe that most future strategies should be covered by existing categories, we also acknowledge that future extensions may be necessary.

Next, we applied our taxonomy to categorize the networking behavior for a broad range of prevalent malware families, as found on MalwareBazaar. To achieve this, we collected 50 different malware families, which account for more than 83% of all submissions to MalwareBazaar during the period from September 1, 2022, to February 29, 2024. Subsequently, we manually inspected more than 2,500 sandbox executions by Recorded Future Triage *and* the corresponding PCAPs to find instances with active C&C traffic within the network capture. For the analyzed family samples, we note that all use a centralized C&C model, and distinguishing between star or hierarchical structures based on the network captures was not feasible. Although most rely on hard-coded IP addresses or domains for rallying, exceptions do exist, including families that use email addresses, URLs to benign sites like Pastebin, or a DGA. Likewise, although most families in our data set use a bidirectional pull mechanism, we also identified various other combinations of unidirectional and bidirectional push/pull mechanisms. Despite the general trend of employing SSL/TLS for malware communication, we found that most samples in our data set communicated using plain carrier protocols, but frequently encrypted the data being transferred. Finally, we note that neither the C&C protocol nor the majority of evasion strategies could be intrinsically evaluated through the network traffic captures we analyzed. Still, we observed a tendency toward text-based protocols and evasion techniques such as encryption or obfuscation.

We have made available the data set utilized for our evaluation, consisting of (unpacked) malware samples and captures of their live network communication behavior as open source on GitHub [5]. By providing this data set, we offer a snapshot of current malware networking strategies, allowing other researchers to develop future analysis methods. The development of techniques that require the capture of live network traffic from a large variety of malware families is likely to benefit significantly from this resource. However, while we consider this a valuable starting point for many new approaches, we recognize that the data set includes only a subset of prevalent malware confined to a specific time period and may become outdated more quickly than desired. Furthermore, since families can utilize multiple techniques even within a single version, network captures may not fully and accurately represent all the techniques implemented.

Acknowledgments: The authors would like to thank the anonymous reviewers of Botconf for their valuable input and their feedback. We also express our gratitude to the Malpedia community for the continued work on a data set that helped identify and collect the data set presented in this paper.

Author details

Steffen Enders

Cyber Analysis & Defense Department
Fraunhofer FKIE
Zanderstr. 5, 53177 Bonn
steffen.enders@fkie.fraunhofer.de

Daniel Plohmann

Cyber Analysis & Defense Department
Fraunhofer FKIE
Zanderstr. 5, 53177 Bonn
daniel.plohmann@fkie.fraunhofer.de

Manuel Blatt

Cyber Analysis & Defense Department
Fraunhofer FKIE
Zanderstr. 5, 53177 Bonn
manuel.blatt@fkie.fraunhofer.de

References

[1] Trend Micro, "Taxonomy of botnet threats," 2006.

[2] Stratosphere. (2015) Stratosphere laboratory datasets. [Online]. Available: https://www.stratosphereips.org/datasets-overview

[3] B. Duncan. Malware-Traffic-Analysis.net. MTA. [Online]. Available: https://www.malware-traffic-analysis.net/

[4] D. Plohmann, M. Clauss, S. Enders, and E. Padilla, "Malpedia: a collaborative effort to inventorize the malware landscape," *Proceedings of the Botconf*, 2017.

[5] S. Enders, D. Plohmann, and M. Blatt. Malware Communication Dataset. Fraunhofer FKIE. [Online]. Available: https://github.com/fkie-cad/malware_c2_dataset

[6] N. Hachem, Y. B. Mustapha, G. G. Granadillo, and H. Debar, "Botnets: lifecycle and taxonomy," in *2011 Conference on Network and Information Systems Security*. IEEE, 2011, pp. 1–8.

[7] S. Khattak, N. Ramay, K. Khan, A. Syed, and S. A. Khayam, "A taxonomy of botnet behavior, detection, and defense," *Communications Surveys & Tutorials, IEEE*, vol. 16, pp. 898–924, 01 2014.

[8] MITRE, "ATT&CK v11," https://attack.mitre.org/versions/v11/, 2022.

[9] D. Plohmann, E. Gerhards-Padilla, and F. Leder, "Botnets: Detection, measurement, disinfection & defence," *European Network and Information Security Agency (ENISA)*, vol. 1, no. 1, pp. 1–153, 2011.

[10] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets." *SRUTI*, vol. 5, pp. 6–6, 2005.

[11] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus," in *2013 8th International Conference on Malicious and Unwanted Software:" The Americas"(MALWARE)*. IEEE, 2013, pp. 116–123.

[12] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 263–278.

[13] B. Farinholt, M. Rezaeirad, P. Pearce, H. Dharmdasani, H. Yin, S. Le Blond, D. McCoy, and K. Levchenko, "To catch a ratter: Monitoring the behavior of amateur darkcomet rat operators in the wild," in *2017 IEEE symposium on Security and Privacy (SP)*. Ieee, 2017, pp. 770–787.

[14] dragos.com, "Suspected Conti Ransomware Activity in the Auto Manufacturing Sector," https://www.dragos.com/blog/industry-news/suspected-conti-ransomware-activity-in-the-auto-manufacturing-sector/, 2022, [Accessed 22-Nov-2022].

[15] AnubisLabs, "Dridex: Chasing a botnet from the inside," BitSight, Tech. Rep., 2015. [Online]. Available: https://cdn2.hubspot.net/hubfs/507516/ANB_MIR_Dridex_PRv7_final.pdf

[16] F. M. Alexis Dorais-Joncas, "Jumping the air gap: 15 years of nationstate effort," ESET Research, Tech. Rep., 2021. [Online]. Available: https://web-assets.esetstatic.com/wls/2021/12/eset_jumping_the_air_gap_wp.pdf

[17] V. Pasca, "A Detailed Analysis of the Gafgyt Malware Targeting IoT Devices," SecurityScorecard, Tech. Rep., 2022. [Online]. Available: https://securityscorecard.com/wp-content/uploads/2024/01/Report-A-Detailed-Analysis-Of-The-Gafgyt-Malware-Targeting-IoT-Devices.pdf

[18] Y. Liu, "Lightweight Emulation based IOC Extraction for Gafgyt Botnets," Qihoo 360 Technology, Tech. Rep., 2020. [Online]. Available: https://vb2020.vblocalhost.com/uploads/VB2020-Liu.pdf

[19] J. Gamblin. Mirai BotNet Source Code. Github (jgamblin). [Online]. Available: https://github.com/jgamblin/Mirai-Source-Code

[20] M. J. Erquiaga. Analysis of an IRC based Botnet. Stratosphere Lab. [Online]. Available: https://www.stratosphereips.org/blog/2019/4/12/analysis-of-a-irc-based-botnet

[21] M. Ashraf. Deep Analysis of Snake Keylogger. Github (x-junior). [Online]. Available: https://x-junior.github.io/malware%20analysis/2022/06/24/Snakekeylogger.html

[22] B. BAKARTEPE and bixploit, "Agent Tesla Technical Analysis Report," EchoCTI, Tech. Rep., 2024. [Online]. Available: https://github.com/echocti/ECHO-Reports/blob/main/Malware%20Analysis%20Report/Agent%20Tesla/Agent%20Tesla%20Technical%20Analysis%20Report.pdf

[23] B. Duncan. Agent Tesla Updates SMTP Data Exfiltration Technique. InfoSec Handlers Diary Blog. [Online]. Available: https://isc.sans.edu/diary/rss/28190

[24] G. Orlando. Malware Analysis - AgentTesla v3. [Online]. Available: https://guillaumeorlando.github.io/AgentTesla

[25] ASEC. Amadey Bot Being Distributed Through SmokeLoader. AhnLab. [Online]. Available: https://asec.ahnlab.com/en/36634/

[26] N. x CAT. AsyncRAT: Open-Source Remote Administration Tool For Windows (RAT). Github (NYAN-x-CAT). [Online]. Available: https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp/

[27] M. Adel. Aurora Stealer deep dive Analysis. d01a. [Online]. Available: https://d01a.github.io/aurora-stealer/

[28] Y. Harakhavik. Warzone: Behind the enemy lines. Check Point Research. [Online]. Available: https://research.checkpoint.com/2020/warzone-behind-the-enemy-lines/

[29] M. Henkel. Decrypting AzoRult traffic for fun and profit. Medium mariohenkel. [Online]. Available: https://mariohenkel.medium.com/decrypting-azorult-traffic-for-fun-and-profit-9f28d8638b05

[30] C. Dong. BAZARLOADER: Analysing The Main Loader. 0ffset Blog. [Online]. Available: https://www.0ffset.net/reverse-engineering/analysing-the-main-bazarloader/

[31] T. D. Report. BazarLoader to Conti Ransomware in 32 Hours. The DFIR Report. [Online]. Available: https://thedfirreport.com/2021/09/13/bazarloader-to-conti-ransomware-in-32-hours/

[32] J. Bader. Yet Another Bazar Loader DGA. Johannes Bader's Blog. [Online]. Available: https://johannesbader.ch/blog/yet-another-bazarloader-dga/

[33] A. Fortuna. How to detect Brute Ratel activities. Andrea Fortuna's Blog. [Online]. Available: https://andreafortuna.org/2023/02/23/how-to-detect-brute-ratel-activities

[34] S. T. R. Team. Deliver a Strike by Reversing a Badger: Brute Ratel Detection and Analysis. splunk. [Online]. Available: https://www.splunk.com/en_us/blog/security/deliver-a-strike-by-reversing-a-badger-brute-ratel-detection-and-analysis.html

[35] S. D. Souza, "Tracking Bumblebee's Development," Botconf, Tech. Rep., 2023. [Online]. Available: https://www.botconf.eu/wp-content/uploads/formidable/2/2023_4889_DESOUZA.pdf

[36] J. Bader. The DGA of BumbleBee. Johannes Bader's Blog. [Online]. Available: https://bin.re/blog/the-dga-of-bumblebee/

[37] A. Bleih. GuLoader Downloaded: A Look at the Latest Iteration. CyberInt. [Online]. Available: https://cyberint.com/blog/other/guloader-downloaded-a-look-at-the-latest-iteration/

[38] A. Osipov. GuLoader Campaign Targets Law Firms in the US. Morphisec. [Online]. Available: https://blog.morphisec.com/guloader-campaign-targets-law-firms-in-the-us

[39] T. Haruyama, "Knock, knock, Neo. - Active C2 Discovery Using Protocol Emulation," VMWare Carbon Black, Tech. Rep., 2021. [Online]. Available: https://jsac.jpcert.or.jp/archive/2021/pdf/JSAC2021_201_haruyama_jp.pdf

[40] A. A. A. Team. Modified CryptBot Infostealer Being Distributed. AhnLab. [Online]. Available: https://asec.ahnlab.com/en/31802/

[41] ANY.RUN. CryptBot Infostealer: Malware Analysis. ANY.RUN. [Online]. Available: https://any.run/cybersecurity-blog/cryptbot-infostealer-malware-analysis/

[42] @0xToxin. DarkGate - Threat Breakdown Journey. 0xToxin Labs. [Online]. Available: https://0xtoxin.github.io/threat%20breakdown/DarkGate-Camapign-Analysis/

[43] M. Choi. Detailed Analysis of DarkGate; Investigating new top-trend backdoor malware. S2W LAB Inc. [Online]. Available: https://medium.com/s2wblog/detailed-analysis-of-darkgate-investigating-new-top-trend-backdoor-malware-0545ecf5f606

[44] G. Palazolo. DBatLoader: Abusing Discord to Deliver Warzone RAT. Netskope. [Online]. Available: https://www.netskope.com/blog/dbatloader-abusing-discord-to-deliver-warzone-rat

[45] B. Duncan. Malspam pushes ModiLoader (DBatLoader) infection for Remcos RAT. SANS ISC. [Online]. Available: https://isc.sans.edu/diary/Malspam+pushes+ModiLoader+DBatLoader+infection+for+Remcos+RAT/29896

[46] M. H. Ali. A deep dive into DCRAT/DarkCrystalRAT malware. Github (muha2xmad). [Online]. Available: https://muha2xmad.github.io/malware-analysis/dcrat/

[47] J. Thompson. Analyzing Dark Crystal RAT, a C# backdoor. FireEye. [Online]. Available: https://www.fireeye.com/blog/threat-research/2020/05/analyzing-dark-crystal-rat-backdoor.html

[48] A. C. Silverio, J. M. Abordo, K. J. Morales, and M. E. Viray. Bruised but Not Broken: The Resurgence of the Emotet Botnet Malware. Trend Micro. [Online]. Available: https://www.trendmicro.com/en_us/research/22/e/bruised-but-not-broken--the-resurgence-of-the-emotet-botnet-malw.html

[49] d00rt. Emutet. Github (d00rt). [Online]. Available: https://github.com/d00rt/emotet_network_protocol

[50] R. Holt. How Emotet is changing tactics in response to Microsoft's tightening of Office macro security. ESET Research. [Online]. Available: https://www.welivesecurity.com/2022/06/16/how-emotet-is-changing-tactics-microsoft-tightening-office-macro-security/

[51] J. Vicente and B. Stone-Gross. Analysis of Xloader's C2 Network Encryption. Zscaler. [Online]. Available: https://www.zscaler.com/blogs/security-research/analysis-xloaders-c2-network-encryption

[52] R. Jullian, "FORMBOOK In-depth malware analysis," Botconf, Tech. Rep., 2018. [Online]. Available: https://www.botconf.eu/wp-content/uploads/2018/12/2018-R-Jullian-In-depth-Formbook-Malware-Analysis.pdf

[53] A. Elshinbary. Deep Analysis of GCleaner. N1ght-W0lf Blog. [Online]. Available: https://n1ght-w0lf.github.io/malware%20analysis/gcleaner-loader/

[54] D. Harley. TDL4 and Glupteba: Piggyback PiggyBugs. ESET Research. [Online]. Available: https://www.welivesecurity.com/2011/03/02/tdl4-and-glubteba-piggyback-piggybugs/

[55] J. Hořejší and J. C. Chen. Glupteba Campaign Hits Network Routers and Updates C&C Servers with Data from Bitcoin Transactions. Trend Micro. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/glupteba-campaign-hits-network-routers-and-updates-cc-servers-with-data-from-bitcoin-transactions/

[56] A. Brandt. Glupteba malware hides in plain sight. Sophos Labs. [Online]. Available: https://news.sophos.com/en-us/2020/06/24/glupteba-report/?cmp=30728

[57] X. Zhang. Analysis of a New HawkEye Variant. Fortinet. [Online]. Available: https://www.fortinet.com/blog/threat-research/hawkeye-malware-analysis.html

[58] I. Migdal. HawkEye Analysis. [Online]. Available: https://github.com/itaymigdal/malware-analysis-writeups/blob/main/HawkEye/HawkEye.md

[59] gbrindisi. Gozi ISFB Sourceccode. Github (gbrindisi). [Online]. Available: https://github.com/gbrindisi/malware/tree/master/windows/gozi-isfb

[60] A. Koren. Ursnif Malware: Deep Technical Dive. Ariel Koren's Blog. [Online]. Available: https://arielkoren.com/blog/2016/11/01/ursnif-malware-deep-technical-dive/

[61] B. Duncan. Wireshark Tutorial: Examining Ursnif Infections. Palo Alto Networks Unit 42. [Online]. Available: https://unit42.paloaltonetworks.com/wireshark-tutorial-examining-ursnif-infections/

[62] A. K. Sood. LokiBot: dissecting the C&C panel deployments. Virus Bulletin. [Online]. Available: https://www.virusbulletin.com/virusbulletin/2020/02/lokibot-dissecting-cc-panel-deployments/

[63] G. Pellegrino, "Deep Analysis of a Recent Lokibot Attack," Infoblox, Tech. Rep., 2021. [Online]. Available: https://www.infoblox.com/wp-content/uploads/infoblox-whitepaper-deep-analysis-of-a-recent-lokibot-attack.pdf

[64] eSentire. The Case of LummaC2 v4.0. eSentire. [Online]. Available: https://www.esentire.com/blog/the-case-of-lummac2-v4-0

[65] G. C. Security. What is Lumma Stealer? Gridinsoft. [Online]. Available: https://gridinsoft.com/spyware/lumma-stealer

[66] FR3D.HK. MassLogger - Frankenstein's Creation. FR3D.HK. [Online]. Available: https://fr3d.hk/blog/masslogger-frankenstein-s-creation

[67] A. Klopsch. Harmful Logging - Diving into MassLogger. Gdata. [Online]. Available: https://www.gdatasoftware.com/blog/2020/06/36129-harmful-logging-diving-into-masslogger

[68] J. F. NanoCore RAT Hunting Guide. Medium the_abjuri5t. [Online]. Available: https://medium.com/@the_abjuri5t/nanocore-rat-hunting-guide-cb185473c1e0

[69] CIRCL. TR-23 Analysis - NetWiredRC malware. CIRCL. [Online]. Available: https://www.circl.lu/pub/tr-23/

[70] P. D. Silva, R. Downs, and R. Olson. New Release: Decrypting NetWire C2 Traffic. Palo Alto Networks Unit 42. [Online]. Available: http://researchcenter.paloaltonetworks.com/2014/08/new-release-decrypting-netwire-c2-traffic/

[71] CyberMasterV. Just another analysis of the njRAT malware – A step-by-step approach. CYBER GEEKS All Things Infosec. [Online]. Available: https://cybergeeks.tech/just-another-analysis-of-the-njrat-malware-a-step-by-step-approach/

[72] M. de Jesus, F. Yarochkin, and P. Pajares. New Panda Stealer Targets Cryptocurrency Wallets. Trend Micro. [Online]. Available: https://www.trendmicro.com/en_us/research/21/e/new-panda-stealer-targets-cryptocurrency-wallets-.html

[73] C. Research. Phorpiex Arsenal: Part I. Checkpoint. [Online]. Available: https://research.checkpoint.com/2020/phorpiex-arsenal-part-i/

[74] A. Bukhteyev. Phorpiex Breakdown. Check Point. [Online]. Available: https://research.checkpoint.com/2019/phorpiex-breakdown/

[75] J. Bader. Phorpiex - An IRC worm. Johannes Bader Blog. [Online]. Available: https://bin.re/blog/phorpiex/

[76] S. Frankoff. PhotoLoader ICEDID. OALabs. [Online]. Available: https://research.openanalysis.net/icedid/bokbot/photoloader/config/2023/04/06/photoloader.html

[77] B. Abdo, B. McKeague, and V. Ta. So Unchill: Melting UNC2198 ICEDID to Ransomware Operations. FireEye. [Online]. Available: https://www.fireeye.com/blog/threat-research/2021/02/melting-unc2198-icedid-to-ransomware-operations.html

[78] P. Trouerbach, K. Merriman, and J. Wise. Fork in the Ice: The New Era of IcedID. Proofpoint. [Online]. Available: https://www.proofpoint.com/us/blog/threat-insight/fork-ice-new-era-icedid

[79] N. Pantazopoulos. The (D)Evolution of Pikabot. Zscaler. [Online]. Available: https://www.zscaler.com/blogs/security-research/d-evolution-pikabot

[80] T. T. Kien and m4n0w4r. [QuickNote] Technical Analysis of recent Pikabot Core Module. kienmanowar Blog. [Online]. Available: https://kienmanowar.wordpress.com/2024/01/06/quicknote-technical-analysis-of-recent-pikabot-core-module/

[81] M. Adel. Pikabot deep analysis. d01a. [Online]. Available: https://d01a.github.io/pikabot/

[82] Intel 471. PrivateLoader: The first step in many malware schemes. Intel 471. [Online]. Available: https://intel471.com/blog/privateloader-malware

[83] D. Schwarz and B. Stone-Gross. Peeking into PrivateLoader. Zscaler. [Online]. Available: https://www.zscaler.com/blogs/security-research/peeking-privateloader

[84] C. François. QBOT Malware Analysis. Elastic. [Online]. Available: https://www.elastic.co/security-labs/qbot-malware-analysis

[85] I. Kenefick, L. Silva, and N. Hernandez. Black Basta Ransomware Gang Infiltrates Networks via QAKBOT, Brute Ratel, and Cobalt Strike. Trend Micro. [Online]. Available: https://www.trendmicro.com/de_de/research/22/j/black-basta-infiltrates-networks-via-qakbot-brute-ratel-and-coba.html

[86] J. Vicente. Tracking 15 Years of Qakbot Development. Zscaler. [Online]. Available: https://www.zscaler.com/blogs/security-research/tracking-15-years-qakbot-development

[87] Embee_research. Quasar Rat Analysis - Identification of 64 Quasar Servers Using Shodan and Censys. embeeresearch. [Online]. Available: https://embee-research.ghost.io/hunting-quasar-rat-shodan

[88] A. Stratton. Raccoon Stealer v2 Malware Analysis. Infosec Writeups. [Online]. Available: https://infosecwriteups.com/raccoon-stealer-v2-malware-analysis-55cc33774ac8

[89] ANY.RUN. Raccoon Stealer 2.0 Malware analysis. ANY.RUN. [Online]. Available: https://any.run/cybersecurity-blog/raccoon-stealer-v2-malware-analysis/

[90] muzi. THE TRASH PANDA REEMERGES FROM THE DUMPSTER: RACCOON STEALER V2. MalwareBookReports. [Online]. Available: https://malwarebookreports.com/the-trash-panda-reemerges-from-the-dumpster-raccoon-stealer-v2/

[91] I. Malihi. RedLine Stealer Malware Analysis. [Online]. Available: https://medium.com/@idan_malihi/redline-stealer-malware-analysis-76506ef723ab

[92] M. Khalil. RedLine Technical Analysis Report. Apophis133. [Online]. Available: https://web.archive.org/web/20230606224056/https://apophis133.medium.com/redline-technical-analysis-report-5034e16ad152

[93] X. Zhang and C. Navarrete. New Variant of Remcos RAT Observed In the Wild. Fortinet. [Online]. Available: https://www.fortinet.com/blog/threat-research/new-variant-of-remcos-rat-observed-in-the-wild.html

[94] B. BAKARTEPE and bixploit, "Rhdamanthys Technical Analysis Report," EchoCTI, Tech. Rep., 2023. [Online]. Available: https://github.com/echocti/ECHO-Reports/blob/main/Malware%20Analysis%20Report/Rhdamanthys/Rhadamanthys-EN.pdf

[95] hasherezade. From Hidden Bee to Rhadamanthys - The Evolution of Custom Executable Formats. Checkpoint. [Online]. Available: https://research.checkpoint.com/2023/from-hidden-bee-to-rhadamanthys-the-evolution-of-custom-executable-formats/

[96] K. Hahn. SectopRAT: New version adds encrypted communication. G Data. [Online]. Available: https://www.gdatasoftware.com/blog/2021/02/36633-new-version-adds-encrypted-communication

[97] BlackPoint, "Ratting Out Arechclient2," BlackPoint, Tech. Rep., 2022. [Online]. Available: https://cdn-production.blackpointcyber.com/wp-content/uploads/2022/11/01161208/Blackpoint-Cyber-Ratting-out-Arechclient2-Whitepaper.pdf

[98] K. Hayashi. Analysis of Smoke Loader in New Tsunami Campaign. Palo Alto Networks Unit 42. [Online]. Available: https://unit42.paloaltonetworks.com/analysis-of-smoke-loader-in-new-tsunami-campaign/

[99] P. Trouerbach. SmokeLoader - The Pandora's box of Tricks. YouTube (BSides Portland). [Online]. Available: https://youtu.be/QOypldw6hnY?t=3237

[100] BitSight. Unveiling Socks5Systemz: The Rise of a New Proxy Service via PrivateLoader and Amadey. BitSight. [Online]. Available: https://bitsight.com/blog/unveiling-socks5systemz-rise-new-proxy-service-privateloader-and-amadey

[101] B. BAKARTEPE and bixploit, "StealC Technical Analysis Report," EchoCTI, Tech. Rep., 2023. [Online]. Available: https://github.com/echocti/ECHO-Reports/blob/main/Malware%20Analysis%20Report/StealC/StealC_Technical_Analysis_Report.pdf

[102] R. Bhat. Neutralizing Tofsee Spambot – Part 3 | Network-based kill switch. Spamhaus. [Online]. Available: https://www.spamhaus.com/resource-center/neutralizing-tofsee-spambot-part-3-network-based-kill-switch/

[103] M. Kotowicz and J. Jedynak, "Peering into spam botnets," CERT.PL, Tech. Rep., 2017. [Online]. Available: https://lokalhost.pl/txt/peering.into.spam.botnets.VirusBulletin2017.pdf

[104] GovCERT.ch. Tofsee Spambot features .ch DGA - Reversal and Countermesaures. GovCERT.ch. [Online]. Available: https://web.archive.org/web/20220916181147/https://www.govcert.ch/blog/tofsee-spambot-features-.ch-dga-reversal-and-countermesaures/

[105] fumik0. Let's dig into Vidar – An Arkei Copycat/Forked Stealer (In-depth analysis). fumik0 blog. [Online]. Available: https://fumik0.com/2018/12/24/lets-dig-into-vidar-an-arkei-copycat-forked-stealer-in-depth-analysis/

[106] A. Holland. Tracking Vidar Infrastructure with Censys. Censys. [Online]. Available: https://censys.com/tracking-vidar-infrastructure/

[107] K. Merriman and P. Trouerbach. Out of the Sandbox: WikiLoader Digs Sophisticated Evasion. Proofpoint. [Online]. Available: https://www.proofpoint.com/us/blog/threat-insight/out-sandbox-wikiloader-digs-sophisticated-evasion

[108] C. Hammond, O. Villadsen, and K. Metrick. Stealthy WailingCrab Malware misuses MQTT Messaging Protocol. IBM. [Online]. Available: https://securityintelligence.com/x-force/wailingcrab-malware-misues-mqtt-messaging-protocol/

[109] I. Lytzki. XWorm Malware: Exploring C&C Communication. ANY.RUN. [Online]. Available: https://any.run/cybersecurity-blog/xworm-malware-communication-analysis/

[110] 0xMrMagnezi. Malware Analysis - XWorm. Medium b.magnezi. [Online]. Available: https://medium.com/@b.magnezi/malware-analysis-xworm-80b3bbb072fb

[111] R. Huessy. MalwareBazaar database. abuse.ch. [Online]. Available: https://bazaar.abuse.ch/

[112] J. Bremer. Triage Sandbox. Hatching.io. [Online]. Available: https://hatching.io/triage/

[113] P. Cowman. Triage Thursday Episode 1: Open registration and a busy week of updates. Hatching.io. [Online]. Available: https://hatching.io/blog/pcapng-https/

[114] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 263–278. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/plohmann

[115] I. Ghafir, V. Prenosil, M. Hammoudeh, L. Han, and U. Raza, "Malicious ssl certificate detection: A step towards advanced persistent threat defence," in *Proceedings of the international conference on future networks and distributed systems*, 2017.