Università degli Studi di Napoli Federico II
Ph.D. Program in
Information Technology and Electrical Engineering
XXXV Cycle

Thesis for the Degree of Doctor of Philosophy

Malicious and Large-Scale Phenomena over the Internet: An Analysis based on DNS

by
Antonia Affinito

Advisor: Prof. Alessio Botta

Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

"Imagination is the highest form of research."

Albert Einstein

Malicious and Large-Scale Phenomena over the Internet: An Analysis based on DNS

Ph.D. Thesis presented
for the fulfillment of the Degree of Doctor of Philosophy
in Information Technology and Electrical Engineering
by

Antonia Affinito

January 2022

Approved as to style and content by

Prof. Alessio Botta, Advisor

Università degli Studi di Napoli Federico II
Ph.D. Program in Information Technology and Electrical Engineering
XXXV cycle - Chairman: Prof. Stefano Russo

http://itee.dieti.unina.it

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Information Technology and Electrical Engineering is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
Parts of this dissertation have been published in international journals and/or conference articles (see list of the author's publications at the end of the thesis).

Napoli, March 9, 2023

_____

Antonia Affinito

Abstract

Cyber security threats and real-life phenomena (*e.g.,* COVID-19 pandemic) are increasingly reflected over the Internet. Hackers usually scan a network to discover active and vulnerable network devices prior to initiating a malicious activity. This is also the approach adopted by botnets, one of the most important, current cyber security threats. These malicious networks of bots more and more use the Domain Name System (DNS) as a tool for their operations.

This thesis provides twofold contributions. The first one addresses the problem of detecting port and net scans in high-speed networks. Big Data analysis techniques are applied to cope with the large volume of data to be processed. Mirai botnet scan is also investigated. Scrutinizing its signature over a six-year period from real Internet traffic reveals the evolution of such botnet and its variants.

The second contribution focuses on DNS as a good observation lens for monitoring the proper operation of the Internet. It focuses on how Internet Service Providers and public DNS resolvers protect users accessing domains associated with such activities. It also shows how the lifetime of malicious domain names may be shorter than the one of benign domains due to take-down efforts of registries. Finally, two case studies on how DNS data can be used to analyze prominent and global real-life events are reported. First, the effect of the COVID-19 pandemic restrictions on network utilization is explored, providing insights into the usage of Internet applications during this period. Second, the impact of the Ukraine conflict on Russian domain infrastructure is presented, investigating its changes before and after the start of this event.

Keywords: Botnet, Cyber Threats, Domain Names, DNS, COVID-19 Pandemic, Russia-Ukraine Conflict

Sintesi in lingua italiana

Le minacce informatiche e gli eventi che hanno un forte impatto sulla società (es. pandemia COVID-19) si riflettono sempre di più sulla rete Internet. In genere, un hacker, prima di effettuare un attacco informatico, scansiona una rete al fine di individuare dispositivi vulnerabili. Questo approccio è adottato anche dalle botnet, che rappresentano una minaccia emergente alla sicurezza informatica. Le botnet utilizzano sempre di più il Domain Name System (DNS) per le loro operazioni di comando e controllo.

Il presente lavoro di tesi fornisce due contributi significativi. Il primo ha l'obiettivo di affrontare il problema del rilevamento delle attività di port e net scan nelle reti ad elevata velocità. A tale scopo, vengono utilizzate tecniche di Big Data per gestire il grande volume di dati da elaborare. Inoltre, si analizzano le scansioni della botnet Mirai. L'analisi della sua signature nel corso di sei anni evidenzia l'evoluzione di tale botnet e delle sue varianti. Il secondo contributo fornito in questa tesi utilizza il DNS come possibile strumento per monitorare il corretto funzionamento di Internet. In primo luogo, si esamina la protezione offerta dai fornitori di servizi Internet e dai resolver DNS pubblici contro i domini malevoli. Successivamente, si mostra come la durata dei domini malevoli sia generalmente inferiore rispetto a quella dei domini benevoli, a causa delle operazioni di rimozione effettuate dai registry. Infine, vengono esaminati due eventi di rilievo per la società attraverso l'analisi dei dati DNS. In particolare, questa tesi tratta l'effetto della pandemia COVID-19 sulla rete Internet, fornendo approfondimenti sull'uso delle applicazioni da parte degli utenti durante tale periodo. Inoltre, viene presentato l'impatto del conflitto in Ucraina sull'infrastruttura di dominio russa, analizzando le sue variazioni prima e dopo l'inizio di questo evento.

Parole chiave: Botnet, Minacce Informatiche, Nomi di Dominio, Pandemia COVID-19, Conflitto Russia-Ucraina

Contents

Acknowledgements

I am extremely grateful to my supervisor, Prof. Alessio Botta, for providing invaluable guidance, support, and inspiration during my PhD journey.

I would like to express my gratitude to Prof. Roland van Rijswijk - Deij, Prof. Anna Sperotto, Prof. Mattijs Jonker and all the other colleagues I met at the University of Twente. Their welcome and expertise contributed to the success of my research.

I would also like to thank Prof. Stefano Russo for his support and valuable advice.

List of Acronyms

The following acronyms are used throughout the thesis.

DNS	Domain Name System
IoT	Internet of Things
DDoS	Distributed Denial of Service
DoS	Denial of Service
TLD	Top-Level Domain
gTLD	Generic Top-Level Domain
ngTLD	New Generic Top-Level Domain
ccTLD	Country Code Top-Level Domain
ISP	Internet Service Provider
CA	Certificate Authority
IDS	Intrusion Detection System
DoH	DNS over HTTPS
DoT	DNS over TLS

ICMP	Internet Control Message Protocol
CT	Certificate Transparency
VPN	Virtual Private Network
C&C	Command and Control
DGA	Domain Generating Algorithm

List of Figures

List of Tables

Chapter 1

Introduction

In the last decade, Internet has been essential for many aspects of daily life, leading users to own more and more various types of devices. Due to the wide volume and variety of devices connected to the network, the Internet has become crucial, especially in terms of security and data privacy. These concerns have intensified mainly with the advent of Internet of Things (IoT) technology. The IoT devices, indeed, generally have limited resources and lack relevant security features, making them targets for cybercriminals [5, 6]. This evolution has caused networks to become increasingly susceptible to malicious activities and massive cyber attacks. The Distributed Denial of Service (DDoS), for example, is one of the common and dangerous network attacks [7]. It involves exploiting plenty of hijacked devices to make a specific network resource inaccessible and unavailable to legitimate users. The identification of these kinds of malicious anomalies is generally entrusted to an Intrusion Detection System (IDS). These systems constantly monitor networks with the aim of identifying malicious threats or intrusion attempts. However, although these tools adopt multiple methods for their detection, they have limited effectiveness in catching newer anomalies [8]. Furthermore, the identification of malicious activities in high-speed networks is also challenging. They are characterized by a huge volume and variety of data to be analyzed, produced by numerous kinds of network devices [9].
Societal, large-scale events are also having a growing impact on the Internet. Two recent episodes, also covered in this thesis, are the COVID-19

pandemic and the Ukraine conflict. On the one hand, these events have led cybercriminals to implement more and more cyber threats. For example, there is evidence of a significant increase in malware during the COVID-19 pandemic and phishing during the Russia-Ukraine conflict [10]. On the other hand, these events have a major impact on the performance of the network itself. In particular, COVID-19 restrictions imposed by governments around the world to stay at home for several months have encouraged people to use the network for various purposes. Some examples are online lessons, remote working, entertainment, and staying in touch with friends and family. However, this increased use of the Internet resulted in network overload, especially for applications that had to handle a large number of users. Regarding the Russia-Ukraine conflict, on March 2022, the Russian government mandated that all websites and network services come under the control of national hosting providers [11]. In addition, some social media were banned, restricting communication, and news dissemination and bringing people to use a Virtual Private Network (VPN) to circumvent the bans [12].

Based on the previous discussion, the primary objectives of this thesis are twofold: first, the analysis of malicious cyber activities at large scale on the Internet; second, the evaluation of the impact of large-scale real-life events on the network. Initially, we address the problem of how to prevent massive network attacks by analyzing scanning activities (*e.g.,* port and network scans). The latter refers to the process of identifying vulnerable hosts on a network and constitutes the preliminary step before an attack. Numerous studies have been published over the years describing various techniques for detecting such network anomalies. Some of these works date back many years (*e.g.,* around the early 2000s [13, 14, 15]) and present approaches that are not suitable for high-speed networks. Other works mainly rely on advanced and new techniques [16, 17]. Our approach, instead, incorporates elements of both categories, implementing an algorithm based on a traditional, statistical method repurposed for Big Data technologies. Together with flow-level analysis of network traffic, this choice addresses the challenge of processing large volumes of data to analyze. This method, used on real network traffic traces, achieves good performance in both detection and execution time. It also allowed us to detect a larger number of anomalies than a reference technique. Another open issue in the detection

of scanning activities is related to botnet scans, a kind of port scan that has become popular with the advent of botnets. A botnet is a collection of network devices (*i.e.,* bots) infected with malware, enabling an attacker (*i.e.,* botmaster) to control them remotely. In the initial injection phase, the botnet scans are carried out using a network of compromised devices, aiming to infect as many devices as possible. In this thesis, we focus on Mirai, one of the popular botnets, and we examine the scanning activities of this botnet over a six-year period. The proposed approach consists of analyzing the TCP SYN packets, included in real network traffic traces, that verify the Mirai signature. Namely, to minimize memory consumption, the cybercriminals set the destination address equal to the TCP sequence number (TCP.seq==IP.dst). We prove how the Mirai signature is nowadays still implemented in botnet scanning activities. In addition, we show that other ports, besides telnet and ssh ones, have been targeted over the years, identifying new variants.
Botnets, like almost all Internet applications, increasingly rely on Domain Name System (DNS) for their operations [18]. For this reason, in this thesis, we use the DNS as a possible lens to observe potential malicious cyber activities. We first analyze malicious domain names (*e.g.,* malware, phishing), focusing on two aspects of the DNS. The first aspect involves investigating how local resolvers - provided by Internet Service Provider (ISP)s - and the public ones - provided by Google and OpenDNS - protect users to access domain names associated with malicious activities. We explore both the traditional DNS queries and the encrypted ones over HTTPS (*i.e.,* DNS over HTTPS (DoH)). The proposed approach reveals that both local and public DNS resolvers achieve a similar level of security. We also inspect the response times of the two categories of resolvers. We show that local resolvers are much faster than public resolvers, even when not considering already cached domains. As a consequence, we prove that we do not have to trade off security and performance. Finally, we show that there are no significant differences between DNS and DoH queries both in response time and code. The second approach we applied to explore malicious domain names is the analysis of the domain name lifetimes. We perform this analysis among the ten largest Top-Level Domain (TLD)s over a ten-year period. In particular, we evaluate the lifetime as a difference between the first and last time that a domain is seen in the zone file. However, there

may be cases where the domain name is parked or in a grace period and, consequently, is not present in the zone file. To account for this limitation, we allow gaps of 90 days before considering a lifetime closed (*i.e.,* 80 days as the sum of all the removal stages of a domain name, plus 10 days of margin). The proposed approach reveals that a significant proportion of domain lifetimes expire before 365 days (minimum registration term). To further explore the possible causes of this duration, we evaluate how many of them are malicious, matching them with those included in the DBL blocklist. As a result, we show that a fraction of malicious domain names is removed from the zone files a few days after appearing in the blocklist. This result is mostly true for some TLDs. Others let the domain names expire naturally. Additionally, by looking at the features of the WHOIS data, we see evidence of bulk registrations related to malicious domain names.

Finally, in this thesis, we examine the effects of the impact of large-scale real-life events on the Internet by looking at DNS data. As mentioned above, two significant events that had a global impact and lasting relevance in the past three years are the spread of the COVID-19 pandemic and the conflict Russia-Ukraine. Regarding the COVID-19 pandemic, we analyze changes in the trends of the most widely used Internet applications before and during the COVID-19 pandemic. We look at several categories (*i.e.,* video, social media, messaging, and collaboration tools) to cover various ways in which the network was used during that period. For this purpose, we use the top 1 million lists provided by Alexa and Cisco Umbrella. Furthermore, the different methods they use to collect data give the ability to understand what type of resource that app was used by (*e.g.,* via browser, mobile phone or television). We implement an algorithm that analyzes the 10K most and least popular domain names of each top 1 million files, covering a period from three months prior to the start of restrictions (*i.e.,* March through April). The proposed approach shows that users primarily access Youtube via browser, and Netflix via other types of network devices (*e.g.,* mobile phones, and TV). Regarding collaboration tools, during the first months of the restrictions, Skype and Microsoft Teams were the most used, followed by Zoom and Webex.

The last event analyzed is related to the regulations of the Russian government to move all the Russian websites and network services inside Russia.

The first regulations, which included the implementation of a Russian national DNS, were enacted in 2019 and implemented by January 2021 [19]. With the invasion of Ukraine, these regulations have become more stringent [20]. Regarding this topic, we inspect the zone files of `.ru` and .рф over a five-year period (*i.e.,* June 2017 - March 2022) and the Certificate Transparency (CT) logs. Our analysis shows that, already in 2018, almost 70% of domain names were fully hosted in Russia. Beginning in February 2022, with the invasion of Ukraine, only a small percentage of domain names moved fully within Russia. Afterwards, we see evidence that some US companies still sell services to Russian customers. Finally, we show how Certificate Authority (CA)s have reacted to the conflict, finding that the American Let's Encrypt manages a great portion of Russian domain names.

1.1 Thesis Outline

The thesis is structured as follows.

Chapter 2 first introduces the concepts of network anomalies, both malicious and benign. It focuses on the scanning activities (*e.g.,* port, network and botnet scans), and it describes the most common disruptive forms of attack. In addition, this chapter provides an overview of the DNS and its features. Concepts related to the DNS namespace, the domain name resolution and life-cycle, blocklists and early take-down actions are discussed.

Chapter 3 addresses the problem of identifying the port and network scans in high-speed networks. Specifically, a system that utilizes a flow-level approach and Big Data technologies is designed and developed. Experimental results confirm the effectiveness of the proposed system, indicating that it achieves better performance than other ground truths, based on complex algorithms. Furthermore, the execution times of this system are short, making it a promising application for a real-time scenario.

Chapter 4 focuses on the analysis of a particular class of port scans: the botnet scans. An overview of six-year of Mirai botnet scans in real traffic traces is presented, examining the features of the TCP SYN packets that verify the Mirai signature. The proposed approach reveals some attempts prior to the first Mirai large-scale attack. It also shows that the

Mirai signature is still being implemented by cybercriminals today to detect vulnerable devices. In addition, this chapter provides insights into the evolution of ports probed over the years, resulting in the discovery of new variants of Mirai.

Chapter 5 explores the local (provided by the ISP) and the public DNS resolvers. Their capabilities to protect users by detecting malicious domain names and the timing performance are investigated. The proposed approach consists of querying a consistent number of benign and malicious domain names at selected resolvers, using both traditional DNS and DoH. Experimental results illustrate that local DNS resolvers are usually faster than public ones. In addition, the protection levels of both categories of DNS resolvers are largely equivalent.

Chapter 6 analyzes the domain name lifetimes of the largest TLDs over a ten-year period. It shows that a significant number of lifetimes is shorter than the minimum registration term (*i.e.,* one year). To further investigate the possible reasons for such short lifetimes, an evaluation of malicious domain names is conducted using blocklist data. Results reveal that a considerable fraction of malicious domains has lifetimes shorter than one year. This chapter also presents evidence that short-lived malicious domains are taken down and removed from the zone files after appearing on the blocklist. Furthermore, an analysis of malicious registration campaigns is reported.

Chapter 7 deals with understanding the impact of the lockdown restrictions, implemented during the COVID-19 pandemic, on the Internet. An algorithm is implemented to analyze changes in domain name usage trends related to widely-used Internet applications belonging to various categories (*i.e.,* video, social media, messaging and collaboration tools). The proposed approach covers a six-month period, including some months prior to and during the COVID-19 restrictions. Experimental results confirm how people found entertainment and work benefits in some Internet applications.

Chapter 8 covers the issue of exploring the impact of the Ukraine conflict on Russian Internet infrastructure. We present an overview of changes in the Russian Internet infrastructure by examining zone files related to `.ru` and .рф and certificate issuance data for the past five years. Results show that a large majority of Russian websites ($\approx 70\%$) were fully hosted

in Russia even before the conflict. In addition, at the beginning of the war, there is only a slight increase in domain names that moved into Russian infrastructure. Furthermore, an analysis related to the TLDs dependencies is reported, highlighting that Russian domains still rely on non-Russian TLDs. Afterwards, an inspection of Western providers is presented, revealing that some companies continue to provide services to Russian customers. Finally, the chapter reports the behavior of CAs regarding the authorization of certificates for Russian domains in response to conflict and sanctions.

Chapter 9 summarizes the thesis contributions.

Chapter 2

Background

In this Chapter, we first introduce the concepts of network anomalies, paying particular attention to malicious ones. More in detail, in Section 2.1, we provide an overview of the most popular cyber threats, and scanning activities (*i.e.,* port and network scans). In Section 2.2, we present the Mirai botnet, its workflow and architecture. Afterwards, we outline the theoretical concepts and definitions related to the DNS in Section 2.3.

2.1 Network Anomalies

In the past several decades, research on anomaly detection has been predominant in a wide variety of applications. The term anomaly refers to a pattern in data that deviates from a normal or an expected behavior [21]. In the context of computer networks, the normality concept should be defined based on different factors related to a network. Network anomalies may be categorized into benign and malicious. A *benign network anomaly* may occur for several reasons, such as device failure, overload of a network, or configuration errors. On the other hand, a *malicious network anomaly* or intrusion, is an activity carried out by an attacker with the goal of corrupting the network or its hosts. Specifically, an intrusion is a voluntary attempt by unauthorized users to access or manipulate information, as well as to deny service to legitimate users or make a system unavailable. Malicious anomalies differ from benign ones because their main goal is to

undermine the three pillars of data security: availability (data availability), confidentiality (data secrecy), and integrity (data not degraded or tampered with).

2.1.1 An Overview of Network Anomalies

In recent years, a significant number of network anomalies, both benign and malicious, have been identified. One of the most famous attacks is the Denial of Service (DoS), which consists of attempts to block legitimate users from accessing the system by making it unreachable [22]. Generally, a DoS attack targets different kinds of network resources (*i.e.,* servers, websites or entire networks) and it may occur in several forms. Two examples are *TCP SYN Flooding* and *Internet Control Message Protocol (ICMP) Flooding*. More in detail, a TCP SYN Flooding takes advantage of the three-way handshake, in which an attacker forwards a substantial number of TCP SYN requests to a server, making it overloaded and unable to process these legitimate requests. Similarly, an ICMP Flooding attack occurs when an attacker overloads his victim by sending it a high number of ICMP packets. In both scenarios, the goal is to make the servers unresponsive and unavailable to legitimate users.

In addition, another form of DoS attack is the DDoS. It involves multiple compromised network devices to conduct a DoS attack amplifying its effect. As we will explain in Subsection 2.2, these network devices (*i.e.,* bots or zombies) are infected with malware software that allows them to be remotely controlled by a malicious actor (*i.e.,* botmaster).

An example of benign anomaly is instead represented by *Flash Crowd*, easily confused with a DoS attack. It consists of an unexpected increase in traffic to a Web site or a network resource. This increase may be, for example, due to important breaking news or the release of certain products [23].

Port and Network Scans

Two typical malicious activities that attackers perform to find vulnerable hosts are *port and network* scans, also known as scanning activities. Detecting these types of anomalies is an important step in avoiding a larger attack, and protecting a network and its resources.

More in detail, a **Network scan**, also referred to as net scan, is an activity prior to an attack with the purpose of identifying active hosts within a network. Specifically, it consists of sending packets to a huge number of IP addresses that can be generated automatically through special tools. The hosts that answer are considered the active ones.

A **Port scan** is an activity that an attacker performs to check all the active services and discover possible vulnerabilities on a host. It involves sending a considerable number of packets to a range of ports of one or a few hosts. The responses help to understand the status of the ports and identify the vulnerabilities of that specific host. There are different kinds of port scans [24, 25]. Some examples are given below:

- *TCP Connect scan*: relies on the three-way handshake to identify which ports are open on a victim host.
- *TCP SYN scan*: consists of initiating the three-way handshake without completing it. More specifically, the attacker sends an SYN segment to a specific port on a host. If an RST is received, then the port is closed. If an SYN-ACK is received, it implies the target port is listening.
- *TCP ACK scan*: is useful to establish the presence of a firewall. Specifically, the attacker sends a packet with the ACK flag set. If an RST packet is received, it means that the port is unfiltered and potentially open. Otherwise, if no responses are received, it implies that the port is filtered.
- *TCP FIN scan*: involves sending a packet with the FIN flag set. If an RST is received, it means that the port is closed. If it is not closed, the victim host is not expected to respond, so it may be open or filtered.
- *TCP XMAS scan*: is similar to the TCP FIN scan except that, in this case, also URG, and PSH are set in the packets.
- *TCP Null scan*: consists in sending a packet without setting any flag.
- *UDP scan*: involves sending an UDP packet. If a "port unreachable" is received, it means that no there is no service listening on that port.

2.2 Understanding Botnets: Workflow and Architecture

In the last years, botnets are becoming increasingly prevalent. They constitute a new type of network anomaly, aiming to launch a large-scale cyber attack, *e.g.,* DDoS attack or spam emails. More in detail, a botnet is a collection of internet-connected devices infected with malware that allows an attacker to control them remotely. The infected devices (*e.g.,* personal computers, IoT devices, mobile devices) and the attacker are referred to as *bot* and *botmaster*, respectively. Botnets usually are designed for malicious activities, including sending spam, stealing data, or performing DDoS attacks. The term bot (short for robot) derives from its ability to automatically execute commands forwarded by the botmaster through a Command and Control (C&C) server, unaware of the device owner.

The life cycle of a botnet typically involves five stages [26]. In the first stage, also known as the *initial injection*, the botmaster seeks to compromise vulnerable devices and expand the botnet by spreading malicious software through methods such as malicious websites, and email attachments. Once the device is infected, the second stage, the *secondary injection*, starts. Specifically, the botmaster executes a shellcode, delivered to the devices by the malware, allowing the attacker to take control of it and add it to the botnet. In the third stage, the *connection*, the infected devices create communication channels with the C&C server. In the fourth phase, the *malicious command and control*, the botmaster sends commands via the C&C channel to the bots, providing instructions to perform a variety of attacks. Finally, in the *maintenance and update* stage, the botmaster constantly updates the attack patterns and IP addresses of the bots to keep a long-lived botnet [27, 28, 29, 26].

Based on the C&C infrastructure, there are typologies of botnets: centralised, decentralised and hybrid [28, 30].

In a *centralized* structure, shown in Figure 2.1, the botmaster generally communicates with the bots via a single C&C channel [31]. This typology of botnet benefits from the simplicity and speed of communication between the C&C server and the bots. However, only a single point of failure is a major disadvantage. Indeed, if the C&C server is detected and taken down, the entire botnet is rendered unusable. Generally, the two protocols most

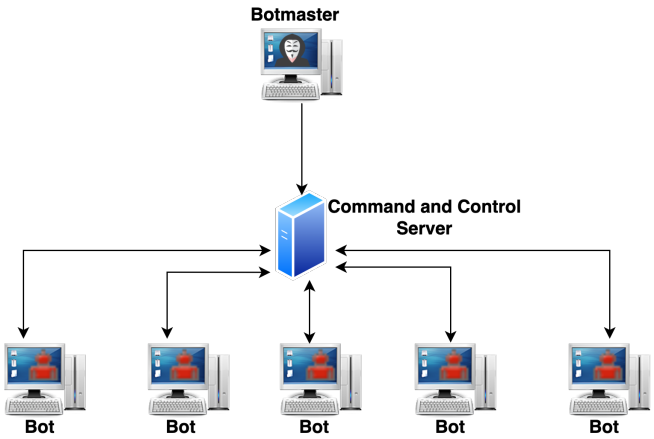


Figure 2.1. Centralized botnet architecture

adopted in a centralized architecture are IRC, HTTP and POP3 [28, 32] [1]. Botnets adopting peer-to-peer (P2P) solutions, also referred to as *decentralized* architecture, have emerged to overcome the limitations of a single C&C channel [35]. In this case, each bot holds multiple connections with other bots and there are multiple C&C servers geographically spread. However, the disadvantage is the major complexity of botnet implementation. The *hybrid* structure, instead, overcomes the P2P typology, diversifying bots into layers based on their role and limiting interactions among them. Each device maintains a fixed list of peers, a few can execute commands to check the situation of the botnet, and others wait for commands from the equivalents to whom they are connected [32, 36].

2.2.1 Mirai Botnet

The number of malicious attacks has intensified with the rapid spread of IoT devices, specifically targeting networked devices and sensors [37]. Most of these devices have limited hardware resources, and lightweight operating systems, and are exempt from regular security updates. Moreover, they are released with vulnerable settings (identical factory credentials for the

[1] The most powerful IRC botnets detected are Spybot, SDBot, and Agobot. Well-known ones that have HTTP-based C&C channels are Spyeye and Zeus [33, 34].

same brand) and easy prey for new botnets. The most famous botnet of recent years, Mirai, has been able to exploit this army of dumb devices to pull off huge DDoS attacks. Mirai was first identified in August 2016 by the white-hat research group [38]. Its main feature is the ability to infect a considerable number of hosts in a short time, conducting attacks with unprecedented volumes of traffic.

Mirai's architecture is decentralized, see Section 2.2. Specifically, it consists of three servers between the bots and the botmaster to establish multiple C&C channels [39]:

- **Command and Control (C&C) Server**. It acts as a bridge between the botmaster and the botnet. This element generally uses two TCP sockets, one on port 23 and the other on port 101 to send instructions to bots for attacks.
- **The Report Server**. It manages a database, which it updates when new bots are added, and it constantly and directly communicates with all members of the botnet.
- **The Loader** receives from the bot the coordinates of the victim device (IP and port) and the credentials to access, download and install the malware.

Figure 2.2 shows Mirai's workflow, which consists of seven phases [2, 3, 1]. In the first phase, referred to as *brute force*, the bots search for new devices to infect. In particular, the bots probe to pseudorandom IPv4 addresses on Telnet ports 23 and 2323. However, with the advent of new Mirai variants, many other TCP ports appeared as targets including 22, 7547, 8000, 8080, 2222, 23231, 37777, 6789, 5555 [40]. The botmaster also provides a blocklist of IP addresses not to be contacted. This list includes multicast addresses and other IPs such as the Internet Assigned Numbers Authority (IANA), the U.S. Department of Defense, and many others [41]. When the bots detect vulnerable devices, they try to bypass their security by applying a dictionary attack [2]. More in detail, each bot has a list of 62 predefined user/password pairs to guess the credentials and gain access to the device. Successful access to the device initiates phase 2.

[2] A dictionary attack is a method to guess passwords by entering every word in a dictionary as a password.

In this phase, the bots notify the victim's address, port, and credentials to the report server. During phase 3, the botmaster, via the C&C server, periodically queries the "Report Server" to retrieve statistics on the status of the botnet. Phase 4 consists of initiating the infection of the detected victim devices. Specifically, loaders log into the devices and instruct them to download the Mirai malware [42].

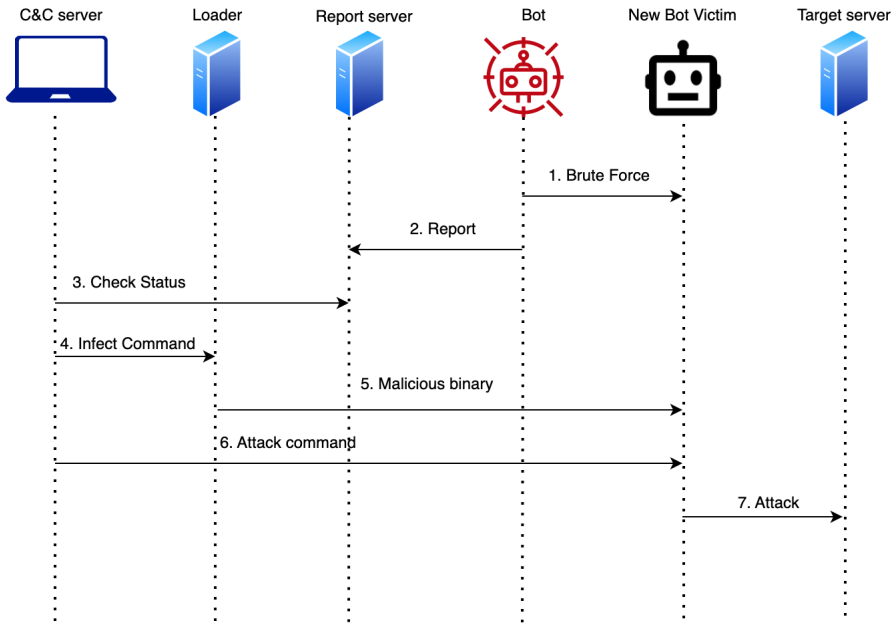


Figure 2.2. Mirai Botnet workflow [1, 2, 3]

In phase 5, the bots download and run the malware according to the instructions provided in the previous stage. During this process, the malware shuts down ssh and Telnet services to protect itself from other possible infections. It also starts interacting with the C&C server to receive instructions. Phases 6 and 7 involve botnet attacks against one or more targets. In particular, in phase 6, the C&C server provides information to the bots about the type of attack and its duration. In the last phase, however, all "firepower" is concentrated against one or more targets, usually using DDoS-type attacks [1].

2.2.2 Mirai Scanner Signature

According to RFC 793, a TCP SYN packet is sent when a new TCP connection is created, including a 32-bit random value in the TCP sequence number field [43]. Nevertheless, during the scanning phase of the Mirai workflow, bots send SYN packets to random IP addresses by setting the TCP sequence number equal to the destination address (*i.e.,* TCP.seq==IP.dst). The reason for this behaviour is that Mirai bots are typically IoT devices with limited resources. Thus, to minimize the memory consumption, the association between the IP address contacted and the TCP sequence number is not stored in memory, just because the destination IP is used as TCP sequence number [44, 45, 46].

2.2.3 Variants Using Mirai Scanner Signature

Mirai was first detected in August 2016: since then, the Mirai botnet has performed its initial scans primarily on service-related ports as Telnet (9 out of 10 requests directed to port 23, 1 out of 10 directed to port 2323) to exploit accesses that are not carefully protected by device manufacturers [38].

From August 2016 to February 2017, the Mirai botnet was capable of infecting more than 600K agents at its peak, mostly IoT devices, with a doubling time of 75 minutes [39] and 15K DDoS attacks have already been associated with Mirai. Among the most famous, we find the attack on a well-known cybersecurity blog by journalist Brian Krebs, which reached a traffic volume of 623 Gbps, an amount of data never recorded before (or never publicly announced) for a DDoS attack [47]. Other ports were then added to the scan pool by new emerging botnets based on the Mirai source code. On September 30th, 2016, the source code of Mirai was first released to the public. As a consequence, many other large DDoS have occurred, such as one towards the French web host OVH equal (1Tbps) [48], or to Dyn on 21st October 2016, DNS providers of high traffic web services such as Twitter, Spotify, Netflix, Reddit, and GitHub [49]. At the end of 2016, a Mirai variant exploit a vulnerability in the CPE WAN Management Protocol (CWMP) implemented in two models of Deutsche Telekom's customer routers interesting nearly a million users [50].

By 2017 Radware discovered that even ports related to SSH service

started to be probed by a botnet called Brickerbot [51]. In September 2017, some articles noted a very dense group of ports contacted by Mirai turns out. This set appears to be a target of a botnet called Reaper that borrows part of the code from Mirai but only focuses on exploiting known vulnerabilities [52, 53]. The Reaper variants do not leverage Telnet brute force with default credentials anymore, but rather leverage HTTP-based exploits of known vulnerabilities in IoT [54]. The reaper variant also uses a combination of nine attacks targeting known IoT vulnerabilities. These attacks affect many popular router brands as well as IP cameras, Network Attached Storage devices, and servers [55].
In November 2017, a new variant of Mirai emerged, Satori, whose peculiarity lies in the way the spreading malware, making it more worm-like [56]. The bot, in fact, was not relying on the loader-scanner mechanism to perform remote planting [57]. Satori adds ports 37215 and 52869 in the scan and drives infected devices to download themselves from the same original URL. The two main vulnerabilities used by Satori, concern one known since 2014 (CVE-2014-8361) [58] for port 52869, and one discovered in December 2017 (CVE-2017-17215) [59]. In addition to the ports related to Telnet service, other ports in the pool of ports contacted by Mirai are those related to HTTP service (80, 8080, 88, 81, and 8000). Edwards et al study and describe the behaviour of a variant of Mirai attacking these ports, Hajime [60].
On May 1, 2018, VPN Mentor disclosed two vulnerabilities against GPON home routers [61]. Also in 2018, the WICKED bot was involved and actively scanning on ports 8080, 8443, 80, and 81 [62]. Since then, at least 5 different botnet families have started to include new exploits based on two vulnerabilities (CVE-2018-10561), (CVE2018-10562) related to HTTP service authentication [63]. Another port that emerges in a new pool of ports of attention from the new Mirai variants Moobot is the ADB port [64]. In this set of ports, we have TCP 34567 of the DVRIP protocol still used to carry out high-profile DDoS attacks, with an average of one hundred attacks per day [65]. Moobot is a new botnet family based on Mirai. Recently it has made quite many releases, according to their C2 protocols and programming languages, we can roughly divide them into *moobot.socks5, moobot.tor, moobot.tor.b, moobot.go, moobot.go.tor, moobot.c*, etc. Not every moobot variant uses this 185 netblock, but we do notice the moobot.c

sample uses 185.244.25.219 as the Downloader [66]

2.3 Domain Name System

The DNS represents an important observation point to study the main issues of current networks, including performance and security. It plays an important role in Internet services. If DNS is disrupted, most communication on the Internet is actually stopped. Its main function is to convert human-readable names (ex: example.com) into their corresponding IP addresses (ex: 93.184.216,34), and so it is considered the phonebook of the Internet. This conversion is accomplished by retrieving information from the corresponding DNS record.

Specifically, a DNS record is a set of information useful to map a domain name to its IP address. The following are the major DNS record types:

- *A record*: maps a domain name to the corresponding IPv4 address.
- *AAAA record*: maps a domain name to the corresponding IPv6 address.
- *MX record*: maps a domain name to the hostname of a server that handles its e-mail.
- *NS record*: includes a list of authoritative nameservers of a domain name.
- *CNAME record*: maps an alias name to a real or canonical domain name.
- *TXT record*: provides text information related to a domain name.

Zone Files

The DNS records are classified in zones, each managed by a specific organization [67]. A DNS *zone file* is a text file that includes all the DNS records of a specific zone, according to a format specified in the RFC 1034 and 1035 [68, 69].

Specifically, at the top of the zone file, there are three main directives [70]:

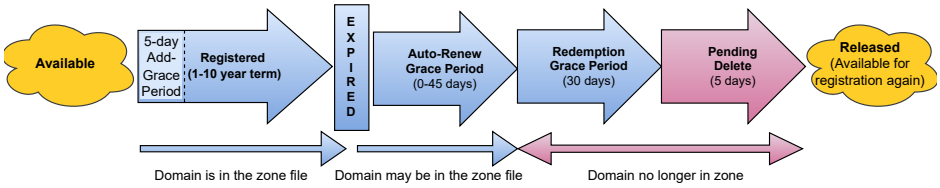


Figure 2.3. A visualization of the life-cycle of a domain, along with all possible states

- **TTL**: Time to Live that corresponds to the time in seconds of the validity of a zone resource record.
- **ORIGIN**: used to define a base name for domain names without a terminating dot.
- **INCLUDE**: include an external file with additional directives.

These directives are followed by the Start of Authority (SOA) record. It is always included in each zone file and it provides significant details regarding the zone. More in detail, it includes the name of the zone, the hostmaster email, the primary nameserver, and the zone serial number. In addition, some timing metrics are provided, such as the time to refresh, retry, expire and the minimum TTL.

2.3.1 DNS Namespace: Top Level Domains

The DNS namespace, first defined in RFC 1034 [69], is a hierarchical inverted tree structure. The root of this inverted tree structure is referred to as the DNS Root. The DNS Root explicitly delegates each individual *zone* under it, typically referred to as a TLD (*e.g.,* `.com` or `.nl`) to organizations, called *registries*, who are responsible for that branch of the namespace, *i.e.,* the *TLD zone*. Registries are typically responsible for administering authoritative nameservers which provide nameservice for all zones under the TLD. For instance, the registry for `.com`, Verisign, operates authoritative nameservers which provide nameserver delegations for `example.com` (typically referred to as a second-level domain (SLD) or registered domain). Those nameservers have authority over all zones under `example.com` (*e.g.,* `www.example.com`). Each of these zones can further

sub-delegate specific branches of the namespace under it.
The TLDs are typically categorized into two types: generic TLDs (Generic Top-Level Domain (gTLD)s) and country-code TLDs (**ccTLDs!** (**ccTLDs!**)). The gTLDs are further divided into two categories: legacy gTLDs (*e.g.,* `.com, .org, .net`), and new gTLDs (New Generic Top-Level Domain (ngTLD)s) (*e.g.,* `.xyz, .loan`) introduced by ICANN in 2012 under the new gTLD program [71, 72, 73]. On the other hand, **ccTLDs!**s are assigned to specific countries (*e.g.,* `.nl, .uk, .de`). Since 2012, ICANN introduced the new gTLD programme, removing many restrictions on the creation of gTLDs [71, 72, 73]. The second gTLD category concerns these new gTLDs (*e.g.,* `.xyz, .loan`) [72].

2.3.2 Domain Name Resolution

Figure 2.4 shows the typical workflow of a domain name resolution. Specifically, the client sends the request to the recursive DNS resolver, which first checks if its cache contains a copy of the DNS record for that domain name. If it contains it, it returns the IP address, otherwise, it forwards the request to the DNS root. Again, the DNS root checks if it has a copy of the DNS record. If it has it, it sends the IP address, otherwise, it responds with the IP address of the top-level domain DNS server. The TLD server verifies its cache and, if it does not possess the DNS record, it sends the IP address of the authoritative nameserver. Finally, the authoritative nameserver responds with the IP address of the domain name.

Encrypted DNS queries In the last few years, to overcome the problem of unencrypted DNS resolution traffic, DoH and DNS over TLS (DoT) solutions have been implemented [74]. In the former case, the DNS query is forwarded on port 853 after establishing a TLS connection. DoH protocol, on the other hand, relies on HTTPS to perform a secure DNS query resolution.

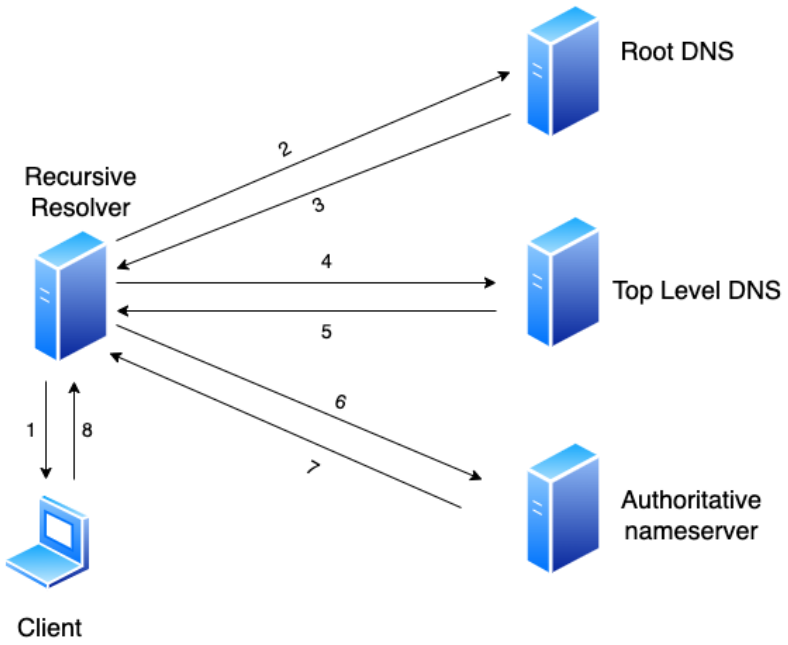


Figure 2.4. Domain Name Resolution

DNS Recursive Resolvers

Nowadays, almost all ISPs provide a DNS recursive resolver. [3] However, a huge number of public or open DNS resolvers are available online and provided for free [75]. The main difference is that the local resolvers resolve domain names only of the users connected to that network. The public resolvers process DNS queries of any user from any location. Many public resolvers are maintained by large companies, such as Google, OpenDNS, and Norton. However, a high number of open and untrusted DNS resolvers are available online because anyone may operate one or more of them. Due to the lack of security measures, open DNS resolvers are usually targets for attacks, such as DDoS attacks and DNS cache poisoning [75].

[3] https://www.quad9.net/news/blog/what-s-the-difference-between-recursive-dns-and-authoritative-dns-2022/

2.3.3 Registries and Registrars

The *registry* is the organization responsible for the administration of a TLD. Typically, the administration of TLDs is delegated to a single organization under contract with ICANN [76]. As part of recent transparency initiatives, ICANN now also mandates that the registries operating a TLD make available via the ICANN Centralized Zone Data Service (CZDS) the *TLD zone file — which includes a list of domains under the TLD and their corresponding nameserver delegations.* The TLD zone files obtained from ICANN CZDS and other sources (Section 6.4) are the basis of this work.

The registries contract with *registrars* to provision new domains. Registrars interface between users looking to obtain domains and the registry administering the domain. A *registrant* looking to obtain a domain name under `.com` would contract with a registrar (*e.g.,* Enom) who in turn interfaces with the registry operating `.com`, Verisign, to query the availability of the domain name and then claim it on behalf of the *registrant.* On successful purchase of a domain, the registrar is then responsible for the domain until it expires or is transferred by the *registrant.* In addition to contracts with the *registry*, *registrars* also have to be accredited by ICANN [77].

2.3.4 Domain Name Life-Cycle

The ICANN registry agreement contract that delegates administration of the TLD also lays out in detail the expected life-cycle of a domain, which includes a number of different possible states. Figure 2.3 provides a visualization of the life-cycle of a domain name in a generic TLD zone [4], and illustrates the following states [78, 79]:

- **Available**: A registrant can use a registrar to find the domain names *available* for registration;

- **Registration**: The registrant can purchase the *available* domain name for a period of at least a year. The registration term may be as long as 10 years. The registrant has a 5-day *Add Grace* period

[4] Since **ccTLDs!** registries have wider latitude in how they administer their zone, the life-cycle for domain names in **ccTLDs!** zones may differ significantly.

during which to undo the registration and receive a refund for the registration fee;[5]

- **Expiration and Renewal**: At the end of the registration term, when a registration is set to expire, the registrant can choose to renew the domain name. On renewal, the registration period (and consequently the expiration date) is extended. The registrant is allowed two grace periods that start after expiration. The first of these grace periods is the *Auto-Renew* period, which ranges from 0 to 45 days. The *Auto-Renew* period allows the registrant to renew the domain name without incurring a penalty;

- **Redemption Period**: After the *Auto-Renew* grace period ends, the *Redemption* grace period starts. In this state, the domain is generally deleted by the registrar, but it still exists in the registry's database. This period, usually 30 days, allows the registrant to renew the domain name with an additional *redemption fee*;

- **Pending Delete**: If the registrant chooses not to renew, the domain will enter the *Pending Delete* state, which is usually 5 days long and during which it is not possible to renew the domain name;

- **Released** After deletion and release, the name can be re-registered. This state is equivalent to the *available* state.

At registration, a registrant procures a domain for a period of at least one year. However, the registrant may choose a longer registration period — anywhere from one to ten years (but always at the granularity of a year). Note, a registrant may transfer a domain following an initial ICANN policy mandated lock of 60 days, but such a transfer requires purchase of at least an additional year of registration beyond the original registration period [78, 81]. Thus, a domain with a lifetime that is not at a granularity of a year (modulo the ICANN mandated grace periods) indicates an action taken by either a registrar or a registry in response to some complaint.

[5] After 2009 a mechanism was introduced to limit abuse of this no-cost grace period, effectively eliminating domain tasting abuse [80].

2.3.5 Early Take Down of Domains

There are a variety of reasons why a domain may not last the one-year duration in the TLD zone file. While there are legitimate reasons for a domain to disappear from the zone files before one year (*e.g.,* a registrant choosing to withhold their domains from being listed), in most cases the disappearance is indicative of take down in response to illegitimate activity. This illegitimate activity can run the gamut from payment fraud to coordinating botnet activity. These take downs can be roughly bucketed into three categories. The first is the "early take down": a registrar discovers an irregularity with the domain registration and takes down the domain. For example, a registrar may discover the registrant used a stolen credit card to purchase the domain. These domains are predominantly taken down before they are involved in malicious activity. The second category is "malicious domain take down": a registrar or registry takes a domain down in response to abuse reports [82].[6] In this case, the domains are taken down after they are involved in malicious activity. The final category is "co-coordinated legal action": law enforcement and other organizations seize large numbers of domains. For instance, in 2011, the US Federal Bureau of Investigation (FBI) seized domains related to Coreflood [84, 85]. Typically, these take downs are targeted at Domain Generating Algorithm (DGA)s associated with malware and botnets. Recently, ICANN made efforts to empower *registrars* and *registries* to unilaterally take down domains involved in ongoing security incidents [86, 87]. In this case, some domains may be taken down by *registrars* or *registries* preemptively before they are involved in malicious activity.

While short-lived domains (domains lasting for less than a year) are indicative of malicious activity, it is important to not use these *solely* as a metric for malicious activity. The "early" and the "malicious" take downs are highly dependent on registrars. While the registrars are required to look into abuse as per their ICANN contract, registrars are routinely overwhelmed, at times by false reporting, leading to long resolution times [83] which may result in domains not being taken down. Consequently, our analyses rely on blocklists as an indicator of malicious activity.

[6] Note, as per the ICANN Registrar Accreditation Agreement, a registrar must maintain an abuse contact to receive abuse reports involving domain names sponsored by the registrar [82, 83]

Chapter 3

Detection of Scanning Activities with Big Data Analysis

In this chapter, we address the problem of detecting port and net scans in high-speed networks. We focus on traditional approaches, previously abandoned due to their limited speed. We rely on Big Data analysis techniques to speed them up and cope with current high-speed networks. The chapter describes our approach and presents an experimental analysis in terms of the detection performance and execution time of a threshold-based algorithm on Apache Spark. We use real traffic traces from the MAWI archive and MAWILab anomaly detectors to compare with our results. The experimental analysis shows that i) the threshold-based algorithm is already able to achieve detection performance higher than MAWILab (in 95% of the considered cases with the best threshold value), currently considered the gold standard in the field; ii) the execution time can be as low as 25 seconds for a 24h traffic trace collected over a 10Gbps link, which makes it usable also in real-time. Moreover, we bridge an important gap in literature providing the research community with a newly labelled dataset, validated using MAWILab and extended with other anomalies not detected by it.

3.1 Motivation

The pervasive use of the Internet has led to a significant increase in the amount of traffic that crosses the network every day. On the other hand, network security and the related, necessary step of traffic analysis, are becoming more and more important [88, 89, 90, 91]. However, the amount of data that has to be analyzed is higher and higher, especially in current high-speed networks. Two typical steps attackers perform to find vulnerable hosts on the network are port and net scan. Such anomalous events represent a good fraction of all the anomalies in current traffic [1]. Detecting them represents an important yet difficult task due to the high volume of traffic to analyze.

Network traffic can be analyzed at several layers of the protocol stack: packet, flow, application, etc. At the flow level, packets relating to the same TCP or UDP communication (*e.g.,* all packets related to an HTTP communication from and to a single host and a web server) are aggregated, and a summary of such group of packets is considered. These summaries can now be provided directly by network devices such as switches and routers, and standard protocols have been defined for this aim (*e.g.,* Internet Protocol Flow Information Export or IPFIX [92]). Working at the flow level seems the most promising approach for coping with the high speed of current and future networks. However, even at the flow level, the analysis of traffic for the detection of anomalies in high-speed networks requires huge computational power or data reduction techniques as flow records still represent a huge quantity of data.

In this chapter, we analyze traffic from high-speed networks using a flow-level approach. The aim is to detect two of the most spread network anomalies *i.e.,* port scan and network scan (or simply net scan). In the former case, an attacker probes a host (the victim) on various TCP/UDP ports to find active and vulnerable services. In the latter case, the attacker scans a group of victim hosts on a single or a small number of ports. Such scanning activities are also associated with worms and botnets [93].

Port and net scans generate a real specific pattern in network traffic. One of the most popular methods for detecting scanning activity is based on the fan-in fan-out proportion of the hosts, *i.e.,* counting the number

[1] Mawilab Documentation: http://www.fukuda-lab.org/mawilab/documentation

of incoming and outgoing flows, and comparing their ratio with a threshold [15, 14]. With this approach, the performance in terms of detection capacity can be high, but the performance in terms of execution times can be very low [94, 15, 14]. Sampling is typically applied to solve this problem, but this involves a significant loss of information. To overcome this problem, we have used Big Data analysis techniques to analyze the whole volume of traffic with a threshold-based algorithm in the shortest possible time. In particular, we use the Apache Spark framework (Spark in the following), which is comparable to Hadoop Map/Reduce but it provides faster results working entirely in the memory.

We concentrated on a simple threshold-based detection algorithm, implemented it in Spark, and performed a large experimental evaluation of its performance by using several real traces. Our results in terms of execution time show that we achieve a processing rate down to 25 seconds for a 24h traffic trace collected on a 10Gbps link, which makes the approach able to easily run in real-time, also in much faster links. Comparing our detection performance with MAWILab [1], we show that our approach achieves fewer false negatives, which is to say that we can uncover more anomalies than the gold standard. For this reason, we also provide an improved dataset publicly on the web with labelled traffic traces, including more port and net scan events than the ones from MAWILab.

The new dataset is updated daily on our project website http://spada.comics.unina.it.

3.2 State of the Art on Scanning Activity Detection

Network analysis is a decisive aspect of security because it helps to identify potential threats within a network [88, 89, 90, 91]. Ascertaining that scanning anomalies are still predominant in these years, we moved on to conduct a study of the literary works to identify approaches for their detection.

Specifically, scientific literature has been focusing on network and port scans for several years. Generally, these anomalies are examined considering that a source of scanning activity shows a very high number of outgoing connections [94].

Zhao et al. [15] proposed an approach based on this consideration. They also proposed a standard hash-based flow sampling algorithm to cope with high connection speeds (10-40 Gb/s). *Dainotti et al.* [95] used wavelet to detect network anomalies and to precisely locate their position inside the traffic, while *Balram et al.* [96] proposed a technique based on packet count through neural networks. *Sridharan et al.* [97] compared the performance of Snort and Bro on backbone traffic and proposed a new approach based on sequential hypothesis testing. *Kim et al.* [14] described a scanning activity in terms of traffic models, working at flow-level and detecting the scanning anomalies through the analysis of variations in the models. *Chan et al.* [16] proposed two machine learning methods, useful for the construction of models detecting network anomalies starting from past behaviour. The approach described by *Wagner et al.* [98] is based on probabilistic measurement of entropy, used to indicate regularity in traffic of network flows. Traffic models used in the three last works can be sensitive to changes in the type of traffic and network. Threshold-based approaches have been widely and successfully used in the literature [99]. In this work, we want to update these approaches to the current transmission rates and network technologies, and without linking the analysis to a specific point in the network. MAWILab team proposed a system to detect attacks or anomalous events, applying a combination of four detectors with different theoretical backgrounds (see Sec. 3.3.1). They calculate a measure of distance from normal traffic using a combination of four techniques, each based on different theoretical backgrounds: Principal Component Analysis (PCA), Gamma distribution, Kullback Leibler (KL) divergence, and Hough transformation. They then use such distance to detect anomalies in MAWI traffic traces, publishing the result of the anomaly detection every day, together with the related traffic trace. This important dataset is currently used as a gold standard in the literature [17]. *Casas et al.* proposed the combined use of a Big Data framework and machine learning algorithms to achieve high performance in terms of speed of execution and detection performance. They analyzed five types of anomalies. We focus on the entire class of port and net scans and use a much simpler detection algorithm. Moreover, we uncover that MAWILab - the ground truth they, as many other works in the literature, use - is incomplete. This clearly jeopardizes the results obtained. Therefore, we also propose an improved

dataset, obtained through a combination of MAWILab and our algorithm.

3.3 The SPADA Algorithm

To detect port and net scan activities, we rely on a traditional approach by applying Big Data techniques. Specifically, we implemented an algorithm, the flow chart of which is shown in Fig. 3.1. Input data are flow-level information. In particular, we focus on the timestamp, the IP addresses, and the transport-layer ports of each flow. For our experiments, we derived the flow-level information processing the packet traces from MAWI with a tool named TIE (Traffic Identification Engine) [2]. This tool combines the packets into flows using five fields: Source IP Address, Destination IP Address, Source Port, Destination Port, Protocol. Then, our algorithm divides the flow-level trace into time intervals of custom duration (*e.g.,* 30 seconds). Afterwards, we use Spark SQL to calculate the ratio between generated and received flows in each interval and from each IP address. This proportion is then compared with a **threshold value**. The IP addresses whose proportion is larger than the threshold are marked as anomalous (see Fig. 3.1).
It is worth noting that, even if very simple, the algorithm is still quite robust. For instance, it will not mark as anomalous hosts that generate a large amount of, even unbalanced, flows (*e.g.,* servers serving popular applications) because a few responses from the other hosts (*e.g.,* the clients) are sufficient to re-balance the equation. As we will see in Sec. 3.4, this simple algorithm is able to detect port and net scan anomalies with high precision and recall and in a very short execution time if run on Apache Spark. It is also worth specifying that the algorithm cannot detect other types of anomalies or more sophisticated attacks by design in its current formulation. But 1) it does not require traffic sampling or modelling, and 2) it is able to detect all types of port and net scans, unlike others that work only for a subset of them [17].

[2]Traffic Identification Engine http://tie.comics.unina.it

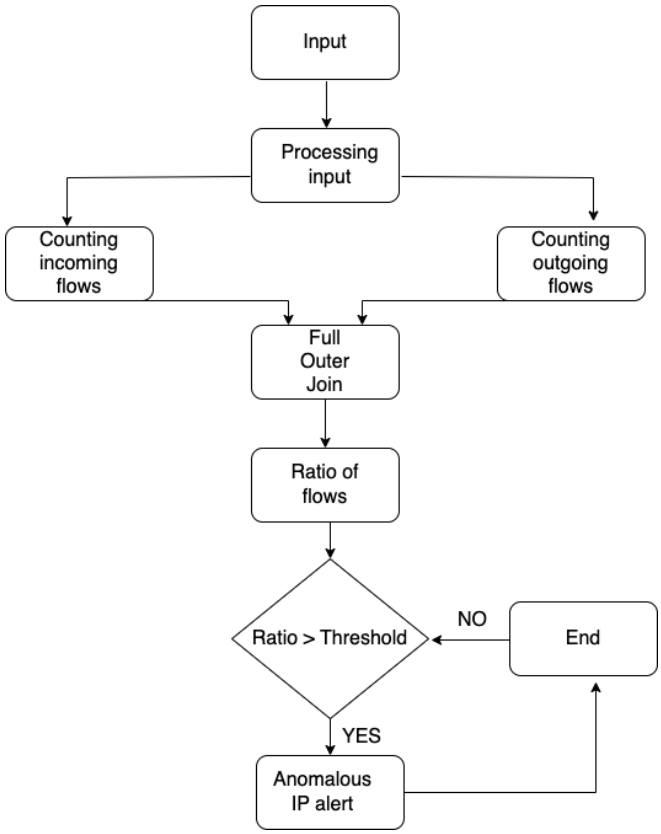


Figure 3.1. Flow chart of the threshold-based algorithm.

Using Apache Spark as Big Data technology

Apache Spark is a platform for fast and efficient distributed processing of Big Data which has almost substituted Hadoop [3]. It is very fast both in data storage and processing because it supports *in-memory* processing, *i.e.,* analyzing data directly in main memory without recurring to mass memories [100].

Spark can work in two different ways: **Batch** and **Streaming**. Both modes [4] have been used in this work. Apache Spark allows data storage in three different types of data structures: Resilient Distributed Dataset (RDD), Dataframe, and Dataset. In this work, the DataFrame structure is used, which is basically equivalent to a table in a relational database. In fact, it is also possible to execute SQL queries on DataFrames.

Apache Spark supports different programming languages, *e.g.,* Java, Python, and Scala. We used Scala for two main reasons: i) Apache Spark is built on Scala and so debugging is easier; ii) Scala is about 10 times faster than others (*e.g.,* Python) to analyze and process data due to the presence of Java Virtual Machine.

3.3.1 Data Sources

We used several real traffic traces from the MAWI (Measurement and Analysis of the Wide Internet) dataset, an archive of real traffic traces provided by the MAWI Working Group [5].

Traces are captured since 2007, and they constitute a very rich dataset that includes different applications and network conditions, and also comprise various known anomalies with global or local impact, periods of congestion, and network reconfiguration. Traffic traces considered are captured on a transoceanic link between Japan and the United States of America. Each trace contains packets captured every day from 14:00 to 14:15 in different locations inside the WIDE network. Traces of 24 and 48 hours are also occasionally collected. A typical 15-minute trace contains 300k-500k unique IP addresses [101]. We use traces captured at Samplepoint-F, a

[3] Welcome to Apache Hadoop! https://hadoop.apache.org

[4] Spark Streaming - Spark 2.3.0 Documentation. https://spark.apache.org/docs/latest/streaming-programming-guide.html

[5] Mawi working group traffic archive. http://mawi.nezu.wide.ad.jp/mawi/

link working at 1 Gbps with a current average load of 650 Mbps that has largely increased in recent years [102].

The MAWI group also created the MAWILab project: a novel approach for network anomaly detection, also implemented in a system that automatically runs every day on a traffic trace from the MAWI repository. MAWILab defines a distance from normal traffic to recognize anomalies in MAWI traffic traces. This distance is calculated through the combination of four anomaly detectors based on different theoretical backgrounds: Principal Component Analysis (PCA), Gamma distribution, Kullback Leibler (KL) divergence, and Hough transformation. These detectors only work on the IP header [103, 104].

The results of these detectors are combined to classify the anomalies into four types:

- **Anomalous** - assigned to all abnormal traffic and should be identified by any efficient anomaly detector;
- **Suspicious** - assigned to all traffic that is probably anomalous but not clearly identified by their method;
- **Notice** - assigned to all traffic that is not anomalous, but has been reported by at least one detector;
- **Benign** - all the rest of the traffic where no detector has labelled it as abnormal.

MAWILab provides the results of the analysis in two files, *Anomalous* and *Notice*. After detecting anomalous behaviours, MAWILab applies a heuristic to assign a label related to the type of anomaly. Possible labels are represented in a tree-based taxonomy, where a root is a generic event and nodes contain an anomaly label.

In the first part of this work, we used the MAWILab archive as a ground truth [105] to validate our method (see Sec. 3.3.3). Afterwards, we verified that many anomalies were not detected by MAWILab, and built a new dataset on which we performed further analysis (see Sec. 3.4.2).

It is worth noting that the MAWILab database helped and still helps a lot of researchers to evaluate the performance of novel anomaly detectors. The availability of traffic traces is already scarce. Labelled traces, including

an indication of anomalies inside them, are very very rare in our research community, and this is a great obstacle to further studies on this topic. For this reason, we decided to also evaluate MAWILab accuracy, and we finally managed to improve it. In particular, our dataset [106] includes a larger set of port and net scans not detected by MAWILab, which we believe is an important contribution to the research community.

3.3.2 Scanning Activities Over Time

Our first question in this work was if net and port scan anomalies are still actual today, so to justify further studies on this topic. To answer this question, we have conducted a longitudinal analysis of the number of anomalies detected by MAWILab over time. We collected and analyzed data regarding the anomalies detected by MAWILab in the last years and dissected them according to the type of anomaly. Fig. 3.2 shows the box plot of the ratio of scanning anomalies over the total number of anomalies, for the years from 2007 to 2017: the x-axis represents the year, and the y-axis represents the percentage of port and net scans over the total number of anomalies detected by MAWILab. The results are aggregated per year, starting from the information available day by day. The graph presents a growing trend, with a large increase in the ratio starting from 2014. This result indicates that the scanning anomalies are increasingly present in the traces, an increase that has been evident especially in recent years. This behaviour motivates our choice to focus on these types of anomalies and calls the research community for always updated data, techniques, and tools for port and net scan detection.

3.3.3 Methodology

The execution of the anomaly detection algorithm, described in Sec. 3.3, provides in the output the IP addresses that are sources of port or net scans. In the first part of our experimentation we have used MAWILab as ground truth, *i.e.,* we have compared the addresses, detected by our algorithm, with the ones in the *Anomalous* and *Notice* files provided by MAWILab.

The IP addresses detected by our algorithm that are also in one of these two files are considered true positives. The results of this analysis

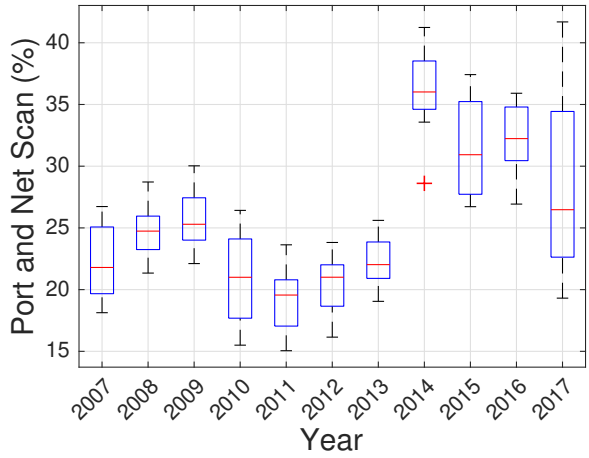


Figure 3.2. Ratio of scanning over all anomalous activities over time during the last 11 years.

are reported in Sec. 3.4.1. We considered as false positives the ones detected by us and not by MAWILab and false negatives the ones detected by MAWILab and not by us. For our positives, besides comparing with MAWILab, we also manually verified that such IP addresses are actually the source of an anomaly. This analysis evidenced the limitations of MAWILab: several anomalies we detected were not present in both MAWILab files (*i.e.,* were also not considered suspicious by MAWILab). As explained in more detail in Sec. 3.4.2, we confirmed this result with several manual inspections and automatic processing of the pcap files. Starting from this important result, we constructed a new dataset extending MAWILab with other anomalous flows and used such dataset as ground truth for another experimental analysis, reported in Sec. 3.4.2.

The detection performance of our anomaly detector is evaluated using two main metrics: **Recall**, also called True Positive Rate, and **Precision**. Several traffic traces have been analyzed in our experimentation. Results reported in the following refer to 60 traces, characterized by a duration of 15 minutes and collected from December 2017 to September 2018. Similar results have been obtained on the other traces. We have carried out mul-

tiple tests on different values of the threshold ranging from 20 to 200. A sensitivity analysis for this important parameter is reported in Sec. 3.4.3. Most of the following results are then presented for the three threshold values that are more interesting: 50, 100, and 200.

3.4 Experimental Results

In this section, we present the experimental results of the proposed approach. We start with a comparison of the detection capability of our system with MAWILab. Next, we present a new dataset that includes the MAWILab anomalies and those found by our approach. We also conduct a threshold sensitivity analysis. Finally, we perform a speed analysis of our system.

3.4.1 Using MAWILab as a Ground Truth

In this section, we analyze the results we obtained using MAWILab as a ground truth. In particular, we used the *Anomalous* and *Notice* files from MAWILab and considered only scanning anomalies, which are the ones our detector has been designed for. All anomalies that are not part of scanning activities have been removed from MAWILab results (*e.g.,* normal events, Denial of Services, Distributed Denial of Services). Since the aggregation of packets into flows does not work well for ICMP, due to unbalanced reduction compared to TCP/UDP, ICMP anomalies are not considered.

Fig. 3.3 shows the values of the Recall and Precision obtained. In particular, we report the box plot of the precision and recall obtained for all the traces considered. Such figures illustrate the median value of the recall ranges from about 0.65 to about 0.4 increasing the threshold value. The median value of the precision ranges from 0.3 to 0.65 increasing the threshold value. Using a threshold value of 100 we can obtain about 0.55 for the recall and about 0.5 for the precision.

These results may seem to indicate that a threshold-based approach does not allow to obtain satisfactory results and cannot be used to detect such anomalies. We then analyzed the false positives in more detail to validate this hypothesis. We performed a manual inspection of the pcap

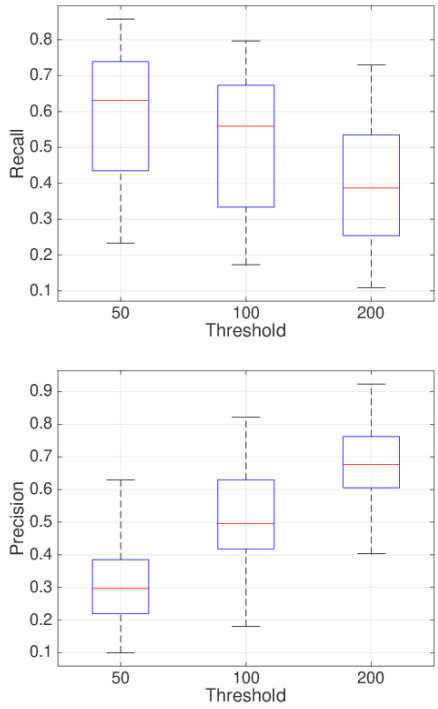


Figure 3.3. Recall and Precision of the threshold-based approach using MAWILab as ground truth.

trace starting from the false positives. Such inspection revealed that most parts of the false positives were actually a source of scanning activity and MAWILab was unable to detect them. For example, we noticed that several IP addresses generate flows with one or two packets, mostly with the TCP SYN and ACK flags set, and they receive zero or a very small number of answers. In addition, the number of useful bytes, *i.e.,* bytes of the TCP payload, is usually zero. These results have led to reconsider them as IP addresses generating port and net scans.
We confirmed this result using several traces. An important implication of this result is that we can not consider MAWILab as a ground truth as done up to now in different works in the literature.

3.4.2 Constructing and Using the New Ground Truth

We built a new dataset expanding MAWILab with the IP addresses that have been found abnormal. In particular, based on the results of the previous analysis, we implemented two heuristic rules to complement MAWILab results for all the cases in which a source of the scanning activity was not detected by MAWILab.

IP addresses that are false positives and trigger such rules are reintegrated into the true positives. The rules are the following: i) An IP address is a generator of **Net Scan** if it generates at least 20 flows towards different IPs of the same subnet; ii) An IP address is a generator of **Port Scan** if it contacts the same destination IP address on more than 10 ports. Using these rules on several traces, we have built a new dataset to evaluate the performance of the threshold-based algorithm detector. This new dataset is obtained by the union of MAWILab results and the list of IP addresses that are considered a source of scanning activities after the application of the rules implemented.

In the experiments described in the following, we compared MAWILab with the threshold-based approach and used the new dataset as a ground truth. Fig. 3.4 shows the difference between the recall of the threshold-based algorithm and the one of MAWILab as a function of the different traces analyzed. The figure shows that the recall of the former algorithm is larger than the one of MAWILab in 95% of the cases when the threshold value is 50 and in 33% of the case with a threshold value of 100. When we increase the threshold value, MAWILab starts to obtain better performance in terms of Recall.

The values of recall for the threshold-based algorithm are reported in Fig. 3.5 (a). The precision of MAWILab is clearly equal to 1 because there is no false positives.Fig. 3.5 (b) shows the precision of the threshold-based algorithm. We can see that for the intermediate threshold value (*i.e.,* 100) the precision is larger than 0.85 in about 70% of the cases, and the median value is about 0.88. Higher precision values can be obtained with higher threshold values.

Comparing the results in Fig. 3.3 and Fig. 3.5 we can see the improvement of the performance of the threshold-based approach with the new dataset. Moreover, we can say that such a very simple approach achieves very high performance, higher than MAWILab which uses a much more

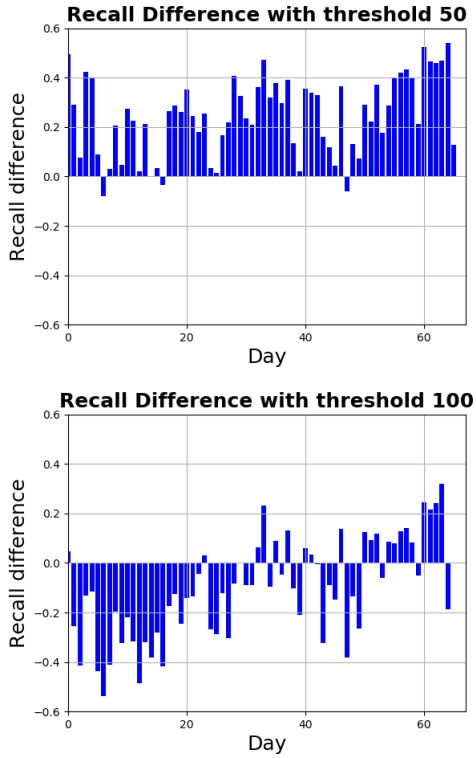


Figure 3.4. Difference between the Recall of the threshold-based approach and the one of MAWILab using the new dataset as ground truth.

complicated and therefore less observable approach.

Summarizing, in this section, we have shown that a very simple detection algorithm can obtain an even better Recall than MAWILab (in 95% of the cases with a threshold of 50 and in 33% of the cases with a threshold of 100), and this is because MAWILAb is not able to detect all scanning activities in MAWI traces.

3.4.3 Analysis of Sensitivity to the Threshold

Fig. 3.6 shows the average values of recall and precision obtained for several threshold values ranging from 20 to 200 using the new dataset as

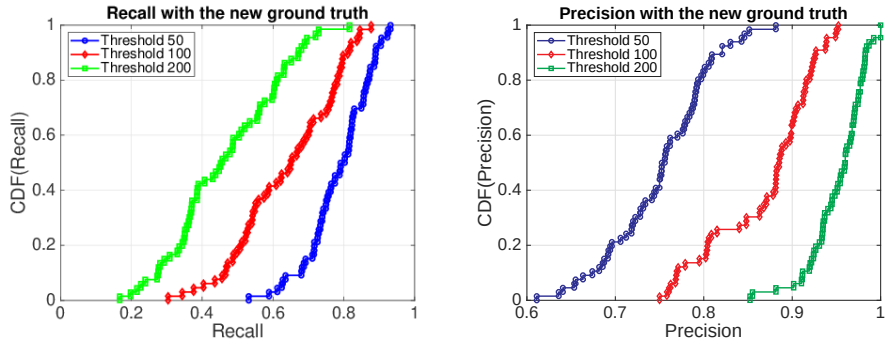


Figure 3.5. Recall (a) and Precision (b) of the threshold-based approach using the new dataset as ground truth.

a ground truth.

This figure shows that the best threshold value depends on which metric you want to maximize. For example, if false positives are annoying for human intervention in the security pipeline, a good threshold value is about 125. For such a value, an average precision of about 0.9 can be obtained. On the other side, if false negatives are more of a problem, 50 is the best threshold value to have a high recall without losing too much precision. An optimal value for both metrics is 80, which allows obtaining an average recall and precision of about 0.85.

3.4.4 Speed Analysis on Amazon EMR

We carried out several experiments on Amazon Elastic Map Reduce[6] to analyze the execution time of the algorithm using various cluster configurations and to understand the impact of the different variables under our control (number of workers, resources per VM, availability zone) on this important performance parameter.

For this analysis, we have chosen MAWI traffic traces that were captured for 24 hours on the Samplepoint-G, the main IX link of WIDE to DIX-IE with a speed link of 10 Gbps. On Samplepoint-G, MAWI provides two 24-hours traces (from the 9th of May 2018 and the 9th of April 2019),

[6] Amazon EMR – Amazon Web Services https://aws.amazon.com/it/emr/

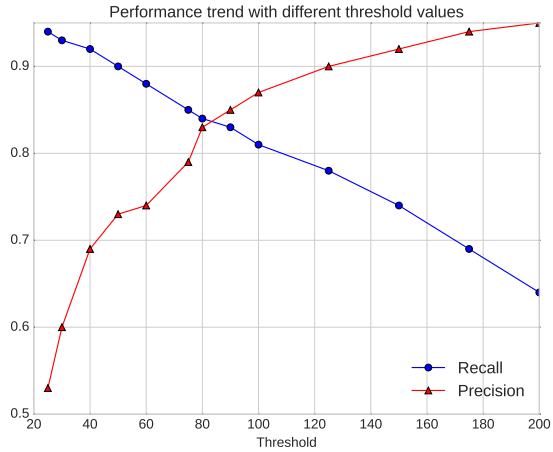


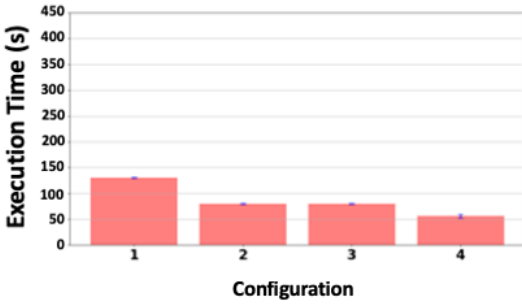
Figure 3.6. Average Recall (blue) and Precision (red) using the new dataset as ground truth for several threshold values.

where the first one is characterized by a smaller average bit rate with respect to the second one.

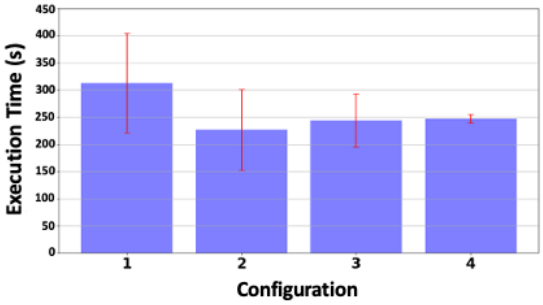
Three Cloud Availability Zones have been considered: Ohio (US), Ireland (Europe), and Tokyo (Asia Pacific). We have instanced four configurations in each of them as shown in Tab. 3.1. Fig. 3.7 and Fig. 3.8 show the execution times obtained by analyzing the two 24-hour traces in each configuration of the three availability zones.

Specifically, Fig. 3.7, related to the trace collected on the 9th of April 2019, shows that the two availability zones with the best execution times in terms of average and variance are Ohio and Tokyo. Ireland, on the other hand, has high average execution times and high variance. Moreover, OHIO and Tokyo show a clear improvement in performance as computing resources increase. Ohio has better results than Tokyo for this trace.

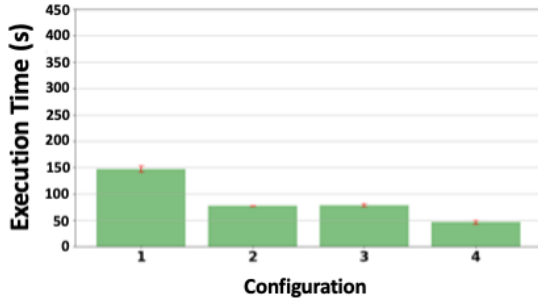
We can observe the opposite behaviour for the trace collected on the 9th of May 2018, which results are reported in Fig. 3.8. Also, in this case, Ireland has low execution times in terms of both average and variance. Like in the previous case, there is an improvement in performance when computing resources increase, although less marked than in the previous case. A significant result of this analysis is the best average time of execu-



(a) Ohio - USA



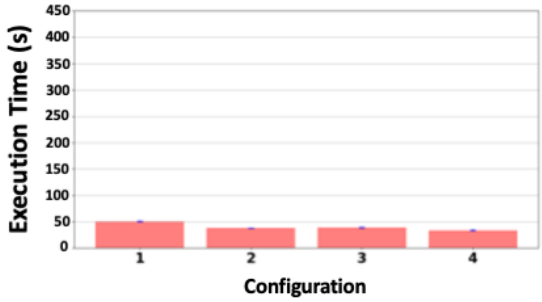
(b) Ireland - Europe



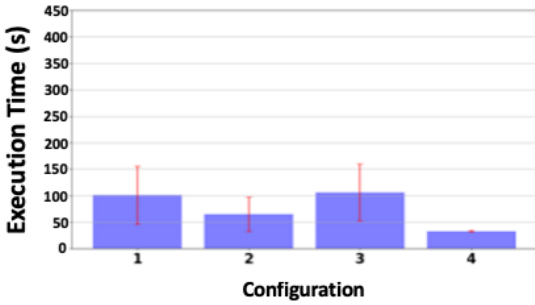
(c) Tokyo - Asia Pacific

Figure 3.7. Execution Time on Amazon EMR - first trace

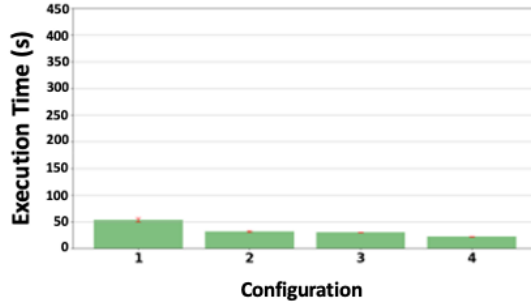
tion of our algorithm on a 24-hour track, processed in about 25 seconds. This average value is obtained in the second track in the Availability Zone of Tokyo with the configuration m5.2xlarge.



(a) Ohio - USA



(b) Ireland - Europe



(c) Tokyo - Asia Pacific

Figure 3.8. Execution Time on Amazon EMR - second trace

Table 3.1. Configurations of EMR Clusters

	Configuration	#Master	#Worker	vcpu	Memory (GiB)
1	m5.xlarge	1	2	4	16
2	m5.xlarge	1	4	4	16
3	m5.2xlarge	1	2	8	32
4	m5.2xlarge	1	4	8	32

3.5 Concluding Remarks

In this chapter, we have first shown that port and net scans, which are well-known malicious activities, are still increasingly spread in recent years. Several research efforts are currently put into complex techniques for anomaly detection, such as deep learning. These techniques can provide good detection performance, but their observability is limited. Our idea, instead, was to recover traditional detection approaches and resort to novel computing frameworks for obtaining the performance required by current network traffic.

In particular, we used a threshold-based algorithm, working at flow-level. The basic idea of this algorithm is to recognize malicious hosts looking at the ratio of their fan-in and fan-out (*i.e.,* the number of outgoing and incoming flows). Though very simple, this approach can obtain good detection performance, but it has scarce performance in terms of processing time. To cope with this problem, exacerbated in high-speed networks, we used Big Data Analytics and Apache Spark in particular. We conducted an experimental analysis with several real traffic traces from the MAWI archive. We also used MAWILab anomaly detection results as a ground truth in the first part, and a comparison in the second one, after recognizing that MAWILab fails to detect several scans.

We first evaluated the precision and recall of the threshold-based algorithm using MAWILab as a ground truth. Results were not satisfactory, in particular in terms of false positives, and pushed us to investigate more in deep. Through manual inspections of several traffic traces, we verified that such positives were actually true rather than false. This drove us to create a new dataset starting from MAWILab and complementing it with several other anomalies identified through heuristic rules devised thanks to the analyses described above.

We then evaluated the performance of the threshold-based algorithm using the new dataset as ground truth and compared obtained results with the ones from MAWILab. This analysis shows that the simpler algorithm can easily achieve higher recall than MAWILab, which is already based on much more complex algorithms. We executed the algorithm also on Amazon EMR to analyze the average execution time on three Availability Zones (Ohio, Ireland, and Tokyo) and four different cluster configurations.

We showed that the fastest availability zones are Ohio and Tokyo and that our algorithm can process 24h of traffic collected over a 10Gbps link in about 25 seconds in the Tokyo Availability Zone with an m5.2xlarge configuration.
Finally, we also set up a system that processes the traces available from MAWI every day and publishes a new dataset, including a comparison with MAWILab [106].

Chapter 4

A Study of Mirai Botnet over a Six-year Period

In this chapter, we investigate the botnet scan as a new form of port scan that has become predominant on the Internet. As explained in Section 2.2, botnet scans involve a network of hijacked network devices to identify and compromise vulnerable devices. Specifically, we present a six-year study of the Mirai botnet scans, one of the most popular botnets. To this end, TCP SYN packets that verify the Mirai signature are analyzed. The botnet master code, indeed, involves sending SYN packets to random IP addresses by setting the TCP sequence number equal to the destination address (*i.e.,* TCP.seq==IP.dst). In this chapter, we prove that the Mirai signature is still implemented by malicious actors. The number of hijacked devices involved in the scanning phase, as well as the number of TCP SYN packets, has increased over time. We also show that cybercriminals always target port 23, followed by port 2323, which receives fewer requests. Instead, ssh trends decrease over time before increasing again in 2022. Finally, we identify some ports that were never contacted until 2019 but received a large number of Mirai-type TCP SYN packets in 2021 and 2022 (*e.g.,* 9530, 5501, 7547, 5555, and so on), which were associated with new variants of the Mirai botnet.

4.1 Motivation

Nowadays, the number of devices connected to the network is continuously growing, reaching a value higher than three times the global population by 2023 [107]. With the rapid spread of IoT devices, the number of malicious attacks has increased, specifically targeting networked devices and sensors [37]. Because of the heterogeneity and lack of prompt security updates of IoT devices, they are targeted by malicious actors for creating large-scale botnets. The latter can be used to perpetrate DDoS attacks [1], mine cryptocurrency [108], steal information and create Botnet-as-a-Service [109] business models. Particularly, one of the most impactful botnets is Mirai, detected for the first time in August, 2016 [1, 39, 110, 40]. It is a type of malware that infects IoT devices (*e.g.,* routers, security cameras) and turns them into bots that can be controlled remotely. The striking features of this botnet are its spreading speed and the huge amount of traffic that may be generated during DDoS attacks [47, 111]. Mirai bots send only TCP SYN packets without performing the 3-way handshake, improving the speed and scalability of the scan [112]. Moreover, the Mirai source code [44] reveals that cybercriminals implement a signature to perform Mirai botnet scans. They set the TCP sequence number equal to the IP destination address (*i.e.,* TCP.seq == IP.dst).
A considerable amount of literature has been published on the detection of the Mirai botnet. Most of them have focused on using data from passive measurements as network telescopes [113, 114, 115] and machine learning to examine activities of botnets [116, 117, 118].
In this chapter, we provide an overview of a botnet architecture, focusing on the Mirai one, explaining its workflow and its features. We conduct an in-depth state-of-the-art study to provide a broader view of the techniques used over the years. Moreover, we explore the Mirai botnet evolution over six year-period by looking at the features of the TCP SYN packets that verify the Mirai signature (Mirai-type in the following). We analyze the changes in the number of source hosts that initiate the scanning activities, the TCP SYN packets they generate, and the relative destination ports. The latter are significant to understand how Mirai has evolved its targets over time and the occurrence of its variants.
In contrast to earlier findings [45] in the literature, we show that the Mirai

signature is applied over years to perform scanning activities and detect new possible victims. More in detail, by looking at the number of TCP SYN packets and the related source IP addresses, we first detect a small number of Mirai scan attemps in the months prior to the first detection. Then, we show a decrease in the number of Mirai scans between the end of 2017 and March, 2020 and then get a further increase until 2022. We also investigate the targeted destination ports, revealing that telnet port 23 is the most contacted over the years. Nevertheless, we show also that Mirai relies not only on telnet and ssh ports. We see evidence of new variants in the network traffic traces we analyzed.

4.2 State of the Art on Botnets

Since 2000, botnets have been the source of the majority of security failures on the Internet, and they have been used to hold the most infamous types of cyber attacks such as spamming, phishing, and DDoS. A comprehensive review broadly discusses the botnet problem, summarizes the previously published studies and supplements these with a wide ranging discussion of recent works and solution proposals spanning the entire botnet research field [26]. In particular, previous research has found that between 100 and 150 million computers were already compromised and part of a botnet at the start of 2007 [119, 120]. Also, in the year 2000, the first botnet to gain public attention was a spammer who sent 1.25 million e-mails in less than a year [121]. Other botnets that had a significant impact include:

- **Hydra** (2008), an open source botnet framework that aimed to infect routers using a built-in list of passwords with the purpose of performing DDoS attacks [122, 123].

- **Aidra** (2012), IRC-based mass router scanner exploits, also used to mine cryptocurrencies, looking for telnet ports to test with default credentials [124].

- **BASHLITE** (2014) is a malware that uses a bash vulnerability to infect Linux-based IoT devices. Most of them were video recorders (DVRs), cameras, routers and Linux servers [125].

- **Remaiten** (2016) is an IRC Bot backdoor that combines the DDoS attack of a Linux malware, Tsunami, and BASHLITE [126].

- **Linux/IRCTelnet**(2016) results from a combination of Aidra (root code), Tsunami (IRC protocol), BASHLITE (infection techniques), and Mirai (credential list) [127].

- **Persirai** (2017) adopts the Universal Plug and Play (UPnP) protocol to spread the malware to other IP cameras [128].

A brief comparison of botnet detection techniques is provided in [29]: in the following paragraph, we report studies that follow an approach using data from passive measurements, darknets, network telescopes and complex machine learning to examine the activities of botnets, in particular Mirai.
A network telescope or darknet is an internet system that allows users to observe various large-scale Internet events. Darknets or network telescopes are specific sources of traffic data, specifically a monitoring system connected to a network of IP addresses, most of which are unused [129].

Torabi et al [113] exposed a significant 26 thousand compromised IoT devices "in the wild," with 40% being active in critical infrastructure. Authors used the analysis of a Network Telescope to infer the type of malicious activity carried out by more than 25k IoT devices obtained through Shodan, a search engine for Internet-connected devices [130]. Pour et al. [115] also presented correlated data coming from a Network Telescope and further information through active measurements detecting more than 14K malicious IoT devices and the attack vector characteristic of many Botnets. Dainotti et al. [131] used the traffic monitored by a Network Telescope correlating it with two other public data sources, to analyze a type of horizontal scanning of a particular botnet. Fachka utilizes this kind of traffic to create a model for inference and prediction of DDoS attacks [132]. Araki [114] used a peculiar solution to obtain aggregated data and cluster botnets by highlighting their characteristics in relation to the hosts that have contacted. Gioacchini et al. studied network traffic to darknets and identified a methodology for grouping IP sources with related activities to detect new attacks and scanning patterns [133].

Lastly, Network Telescope payloads can be analyzed using a custom Deep Packet Inspection (DPI) technique to dissect and analyze the packets

and machine learning to classify the sources behind the campaigns and identify threat actors such as botnets, malicious attackers, or researchers, and establish a methodology to rank campaigns to prioritize our analysis [134].

Machine learning techniques in fact are often used as a detection method for botnets:

- Nakip et al. have studied the Mirai botnet developing a detection method using Neural Network to intercept SYN attacks [116]
- Cruz et al. proposed a solution to find Mirai in IoT with Machine Learning [117]
- Shao et al. offered a near real-time solution to process traffic data and detect malicious traffic through predictive algorithms[135]
- Alauthman et al. demonstrated a technique that combines reinforcement learning and a traffic reduction method to create a malicious traffic detection mechanism that constantly learns new features and achieves detection rates above 98% [136]
- Wang et al. studied a methodology to detect traffic from existing botnets and emerging ones using a detection model that performs a hybrid traffic analysis based on machine learning algorithms [137]
- Almutairi et al. proposed a combined detection technique that blends traffic flow and host activity analysis to detect emerging botnets by distinguishing malicious from legitimate traffic [118]
- Karthik et al. designed an innovative algorithm to prevent high-rate attacks by the Mirai botnet [138]
- Jaafar et al. presented a machine learning-based mechanism for detecting infected IoT devices which analyzes the network traffic and power consumption [139]

To perform early detection the aforementioned papers are mostly verifying packet size, inter-transmission times of the packets and the total number of packets that are transmitted in a time window of a certain duration. Few

papers present solutions based on fingerprinting or early signature detection of Botnet as Mirai. Only one interesting work proposes an adaptive network layer that uses characteristics of the malware behavior to scan the Mirai botnet for their signature: TCP packets in fact appear to be instantiated using the same value for the destination IP address and TCP Sequence number. The network behavior of both Mirai and Bashlite samples were analyzed and scanned for botnet signature for a period of 24 h [140].

4.3 Data Sources and Methodology

Data Sources We rely on real network traffic traces provided by MAWI (Measurement and Analysis of the Wide Internet) Working Group[1]. As for port and net scan analysis, Section 3.3.1, we select the traces captured at Samplepoint-F, a link working at 1 Gbps with a current average load of 650 Mbps that has vastly increased in recent years [102]. We analyze the MAWI traces from March, 2016 (*i.e.,* a few months before the first Mirai detection [39]) until November, 2022. However, because of the large number of traces to be analyzed over six years, we investigate two traffic traces per month randomly. The idea was to have more variability in the dataset, without being biased by a specific day of the week or month.

Methodology IP addresses of MAWI network traffic traces are anonymized using applying *Crypto-PAn* algorithm [141]. Moreover, the mapping between the anonymized and original IP address is consistent only within a single trace for daily traces. To overcome the problem of analyzing anonymized IP addresses, we pre-process MAWI traces following the methodology presented by Blaise et al. [142]. The authors propose a solution to detect botnets in the scanning and fingerprinting stages. The approach consists of generating an anomaly detection system using an algorithm based on a z-score measure. We rely on their solution to de-anonymise IP addresses belonging to 9 subnets. Afterwards, we filter all the TCP SYN packets that verify the Mirai signature (TCP.seq==IP.dst), in order to extract all the relevant features for our analysis.

[1] https://mawi.wide.ad.jp/mawi/

4.4 Mirai Botnet Evolution From 2016 Until 2022

In this section, we illustrate our experimental results. We start with an overview regarding the evolution of the Mirai occurrences over a six-year period (2016-2022) in the MAWI dataset. We inspect the TCP SYN packets that verify the Mirai signature, the relative source hosts that initiate the Mirai scans and the destination ports involved in the scanning process.

As explained in Section 2.2.2, Mirai botnet scans can be identified by checking if the destination IP address matches the sequence number in a TCP SYN packet (*i.e.,* TCP.seq == IP.dst). We investigate this pattern in 159 traces of the MAWI dataset from 2016/03 to 2022/11. Specifically, as described in Section 4.3, we analyze two traces per month to cope with the problem of a large amount of data to be processed.

TCP SYN packets and source hosts. Figure 4.1 shows the source IP Mirai-type addresses and TCP SYN packets rates over time.
The first interesting observation is related to the months prior to Mirai first detection (*i.e.,* August, 2016 [39]), where the rates of the SYN packets and source IP addresses are approximated to zero. However, these low but not negligible values (*e.g.,* in March, 2016: 221 Mirai-type SYNs of 1.2M total SYNs and 62 Mirai-type source hosts of 1.2K total source hosts), may indicate a few Mirai scanning attempts before the first real attack. Another possibility is due to TCP SYN packets that randomly have a TCP sequence number that starts as one of the subnets monitored by Mawi. The first spike occurs in August, 2016 (*e.g.,* 346K Mirai-type SYNs of 703K total SYNs, and 95K Mirai-type source hosts of 260K total source hosts on 08/15/2016), confirming the first detection of the Mirai botnet [39].The spread of the Mirai botnet is rapid from August until December, 2016, peaking in November with nearly 5.7M of 7.5M SYNs and 193K of 245K source hosts. Proof of this, several articles report an increase in this malware in November, 2016 [143, 144]. Beginning in December, 2016, the number of Mirai packets and the related source hosts gradually decrease over time, reaching a near-steady trend from November, 2017 until March, 2020. The trend of the source Mirai-type hosts is also confirmed by Antonakakis et al. [39]. From March, 2020 (*i.e.,* , 05/29/2020), the trends show a marked decrease in the rate of SYN packets. Inspecting the dataset, we notice that this decrease is due to an increase in the total number of

SYN packets while the Mirai-type ones are roughly unchanged. Another interesting finding is related to the increase of source IP addresses in that period: a higher number of Mirai-type is involved in the Mirai botnet scans with a lower amount of total ones. In addition, we observe that, as of February, 2022, one of the subnets analyzed is no longer present in MAWI's traffic traces. Therefore, the number of malicious or infected increases. Consequently, our findings show that in 2022 the number of Mirai scans is higher.

In summary, we show that cybercriminals are still implementing the Mirai signature in the scanning phase, in contrast with what other works have reported [45]. In addition, we show how the number of hijacked devices involved in the scanning phase has increased over time.

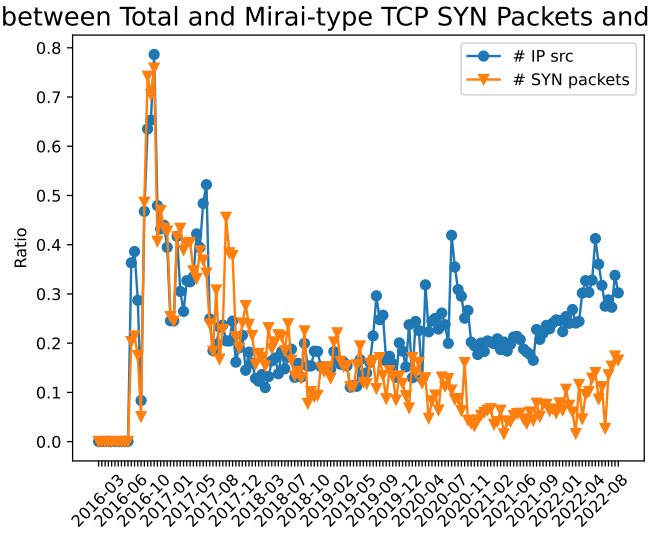


Figure 4.1. March, 2016 - November, 2022: Ratio between the number of a) total TCP SYN packets and TCP SYN Mirai-type packets; b) total source IPs and source Mirai-type IPs

Analysis of ports involved. To further investigate the scanning patterns of the Mirai botnet, we look more in-depth at the TCP SYN Mirai-type packets by examining the destination ports to which they are directed.

As described in Section 2.2.1, during the scanning phase of the Mirai workflow, cybercriminals exploit vulnerabilities principally in ssh (22, 2222) and telnet (23,2323) ports. Figure 4.2 shows the number of TCP SYN Mirai-type packets received on telnet and ssh ports over six-year period. What is striking in this plot is that before and during the first Mirai spread (*i.e.,* August, 2016), port 23 is by far the most scanned, followed only by port 22 with far fewer requests (around 10 requests) received on only a few days. Since September, 2016, also the 2323 port has started to be targeted as well. In particular, the trends of the two ports related to telnet - 23, 2323 - follow almost the same pattern, except that port 2323 has a smaller order of magnitude of requests. This is because the scanner module (scanner.c) of the Mirai source code is implemented to send an SYN packet to a random address one time out of ten to port 2323, and the remaining nine times to port 23.
Moreover, Figure 4.2 shows that the number of TCP SYN packets redirected to the ssh - 22, 2222 - ports peaks in September, 2017. As for telnet port trends, ssh ports follow almost the same pattern, except that port 2222 has fewer requests. In contrast to the telnet ports, from November, 2019 both ssh port trends started to decline rapidly over time, indicating that the cybercriminals tested the vulnerability of other ports. However, beginning in July, 2022, the two ports were targeted again, especially port 22. To inspect the reason behind the rapid decline in the trends related to the ssh ports, we examine the number of destination ports of TCP SYN packets, shown in 4.2. More specifically, we analyze ports that receive at least two TCP SYN to exclude cases where randomly the TCP seq matches the IP dst. The trend is in contrast to previous studies that claimed that the Mirai botnet scans were targeted more to telnet and ssh ports [39]. Further on, as of November, 2018, the plot shows a peak and a subsequent increase in ports receiving at least 2 TCP SYN packets. Fewer ports, instead, using the Mirai signature are contacted as of January, 2021.
To further investigate the ports adopted in the scanning phase and the increase in the number of ports contacted as of the end of 2018, we focus on the most 15 contacted ports per year. Confirming the trends in Figure 4.2, the telnet ports 23, 2323 - are in the first three positions every year. Another interesting observation is related to the considerable increase in requests for ports 37215 and 52869 at the beginning of 2018,

shown in Figure 4.4. Also, these two ports had never been contacted in previous years in MAWI traces we have analyzed. These two ports may be related to the evolution of Satori, which started its propagation at the end of 2017 [57, 56]. In this plot, we show that, except for a decrease around a few months in 2021, this variant is still active and adopts the Mirai signature. Moreover, we found that port 443, related to HTTPS services, is the port most contacted in the months prior to the first Mirai detection (August, 2016), also more than the telnet ports. To the best of our knowledge, this analysis is the first to exhibit this outcome. In addition, we see a significant increase in the TCP SYN Mirai-type packets for ports 80, 8080, 8081, and 8088 and other ports related to the HTTP services at the beginning of 2018. This result is also confirmed by the evolution of Repair and Wicked botnets, starting their propagation at the end of 2017[61, 62]. In our dataset, we see also evidence of the TR-069 and Android Debug

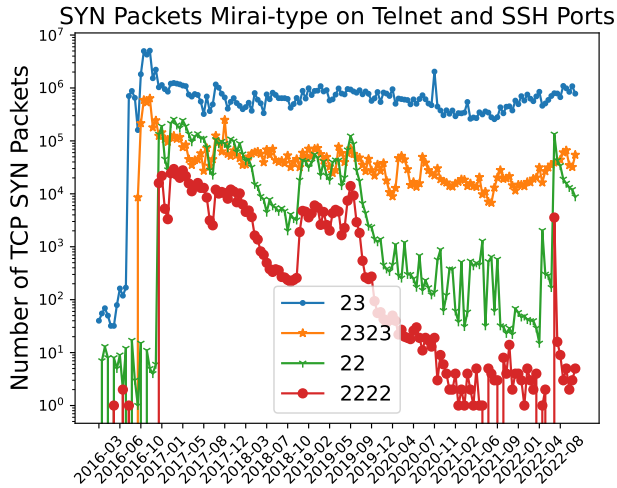


Figure 4.2. Number of SYN Mirai-type packets on Telnet and SSH ports

Bridge Mirai variants [145], by looking at the ports 5555 [2] and 7547[3]. We do not report their trends for brevity. More in detail, port 5555 receives

[2]Working port of ADB debugging interface on Android device.

[3]Port associated with TR-069 - application layer protocol for remote management of end-user devices.

a considerable number of TCP SYN Mirai-type over six years period. Instead, port 7547 records a decrease from 2017 and 2021 and a significant increase in 2022. In the end, we find that ports 5501 and 9530 starts to be targeted at the end of 2020, getting more and more requests in 2021 and 2022. The considerable number of TCP SYN packets to port 5501 may be related to the scans carried out by Priority Threats actors reported by Juniper Networks[4]. The huge number of scans to the 9530 port, instead, may be related to the LeetHozer botnet[5].

In summary, we show that Telnet port 23 is the most contacted one over a six-year period. In addition, as of the end of 2018, multiple ports have been targeted related to new variants of Mirai.

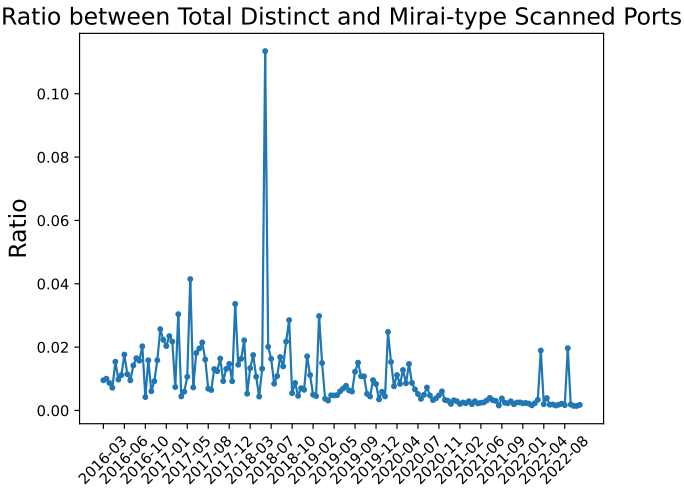


Figure 4.3. Ratio of the number of total distinct ports to Mirai-type ports receiving at least 2 Mirai-type TCP SYN packets

[4] https://blogs.juniper.net/en-us/security/priority-threat-actors-adopt-mirai-source-code.

[5] https://blog.netlab.360.com/the-leethozer-botnet-en/

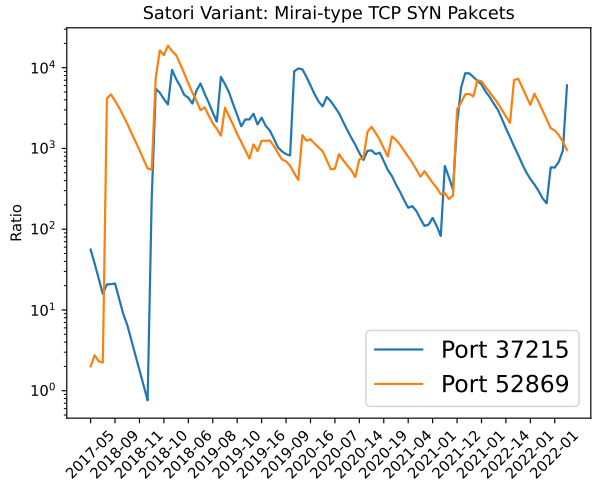


Figure 4.4. Number of TCP SYN Mirai-type packets on 37215 and 52869 ports - Satori variant

4.5 Concluding Remarks and Limitations

In this chapter, we conducted a study of Mirai botnet scan detection over a six-year period. Specifically, we inspected MAWI network traffic traces by examining the Mirai signature (*i.e.,* TCP.seq == IP.dst). We showed that the Mirai signature is still implemented by malicious actors, in contrast to what was reported by other works [45]. Particularly, we see an increase over time in the number of hijacked devices involved in the scanning phase as well as the number of TCP SYN packets. Sticking to the Mirai code, we looked at the number of requests Mirai-type for telnet (*i.e.,* 23, 2323) and ssh (*i.e.,* 22, 2222) ports. We show that port 23 is always the most targeted by cybercriminals, followed by 2323 with fewer requests. Trends in ssh, instead, decrease over time and increase later in 2022. In addition, we identified some ports that were never contacted until 2019 but with a large number of TCP SYN Mirai-type packets in 2021 and 2022 (*i.e.,* 9530, 5501, 7547, 5555, etc.), related to new variants of Mirai botnet.

The first limitation of our interference ability was that the IP addresses of MAWI datasets were anonymized and, consequently, it was not possible

to analyze how the IP addresses of hijacked devices change over time. Another limitation is related to the possibility of analyzing only 9 subnets. Indeed, as we explained in Section 4.3, to check the Mirai signature in the TCP SYN packets, we performed the operation of de-anonymization by applying the method used by Blaise et al. [142]. Finally, the duration of the network traffic trace is only 15 minutes per day. Therefore, we showed only a portion of Mirai's traffic during the day. Nevertheless, we believe that the investigation of Mirai scans by looking at the Mirai signature can help network operators to reduce the number of harmful devices. Future studies should aim to replicate results in a larger dataset, looking at how malicious IP addresses change over time.

Chapter 5

Local and Public DNS Resolvers: Timing and Security Performance

In this chapter, we analyze the behaviour of DNS resolvers provided by three main Italian ISPs and contrast them with open, public resolvers provided by Google and Cisco. We consider two aspects. The first one is the time spent performing a query and obtaining a response from the resolvers, which has a considerable impact on the performance of most applications on the Internet. The second one is the capability to recognize domains associated with malicious activities, blocking related requests to protect users. The DNS response time is generally shorter for local resolvers since they are closer to the users. On the other hand, public resolvers are typically considered more efficient in detecting malicious domains. We performed a large number of DNS queries towards the different resolvers, both local and public, using different sets of domain names and different Internet access networks from main Italian providers. Our results confirm that the response time of local resolvers is shorter than that of public ones. However, they also show that, unexpectedly, the protection level of local resolvers is largely comparable with the one of public resolvers. Consequently, you do not have to trade off security against performance. In addition, we study the impact of DNS over HTTPS, and we unveil the different mechanisms implemented to block users from accessing malicious domains and assess

the impact of caching on the obtained results.

5.1 Motivation

Translating domain names into their associated IP addresses is the main task of the DNS. This is an indispensable component of the Internet, distributed over a global network of servers that are constantly in communication with each other to bring users to their websites or network resources [146]. DNS is an emerging topic in literature for its decisive impact on the performance of almost all internet activities. Plenty of previous work focused on the performance of DNS resolvers from several vantage points, also obtaining contrasting results [147, 148, 149, 75, 150]. On the other hand, millions of new domain names are registered every day, including the ones used by attackers to redirect victims to malicious destinations like malware, spam, phishing, and other insecure contents [151]. DNS traffic contains several meaningful features to identify domain names associated with such malicious activities. This is why DNS is more and more used to protect users from possible threats [152], together with other techniques based on *e.g.,* machine learning and artificial intelligence [153, 106].

In this chapter, we study the behavior of resolvers provided by three Italian commercial ISPs (TIM, Wind, and Fastweb) - and two public ones offered by Google and OpenDNS. We collected a dataset performing a high number of queries towards these resolvers, looking at domain names from four different categories: Top 1 million, Command and Control (C&C), Malware, and Phishing. We also considered encrypting DNS queries over HTTPS, a.k.a. DoH [154, 155]. Our analysis focuses on two relevant aspects. The first one is related to the *Response Time* (RT), which is the time required to get a DNS response after issuing the request. RT has a strong impact on the performance of most Internet applications. The second one concerns the *Response Code* (also RC or RCODE in the following), included in the DNS response, useful to figure out if the DNS server provides an IP address or not, *e.g.,* to protect users from malicious hostnames. For example, an "NXDOMAIN" message can indicate either that the domain name does not exist or that it is related to a malicious activity, blocked by the resolver. Google and OpenDNS feature several strategically located resolvers that rely on anycast. But their response time is typically

higher than the local resolvers [149], which are closer to the users. On the other hand, Google and OpenDNS are expected to detect more malicious domain names because they have a wider view of the network traffic than local resolvers. The results of the response time analysis provide many interesting insights. First, we confirm that local DNS resolvers are faster than public ones: Fastweb is up to 86ms faster than Google, and up to 129ms quicker than OpenDNS. Google is generally faster than OpenDNS, contrary to what has been reported by other works [149, 156, 157]. Additionally, we uncover that there are no significant time differences between DNS and DoH both for Google and OpenDNS. We also studied the effect of the caching in the home router and saw that up to 40% of the domains can be cached at a such router for up to 4 hours. The results related to the response codes show that resolvers use different approaches to block dangerous destinations, *e.g.,* Fastweb, TIM, and Google return "NXDOMAIN", while Wind provides a "0" RC, but the IP address is related to a courtesy page. Also, OpenDNS achieves slightly higher performance with malware and phishing domains. This is somewhat expected as we use domain names collected by Cisco Umbrella, the same company owing OpenDNS. Furthermore, we do not find significant differences between DNS and DoH both for Google and OpenDNS. The most unexpected result however is that all the resolvers considered protect from malicious domains with comparable performance. That is to say that it is not necessary to trade off performance and security, at least not anymore.

The contributions of this work can be summarized as follows: i) we perform an analysis on the DNS resolvers using a large dataset of domain names, including most popular as well as malicious domains; ii) we show that Italian local resolvers generally provide a comparable level of security of open, public resolvers used from all over the world and deployed by large companies like Google and Cisco; iii) we unveil the mechanisms employed by the resolvers to protect users; iv) we show that using local resolvers can save up to about 130ms on average for each DNS resolution; v) we contrast previous findings in the literature (*e.g.,* [149]); vi) we show the impact of caching on obtained results.

5.2 State of the Art on the Performance of DNS Resolvers

In recent years, there has been an increasing number of works studying DNS resolvers [147, 148] also with a focus on a comparison between public and commercial ones [149, 75, 150]. An interesting analysis of the response times and the addresses returned by the local resolvers against Google and OpenDNS was conducted by Ager et al. [149]. They claim that Google and OpenDNS, in some cases, outperform the local resolvers in terms of the observed response times. This result is in contrast with ours: local resolvers we consider in this work typically show response times smaller than public resolvers. Moreover, we analyzed in much more detail the capability of such resolvers to block dangerous domains. Current literature pays particular attention to open DNS resolvers. Kuhrer et al. performed a long-term, large-scale analysis in order to study the changes over time and classify the resolvers according to several features like device type and software version. They have also deepened the DNS responses correctness querying the "A" record of 155 domains, divided into 13 categories, towards 22 million open DNS resolvers [75]. Dagon et al. carried out a similar analysis, but they only analyzed some samples of the DNS responses, and they did not provide detailed statistics except for Chinese splash pages [158]. Another interesting analysis on open resolvers was conducted by Park et al. [150], who compared their previous findings in 2013 showing that the number of resolvers providing incorrect responses is almost the same, those providing malicious responses have increased.
Companies have been focusing on comparing local and public resolvers. They claimed that it is more convenient to use public DNS than local ones, both for their response time and security protection [156, 157]. Our results show that their protection level is largely comparable while local resolvers are always faster than public ones. Google is generally faster than OpenDNS from our vantage points, and this outcome is also confirmed by other works [156, 157]. DNSPerf, instead, shows the opposite behavior: OpenDNS is faster than Google [159]. Different protocols have been implemented to encrypt DNS queries. They provide security and privacy, and they allow clients to send DNS queries to public DNS resolvers, preventing the ISP from seeing such queries [74]. Several works analyzed

possible differences in performance between DNS and other encrypted DNS protocols, like DoH. Some of them claimed that the DNS response times are higher than those of DoH [74, 155]. Other works, instead, claim that it is not simple to choose the best DNS protocol for all clients because DoH response times can be both longer and shorter than the traditional one [160]. We compare the performance of standard DNS (UDP port 53) and DoH and do not observe significant differences.

5.3 Datasources and Methodology

In this section, we describe our experimental setup and the datasets we used. We relied on the PyDig [161], a tool written in Python, to perform queries towards DNS servers and exercise various existing and emerging features of the DNS protocol. This tool features queries through DNS over TLS and DNS over HTTPS. We queried a considerable number of domain names, divided into two different categories. The first one is related to Cisco's Top 1 Million lists, while the second one contains malicious domains collected by Cisco Umbrella analysts.

Concerning the first category, Cisco provides, every day, the list of the first one million domains (Top 1 Million) most commonly queried from all over the world to OpenDNS resolvers [1] [162]. We relied on this list following the suggestions provided by Scheitle et al.[2] [163]. To examine more in-depth the performance achievable in a wide set of conditions, and, consequently, for the purpose of having variability in our data, we created the dataset as follows. We pulled out the first and last 10,000 rows from the top 1 Million list, resulting in a dataset of 20,000 domains with the most and least common ones. We suspected that the most popular domain names might be benign with a higher probability, unlike the less popular ones that might be benign with a lower probability. This assumption was also derived from the work by Scheitle et al., stating that the Cisco Umbrella list contains test domains or several domains with non-authorized gTLDs [163]. In addition, the clients adopting OpenDNS are not only

[1] https://umbrella.cisco.com/blog/cisco-umbrella-1-million

[2] The authors observed that there is a high daily fluctuation in the Top1Million. Therefore, it is important to specify the day on which we downloaded the Top1Million file: 09/10/2020.

PCs but also mobile and IoT devices. To further investigate the nature of the first and last 10k domain names included in the Top 1 Million. We rely on the "Domain Status and Categorization" API. This API belongs to the groups of APIs provided by Cisco Umbrella Investigate and used in several works [164, 162]. It returns the domain status, indicating whether a domain has been flagged as malicious by the Cisco Security Labs team (score of -1 for status), it is believed to be safe (score of 1), or it is still undecided (score of 0)[3]. 78.5% of these domains is benign, a small percentage (0.1%) of them is malicious, and the nature of the rest of the hostnames (21.4%) is not further specified by Cisco Umbrella. In conclusion, the Top1Million dataset mostly consists of benign hostnames. We will refer to this dataset simply as TopCisco in the following.

The second category of the dataset is characterized by malicious hostnames collected by Cisco security analysts. Thanks to a collaboration with Cisco, we had access to a wide list of malicious domain names blocked by the Cisco Umbrella platform. The list is split into three different datasets, containing different kinds of malicious activities: *C&C* - domains associated with a Command & Control systems of botnets; *Malware* - domains associated with malware threats; *Phishing* - domains associated with phishing pages.

In summary, the datasets adopted for our analysis are four: **TopCisco**, characterized by 20,000 domains, mostly benign; **C&C**, characterized by 16,021 domains associated with Command & Control activities of botnets; **Malware**, characterized by 81,217 domains associated with malware activities; **Phishing**, characterized by 658 domains, associated with phishing pages. We expect that the OpenDNS resolver shows a higher protection level than the other ones because the datasets contain domain names blocked by Cisco Umbrella, which is also the company owing OpenDNS.

5.3.1 DNS Resolvers Analyzed

We selected three commercial Italian ISPs and two public DNS resolvers by Google and OpenDNS. **Google** is a free, public, and open resolver adopted by a high number of users, available at 8.8.8.8 and 8.8.4.4

[3] https://docs.umbrella.com/investigate-api/docs/domain-status-and-categorization-1

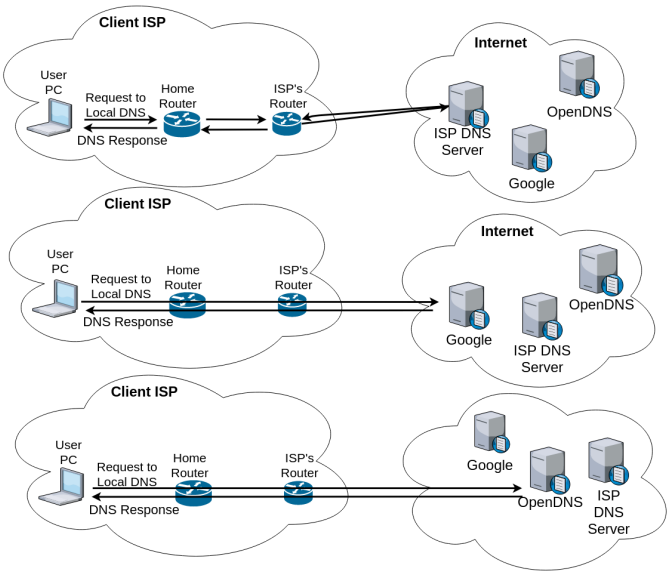


Figure 5.1. Three scenarios of our system architecture

IP addresses. It provides two different DoH APIs [165]. We utilized the one at endpoint `https://dns.google/dns-query`. **OpenDNS** is a free, public, and open resolver, founded in 2005 and currently owned by Cisco. It is available at 208.67.222.222, 208.67.220.220, 208.67.222.220, 208.67.220.222 IP addresses and it also provides DoH endpoints to implement the DNS over HTTPS [166]. We used endpoint at `https://doh.opendns.com/dns-query`. As local resolvers, we adopted the ones provided by **TIM**, **Wind**, and **Fastweb**, three of the major and most used Internet Service Providers (ISPs) in Italy. We carried out tests in two different cities - Naples and Rome. As reported by the Fair Internet Report, `https://fairinternetreport.com/Italy/Rome`, in both cities TIM, Wind, and Fastweb are three of the fastest providers in Italy. We conducted all experiments under three residential fibre Internet access networks by these operators.

Figure 5.1 shows how DNS queries have been performed from the same ISP network to the different resolvers. In the first case (top diagram in the figure), queries are issued towards the local resolver, *i.e.*, the default

resolver provided by the DHCP of the home router. We remark that for the security analysis, it is not important to distinguish the responses arriving from the Home Router (*i.e.,* from the cache) from the responses arriving from the ISP resolver because the former resolver gets information about domain names from the ISP one. In the response time analysis, instead, the time from the User PC to the Home Router, connected to each other by a wi-fi connection, is typically smaller than the time between such PC and the ISP resolver. In the second case, in Figure 5.1 (middle diagram in the figure), the client sends DNS requests to Google resolver directly. This scenario differs from the previous one because the Home and ISP routers are crossed from the DNS request only at the IP level. Thus, they are not involved in the dynamics of the application/DNS layer. The third scenario is similar to the second one except that requests are issued to the OpenDNS resolver.

The second and third scenarios were applied to both DNS and DoH endpoints. The three scenarios were repeated under the three ISP networks: TIM, Wind, and Fastweb. In summary, under each ISP network, the queries were sent to: local DNS, Google DNS and DoH, and OpenDNS DNS and DoH. We performed the queries towards Google and OpenDNS resolvers under three ISP networks, aiming to investigate the impact of the network on the DNS resolvers.

5.4 Experimental Results

The DNS queries have been issued with Pydig specifying the DNS resolver address and the record type (*e.g., pydig(www.example.com, 8.8.8.8, A)*). The responses obtained by this tool include information related to a Resource Record In particular, it returns the following fields: **Response code** - specifies the outcome of the response. There are some common return codes that can be returned when issuing a DNS query (*e.g.,* '0', '3', etc.) [4]. Other rare codes can appear in a few circumstances [4]. **IP** - contains one or more IP addresses associated with the requested domain name. It may also be null or contain a CNAME field. **Size** - represents the total size of the DNS response. **TTL** - specifies the Time To Live

[4] https://support.umbrella.com/hc/en-us/articles/232254248-Common-DNS-return-codes-for-any-DNS-service-and-Umbrella-

Table 5.1. Results of the equations 4.1, 4.2, 4.3, 4.4, 4.5. All values are expressed in ms.

i	j	Min	Median	Mean	Std_Dev	10th	25th	75th	90th
Fastweb	**Google DNS**	82	83	86	104	90	89	76	116
Fastweb	**OpenDNS DNS**	118	126	129	146	129	131	117	244
Google DNS	**OpenDNS DNS**	36	42	43	425	38	42	40	127

(TTL) of how long a record is cached in a DNS server. **Response time** - is the total amount of time to perform a query and receive the response. **Exception** - is true if an exception occurs during the DNS request.

We focused on the response code and the response time fields. The response code has been selected to study the resolver's capability to distinguish benign and malicious domains. The analysis of the response time is aimed at understanding the timing performance of a DNS resolver.

5.4.1 Analysis of the Timing Performance

In this section, we focus on the response time of the DNS queries. Since PyDig is written in Python, an interpreted language, we verified the impact of the tool on the obtained values. We evaluated the difference between the response time provided by the Tshark tool and the one provided by PyDig on a sample of domain names. The average difference we observed in our setup is about 0.002s. This value may be significant for some experiments. However, it does not affect our analysis because we are using the same tool for each experiment, and we are interested in comparing the different resolvers. Figures 5.2, 5.3, 5.4, 5.5 show the comparison between local ISPs, Google (DNS,DoH), and OpenDNS - DNS, DoH - resolvers under the three ISP networks and for the four datasets.

We can make some interesting considerations. The first observation is related to the response times of the local resolver, which are smaller than those of public ones for each dataset and under the three networks. This is in contrast with the results reported by other works [149, 156, 157]. Another interesting finding is related to the Google resolver speed compared to that of OpenDNS. In all the figures mentioned above, Google-DNS and Google-DoH have slightly smaller response times than OpenDNS-DNS and OpenDNS-DoH. Besides that, under each network and for each dataset, Google DNS and DoH present similar response times, as, for example,

shown in the overlap between the curves representing them. This outcome is in contrast with previous findings [74]. Similar behavior is shown by OpenDNS DNS and DoH. Some exceptions are visible for OpenDNS under the Wind network. In particular, Figures 5.3 (b), 5.4 (b), and 5.5 (b), show that OpenDNS-DNS is slower than OpenDNS-DoH in this case. Based on these considerations, we can infer that local DNS resolvers are faster than public ones, and, from our vantage points, Google is slightly faster than OpenDNS.

We also evaluated the distance between the curves aiming to compare response times between public and commercial resolvers to see how much time we can save by using local resolvers instead of public ones. For the sake of brevity, we report only the results related to Figure 5.4 (c). The differences between the curves have been calculated with the (4.1), (4.2), (4.3), (4.4), and (4.5), where F(RT)(t) is the response time function and i and j represent the various adopted resolvers. In particular, we computed: the difference between the minimum values from the two CDFs in (4.1), between the median values in (4.2), the mean values in (4.3), the standard deviation values in (4.4), and the 10-25-75-90th percentile values in (4.5). The results related to the four equations are reported in Table 5.1.

$$RT_i - RT_j : F(RT_i) = min(F(RT)) \qquad i \neq j \qquad (4.1)$$
$$RT_i - RT_j : F(RT_i) = median(F(RT)) \qquad i \neq j \qquad (4.2)$$
$$RT_i - RT_j : F(RT_i) = mean(F(RT)) \qquad i \neq j \qquad (4.3)$$
$$RT_i - RT_j : F(RT_i) = std\_dev(F(RT)) \qquad i \neq j \qquad (4.4)$$
$$RT_i - RT_j : F(RT_i) = [10, 25, 75, 90]thF(RT) \qquad i \neq j \qquad (4.5)$$

An interesting aspect is related to the `Min` column (as shown in Table 5.1) where, in the best case, the Fastweb client is 82ms faster than Google DNS, 118ms faster than OpenDNS DNS, and Google is 36ms faster than OpenDNS DNS. On average, Fastweb is quicker by 86ms and 129ms than Google DNS and OpenDNS DNS, respectively. Google is 43ms faster than OpenDNS DNS. We report only the results related to the DNS protocol because those obtained with DoH are similar in most cases and not reported for brevity. We also looked at the impact of security on perfor-

mance, analyzing the trend of the response times split by the four datasets and the response codes. We have not found relevant differences in the results obtained from this analysis.

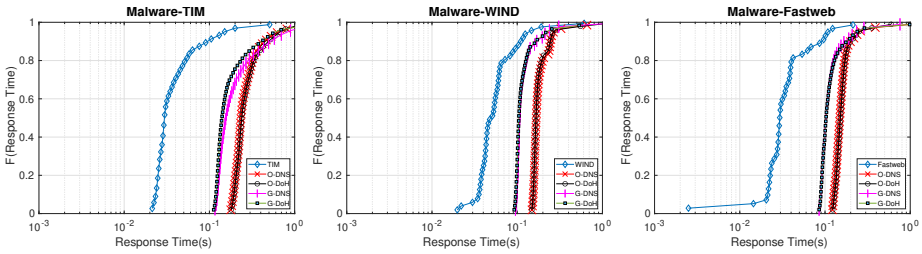


Figure 5.2. Malware - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

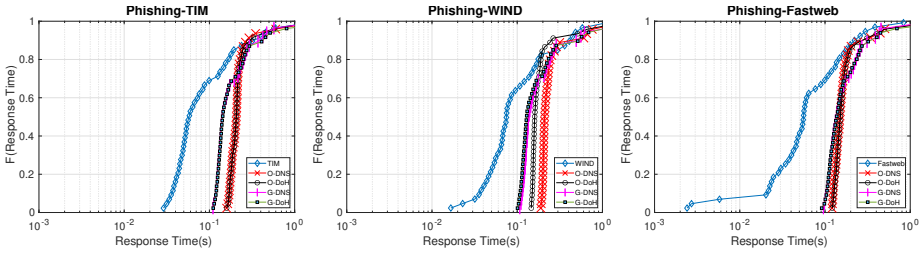


Figure 5.3. Phishing - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

A further remaining issue relates to the impact of the DNS caching mechanisms on the obtained results. DNS caching can occur at different levels in a DNS lookup. The first two steps involve the operating system and the browser, and so they are related to the client. The other levels are associated with the resolver, root server and TLD server. To investigate the DNS caching impact on our experiments, we extracted 100 domains from the datasets, characterized by different TTL values, and therefore, presumably, different caching times [167]. We executed queries at different time intervals. In particular, the first execution took place after restarting the home routers of the clients used for the experiments (time 0, called *baseline* in the following). Then, we performed queries after 1 minute, 10 minutes, 1 hour, 4 hours and 24 hours. Figure 5.6 shows the CDFs of the

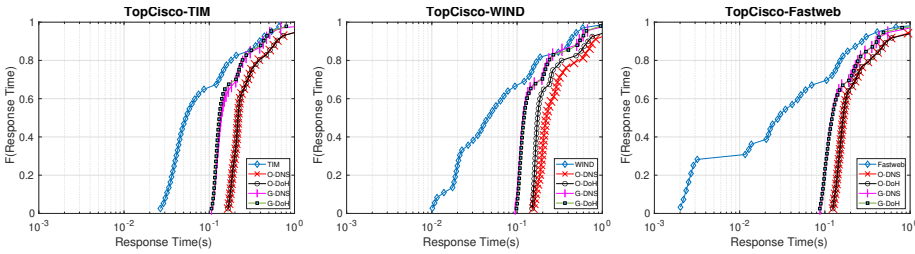


Figure 5.4. TopCisco - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

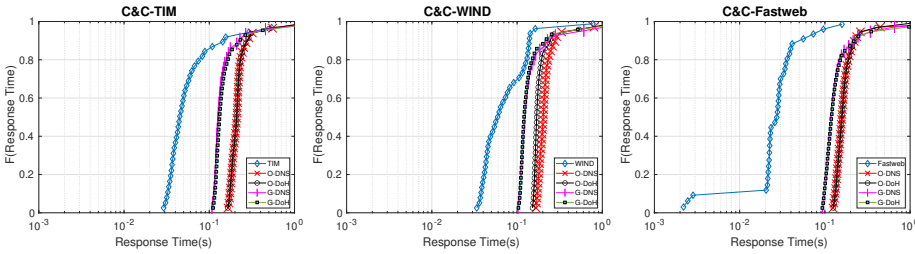


Figure 5.5. C&C - Comparison local DNS with Google and OpenDNS for (a) TIM, (b) Wind, (c) Fastweb

response times obtained. The response times related to 1m, 10m, 1h, and 4h are larger than the baseline and 24h. The number of domains kept in the cache is large for times up to one hour, decreases after 4 hours and reaches almost zero after 24 hours. After manual analysis, we have also reported a black dotted line in the plot to illustrate the DNS responses coming from the home router (response time equal to 4ms). Excluding domains cached in the router and comparing Figure 5.6 and Figure 5.4, we can claim that the local resolver is still faster than the public ones and considerations reported in the previous sections are still valid. Similar experiments and comparisons were also performed with TIM and Wind clients obtaining comparable considerations.

5.4.2 Analysis of Security Performance

The purpose of the response code analysis is to investigate the level of security service provided by the resolvers, to study how much they can

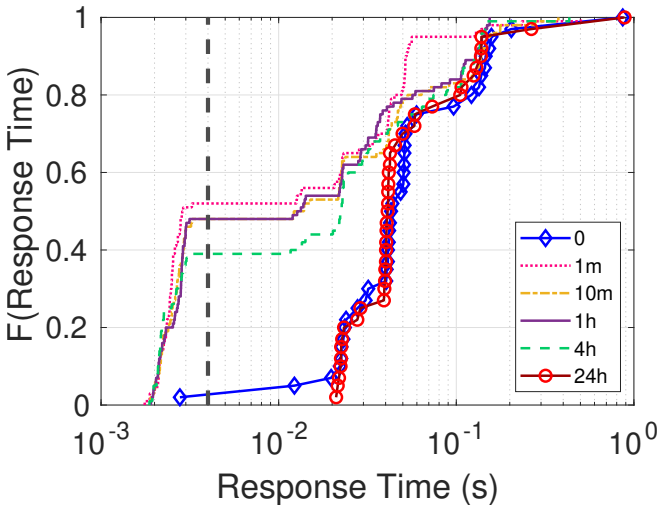


Figure 5.6. Caching

Table 5.2. Response codes identified in our experiments [4]

DNS RC	Description	Class
0	DNS Query completed successfully	Positive
3	Domain Name does not exist	Negative
5	The server refused to answer	Negative
2	Server failed to process the query	Negative
Null	Other exceptions	Negative

protect users from possible threats.
Table 5.2 illustrates the DNS response codes obtained in our study. We identified four types of response codes and other exceptions due to network failures. The results differ from the ones shown by Park et al., who claim that codes "0", "3", "5", and "2" decreased, and the remaining ones increased in the last years [168]. In particular, the "0" label occurs when the query is completed successfully [4]. In this label, we also included the cases in which the DNS server does not know the IP address of a host, and it returns another domain name through which the same destination can be reached (CNAME). In addition, in the same label, we added the case in which the response code is "0", but the DNS servers provide the SOA record (Start of Authority). Since the label "0" implies that the query was executed correctly and the DNS response has the IP addresses or information useful to obtain them, it represents the positive class of the confusion matrix. The latter is calculated to obtain a synthetic measure of the security capability of the resolvers [169]. The other response codes, "3", "2", "5", and "Null" are errors. Therefore, we classify them as belonging to the negative class.
We summarize the occurrences of the response codes through bar plots showing a graph for each resolver adopted (locals, Google, OpenDNS), as depicted in Figures 5.7, 5.8, 5.9. There are four subplots in each figure relating to the four datasets: TopCisco, C&C, Malware, and Phishing. The bars in each subplot refer to a different network ISP: TIM, Wind and Fastweb. Results obtained with DoH are the same as those obtained with standard DNS and are not reported for brevity. Figures 5.7, 5.8, 5.9 show that, for each provider, when the dataset is the TopCisco, the occurrences of "0" are more than 10K; the number of "3" is above 1K; the amount of "2", "5" and "None" differs for each resolver. The TopCisco results are in line with the expected ones because this dataset is mainly characterized by benign and existent hostnames. Therefore, we suspected that the number of "0" codes, and thus the number of queries to legitimate clients, was greater than the others. In addition, we can remark that, for the provider Wind, the number of "3" codes is lower than the other two local providers because there is a slightly higher amount of "0" and "Null". We point out that the highest percentage of "0" codes in the Wind provider is related to the courtesy page.

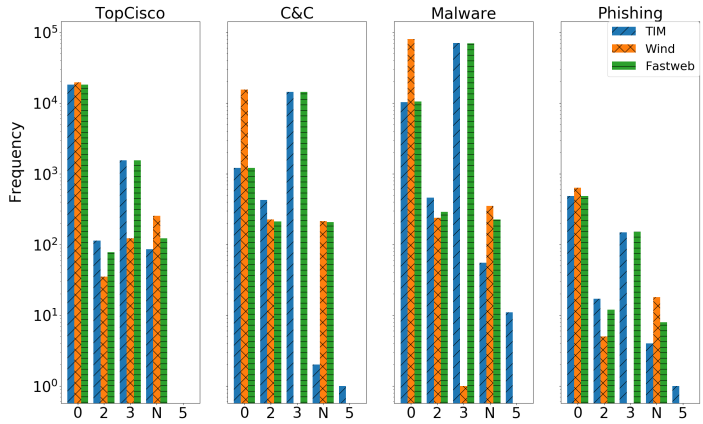


Figure 5.7. Results obtained under three different ISP networks using the resolvers provided by their ISPs (*i.e.*, using the local resolver).

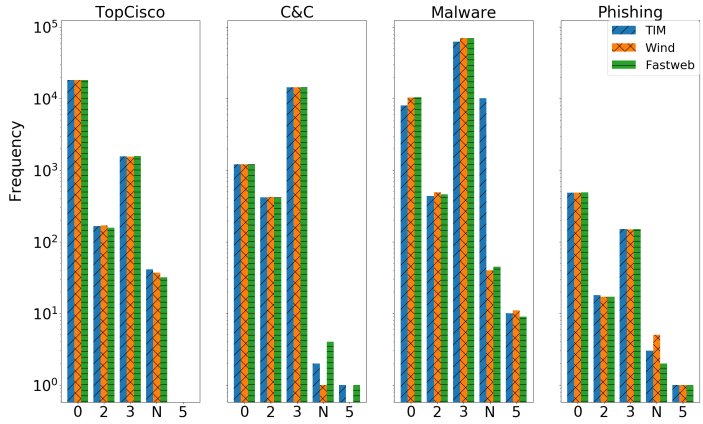


Figure 5.8. Results obtained under three different ISP networks using the resolver provided by Google (*i.e.*, using a public resolver).

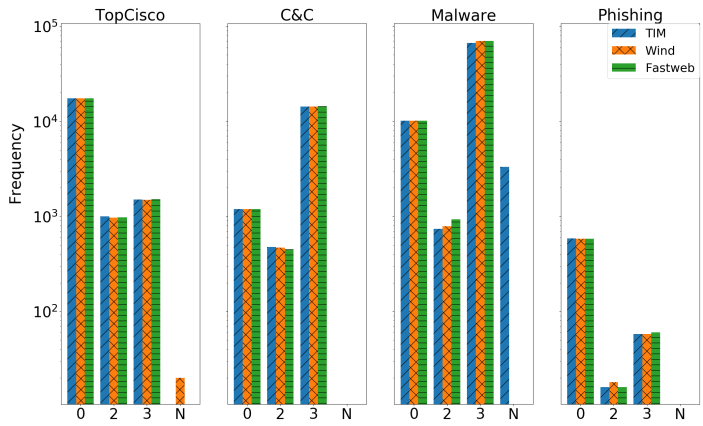


Figure 5.9. Results obtained under three different ISP networks using the resolver provided by OpenDNS (*i.e.,* using a public resolver).

Considering the C&C dataset, the plots illustrate that the Wind resolver has the highest number of "0" codes and zero occurrences of "3", unlike TIM and Fastweb. The latter two are characterized, indeed, by a lower value of "0" occurrences, and also include a significant value of "3" occurrences.

Similar observations can be found regarding the Malware dataset. Looking at the two datasets in Figure 5.8, we can mark that Google resolver acts like TIM and Fastweb, returning a high number of "NXDOMAIN" rather than "NOERROR". OpenDNS, instead, reported in Figure 5.9, returns a high number of "3", but also a small percentage of "0" codes with an IP address related to a courtesy page. When the dataset is Phishing, the number of "0" codes is higher than the number of "3" codes in all of the cases. In summary, all the DNS resolvers return the "NOERROR" message when the domain name is benign. Instead, when the domain is included in the C&C or Malware dataset, Wind returns the "NOERROR" message with a courtesy page IP address, while the other DNS resolvers return the "NXDOMAIN". Lastly, when the domain is related to a phishing client, all resolvers return a high number of "NOERROR" and a smaller number of "NXDOMAIN". A "NOERROR" message in the response does not always mean that everything is correct. For example, the code "0" is even common when the DNS resolver returns the IP address of a cour-

tesy/splash page to prevent the user from accessing a potentially dangerous resource. We found that two DNS resolvers return a courtesy page in some cases: Wind and OpenDNS. For the analysis performed, it is not needed to distinguish between the two cases above because the "NXDOMAIN" message still protects the user.

We further investigated the results about the "3" and "0" codes for each dataset and provider. We obtained that OpenDNS detects Malware and Phishing domain names better than Google. This behavior occurs similarly for each residential network (Tim, Wind, and Fastweb). For the sake of brevity and because it is the residential network with the fewest failures, we report only the comparison results for the Fastweb network. We filtered out the "0" code with a courtesy page address in the response. Specifically, about the Malware dataset, Google presents 0.09% more than OpenDNS for code "3", which presents 0.52% less than Google for code "0". About the Phishing dataset, Google presents 13.68% more code "3" than OpenDNS, which presents 53.5% less than Google for "0" code. We also investigated the local DNS resolvers against Google and OpenDNS. We obtained that local resolvers present higher performance than Google, as confirmed also by other works [156, 157]. Therefore, they are performing worse than OpenDNS, except for Wind provider, which shows a similar behavior about the courtesy page.

F-Measure and Accuracy

In the following, we report additional information regarding the security level of the resolvers. We calculated the **F-score** and **accuracy** as measures of their capability to detect malicious domains. The first step in calculating these two metrics consists in determining a confusion matrix characterized by positive and negative classes. The column "Class" of Table 5.2 summarizes the class of each RC received in our DNS responses. More specifically, if we perform a query to a benign IP, we expect a "0" code in the response. Conversely, with a malicious domain name, we should get a non-"0" rcode in the response. This is why the code "0" belongs to the positive class; the "3, 2, 5, Null" codes to the negative class. Since the domain names of the TopCisco dataset are characterized mainly by benign domains, the corresponding DNS responses include a large number of "0" labels. Consequently, this dataset belongs to the "positive" class.

The other three lists contain only malicious hostnames and, hence, the corresponding DNS responses consist of a high amount of "3,2,5, Null" codes. For this reason, they belong to the "negative class". Summarizing, the elements of our confusion matrix are the following: `True Positive` (TP): IP addresses obtained querying domains from the TopCisco dataset; `True Negative` (TN): NXDomain obtained querying domains from the C&C, Malware and Phishing lists; `False Positive` (FP): IP addresses obtained querying domains from the C&C, Malware and Phishing list; `False Negative` (FN): NXDomain obtained querying domains from the TopCisco dataset. For Google and OpenDNS, we report only the results obtained until the Fastweb network. We see that Google presents an F-score equal to 71%; OpenDNS 73%, Fastweb 72%, Wind 28% and TIM 72%. Concerning the accuracy measure, TIM, Fastweb and Google reach a percentage of 88.1%; OpenDNS has an accuracy of 87.7%; Wind is accurate to about 17.5%. As mentioned before, the F-score and accuracy values for the Wind resolver are lower than the others mainly because of the huge amount of courtesy pages contained in the DNS response with a "0" code. In addition, OpenDNS shows a lower accuracy value than the other resolvers because it applies a hybrid approach.

Analyzing recurrent IP addresses

We also examined the IP addresses with a high number of occurrences. We report those obtained with the Wind resolver related to the Malware dataset: the IP address 40.68.249.35 occurs slightly less than 100.000 times, 86% of the times in our experiments. We checked that it corresponds to the IP address of a courtesy page. Other interesting IP addresses are: 216.218.185.162, 64.70.19.203, 34.102.136.180, 35.102.136.180. These IPs are consistently reported by all the resolvers. Querying the Whois tool, we discovered that the first IP belongs to Hurricane Electric LLC. The second one is related to CenturyLink Communications, LLC. The third and the fourth ones belong to Google LLC. These IP addresses are obtained only from the C&C and Malware datasets.

Hurricane Electric has already been traced back to malicious DNS activities. Anyone could register for a free account with Hurricane Electric's hosted DNS service. It is possible to register a zone and create A records, even causing the hijacking of legitimate domains because the provider does

not check if zones created by their users have already been registered (*e.g.,* see [170]).

Hurricane Electric the address is `216.218.185.162` and the hostname is *216-218-185-162.sinkhole.shadowserver.org*. Shadow server[5] is a non-profit security organization that gathers and analyzes data on malicious Internet activity, including malware and botnet. They provide a sinkhole service used for spoofing DNS requests to prevent the resolution of malicious hostnames. It can be accomplished by configuring DNS resolvers that return a sinkhole address for a specific domain name. One of the nameservers of the Shadowserver operator is `sinkhole.shadowserver.org` - `216.218.185.160/29`, that we found in our results [171]. All domain names have a .xyz top-level domain (TLD) and a TTL value equal to 21599.

CenturyLink Communication We have also investigated the IP address related to Century Link Communication. We performed reverse DNS lookup queries and got PTR records from IP addresses with the dig tool. The domain name related to this IP address is `mailrelay.203.website.ws`, useful to register a new .ws domain.

Google LLC The last two IP addresses belong to Google LLC. In more detail, the first IP address `34.102.136.180` is related to the `180.136.102.34.bc.googleusercontent.com` domain name. This domain is adopted for multiple purposes, like cached copies of websites visited by the Google search engine and storing static content including images [172]. In different cases, hackers hide malicious code inside image files that are rarely scanned for malware [173].
In conclusion, we observed that OpenDNS is slightly slower than Google and local resolvers. However, local resolvers are faster than public ones. Moreover, all resolvers analyzed protect users from most malicious domain names. OpenDNS provides a higher level of protection for Malware and Phishing domain names than Google and local resolvers. Wind presents a behavior similar to the one of OpenDNS.

[5] `https://www.shadowserver.org/`

5.5 Concluding Remarks

In this chapter, we investigated the behaviour of different DNS resolvers. In particular, we evaluated their capability to recognize malicious domains (*i.e.,* to protect clients), and the response time between them and their clients. We focused on two classes of resolvers: local DNS resolvers from main Italian ISPs (TIM, Wind, and Fastweb), and public resolvers by Google and OpenDNS. We based our analysis on the *Response Time* and *Response Code* obtained from the queries. The first one has been used to understand the speed of resolution of a domain name. The response code has been used to study how much a DNS resolver can recognize a domain name associated with malicious activity.
The results about the **Response Time** show that: (i) the local DNS resolvers are generally faster than public resolvers; (ii) Google is slightly faster than OpenDNS; (iii) there are no significant differences between DNS and DoH of both Google and OpenDNS. We have also computed the time we can gain using a resolver in spite of another, obtaining that: (i) Fastweb is 86ms faster than Google on average; (ii) Fastweb is 129ms faster than OpenDNS on average; (iii) Google is 43ms faster than OpenDNS in average. We also show that the increased speed of local DNS resolvers against public ones is confirmed even if we exclude domains cached at the home router.
The results about the **Response Code** show that some local DNS resolvers and Google return an "NXDOMAIN" message for malicious domains. Other resolvers, instead, provide a "0" RCODE with a courtesy IP address. OpenDNS behaves in a hybrid manner. The resolvers analyzed achieve good security levels, protecting users from most malicious domain names. In addition, OpenDNS achieves a slightly higher level than local resolvers and Google with malware and phishing domain names. This outcome is somewhat anticipated as we use domain names provided by Cisco Umbrella, the same company owing OpenDNS. In addition, both the DNS and DoH protocols tested with Google provide the same results in terms of RCODE. The same behavior is also observed with the two protocols tested with OpenDNS. We also examined security capabilities as a function of the dataset and we obtained no significant differences.
We believe that our analysis is first and foremost useful for the scientific

community and network operators to gain a better knowledge of the DNS and how to improve it. In addition, our results may provide insights to users in choosing the most appropriate DNS and, more generally, to the community on how the DNS works, which is far beyond just translating domains into IP addresses, as originally conceived.

Chapter 6

Lifetime of Benign and Malicious Domain Names

In this Chapter, we explore domain name lifetimes at scale and over a ten-year period. The DNS is essentially a hierarchical and distributed database that involves – and is operated by – many independent parties that fulfill various roles. Top-level domains such as `.com` and `.co.uk` are run by *registries*. *Registrants* can register domain names, usually through so-called *registrars*, but sometimes directly with the TLD registry. Domain names go through a well-defined life-cycle and names that are only short-lived in ways break expectations. Specifically, in this Chapter, we focus on ten prominent TLDs and observe that under most, the vast majority of lifetimes (95%) last exactly the minimum registration term of one year. The exception to this is `.com`, which sees 40% of lifetimes renewed for at least one more year. We also identify lifetimes that are suspiciously short-lived (*e.g.,* 80% under `.xyz`). Using blocklist data we confirm that about 25% are reportedly malicious and study indicators if names are taken down and how quickly. Finally, we empirically study malicious name registration campaigns and show that this involves registrars that offer bulk registration options.

6.1 Motivation

The Internet Corporation for Assigned Names and Numbers (ICANN) determined a well-defined life-cycle for domain names that nominally leads to domain name lifetimes of yearly granularity. In most cases, the lifetime of a domain name is under the direction of its registrant, with whom rests the decision whether or not to renew the registration. However, there are other possible factors, notably if domain names are used for abusive purposes and taken down.

While the DNS and domain abuse are extensively studied in the literature, the area of domain name lifetimes is arguably still dim. In this work, we take steps towards closing this gap. We analyze domain name lifetimes under the ten largest top-level domains in CAIDA's DNS Zone Database [174] across a time span of ten years. To empirically validate the idea that shorter lifetimes can be the result of abuse take-down efforts, we use a large blocklist feed of malicious names and demonstrate that many short-lived names are indeed malicious.
We make the following contributions in this chapter:

- We perform an analysis of domain lifetimes among 10 of the largest TLDs over a ten-year period, showing that one-year lifetimes predominate (∼95% of lifetimes last exactly one year) in most TLDs except `.com`, where 40% of the domains have longer lifetimes;
- Using blocklist data, we evaluate the prevalence of malicious domain names across the TLDs and reveal that a large fraction of malicious names have shorter-lived lifetimes. We also show that malicious names are substantially shorter-lived in some TLDs compared to others (*e.g.,* 80% of malicious `.xyz` names live shorter than the minimum registration term of one year);
- We show signs that malicious names are acted upon and provide insights into take-down times, while we also provide indications that some malicious names are not acted upon and are left to linger;
- We identify a number of malicious registration campaigns and empirically show that such campaigns can include registrars that offer bulk registration options.

All in all, our findings help shed light on domain registration practices and the use of domain names and malicious behaviors. We also shed light on operational practices by studying indicators of the presence (or absence) of take-down efforts.

6.2 State of the Art on Domain Name Lifetimes

Domain name abuse is extensively discussed in the literature. For malicious registration detection, several works go beyond blocklists to find additional ways to detect malicious domain names. Sun *et al.* [175] propose a methodology named *HinDom* to detect malicious domain names using a classification based on relationship between clients, domains and IP addresses. Their methodology was able to detect a long-buried botnet and several malicious domains in a real-world scenario. Using an Extreme Learning Machine, Shi *et al.* built a malicious domain detector that uses several features (*e.g.,* length of domain, entropy, number of IP addresses) and achieves an accuracy greater than 95% [176]. Hason *et al.* used similar features to build a classifier of malicious domain names, also achieving an accuracy of 95.2% [177].

Bilge *et al.* built a system to detect malicious names, adopting machine learning techniques based on passive DNS data [178]. Combining 15 behavioral features, their system identifies a large number of malicious hosts. Vinayakumar *et al.* assessed the efficacy of using deep learning to detect malicious domain names [179]. They applied CNN (Convolution Neural Network) and RNN (Recurrent Neural Network) approaches to a large volume of DNS logs.

Previous studies have also explored malicious campaigns registered in bulk for large-scale attacks [180, 181, 182]. Cybercriminals register considerable numbers of domains to quickly replace detected domains and recover from take-down efforts [183]. Vissers *et al.* examined malicious campaigns in the registration data related to the `.eu` TLD [184]. Looking at domain names with the same registrant and registry information, they found that 80.04% of short-lived domain names could be tied to 20 campaigns. Furthermore, they claim that these campaigns differed in terms of duration: from one month to a year and beyond. Their results are in line with ours. Indeed, we detect several campaigns characterised by malicious

names with overlaps in features. Contrary to their analysis of only the `.eu` TLD, we investigate a selection of 10 TLDs that represent a sizable part of the global namespace.

Regarding domain name lifetimes, Foremski *et al.* analyzed malicious short-lived domain names, finding that 9.3% of new domains were deleted in the first seven days, with a median lifetime of 4 hours and 16 minutes. Their study leverages the NOD (Newly Observed Domain) service based on passive DNS observation and active DNS measurements [185]. In addition, they inspected several possible causes of deletion, stating that blocklisting is responsible for 6.7% of it. As with Foremski *et al.*, we study domain name lifetimes, focusing on the ten largest TLDs, and we use the DBL blocklist to identify malicious domain names. Unlike this work, we examine all domain name lifetimes (not only the newly observed domains and malicious ones) included in CAIDA's Zone Database over ten years. In addition, to further investigate the causes of their short lifetime, we examine the presence and the lifetimes of malicious domain names in 2018-2021. Barron *et al.* [186] show that early deletions of domain names are significantly correlated to potentially malicious activities, and we have similar findings. The authors also show that short-lived malicious domain names tend to be longer and more pronounceable or prone to typo-squatting. We examine related characteristics of malicious domains and confirm largely similar results.

Finally, Korczynski *et al.* [73] reveal that abuse activity shifted from legacy gTLDs to newer gTLDs, in part due to registration prices. In our work, we show also that legacy gTLDs still include a considerable number of malicious domains. Lauinger *et al.* examined the WHOIS records of domains about to be deleted in DNS zone files during the stages of the expiration and re-registration [187]. They found that registrars implement different cancellation techniques that are not always compatible with the life cycle of domains. In contrast with our work, they analyze fewer TLDs and do not inspect the expiration and re-registration of malicious domain names. Finally, an interesting study regarding the new domain name registrations related to COVID-19 domain names was conducted by ICANN [188]. They found that these domains were also used for malicious purposes, around 1.8% being flagged.

6.3 Methodology

Lifetime Inference. Central to this chapter is our ability to infer domain lifetimes. We devised a relatively straightforward methodology that uses zone files. Recall from Section 2.3 that for a domain name to functionally exist, the parent zone (*i.e.,* registry) typically delegates authority to a name server of choice of the domain name owner (*i.e.,* registrant). We assume that if a domain name is "alive", its nameserver delegations will be present in the zone file. This assumption does not always hold. There could be cases in which `NS` records are absent, for example when a domain is parked or in a *grace period*. To account for these blind spots in zone files, our methodology allows for gaps of at most *90* days before considering a lifetime closed. We choose this value arguing that it is sufficient to capture temporary disappearance, *e.g.,* during one or both of the possible grace periods, but not so long as to capture re-registration after release. The *90*-days threshold includes a margin of *10* days over the *80* days domain removal scenario defined in subsection 2.3.4, to account for possible errors in zone file collections.

Because of the granularity of our data sources (Section 6.4), we consider lifetimes in terms of multiple days. As we will show in Section 6.5.5, WHOIS data for malicious domains validate that our assumptions provide a good estimation of domain lifetimes. Note that the lifetimes that we define and consider in this chapter are *closed lifetimes*. More specifically, for a given domain, these are the lifetimes for which we are able to observe the start and end, because the domain creation and expiration dates fall within the boundaries of our data.

Malicious Domain Names. The other important part of our methodology relates to how we consider and analyze *malicious* domain names. To make a determination of maliciousness, we rely on a blocklist (Section 6.4) as input. To characterize malicious names and study the presence and properties of such names under various top-level domains, we consider the registered domain name part. We extract the registered domains from blocklisted names with Public Suffix List even though they may contain additional labels. This puts the considered entries at the same level as the names (technically, zones) in `NS` records in TLD zone files, which in most cases do not contain deeper levels of nesting. We note that this choice

could lead to classification errors for registered domain names that are in the parent zone to both malicious and non-malicious names (consider, *e.g.,* the shared suffix under Dynamic DNS service providers). Nevertheless, we argue that the number of third-level domain hosting services compared to the number of second-level domains is negligible. In fact, they are managed by established companies that are not likely to have short-lived domain names. We, however, consider registered names that expire, which are less likely to introduce such classification errors.

6.4 Data Sources

We use two data sets for this analysis, together with supplementary data. We obtain the primary data sets from two sources: zone file data and malicious domain names.

Zone files. We use data from CAIDA's DNS Zone Database (DZDB)[174], which is built on a sizable collection of TLD zone files and captures the history of domain names, name servers and IP address records. Following the inception of ICANN's Centralized Zone Data Service (CZDS), most of the newer gTLDs were added to DZDB, which currently contains approximately 210 million names. Our analysis of the lifetime behaviors of domain names involves a sizable part of the DZDB data. We consider a time period of roughly ten years, starting at the earliest DZDB data (2011/04/11 – 2021/02/14).

For our analyses, we consider the Top 10 TLDs in DZDB in terms of total size ranking since 2011. The Top 10 is representative (they cover 87% of all the SLDs in our entire data set) and allow us to provide insights into administration policies for individual TLDs. Table 6.1 shows the Top 10 **TLD**s, a summary of DZDB data available for them, and the number of lifetimes that we infer. Taking `.com` as an example, we infer 169M lifetimes throughout the ten-year period. Relative to the total of 157M unique `.com` names, this shows that for some names we infer altogether new registration (and another lifetime), as per our methodology (see Section 6.3).

Blocklists. As an indicator of malicious activity, we rely on the Domain Block List (DBL) maintained by the Spamhaus project. Our data set consists of daily snapshots of the DBL feed from 2018/01/01 to 2021/02/14. While a single blocklist is a narrow window into malicious domain re-

Table 6.1. Top 10 TLDs data set, showing CZDS start and end dates, the number of lifetimes segments inferred per TLD, and the number of unique domain names involved

TLD	Start Date	End Date	# LT Segments	# Names
.com	2011/04/11	2021/02/14	168.9M	156.5M
.net	2011/04/11	2021/02/14	20.6M	19.4M
.info	2011/06/06	2021/02/14	16.3M	15.7M
.org	2011/05/08	2021/02/14	12.7M	12.1M
.xyz	2014/03/31	2021/02/14	12.8M	12.2M
.top	2014/08/04	2021/02/14	12.1M	11.7M
.icu	2015/06/24	2021/02/14	5.6M	5.6M
.biz	2011/05/06	2021/02/14	4.9M	4.6M
.us	2011/05/06	2021/02/14	4.6M	4.4M
.loan	2015/03/30	2021/02/14	4.6M	4.6M

lated activity, we find this window illuminating. Since our DBL data set starts in 2018, we only consider DZDB data from 2018 onwards for our analysis in Section 6.5.2. However, a limitation of this data set is that it does not include the type of malicious activity associated with the domain name. Consequently, we cannot show the trends of the lifetimes by varying the malicious activity. To overcome this limitation, we relied on the "Domain Status and Categorization" API provided by Cisco Umbrella Investigate [189], but it did not give us enough data to support this analysis.

WHOIS data. We rely on data provided by Cisco Umbrella to investigate malicious domain name registration campaigns (Section 6.5.5). Their Investigate API gives a complete view of a domain name, IP address, ASN, and malware file details to help identify misused infrastructure and predict future threats [190]. Relevant to our work, the provided WHOIS data includes registration information for domain names, including creation date; registrant organisation, city and country; and registrar name and IANA ID.

6.5 Experimental Results

In this section, we present our results. We start with an overview of lifetimes for domains under the Top 10 most populous TLDs in our data (Section 6.5.1). We then investigate malicious name lifetimes (Section 6.5.2) and suspicious short-liveness (Section 6.5.3). Next, we look at post-blocklist life and possible take down actions (Section 6.5.4). Finally, we investigate malicious registration campaigns (Section 6.5.5).

6.5.1 Lifetime of Domain Names

As explained in Section 2.3.4, a domain name can go through five different life-cycle states, of varying lengths, which together form the *lifetime* of a domain name. We expect most domain names to be visible in the zone files while *registered*. This expectation allows us to evaluate the lifetime of domain names as the time from when a domain is first and last seen in the zone file. In our methodology (Section 6.3), we consider a lifetime to have ended when, after appearing, it is absent from the zone file for 90 days or longer. We treat reappearance beyond this point as an altogether new registration.

We infer domain name lifetimes for the Top 10 TLDs. Figure 6.1 shows CDF plots for domain names under `.com`, `.icu`, `.xyz`, `.loan`, and `.us`. For this analysis, we consider domain names in zone files that have valid first-seen and last-seen values in the period 2011/01/01 through 2021/02/14, capping the lifetime at roughly 3700 days. Therefore, our analysis does not include domain names still active at the last collection time. For clarity, we do not plot the other five TLDs, but they display similar trends as we further detail below. The results show that a considerable number of domain names are registered for lifetimes of one year in most TLDs, with all TLDs showing a sharp increase around 410 days: one year plus the *Auto-Renew* grace period of 45 days.[1] Moreover, zones also see lifetimes that are under the minimum registration term of one year, which may be the result of take-down efforts (see Section 2.3.5).

For `.com` domains, 60% of their lifetimes are at most a year (101M of 169M lifetimes in Table 6.1), and about 20% of `.com` names involve

[1] The edge is slightly slanted because registered names may take a few days to appear in the zone file, as we will show in Section 6.5.5.

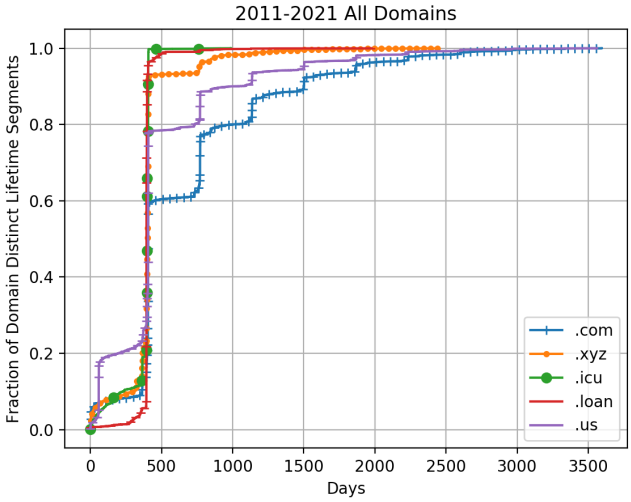


Figure 6.1. Domain name lifetime in selected Top 10 TLDs under consideration

lifetimes of three years or longer. The TLDs `.org`, `.net`, `.info` and `.biz` all show similar trends (not plotted). These zones belong to the first set of ICANN gTLDs, originally created between 1985 and 2001 [191]. A comparable trend occurs for the domain names of the Country Code Top-Level Domain (ccTLD)s `.us`. In this case, 78% of their lifetimes last at most one year (3M of 4.6M lifetimes), and around 20% of `.us` lifetimes are longer than three years. Moreover, `.us` includes roughly 20% of domain names with a lifetime less than 70 days, in contrast with the `.com` and the analogous TLDs where this value is significantly lower (0.08% of 169M lifetimes). In contrast, for `.xyz`, about 93% of lifetimes (11.9M) are about one year, 95% of at most about two years, and only a small percentage of domains remain registered three years or longer. The `.top` TLD (not plotted) presents a similar trend to `.xyz`. Indeed, around 2019, these were the new gTLDs with the most number of registrations [192].

An interesting behavior seen relates to `.icu`. This TLD was created in 2015, but the first domain names under it were registered around 2018. Therefore, we have a three-year observation period for this TLD. The trend that becomes apparent is that most lifetimes are one year. The

same applies to .loan, except that .loan includes fewer domains with a duration of less than 400 days than .icu. We have also evaluated the number of domains still active at the end of the collection period, and found that .com is still the TLD with the highest number of domains.

6.5.2 Malicious Domain Names

To better understand possible causes for patterns in lifetimes, and considering that lifetimes can be cut short as a result of take-down efforts (see Section 2.3.5), we match domain names from the zone files with those included in the Spamhaus DBL blocklist. Note that, for now, we consider *any* malicious name, regardless of the duration of its lifetime. In a later section, we will focus on short-lived names in particular.

Our overall lifetime analysis and Figure 6.1 capture a ten-year period. As we obtained DBL data from Jan 1, 2018 onward, we can only match domain names registered after this date against DBL inclusion. For this reason, going forward we consider zone files data for 2018 and onward.

The lifetime of malicious domains is usually considerably shorter than that of benign names [176, 193]. Malicious names are deactivated once revealed or because hackers want to minimize blocklist interference. For example, many spam domains are only active for one day, in an attempt to avoid detection and from being added to blocklists [180, 184].

We calculate the percentages of malicious domains in the Top 10 TLD data for 2018 and beyond and extract malicious lifetimes. Table 6.2 summarizes the results. We show the total number of names and lifetimes inferred as before (Table 6.1). The .biz TLD contains the highest percentage of malicious domain names (28.46% of 950K), followed by .top and .us. While lower, .loan and .info are still above 10%. Under the largest TLD .com, 7.5% of domains are malicious. Spamhaus estimates an abuse score for each TLD based on the prevalence of malicious domains[2]. Our findings are largely in line with these scores: the current Spamhaus scores identify .biz, .top and .us as most-abused, and .org as least.

Figure 6.2 relates specifically to the lifetimes of malicious domain names. We show only the CDFs related to .com, .xyz, .icu, .loan, .top, .biz. Lifetimes for malicious names under the other TLDs show trends similar

[2] https://www.spamhaus.org/statistics/tlds/

Table 6.2. Top 10 TLDs data set with malicious names, showing the number of lifetimes segments inferred and unique names in CZDS data for 2018+, as well as the malicious figures

TLD	# Total LT Segments	# Total Names	# Malicious LT Segments	Malicious Names (%)
.com	32.7M	32.2M	2.5M	7.53%
.net	2.6M	2.6M	201K	7.62%
.info	2.0M	2.0M	256K	12.41%
.org	1.7M	1.6M	51K	3.05%
.xyz	3.5M	3.4M	233K	6.62%
.top	6.2M	6.1M	1.3M	21.56%
.icu	5.7M	5.6M	244K	4.27%
.biz	928K	950K	270K	28.46%
.us	794K	790K	156K	19.67%
.loan	2.0M	2.0M	240K	12.20%

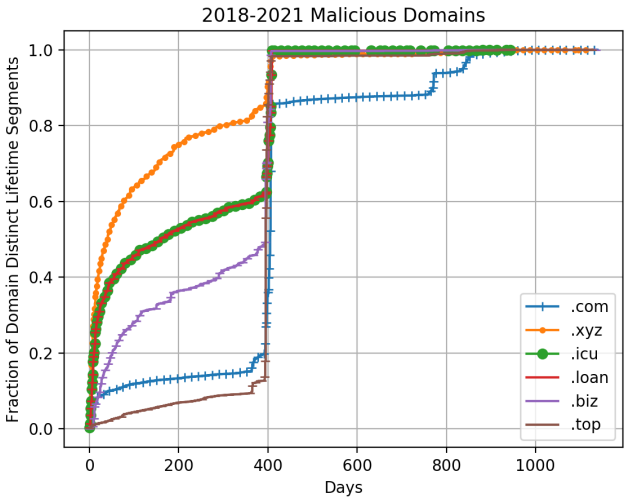


Figure 6.2. Domain name lifetime in various Top 10 TLDs under consideration for names that are reportedly malicious

to the counterparts of these TLDs we reported in Section 6.5.1. Malicious domain names in `.xyz` generally have shorter lifetimes than those under other TLDs. The TLD `.icu` is next in rank. The TLD `.loan` sees a considerable number of malicious domain names that have a lifetime of around one year, followed by `.biz`. The TLD `.top` includes a high percentage of malicious domain names (*e.g.,* 21.56%) with longer lifetimes than the other TLDs (*i.e.,* 12% of malicious `.top` domain lifetimes are *shorter* than 365 days). More specifically, 97–99% of malicious `.xyz`, `.icu` `.loan`, `.top`, `.biz` domain lifetimes are *shorter* than 410 days. For `.com` it is 86%. The `.xyz` TLD could stick out for multiple reasons. First, we see that malicious `.xyz` domains are less likely to be renewed in general (Figure 6.1). Second, as we show in Section 6.5.4, malicious `.xyz` names are acted upon quicker compared to other TLDs.

6.5.3 Short-Lived Domain Names

We now focus on domain name lifetimes of 364 days or shorter. We chose this threshold because it captures domain names that live less than the minimum registration term, considering the minimum of 0 days under grace (Section 2.3.4). For the overall DZDB data (*i.e.,* starting in 2011), 6.19% of lifetimes are 364 days or shorter. For 2018 onward, which aligns with the DBL data available to us, the percentage is 19.57%: 11.3M lifetimes involving 11.0M unique domain names. We cannot make a strong inference from the relative increase in percentages, but do note that anecdotal evidence suggests increases in domain name abuse [194]. In addition, although the ICANN report shows an increase in the number of registrations and a decrease in the number of abuses from 2017 to 2022, we see a drop in the number of new registrations from 2018 to 2020 [195]. Furthermore, the percentage of lifetimes less than 364 days is 15.5% in 2018 and 16.2% in 2019. We cannot estimate this percentage in 2020 because our data set lasts until February 2021. Considering DBL data, we confirm that 24.27% of short-lived lifetimes involve malicious domain names. These ∼1.3M lifetimes involve almost the same number of domain names, and hence we rarely encounter malicious names for which we infer multiple (short-lived) lifetimes.

We calculated the percentages within each Top 10 TLD to investigate how they compare. We find that `.biz` has the highest percentage: 34%.

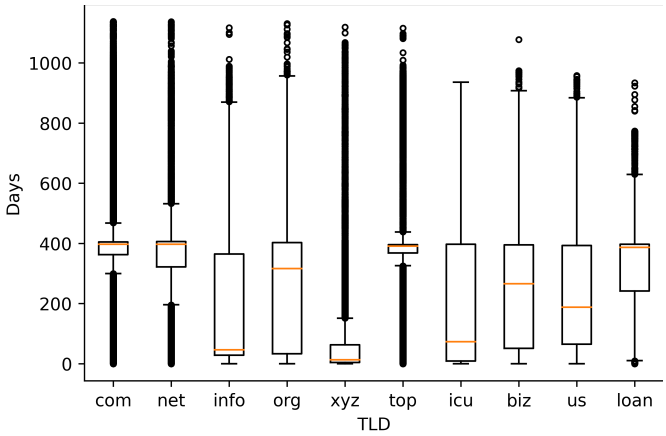


Figure 6.3. Number of days elapsed between the insertion of malicious names on the blocklist and their removal from the zone file

Recall from Section 6.5.2 that this TLD also sees the highest percentage of malicious names. The `.icu` and `.top` TLDs contain the lowest percentages of short-lived domain names. At the same time, however, if we consider strictly malicious domain names in these TLDs, we see that many fit the short-lived criterion (see Figure 6.2).

6.5.4 Post-Blocklist Life and Removal

We investigate how much longer domain names live after appearing on the blocklist, noting that removal can be the result of take-down efforts. To this end, we look at the number of days between DBL insertion and removal from the zone file.

First, we consider any malicious name (*i.e.,* not necessarily short-lived ones), including names that naturally expire.

Figure 6.3 shows the resulting boxplots for the Top 10 TLDs. The TLDs `.com`, `.net`, and `.top` see median deletion times of 379, 379, and 387 days, respectively. These values are close to 410 days (one year plus the auto-renew grace period), which is the minimum lifetime of a domain if it is not renewed. Therefore, this plot shows that these three TLDs include most blocklisted names that may have naturally expired rather than being acted upon (*e.g.,* by registries or registrars). The TLD `.xyz`

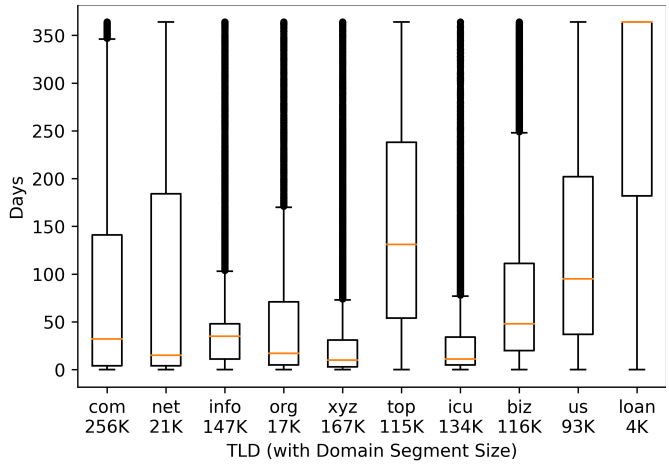


Figure 6.4. Number of days elapsed between the insertion of short-lived, malicious names on the blocklist and their removal from the zone file

shows the opposite: a median of just 13 days. With the exception of `.xyz`, the upper quartiles are close to the one-year mark, suggesting that a long tail of names under most TLDs naturally expire. Finally, looking at 95-percentiles, we see that there are malicious domains that live for multiple years before expiring.

Second, we consider short-lived malicious names, postulating that malicious names that do not live for the minimum registration term of one year are likely to have been taken down. Figure 6.4 shows the resulting boxplots. The `.xyz` TLD again shows the lowest median value (10 days here), indicating that malicious domain names are removed from this zone shortly after being blocklisted. The short boxplot for `.xyz` also suggests that few malicious domains live anywhere near the minimum registration term.

We observe different behavior for `.top`, which sees a high median value of 131 days. Its relatively tall plot and upper quartile shows that some malicious names live for a considerable amount of time after being blocklisted. Similar observations can be made for several other TLDs such as `.net` and `.us`, although not as pronounced. With the exception of `.top`, the results are comparable for the situation in which we considered any

malicious name, regardless of whether they are short-lived. Considering Figure 6.4, we conclude that, for suspected take down efforts, the median removal time is largely between 0 and 2 months. Given that the 4k short-lived malicious names represent only a tiny fraction of the malicious `.loan` names (0.02%), we do not consider its results representative.[3]

As only 24.27% of the short-lived lifetimes involve malicious names in DBL data, we consulted two parties — a ccTLDs registry and a large global registrar — about other possible reasons for domains being short-lived. The registry stated that the blocklist perspective only accounts for a subset of short-lived domains, but what is missed is still due to abuse. The registrar indicated that malicious domains can be re-registered with them after being taken down and after the expiration of the redemption period. Finally, we note that some registrars, such as Freenom, provide an API to security researchers to immediately take-down free domains following signs of abuse[4]. We do not know if such mechanisms are available for the TLDs that we considered. However, it could help explain differences in take-down timings.

6.5.5 Investigating Malicious Campaigns

Some cybercriminals register a considerable number of domain names for malicious purposes at once [196, 184]. There are registrars that make this possible by offering bulk registration options. To investigate, we study malicious name registrations over time and look for signs of bulk registration. We cross-reference DZDB and DBL data and calculate how many new malicious registrations occur every day. Figure 6.5 shows the results for *.com*, which usually sees 1K – 10K malicious registrations daily and also contains a pronounced spike on July 28, 2018. For other TLDs (not plotted) we observe lower daily averages and also occasional spikes.

We investigate the suspicious spike, which involves 27k malicious names, for possible causes. Related work has shown that maliciously registered names in bulk can involve overlap in WHOIS features [184]. For this reason, using Cisco Umbrella data, we look for overlap in: the registrant organization, city, and country; and the registrar name and IANA ID.

[3] As we show in Section 6.5.2, the malicious names that we found in `.loan` are typically longer lived. One possible reason is that it does not react to abuse notifications.

[4] https://www.freenom.com/en/antiabuse_api.html

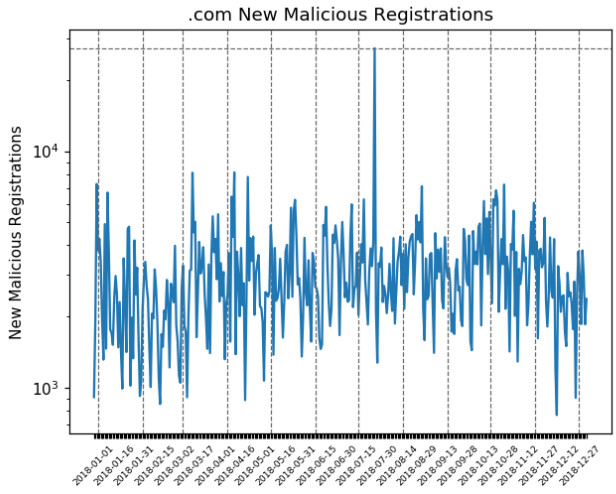


Figure 6.5. New Malicious Registrations in 2018 - .com TLD

This identifies a campaign characterized by 4362 domain names, which can be tied to a single registrant organization in Malaysia, and the registrar GoDaddy. Furthermore, a considerable number of domains related to this peak (around 17K) were registered by the registrar Alibaba. Both registrars offer bulk registration. We looked for visually prominent spikes for other Top 10 TLDs as well (not plotted). Two peaks occurred on 2018/03/01 and 2018/05/16, respectively, involving 83k and 52k malicious domain name registrations, 93% and 97% of which are under `.top`. In both cases, Alibaba was also the registrar used, and the names share a single Chinese registrant. Consequently, all malicious spikes analyzed were triggered by a significant number of registrations performed by the Alibaba registrar.

We also looked beyond spikes and examined 12 "average" days of malicious `.com` registrations, one per month, equally spaced over a year. Figure 6.5 marks the dates with dashed vertical lines. We identify several smaller campaigns with an average of about 3k daily registrations. We find registrar overlap for GoDaddy, GMO Internet, PDR Ltd. d/b/a, or Xin Net Technology Corporation. Finally, we looked at the malicious names to further confirm commonality. Using Levenshtein distances we observe

that in some campaigns, names differ by only a few characters.

Lifetime inferences. We extracted from WHOIS data the creation dates for domain names involved in the 12 snapshots and three peaks, and compared them with the registration date that we infer from zone files. This comparison reveals that 80% of domains were registered up to one day and 97% up to two days earlier. This shows that, in most cases, our zone file approach to inferring the date on which a domain name's lifetime starts is reasonable. (Recall from Section 2.3.4 that domain name owners may withhold names from the zone files.)

6.6 Concluding Remarks

In this chapter, we analyzed domain name lifetimes. We showed that among a representative selection of TLDs, initial ICANN gTLDs (*e.g.,* `.com`) exhibit a higher renewal rate than newer gTLDs (*e.g.,* `.icu`). We also see signs that a non-negligible number of domain names do not live as long as the minimum registration term of one year. To investigate possible causes, we examined the presence and lifetimes of malicious names. Half of the TLDs considered involve substantial numbers of malicious names (*i.e.,* 12.20–28.46%). Moreover, malicious names in some TLDs live longer than in others. We see indications that domains are subjected to take-down efforts, finding also that in some TLDs this takes place quickly after domains have appeared on a blocklist. Finally, we looked at malicious registration campaigns. We empirically identified a number of them on the basis of WHOIS feature overlap (*e.g.,* registrant or registrar) and also found indicators that some registrars are used regularly to this end. We believe that the investigation of the malicious campaign may be applied also in the threat intelligence or cybersecurity fields. Specifically, the security level of a domain may be pre-estimated by observing its registration features and also whether it belongs to a bulk registration. Future work can extend the coverage of TLDs to less popular ones beyond the Top 10, and increase the coverage of malicious names, *e.g.,* considering other blocklists like VirusTotal and Cisco Umbrella Domain status.

Chapter 7

Impact of COVID-19 Restrictions on Internet Application Usage

In this Chapter, we analyze the impact of COVID-19 pandemic restrictions on the usage of Internet applications through DNS data. The emergency related to the Coronavirus has impacted everyone's life. From the first weeks of 2020, in China, and for weeks later, in other countries of the world, isolation and social distancing measures have been adopted to avoid the spread of the virus, forcing people worldwide to isolate themselves in their homes.

We provide insights into the use of different categories of Internet applications. We use two complementary sources of information: the lists from Alexa and Cisco Umbrella regarding the top 1 Million websites and domains used worldwide. Our results show that, during the first lockdown period, the most used applications have been Youtube followed by Netflix, Facebook, Whatsapp and Skype. This shows how users have looked for consolation in entertainment apps such as youtube, and Netflix, and in social media like Facebook. App of messaging services and collaboration, like WhatsApp and Skype, have been used to communicate with friends and families while also used for smart working. Contrasting the results from the two lists, we also uncover important differences in the usage of different kinds of devices. We believe that the COVID-19 pandemic repre-

sents a very interesting situation from the network utilization point of view and we shed light on how such a situation impacted the use of Internet applications.

7.1 Motivation

The global emergency related to Coronavirus (COVID-19), which spread in the last few years, has changed the life of every person, leading to an overload of Internet traffic as well. Schools of all grades have adopted distance learning, and any kind of office has started remote working. Lots of calls and remote connections to the office devices were held. Besides, the global pandemic and sequential lockdown have led to the high use of the Internet and other online services also for leisure.

In this chapter, we analyze the changes in Internet usage by looking at the most searched domains and accessed websites. We consider two datasets: the first one is provided by Cisco Umbrella [1]) and Alexa [2]. Every day, they provide a list of the top 1 million most popular domains and websites according to their ranking. The two providers of the lists adopt different methods for such ranking: the Umbrella list contains the most queried domains based on passive DNS, and Alexa's list contains the most popular sites visited by people that use Alexa's browser extensions. We call the two lists simply Umbrella and Alexa in the following. We analyze the last two months of the year 2019 and the first four of 2020, looking at the trends of the most popular applications, divided by category, to spot changes in application usage during the lockdown. This analysis was born out to understand how users spent their time during a period when they were forced to spend their time at home. For example, videoconferencing and messaging applications have been widely used for both business and entertainment with friends and family. But also entertainment applications, included in the video and social media categories, have provided moments of lightness and entertainment. Thus, the categories we consider are video, social media, messaging, and collaboration.

Our results confirm some results covered by the press, but also show interesting differences when contrasting Alexa and Umbrella. For exam-

[1] https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/
[2] https://toplists.net.in.tum.de/archive/alexa/

ple, we see that: 1) in the video category, youtube.com always occupies position 2 in Alexa, higher than netflix.com, which, in turn, occupies a higher position than youtube.com in Umbrella; 2) in the social media category, the domain facebook.com occupies higher positions in the Alexa with respect to Umbrella, where the most popular domain is twitter.com; 3) in the messaging category, telegram.org presents an interesting change in Umbrella, where it scales different positions in the ranking. The domain whatsapp.com features an increase and a decrease respectively in Alexa and Umbrella; 4) in the collaboration tool category, skype.com has the best performance in Umbrella, followed by zoom.us and webex.com. We believe that these results and the analysis presented represent unique contributions that will be difficult in the future time. At least we hope they will.

7.2 State of the Art on the Impact of COVID-19 Restrictions over the Internet

Scientific literature and several press and Internet companies have been focusing on the changes in Internet usage with the emergence of the world pandemic. There are works related to this topic, in different contexts, made with different datasets, and with different methodologies.

An interesting analysis was made by App Annie, an important analysis society, demonstrating how some conference applications, including Houseparty, Zoom, Hangouts Meet, and Microsoft Teams, recorded a high number of downloads in the week of 14-21 March 2020 worldwide, *i.e.,* 62 million between IOS and Google Play [197]. The New York Times reported an analysis of the usage of different applications by American users. The analysis was conducted by SimilarWeb and Apptopia, two providers of online data. They showed that Facebook, Netflix, and Youtube were more used by websites than by phone apps. In fact, spending time at home mostly at laptop computers, Americans are beginning to appreciate large computer screens rather than small smartphone screens. SimilarWeb and Apptopia have also registered a visible increment on those applications,*i.e.,* Google, Duo, and Houseparty video chat, which allows groups of friends to participate in a single video chat and play together. Increases were also recorded for videoconferencing applications such as Meets, Mi-

crosoft Teams and Zoom for smart working and virtual classrooms. One application not impacted by the crisis was Tiktok, which continued its rise even after the pandemic began [198]. Statista, a web portal for statistics that collects a large amount of data on multiple topics, has shown the ranking of the most downloaded apps in the Google Play store after the coronavirus (Covid-19) outbreak in France as of April 3, 2020, by several downloads [199]. They have demonstrated that the five applications most downloaded in France during this period of lockdown were: Whatsapp, Zoom, TikTok, Houseparty, and Skype, in this order. The analysis of network traffic to know the needs of users in terms of application usage is a topic already frequently covered in the scientific literature. Martino et al. show a longitudinal view of Internet traffic in five years, relying on data collected by a nationwide ISP infrastructure. Through this data, they studied the evolution of the Internet in order to evaluate which services became popular and which get abandoned [200]. They demonstrated that video content drives the bandwidth demand and that users of social messaging applications, like Instagram, consume more and more traffic. In fact, the traffic of each Instagram user is already comparable to the traffic of video-on-demand users, such as Netflix or YouTube. Authors of [201] focused on the traffic of a year about three European countries through five vantage points with different access technologies. Their scope was to perform measurements of "What the user does with the Internet", for example, they studied the popularity of the applications related to different categories (streaming Services over HTTP, File Hosting, Social Networking, Web and Peer-to-Peer). Favale et al. analyzed the impact of the quarantine period on the Politecnico di Torino campus network, focusing on collaboration and remote working platforms usage, remote teaching adoption, and looking for changes in unsolicited/malicious traffic. They have demonstrated that on the Polito campus there are no big problems. They have encountered a few cases of poor performance, probably related to people connected through 3G/4G operators [202].

7.3 Data and Methodology

In this work, we rely on two popularity lists provided every day by Cisco Umbrella and Alexa: two different kinds of lists of the top 1 million most

popular domains and websites. The two lists are created with different processes, discussed in more detail in Section 7.3.1. The **Cisco Umbrella Top 1M**, provided free of charge, is created through the DNS traffic to OpenDNS [3], its DNS resolver characterized by 100 billion daily requests from 65 million unique active users in more than 165 countries [203, 204]. The collection method of domains gathered by Cisco Umbrella consists of capturing not only browser-based traffic, such as Alexa, but also keeping track of internet activity on any port and not just port 80. The algorithm adopted to build the popularity list is relying on the unique number of IP clients visiting that domain compared to the sum of all requests to all domains [204]. **Alexa's top sites** is a list of websites ordered by Alexa Traffic Rank [205], generally provided for a fee. We have obtained these files through the project realized by Scheitle et al. [203], that they share the code, data, and additional insights into the following website https://toplists.github.io/.

The ranking provided by Alexa is generated by capturing data from Alexa's browser plugin on 25000 different browser extensions over the past three months from millions of users [206, 207]. "Alexa's Traffic Ranks are based on the traffic data provided by Alexa's global sample over a rolling 3-month period. Traffic Ranks are updated daily. A site's ranking is based on a combined measure of Unique Visitors and Pageviews. Unique Visitors are determined by the number of unique Alexa users who visit a site on a given day. Pageviews are the total number of Alexa user URL requests for a site. However, multiple requests for the same URL on the same day by the same user are counted as a single Pageview. The site with the highest combination of unique visitors and pageviews is ranked #1. Additionally, we employ data normalization to correct for biases that may occur in our data" [208].

7.3.1 Alexa and Cisco Umbrella Top 1 Million

As mentioned in Section 7.3, These top lists are created with different processes and data sources, generating different rankings. The first main difference between the Top 1M's lists is that the Umbrella list contains subdomains (like hangouts.google.com) while the Alexa list includes only

[3] https://www.opendns.com/

the top domains (like "google.com"). For this reason, the Umbrella list holds fewer main domains than the million main domains of the Alexa list [209]. Furthermore, Alexa receives web browsing data from users who have installed one of the many extensions of the Alexa browser. Cisco Umbrella, instead, builds these statistics from DNS queries sent via OpenDNS. Thus, Alexa analyzes data related to HTTP traffic from web browsers. On the other hand, the Cisco Umbrella Top 1 Million lists are based on DNS traffic data collected from various network devices, such as routers, firewalls, and endpoints. Studying both the Alexa and Cisco Umbrella Top 1 Million lists allows us to gain a comprehensive view of the usage of these applications. Besides, Umbrella's list contains several domains with non-authorized gTLDs (.mail) or test domains (www.example.com), not present in Alexa's list [209]. In the scientific literature, several works focus on the differences between the Top 1 million lists (*e.g.,* Cisco Umbrella and Alexa), helpful to understand the most used Internet domains.

Scheitle et al. studied the extent, nature and evolution of top lists used by research communities. They focused on three different popular top lists - Alexa, Cisco Umbrella and Majestic - and they evaluated their structure, stability, significance, ranking mechanisms and the research result impact. In particular, related to the difference between Cisco Umbrella and Alexa lists, they specified that the clients using OpenDNS are not only PCs but also mobile and IoT devices. Some clients that perform DNS queries to OpenDNS may be bogus, and nonexistent, unlike Alexa which captures data related to websites visited by users. But, in general, Umbrella's list can be useful to analyze DNS traffic while Alexa's list can be interesting for a study focused on the human web [203]. Le Pochat et al. [207] detected a change in the way Alexa composes its lists: the data is averaged over a single day, causing half the list to change every day. Instead, for the Cisco Umbrella list, only 49% of domains respond with HTTP status code 200.

7.3.2 Our Approach

As mentioned in Section 7.3, we have analyzed two different lists of the most popular domain names to observe the trend of the most used applications during the lockdown time. We retrieved Cisco Umbrella and Alexa daily top 1 million lists of the final two months of 2019 (November and December), as well as the initial four months of 2020 (January, February,

March, and the first thirteen days of April).

The aim of this study was to investigate the utilization of applications across various categories (Section 7.4) by users before and during the spread of COVID-19. We have analyzed only the first 10K domains of each list, aiming to focus only on the most popular domains. As the beginning of the COVID-19 pandemic in the world dates back to the first days of January [4], we have fixed the last two months of 2019 as the baseline. Subsequently, we compared each week of the first four months of 2020 with the baseline in order to discern how the most popular domains evolved each week in 2020 compared to the final months of 2019.

Algorithm 1 Create Result File

for i=0 to N **do**
  $Y \leftarrow Domain_1 i$
  $SearchYinFile_2$
  if $YinFile_2$ **then**
    $Position \leftarrow Position_1 + Position_2$
    $POP(YinFile_2)$
  else
    $Position = Position_1 + 100K$
  end if
end for
for j=0 to M **do**
  $Y \leftarrow Domain_2 i$
  $Position = Position_2 + 100K$
end for

Figure 7.1 summarizes the steps taken to obtain our results; the comparison component is zoomed in the pseudocode shown in Figure 1. We have selected the domain names related to the most frequently used applications by users, divided into categories 7.4. To track the popularity of these domains during the days of the week in 2020 and baseline months (November and December), we calculated each domain's mean position and variance, using the approach shown in the pseudocode 1. This allowed us to monitor the dispersion of the data around the mean position. First, for each week of 2020 and 2019, the first two files related to the first two days of the week are compared. The result of this comparison is com-

[4] https://www.nytimes.com/article/coronavirus-timeline.html

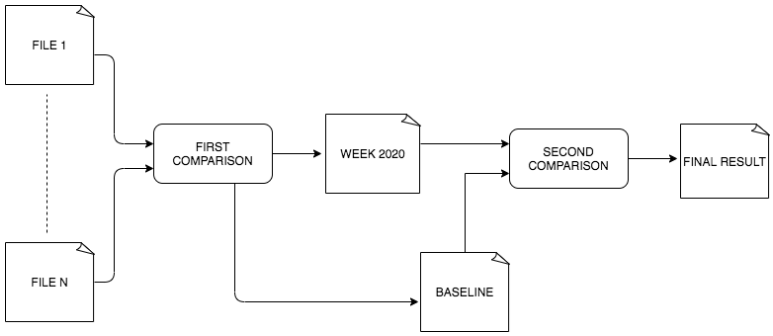


Figure 7.1. Flow chart of our approach

pared with the third day of the week and so on until all the files of the week are analyzed. The comparison between the two files, also reported by the pseudo-code, is structured as follows: each analyzed domain is searched in both the first and second files. If it is present in both files, the position in the resulting file is given by the sum of the two positions. If the domain is not present in a file, the final position will be equal to the position of the domain reported in the file in which it is present + 100k. We assign the position a value of "100K" because, in this way, we signal that the domain is not included in the top ten thousand domains on that date. Then, we performed a comparison between the resulting files of each week and those of the baseline with a similar procedure.

7.4 Experimental Results

Starting from the resulting files, obtained by applying the approach explained in 7.3.2, we have analyzed several domains related to different categories of most used applications, both web and mobile applications. We considered the following categories and related domains:

- **Social Media**: "facebook.com", "linkedin.com", "twitter.com", "snapchat.com", "instagram.com", "tiktok.com";
- **Video**: "netflix.com", "youtube.com";
- **Messaging**: "whatsapp.net", "whatsapp.com", "telegram.org"

- **Collaboration Tool**: "zoom.us", "teams.microsoft.com", "skype.com", "webex.com", "hangouts.google.com".

We have chosen these categories to analyze the change in both entertainment and leisure applications (*i.e.,* video, social media, and messaging) and collaboration applications, helpful for distance learning and the smart working to which the whole world had to bend.

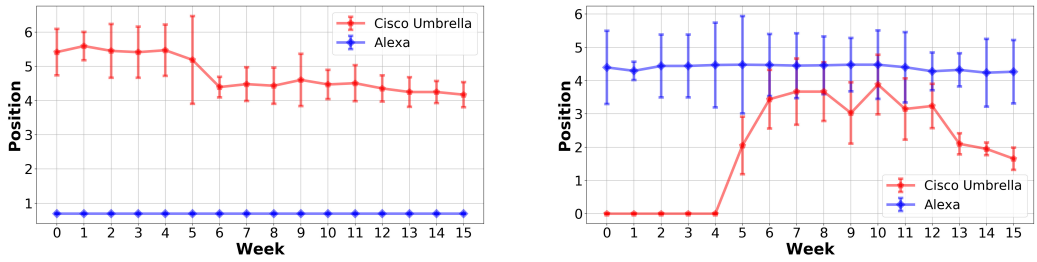
Table 7.1. Time Frames

Label	Time Span
0	Nov - Dec 2019
1	1 Jan - 5 Jan 2020
2	6 Jan - 12 Jan 2020
3	13 Jan - 19 Jan 2020
4	20 Jan - 26 Jan 2020
5	27 Jan - 2 Feb 2020
6	3 Feb - 9 Feb 2020
7	10 Feb - 16 Feb 2020
8	17 Feb - 23 Feb 2020
9	24 Feb - 1 Mar 2020
10	2 Mar - 8 Mar 2020
11	9 Mar - 15 Mar 2020
12	16 Mar - 22 Mar 2020
13	23 Mar - 29 Mar 2020
14	30 Mar - 5 Apr 2020
15	6 Apr - 13 Apr 2020

We will analyze each category in detail in the following subsections. For each domain, we have reported the error bar plot, where we plotted the average position and the variance for each week. In particular, the x-axis represents a specific time frame, whose order is listed in the table 7.1, and the y-axis shows the average position in the logarithmic scale.

7.4.1 Video Applications

In the video section, we dwelt on Youtube and Netflix, both accessible by both browsers and applications. There are a lot of domains related



(a) "youtube.com" **(b) "netflix.com"**

Figure 7.2. Video Category

to them, but we have considered "youtube.com" and "netflix.com". Figure 7.2a shows the trend of the domain "youtube.com". In Alexa, this domain presents a regular trend; in particular, it has a constant average and zero variance. It always holds the second position on the list. Instead, in Cisco Umbrella, this domain always occupies the first position until the last week of January, then, later, lower positions in the ranking than the Alexa list with a dispersion of data around the average value. After the second week of February, this domain gains some positions. The "netflix.com" domain, 7.2b, occupies low positions in Alexa rather than Cisco Umbrella, where the trend is constant in the two last months of 2019 and in every week of January 2020, and then there is an increase in the next weeks. Therefore, in the video category, the two domains analyzed have opposite behaviour in the two datasets: for Alexa, the more popular domain is "youtube.com", for Umbrella, instead, is "netflix.com". The two domains, related to video streaming, were highly searched during the lockdown. In summary, "youtube.com" has been most searched by users via browsers and not by other types of applications such as (mobile phones, televisions and so on), in accordance with SimilarWeb and Apptopia [198]. We can observe the opposite behaviour in the "netflix.com" domain.

7.4.2 Social Media

In the social media category, we have analyzed the six domains related to the most known applications: "facebook.com", "instagram.com",

"tiktok.com", "snapchat.com", "linkedin.com", "and twitter.com". In the Top 1M files of Alexa, the domain "facebook.com" occupies the highest position in the list, against to Cisco Umbrella dataset where the most popular domain, in this category, is "twitter.com". Besides, the domains "instagram.com", "linkedin.com", "twitter.com" and "tiktok.com" achieve better positions in Alexa's list and not in Cisco Umbrella's dataset. In contrast, "snapchat.com" occupies the upper position in Cisco Umbrella's list. Another significant aspect is related to the domain "facebook.com", which holds better positions in Alexa except for an overlap in the weeks of February and March. In all of the cases, there is an evident high value of standard deviation.

During this period of lockdown, the social media that has evolved the most has been "tiktok.com", with a decrease in the trend, and so rising in the ranking, from the first weeks of February. This decrease is more evident in Cisco Umbrella than in Alexa, probably because the greater use was due to applications from different types of devices and not by browser. In fact, different newspapers testify to the wide use of this application by users [210]. Another interesting trend to observe is "facebook.com", which shows how, during this lockdown period, this domain gained several positions in the Umbrella list between February and March. During the same weeks, in Alexa's list, this domain loses some positions in the ranking. We justify this behaviour with the same motivation as "tiktok.com", so more use by applications, through for example a mobile device, and not by browser, in contrast with the analysis reported by the New York Times [198].

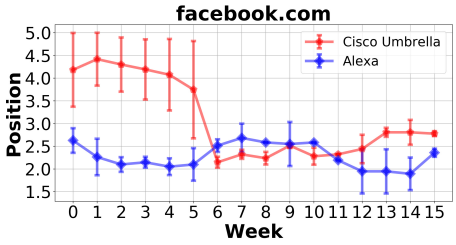
7.4.3 Messaging Applications

In the messaging category, we have analyzed three domains: "whatsapp.com", "telegram.org" and "whatsapp.net". The last domain is in the top 1 million of Cisco Umbrella and not on Alexa's list. Both for Umbrella and Alexa, the messaging domain most popular during this period of the pandemic is "whatsapp.com", followed by "telegram.org". More in detail, both "whatsapp.com" and "telegram.org", figures 7.4 (a) (b), take lower positions in the ranking in Alexa against Umbrella, probably for the same motivation of the video category 7.4.1. Therefore, for these two domains, the number of HTTP requests by the browser is greater than the number

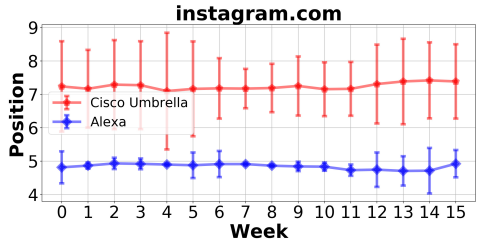
of queries to the OpenDNS server, which includes applications on different port numbers. The "whatsapp.net" domain (Figure 7.7 (c)) presents a high dispersion of data around the mean value, but the trend is fairly constant. During the lockdown period, we notice a significant change in telegram application usage, related to the "telegram.org" domain. This change is featured in the Umbrella dataset where this domain wins many positions in the rankings. Instead, for the "whatsapp.com" domain, there is an increase and a decrease respectively in Alexa and Cisco Umbrella lists. The "whatsapp.net" domain (Figure 7.7) has not undergone excessive variations during the quarantine.

7.4.4 Collaboration Tools

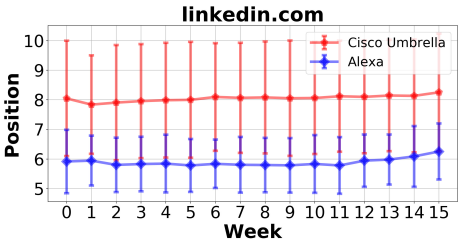
In the collaboration category, we have analyzed five domains related to the most popular platforms of collaboration: "skype.com", "zoom.us", "webex.com", "teams.microsoft.com", "and hangouts.google.com". Among these domains, only three are included in both datasets: "zoom.us", "skype.com", "and webex.com". The two remaining ones, *i.e.,* "teams.microsoft.com" and "hangouts.google.com", are contained only in the Cisco Umbrella's list because in the files of the top 1 million of Alexa are not present subdomains 7.3.1. The collaboration domain with the best performance (Figure 7.5) is "skype.com" in the Umbrella's dataset, followed by "zoom.us" and "webex.com" where there is an overlap between the two lists. In particular, we can notice a relevant aspect in the trend of zoom.us in correspondence of week 12: March 16th - March 22nd. In this interval, the domain acquired many positions in the rankings which may be justified considering that, during the first two weeks of March, the beginning of the use of this tool by companies, universities, and schools but also for entertainment was recorded. Instead, among the two domains contained only in Umbrella's list, the one with the highest position in the ranking is "teams.microsoft.com" (Fig. 7.6.



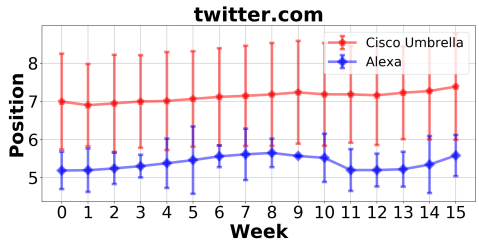
(a) "facebook.com"



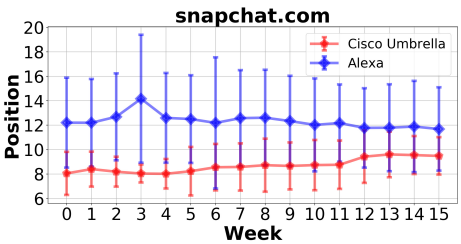
(b) "instagram.com"



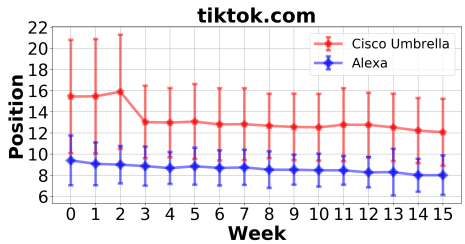
(c) "linkedin.com"



(d) "twitter.com"

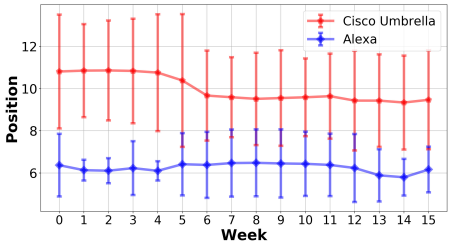


(e) "snapchat.com"

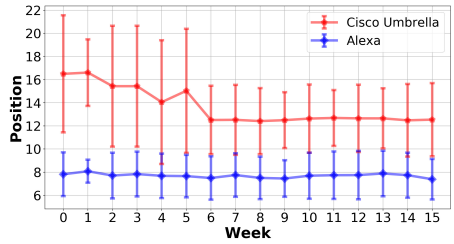


(f) "tiktok.com"

Figure 7.3. Social Media Category

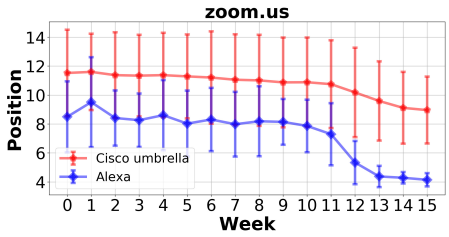


(a) "whatsapp.com"

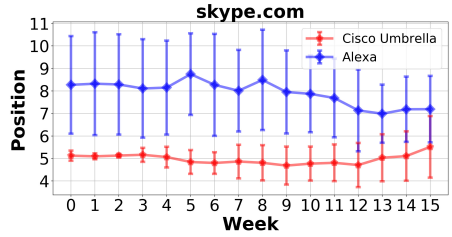


(b) "telegram.org"

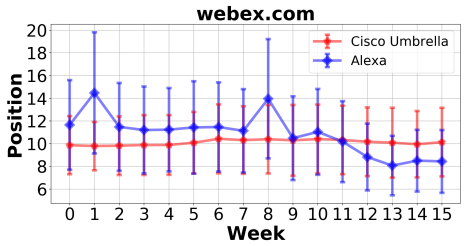
Figure 7.4. Messaging Category - common domains



(a) "zoom.us"

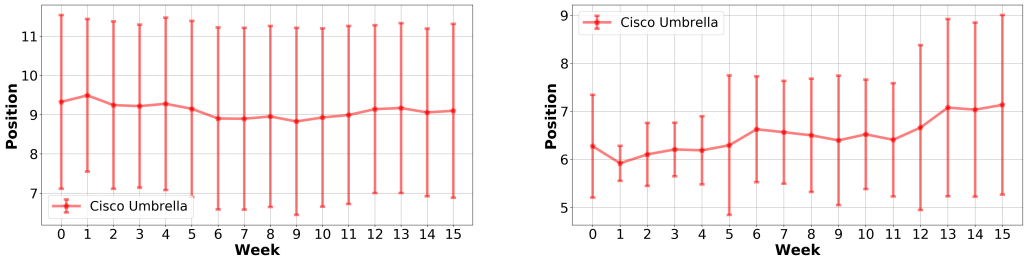


(b) "skype.com"



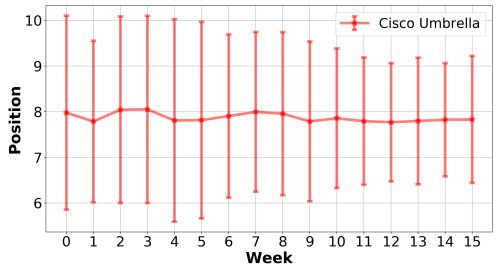
(c) "webex.com"

Figure 7.5. Collaboration Tool Category - common domains



(a) "hangouts.google.com" **(b) "teams.microsoft.com"**

Figure 7.6. Collaboration Tool Category - Umbrella's domains



"whatsapp.net"

Figure 7.7. Messaging Category - Umbrella's domain

7.5 Concluding Remarks

Web and mobile applications played a significant role during the lockdown period when the whole world was forced to hole up in their homes and go out only for basic needs. Consequently, most employees were forced to adopt smart working, while schools and universities embraced distance learning, leading to the widespread use of video conferencing applications. Not only for work and distance learning, but several applications have also been essential in your free time and to keep in touch with friends and relatives.

In this work, we have analyzed the trends of different applications, belonging to different categories, *i.e.,* video, social media, messaging and collaboration tools. For this analysis, we have adopted two different, complementary data sources: the top 1 Million lists provided every day by Cisco Umbrella and Alexa, containing the most popular domains and websites respectively. While confirming some results covered by the press, our results also show interesting differences noticed contrasting Alexa and Umbrella, *e.g.,* youtube.com always occupies position 2 in Alexa', higher than netflix.com, which, in turn, occupies a higher position in Umbrella; facebook.com occupies higher positions in the Alexa with respect to Umbrella, where the most popular domain is twitter.com; whatsapp.com features an increase and a decrease respectively in Alexa and Umbrella; skype.com has the best performance in Umbrella, followed by zoom.us and webex.com. These results are related to the use of different devices by the users, which are differently captured by the two lists. In our ongoing work, we are looking at the possibility to dissect the analysis at the Country level, as the lists used do not provide such information. Moreover, we are using a systematic approach [211] to understand the performance of the network during the lockdown. Preliminary results show that despite the major change in network access, use and operation conditions, no significant impact on the applications and network performance has been observed.

Chapter 8

Impact of Ukraine Conflict on Russian Internet

In this Chapter, we investigate the impact of another global, societal event through DNS data. The hostilities in Ukraine, indeed, have driven unprecedented forces, both from third-party countries and Russia, to create economic barriers. In the Internet, these manifest both as internal pressures on Russian sites to (re-)patriate the infrastructure they depend on (*e.g.,* , naming and hosting) and external pressures arising from Western providers disassociating from some or all Russian customers. While quite a bit has been written about this both from a policy perspective and anecdotally, this chapter places the question on an empirical footing and directly measures longitudinal changes in the makeup of naming, hosting and certificate issuance for domains in the Russian Federation.

8.1 Motivation

On February 24, 2022, Russian forces invaded Ukraine, leading to the the largest refugee crisis in Europe since World War II. Unlike Russia's 2014 annexation of Crimea or ongoing support for separatists in Ukraine's south-east, this escalation produced a strong global response — particularly from Western countries. In addition to providing military and financial support for Ukraine, Western countries imposed broad economic sanctions against Russian entities, including the Russian Central Bank,

imposed export controls to deny Russia access to strategic material, seized or froze property and assets held abroad, and imposed flight bans and travel restrictions. In addition to these government actions, a broad array of roughly 1,000 private sector companies independently restricted or exited the Russian market [212].

The Internet has not escaped this conflict. For example, the US Office of Foreign Asset Control (OFAC) started listing particular Russian corporate Web sites on its Specially Designated Nationals (SDN) list of sanctioned entities [213]. Independent of these particular sanctions, many western Internet service companies have decided — for some combination of moral principle, reputational risk and/or economic volatility — to broadly disengage from the Russian market. While some have simply halted new sales to Russian customers (*e.g.,* , Amazon, Microsoft, Google [214], GoDaddy [215]), others, such as Cogent, have stopped providing service to Russia entirely [216]. Ukraine has advocated for such actions and on March 1st, 2022, their Deputy Prime-Minster formally requested that ICANN revoke the `.ru`, .рф and `.su` domains, support the revocation of all TLS certificates for those domains and shut down DNS root servers located in the Russian Federation [20].

These actions have reinforced Russia's long-held concerns about threats to their "Internet sovereignty", leading the government to take proactive steps to repatriate key services.[1] In March 2022, Russian authorities mandated that all state-owned websites and services switch exclusively to domestic ISPs, DNS operators and hosting providers [11]. Similarly, the Russian Ministry of Digital Development announced that it was standing up an independent state-operated CA whose root certificate would be trusted by Russian browsers (VK Atom and Yandex.Browser).[2] Russian private sector operators have also started to anticipate third-party disengagement: RU-CENTER, Russia's leading registrar and hosting provider, advised customers "operating in sectors subject to international sanctions"

[1] Russia has a long of history of trying to exert control over its domestic Internet, including requirements for domestic data storage and surveillance [217] and the ability, recently tested by communications regulator Roskomnadzor, to actively disconnect the country from the global Internet if needed.

[2] The timing of this action appears to have been related to DigiCert's revocation of Russian Bank VTB's TLS certificate — presumably in response to VTB's sanctioning by the US OFAC.

to "purchase certificates by GlobalSign, a Japanese certification authority" [218].

These internal re-patriation pressures from the Russian government, combined with the risk of further shunning by Western service providers, suggest an unprecedented environment for the Russian operators and their enterprise customers. It would be entirely reasonable to hypothesize that these forces are driving Russian sites to rapidly decouple from non-Russian infrastructure. This chapter is an attempt to put this question on an empirical footing.

In particular, we explore the longitudinal changes in the infrastructure used by Russian sites — notably DNS, hosting, and TLS certificate issuance — before and after the invasion of Ukraine. Our analysis combines five years of daily `.ru` and .рф zone transfer data, with contemporary active measurements and historic certificate issuance data. We explore the extent to which such sites have experienced significant patriation of their infrastructure and, to the extent such changes exist, whether they can be best explained by the actions of service providers outside Russia or by the anticipatory decisions made by Russian site operators themselves.

8.2 State of the Art on Russia Internet Infrastructure

The relation between state political interests and Internet communication has become an important field of study, ranging from analyses of global state censorship [219, 220] to the use of blocking, denial-of-service attacks and wholesale closing of Internet access to control opposition forces [221, 222]. Specific to Russia, Moyakine *et al.* [223] explore the 2015 *Yarovaya* counter-terrorism law, which mandated extensive surveillance requirements on Russian telecommunication providers and its impact on the communication of vulnerable groups. Epifanova and Dietrich [217] explore Russia's contemporary goals for "digital sovereignty", both for controlling domestic communication and to reduce dependence on foreign IT services. This goal is evident in empirical studies by Zembruzki *et al.* [224] and Liu *et al.* [225], who analyze the centralization of hosting and e-mail service with a small number of Western providers, but show that Russia bucks this trend with a heavily centralized infrastructure. Ramesh *et al.*

[226] analyze the centralized blocking policy dictated by Roskomnadzor to characterize Russian content blocking and the differential experience between residential and business customers.

8.3 Data Sources

We use DNS measurement data of all domain names registered under the Russian Federation country code top-level domains (**ccTLDs!**s) `.ru` and .рф[3] over a nearly five-year period (1803 days). The exact period of our study is June 18, 2017 through May 25, 2022, meaning the data extends years before Russia's invasion of Ukraine on February 24, 2022, and also extends 90 days forward of this point.

The DNS measurements were provided by the OpenINTEL project, which uses daily zone file snapshots as seeds to actively query all registered domain names under a TLD for a selection of DNS resource records [227].[4] The collected data include each domain's `NS` records (to investigate whether name service is delegated outside `.ru` and .рф), as well as the `A` record resolution for both their name servers and apex domain. We geolocate each of the resulting IP addresses, using contemporaneous results from the IP2location service [228], to provide a proxy for the physical hosting of each domain's DNS infrastructure and Web site, respectively.[5] Our dataset contains *11.7M* unique Russian Federation domain names, and *13.3k* and *9.5k* unique networks (AS numbers) that, respectively, hosted domain apexes or authoritative DNS infrastructure.

We also collected longitudinal certificate data for the `.ru` and .рф domains using both historic certificate transparency logs, as well as active scans by Censys [229] of Internet Web sites during the collection period.[6] Finally, we label 107 unique domains as being specifically *sanctioned* based on their appearance on either US OFAC SDN [213] or UK sanctions

[3] .рф is the Cyrillic code for Russian Federation. The internationalized domain name form of this **ccTLDs!** is `.xn-p1ai`.

[4] https://openintel.nl/coverage/

[5] We note that there is a small percentage of disagreement in country-level geolocation and inferences made regarding relocation may "lag behind," in particular when IP address (space) of hosting or DNS infrastructure is moved rather than changed.

[6] We consider a certificate to "match" if either its *Common Name (CN)* or *Subject Alternative Name (SAN)* fields include a domain name under a `.ru` or .рф TLD.

lists [230].[7]

8.4 Impact on DNS Ecosystem

In this section, we first provide historical context for the DNS infrastructure supporting .ru and .рф domains, and then focus on activity surrounding the 2022 invasion for all Russian domains, sanctioned Russian domains, and the actions taken by major Western providers.

8.4.1 Historical Context

For historical context, we start by characterizing the long-term locations of Russian domain hosting and name server infrastructure across our full data set from June 18, 2017, to May 25, 2022. We label a domain as *fully* Russian-hosted if all of its A records geolocate inside the Russian Federation, *partial* if only a subset are in Russia, or *non* (Russian) if all such records are located outside the Russian Federation. Name service is similarly labeled based on geolocating the authoritative name servers for the domain.

Historically, the fraction of domains hosted in Russian networks only fluctuates mildly over our period of study. For example, on June 18, 2017, *71.0%* of .ru and .рф names are *fully* hosted in Russia, *0.19%* are *partial*, and *28.81%* are *non* Russian. This hosting breakdown does not change significantly until the Ukrainian invasion in February 2022. At that point, there is a *slight* increase in both *fully* and *partial* domains driven by flight from the US and other Western countries to a combination of Russia and the Netherlands.

The name server infrastructure for Russian domains is also relatively stable over the long term, but shows a more pronounced change once the conflict starts. Figure 8.1 shows this longitudinal name server breakdown in more detail. For all domain names registered under .ru and .рф, it displays whether their delegated name servers are *fully*, *partially* or altogether not located inside Russia.[8] The black curve shows the total number

[7] While the US OFAC subsequently issued license exceptions for a range of Internet services on April 22, 2022 [231], we have not observed clear changes in certificate issuance behavior in response to this modified policy.

[8] The dip on March 22, 2021 is a measurement outage.

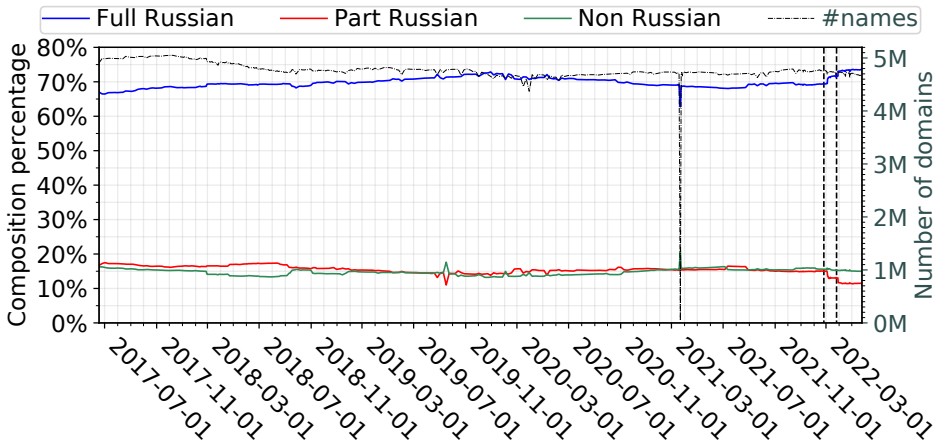


Figure 8.1. Country composition of DNS infrastructure of `.ru` and .рф domain names. *Full* means the authoritative name servers fully geolocate to Russia. *Non* means the servers altogether do not. *Part* means they partially do.

of Russian domains (right ticks). As points of reference, we divide recent months into three time periods: pre-conflict (before February 24, 2022), post-sanctions (after March 26, 2022), and pre-sanctions (the period in-between). We delineate these periods in the graphs with vertical dashed lines.

On June 18, 2017, there are just under $5\,M$ registered domains, *67.0%* of which have name servers *fully* located in Russia. This breakdown, along with the roughly equivalent levels of *partial* and *non* domains, is stable over time, suggesting that internal patriation pressure in the years immediately prior to the 2022 conflict have had little bearing in practice. Changes do become apparent in February 2022, when many domains with name servers *partially* outside Russia clearly transition towards *fully* Russian. However, in historical context, these changes are minor. For our most recent data, *73.9%* names are *fully* Russian, only a *6.9%* change over the five-year period.

One aspect of Russian domain infrastructure that becomes less Russian-focused over time are the TLD dependencies of Russian domains. We extract the TLD of each name server to which `.ru` and .рф domain names

delegate authority. If all of a domain's name servers are exclusively registered under the Russian Federation TLDs, we consider the TLD dependency *fully* Russian. Similar to prior categorizations, if only a subset are Russian TLDs, we consider it *partial*, otherwise we consider it *non* Russian.

Figure 8.2 shows the name server TLD composition breakdown over time. Perhaps counter-intuitively, there is a slight downward trend in *fully* Russian (a net reduction of *6.3%* comparing extrema), and an increase in *partial* (a net increase of *7.9%*). Over time, Russian domains increasingly delegate to name servers whose names are in non-Russian TLDs, implicitly increasing their dependence on external infrastructure, which could become subject to Western sanctions. Figure 8.3 shows a longitudinal breakdown of specific TLDs under which authoritatives of Russian domains are registered. We show the Top *5* TLDs (out of a total *270*). Unsurprisingly, most Russian domains delegate to name servers with a name in `.ru`: *78.3%* on May 25, 2022. Second is `.com` with *24.7%* of Russian domains (a net increase of *7.5%* over the five-year period). Next in rank are: `.pro` (*12.4%* up from *8.8%*), `.org` (*9.2%* up from *8.2%*), and `.net` (*7.3%* down from *9.1%*). The remaining TLDs see $<$*1.0%* each (on May 25).

TLD dependency trends also change at the start of the conflict. Both *fully* and *partial* Russian compositions (Figure 8.2) increase very slightly (by *0.2%* and *0.5%*, respectively). As a result, the small fraction of Russian domains that changed from a *non* composition are less exposed to potential Western interference. Those that remain could become unresolvable in case the authoritatives stop providing service or Russia disconnects itself from the global Internet.

8.4.2 Recent Activity

In the post-conflict period, Russian domains have experienced more movement in their hosting networks, but the movement has almost entirely been among networks outside of Russia. Figure 8.4 shows a selection of providers networks that host `.ru` and .рф domain names. The Russian ASNs have stable and consistent customer bases over time, together accounting for *38%* of Russian domains at the start and *39%* at the end. The other stable curve is Cloudflare, which accounts for nearly *7%* of Russian domains throughout this period. The networks that *do* experience

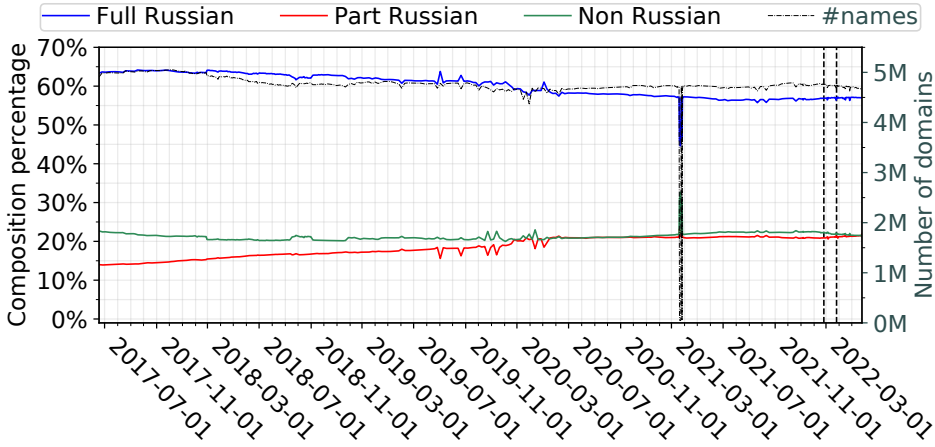


Figure 8.2. TLD dependency composition of `.ru` and .рф domain name authoritatives. *Full* means the name servers are all registered under Russian TLDs. *Non* means none are. *Part* means some but not all are.

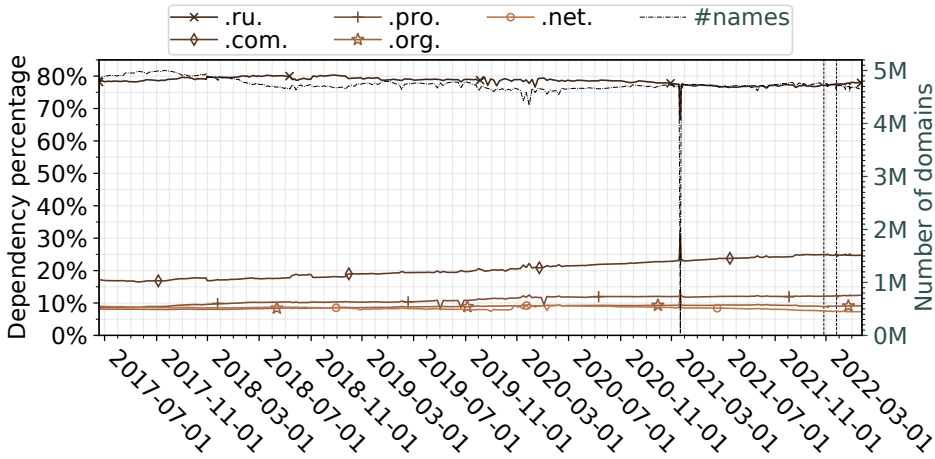


Figure 8.3. Top *5* TLDs used by authoritative name servers of `.ru` and .рф domain names. The other *265* TLDs (not shown) see *<1%* each.

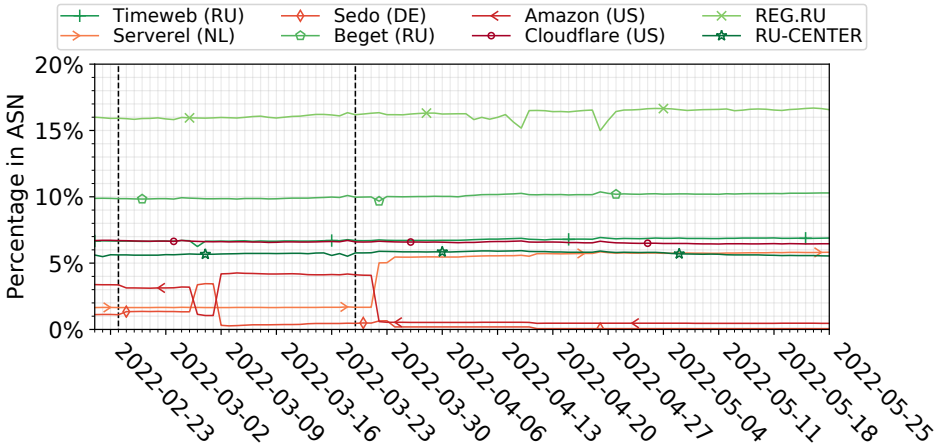


Figure 8.4. Hosting networks of `.ru` and .рф domain names (Top ASNs). The share of Russian domain names that each network hosts is shown. The vertical dashed lines delineate the *pre-conflict*, *pre-sanctions* and *post-sanctions* periods.

movement correspond to `.ru` and .рф domains that switch back and forth between Amazon (US) and Sedo (Germany), and then ultimately move to Serverel (Netherlands). This dynamic is, in part, driven by business reactions to the conflict, which we discuss further in Section 8.4.4.

Russian domains have also experienced changes regarding where their DNS infrastructure is hosted, with noticeable movement starting during the pre-sanctions period and continuing post-sanctions. A significant change involved Netnod, a Swedish DNS provider, and RU-CENTER, a large Russian domain name registrar and (former) Netnod customer. Due to IP address reconfigurations on March 3rd, Netnod stopped providing service for *76 k* Russian domains, which quickly changed from *partial* to *fully* Russian DNS infrastructure (Figure 8.1). We observe other large transitions at the end of March involving migration out of the networks of Hetzner (Germany) and Linode (US). One non-Russian network that hosts DNS infrastructure for a substantial number of Russian domains is Cloudflare, and this network sees little change since the conflict started.

8.4.3 Sanctioned Domain Names

We now focus on domain names specifically tied to Russian entities that were sanctioned by the US and UK.

Note that the potential for impact on the hosting of these domains is inherently slight as *101* of the *107* sanctioned domains (*94.4%*) were already hosted exclusively in Russian ASNs before the conflict on February 24, 2022. Three more became *fully* Russian hosted by May 25, 2022,[9] and the final three have remained *fully* hosted in Germany, Czech Republic, and Estonia.

However, the name server infrastructure for these sanctioned domains *has* experienced significant movement. Figure 8.5 shows the country composition of the authoritative name servers for these domains over time. The three colored curves again distinguish among *fully*, *partial* and *non* Russian composition, and the black curve shows the daily total number.

On February 24, 2022, *34.0%* of sanctioned domains are *partial* and *5.2%* *non* Russian. This situation drastically changes by March 4, 2022 when the vast majority (*93.8%*) of the DNS infrastructure for the sanctioned domains are strictly hosted in Russia. Note that for the *partial* sanctioned domains that changed to *full*, nearly all of them had an authoritative name server hosted by Netnod (in Sweden) until the change to *full* Russian on March 4.

8.4.4 Actions taken by Providers

A number of Western providers publicly stated the business actions their company would take in response to the conflict, either in voluntary protest or for alignment with sanctions. Using our DNS data, we examine the extent and effect of the business actions taken by four major Western providers.

Amazon – On March 8, 2022, Amazon reported that it would no longer be accepting new Russian or Belarusian AWS account registrations [232]. Since that time, we see significant changes in the makeup of `.ru` and .рф domains resolving to Amazon's ASN (`AS16509`), including the surprising appearance of newly hosted domains from these TLDs.

[9] These three domains were previously hosted exclusively in Germany or Poland.

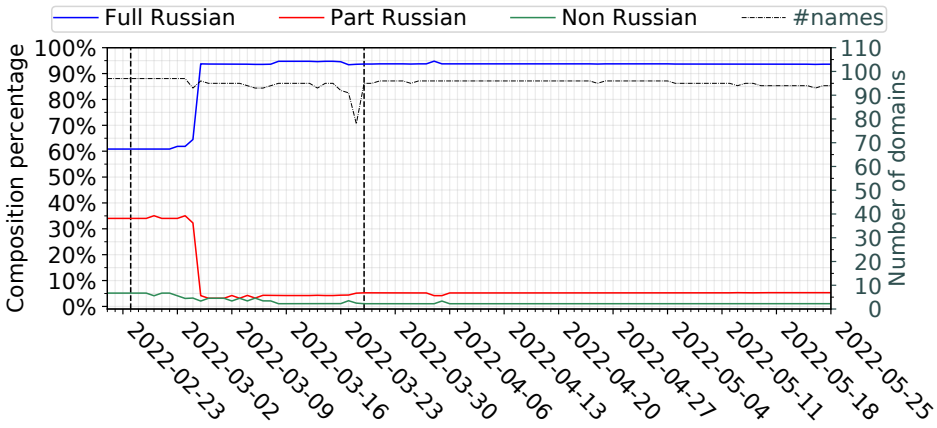


Figure 8.5. Country composition of DNS infrastructure authoritative for *sanctioned* Russian domains, broken down in fully, partially, and not geolocated to Russia. Significant movement is seen in the *pre-sanctions* period.

Figure 8.6 displays the movement of Russian domains that originally resolved to Amazon's ASN on March 8, 2022. By May 25, 2022, more than half of these domains relocated to other ASNs, but we do not know whether this reflects Amazon's initiative or independent customer decisions. A little under half (*43%*) remained, but this set also includes *574* newly registered `.ru` and .рф domain names (confirmed using Cisco's *Whois Domain* API) and *988* existing domains that relocated to Amazon. While this influx of *1.5k* `.ru` and .рф names appears inconsistent with Amazon's statement, it is possible that these domains are owned by existing customers.[10]

Sedo – On March 9, 2022, it was reported that *Sedo* was "pulling the plug" on Russian domains [233]. Sedo followed through on its stated intention, although the plug was not pulled completely. Figure 8.7 shows the significant movement of `.ru` and .рф name hosting from Sedo's `AS47846`. Starting on March 8, 2022, *164k* `.ru` and .рф domains resolved to Sedo's ASN (`AS47846`). By May 25, 2022, *160k* (98%) had relocated to a different

[10]Using Cisco's *Whois Domain* API, we found registrant information for a subset of these domains (≈*1/6th*). Manual inspection revealed that some registrations were business-as-usual or defensive, by non-Russian, existing Amazon customers (*e.g.,* , Disney registered various brand names such as `thorloveandthunder.ru` and `blackpantherwakandaforever.ru`).

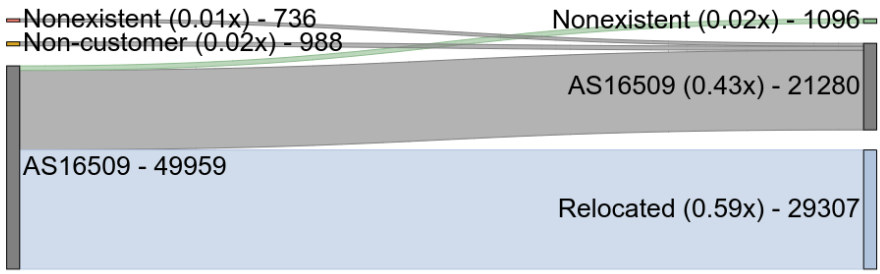


Figure 8.6. Russian domain name movement in *Amazon*'s `AS16509` (comparing 2022-03-08 and 2022-05-25).

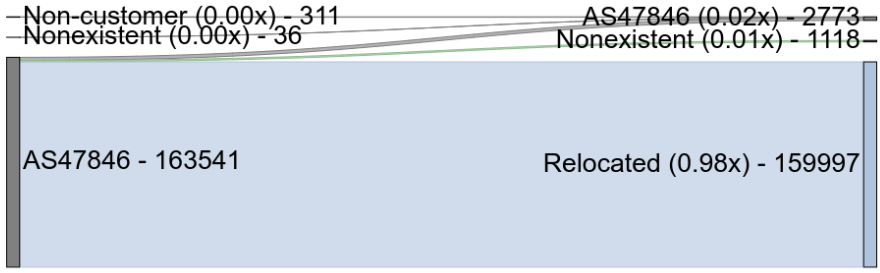


Figure 8.7. Russian domain name movement in *Sedo*'s `AS47846` (comparing 2022-03-08 and 2022-05-25).

ASN, *2.7k* (1.6%) remained, and *311* external domains relocated to Sedo.

Cloudflare – Cloudflare wrote in a March 7, 2022, article that it was complying with sanctions [234]. It also expressed that, in consultation with government and civil society experts, the company would not terminate Cloudflare's services inside Russia. The domain resolutions confirm that the company is doing business as usual. Starting March 7, 2022, nearly *315k* `.ru` and .рф names resolved to `AS13335`. On May 25, 2022, a little over *296k* (*94%* of the original set) remain in Cloudflare's AS, and *34k* Russian domains newly appeared. This activity is consistent with the sentiment expressed by Cloudflare's CEO Matthew Prince, that "Russia needs more Internet access, not less" [234].

Google – On Thursday, March 10, 2022, a Google spokesperson was reported as saying that the company would stop accepting new customers in Russia [235], but declined to comment if existing cloud customers in

Table 8.1. Issuing activity of Certificate Authorities in the three-time periods in 2022.

Pre-Conflict			Pre-Sanctions			Post-Sanctions		
Issuer Org.	**# Certs**	**(%)**	**Issuer Org.**	**# Certs**	**(%)**	**Issuer Org.**	**# Certs**	**(%)**
Let's Encrypt	6,586k	91.58%	Let's Encrypt	3,285k	98.06%	Let's Encrypt	5,458k	99.23%
DigiCert	244k	3.40%	GlobalSign	25k	0.76%	GlobalSign	28k	0.52%
cPanel	153k	2.13%	cPanel	11k	0.34%	Google	13k	0.24%
Other CAs	207k	2.89%	Other CAs	28k	0.84%	Other CAs	422	0.01%

Russia would see action taken. Starting on March 10, 2022, *17.7k* `.ru` and .pф domains resolved to Google's ASN (`AS15169`). By May 25, 2022, *57.1%* (*10.1k*) of these domains had relocated to a different ASN, but most of these (*75.2%*) had simply relocated to a different Google ASN (`AS396982`).[11] In this period, a small number of external Russian domains (*187*) and newly registered domains (*184*) relocated to Google. As with Amazon, while seemingly inconsistent with Google's stated policy, it is possible that this influx of domains was created by existing customers.

8.5 Impact on Web PKI Ecosystem

In the modern Web ecosystem, TLS certificates are crucial infrastructure for securing domains. In this section, we examine how Certificate Authorities (CAs) have reacted to the conflict and sanctions in terms of the certificates they authorize for Russian domains.

On the one hand, the conflict and sanctions *have not* significantly undermined the number of certificates issued for `.ru` and .pф domains from global CAs. For our three time periods in 2022, CAs issued *130k* certificates per day on average pre-conflict, *115k* certificates per day *pre-sanctions*, and *115k* per day *post-sanctions*. However, individual CAs *have* reacted very differently to the conflict, and in this section we characterize the behavior of global CAs who issue and revoke certificates, as well as the effect of the new Russian Trusted Root CA.

[11] Using OpenINTEL DNS measurement data of non Russian Federation domain names, we observe significant relocation from `AS15169` to `AS396982` for names under other TLDs too (around March 16). As such, we conclude that this intra-Google relocation did not occur because the *8.5k* (*75.2%* of *10.1k*) domains are Russian.

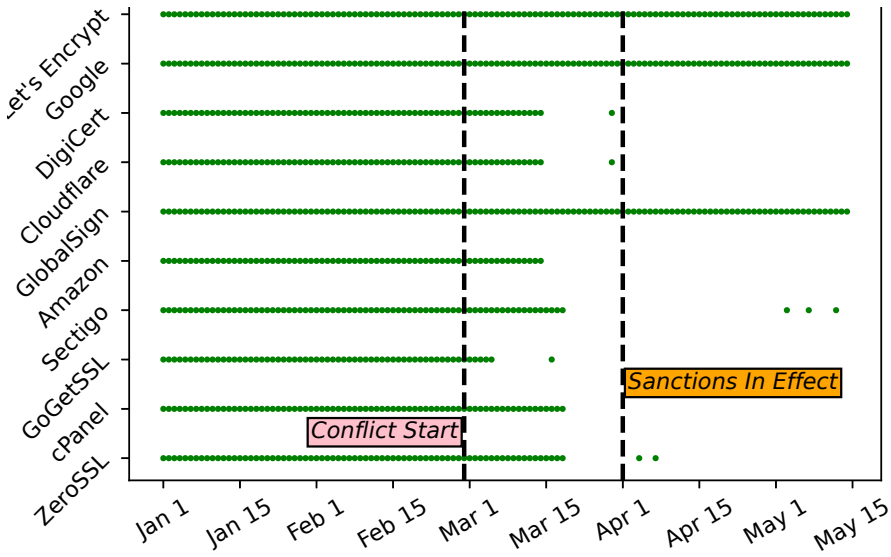


Figure 8.8. Timelines for CAs issuing new certificates for `.ru` and .рф domains. A green dot indicates the CA issued at least one certificate on the day.

8.5.1 Shift in Certificate Issuance

We use the Certificate Transparency (CT) logs indexed by Censys [229] to obtain the TLS certificates securing an `.ru` or .рф domain from January 1, 2022 to May 15, 2022. For each certificate, we extract the Issuer Organization term from the Issuer DN field to identify the CA responsible.
Figure 8.8 shows timelines for when the top 10 CAs issue new certificates for Russian domains. A green dot indicates that the CA issued at least one certificate for a `.ru` or .рф domain on that day. Six of the ten top CAs for Russian domains stopped issuing certificates altogether after the conflict started or sanctions were imposed. The three CAs that continue issuing certificates are now the only major issuers for `.ru` and .рф domains. Since CAs typically issue certificates under different Common Names (CNs) (*e.g.,* , DigiCert issues certificates under CNs RapidSSL and GeoTrust), we suspect the isolated dots are likely a result of CAs not preventing issuance from their lesser known CNs.
Table 8.1 shows the number of issued certificates in each of the three time

Table 8.2. Revocation activity by the five CAs with the most revocations.

	.ru and .рф Domains		Sanctioned Domains	
Issuer	**Issued**	**Revoked**	**Issued**	**Revoked**
Let's Encrypt	15M	10k (0.06%)	16k	196 (1.19%)
DigiCert	247k	2.1k (0.80%)	308	308 (100%)
GlobalSign	95k	1.6k (1.68%)	905	23 (2.54%)
Sectigo	96k	5.1k (5.15%)	164	164 (100%)
ZeroSSL	56k	165 (0.30%)	82	2 (2.43%)

periods for the top three issuing CAs in each period. Overall, the effect of the conflict has been to further concentrate certificate activity to just three CAs. While Let's Encrypt already dominated the market before the conflict, it increases its share to more than *99%* afterwards. Pre-conflict there was a long tail of CAs issuing certificates, but post-conflict only three CAs effectively participate.

8.5.2 Revocation Activity

Issued certificates only paint half the story: not only have many CAs stopped issuing new certificates, but some have responded by also fully revoking sanctioned domains. Using the the Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) state as indexed by Censys, we tallied the revocations for certificates securing `.ru` and .рф domains across all CAs whose validity ended after February 25, 2022.
Table 8.2 shows the breakdown of domains issued and revoked by the top five CAs with the most revocations. Significantly, both DigiCert and Sectigo have revoked the certificates for all of the sanctioned domains that they issued, apparently choosing to remove any risk of engagement. Although we have no insight into individual CA policy decisions, we note that all CAs have significantly higher revocation rates for sanctioned domains than other `.ru` and .рф domains. We also suspect some revocation activity may be initiated by the sanctioned domains themselves as they navigate the sanctions by testing different CAs.

8.5.3 Russian Trusted Root CA

The creation of the Russian Trusted Root CA by Russia's Ministry of Digital Development received significant attention when announced. In addition to being a state-run CA, it does not record its issued certificates in the CT logs and is not trusted by major browsers.[12] To evaluate the initial impact of this new Russian CA, we used the Censys Universal Internet Data Set (CUIDS), which performs daily Internet-wide IP scans that index all TLS certificates returned from responding IP addresses.[13] Using these results we identified all TLS certificates containing the Russian CA in their certificate chain, between its inception and May 15, 2022.
The certificate scans show two trends. First, very few sites are offering certificates from the Russian CA: only 170 unique certificates from the Russian CA are seen in the CUIDS data. For context, all other CAs issued more than *800 k* certificates for Russian domains in the same time period. While the metrics are not the same — far more certificates are issued than are in active use — the small number of active certificates from the Russian CA indicates it has yet to have a significant impact on the overall Russian domain ecosystem. Second, as expected, the certificates all secure Russian-related entities, many of which are sanctioned domains. The 170 certificates secure 130 `.ru` and 2 .рф domains while the remainder, in a long tail of other TLDs, are affiliated with Russian sites. Based on the issuance times, the certificates seem to be issued over a period of a few weeks. Of the 170 certificates, 36 secure sanctioned domains (thus accounting for *34%* of the sanctioned domain list).

8.6 Concluding Remarks

The Russian government has long understood their potential exposure to foreign-operated Internet services. Government efforts to establish a "sovereign Internet" have included a range of regulatory requirements on service providers, including requirements for domestic storage of data on Russian citizens, the use of Russian-controlled DNS root instances, as

[12] Russian citizens were instructed to either use a state-approved browser or to configure their browser to accept the new CA.

[13] Since active scans of certificates are likely a subset of issued certificates, the scans represent a lower bound.

well as increasing pressure to prefer the use of domestic information and communications technology (ICT) services [217]. Perhaps most inflated is Russia's purportedly-tested capability to disconnect from the global Internet. Thus, even though Russia may have underestimated the magnitude of Western response to its invasion of Ukraine, it is clear that they understood the Internet could be a potential pressure point.

Indeed, we have clear empirical evidence of this pressure, with many thousands of Russian sites losing access to a range of Western service providers, *e.g.,* , Netnod for DNS hosting, Sedo for site hosting, and DigiCert and Sectigo for certificate issuance. However, these issues have been far from existential. First, Russia enjoys the benefits from high levels of pre-existing domestic provisioning. The vast majority of Russian sites ($\approx$*70%*) were fully hosted in Russia with entirely domestic name servers long before the start of the conflict.[14] Thus, while we see changes in single digit percentages, when measured against the entirety of the Russian Internet, these are modest effects. Second, for those Russian sites who have made use of non-Russian infrastructure, there are many providers who continue to service Russian customers, both within Russia and without. Thus, while prominent Western providers chose to leave the Russian market, virtually all of the impacted sites quickly found new providers. Moreover, we see little evidence of spontaneous or anticipatory repatriation by Russian domain operators who have not been forced to act.

Finally, we note that certificate issuance represents the one area of significant exposure for Russia. The near-complete control Let's Encrypt holds in securing `.ru` and .рф sites is startling. While Let's Encrypt has a public interest mission that provides free CA service to all comers, it is also a US entity and subject to US law and export control restrictions. Moreover, Russia does not appear to have anticipated this issue by establishing domestic CAs with similar capabilities and, most importantly, established trust relationships with the major browsers.

[14] Our data extends back to 2017, so we cannot establish if this domestic Internet service centralization represents Russia's longer-term state of affairs.

8.7 Ethics

In this chapter, we attempt to contextualize changes in underlying infrastructure of `.ru` and .рф domains as a result of push and pull from competing forces (internal and external to Russia) and the vision of "cyber sovereignty". While this type of analysis — identifying trends in infrastructure — does not raise ethical objections, we accept the sensitivities around the conflict and the implications of sanctions on the global Internet may raise concerns. While we recognize these concerns, we believe full transparency is the way ahead.

Chapter 9

Conclusions

This thesis provided two main contributions: the analysis of the cyber threats and the impact of global societal events over the Internet, allowing a better understanding and optimal operation of this fundamental network.

First, we have focused on how to prevent a network attack by identifying scanning activities (*i.e.,* port and network scans) in high-speed networks. To deal with the challenge of the large quantities of data to be analyzed, we have designed and developed a system exploiting Big Data techniques. This system is based on a statistical method and inspects network traffic traces at the flow-level. It achieves good performance in terms of precision and recall. We have also analyzed the execution time on three Amazon EMR availability zones. We have found that this system analyzes a 24-hour traffic trace in about 25 seconds, which is promising for real-time deployment. Finally, our system has detected also a greater number of scanning activities than a reference technique. Botnets also use scanning activities in order to find vulnerable devices. In this thesis, we have focused on the Mirai botnet that implements a particular signature in TCP SYN packets (TCP sequence number sets as destination address, TCP.seq==IP.dst). We have inspected the TCP SYN packets that verify this signature, the hosts that initiate these scanning activities, and the ports targeted. In contrast with the previous works, we have shown that this signature is still implemented in the initial scanning phase. Furthermore, we have identified new variants that probe many vulnerable ports.

Second, we have analyzed malicious domain names and the effects of

real-life events by studying a small fraction of Internet traffic: the DNS and its features. We have investigated the security performance of local DNS resolvers, provided by three main Italian ISPs, and two public ones, provided by Google and Cisco Umbrella. Specifically, we have performed a considerable number of DNS queries, using benign and malicious domain names. We have conducted this experiment considering both traditional DNS protocol and encrypted queries over HTTPS DoH. Our analysis has shown that local and public resolvers achieve approximately the same protection level. In addition, we have looked at the response times of these resolvers. We have shown that local resolvers are faster than public ones, also excluding cached domain names. Also, we have studied the domain name lifetimes related to top 10 TLDs over a ten-year period. Specifically, we have analyzed the last and first time a domain name was seen in the TLD zone files, considering cases where the NS record is not present in the file zone. With this approach, we have noticed that a noteworthy percentage of domain names last less than the minimum registration term (*i.e.,* one year). To further investigate the possible causes of such a short duration, we have compared these domain names with those included in the DBL blocklist. We have found that a high number of short-lived domain names are also used for malicious network activities. In addition, with further analysis, we have shown that some TLDs quickly remove malicious, short-lived domains from the zone files after appearing in a blocklist, while others do not. Finally, we have identified some malicious campaigns by looking at WHOIS data.

The last two issues addressed in this thesis are related to the study of the impact of large-scale, real-life events on the Internet. We have focused on understanding how the COVID-19 pandemic restrictions impacted several categories of Internet applications: video, social media, messaging, and collaboration tools. To this end, we have used the top 1 million lists provided every day by Cisco Umbrella and Alexa. We have implemented a system that re-evaluates every day the score of the domain names analyzed taking into account previous scores. With this approach, we have shown that, during the COVID-19 period, individuals used these applications for several purposes (personal and work). Netflix and Youtube were the most used, followed by Facebook and Skype. An interesting increase in the ranking is also shown by Whatsapp, Zoom, and Webex.

Regarding the Ukraine conflict, we have examined the regulations imposed by the Russian government in recent years, including the implementation of the Russian national DNS. To this end, we have analyzed the zone files of `.ru` and .рф TLDs over a five-year period. We have shown that, even before the announcement of these regulations (*i.e.,* 2019), a large percentage (almost 70%) of domain names were geolocated fully within Russia. However, in February 2022, there was a slight increase in these Russian domains. In contrast, we have also shown that there is a slight decrease in the full Russian TLDs, highlighting their dependence on external infrastructure. In addition, we have also examined the actions taken by Western providers to not accept new Russian customers after the Russian invasion of Ukraine. However, we have found that some companies accepted newly registered domains. In the end, by inspecting the CT logs, we have demonstrated that the American society Let's Encrypt manages a large percentage of `.ru` and .рф domain names.

In summary, this thesis contributes to shed light on activities that significantly impact the Internet, including recent, cyber security threats and global societal events. The findings can be useful for network operators, policymakers and researchers interested in better understanding today's Internet.

Bibliography

[1] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[2] R Vinayakumar, Mamoun Alazab, Sriram Srinivasan, Quoc-Viet Pham, Soman Kotti Padannayil, and Ketha Simran. A visualized botnet detection system based deep learning for the Internet of Things networks of smart cities. *IEEE Transactions on Industry Applications*, 56(4):4436–4456, 2020.

[3] Burair Hameed, Selvakumar Manickam, and Kamal Alieyan. Internet of Things botnet (Mirai): A systematic review. pages 607–616, 07 2019.

[4] Domain Name System (DNS) parameters. https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6. (Accessed on 01/25/2021).

[5] Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, and Gérôme Bovet. Federated learning for malware detection in IoT devices. *Computer Networks*, 204:108693, 2022.

[6] Rajesh Kumar and Rewa Sharma. Leveraging blockchain for ensuring trust in IoT: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8599–8622, 2022.

[7] A review of DDoS attack detection and prevention mechanisms in clouds. In *2022 24th International Multitopic Conference (INMIC)*, pages 1–6. IEEE, 2022.

[8] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, 2022.

[9] M Mazhar Rathore, Anand Paul, Awais Ahmad, Seungmin Rho, Muhammad Imran, and Mohsen Guizani. Hadoop based real-time intrusion detection for high-speed networks. In *2016 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE, 2016.

[10] The latest cyber crime statistics (updated january 2023) | aag it support. https://aag-it.com/the-latest-cyber-crime-statistics/. (Accessed on 01/14/2023).

[11] Julija Tišina, Anastasija Gavriljuk, Venera Petrova, and Nikita Korolev. Authorities Isolate Networks. *Kommersant*, 2022. Accessed: 2022-05.

[12] Russia is restricting social media as war in ukraine continues : Npr. https://www.npr.org/2022/03/07/1085025672/russia-social-media-ban. (Accessed on 01/14/2023).

[13] Jayant Gadge and Anish Anand Patil. Port scan detection. In *2008 16th IEEE international conference on networks*, pages 1–6. IEEE, 2008.

[14] Myung-Sup Kim, Hun-Jeong Kong, Seong-Cheol Hong, Seung-Hwa Chung, and James W Hong. A flow-based method for abnormal network traffic detection. In *2004 IEEE/IFIP network operations and management symposium (IEEE Cat. No. 04CH37507)*, volume 1, pages 599–612. IEEE, 2004.

[15] Qi Zhao, Jun Xu, and Abhishek Kumar. Detection of super sources and destinations in high-speed networks: Algorithms, analysis and evaluation. *IEEE Journal on Selected Areas in Communications*, 24(10):1840–1852, 2006.

[16] Philip K Chan, Matthew V Mahoney, and Muhammad H Arshad. Learning rules and clusters for anomaly detection in network traffic. *Managing Cyber Threats: Issues, Approaches, and Challenges*, pages 81–99, 2005.

[17] Pedro Casas, Francesca Soro, Juan Vanerio, Giuseppe Settanni, and Alessandro D'Alconzo. Network security and anomaly detection with BigDAMA, a Big Data analytics framework. In *2017 IEEE 6th international conference on cloud networking (CloudNet)*, pages 1–7. IEEE, 2017.

[18] Khoh Choon Hwa, Selvakumar Manickam, and Mahmood A Al-Shareeda. Review of Peer-to-Peer botnets and detection mechanisms. *arXiv preprint arXiv:2207.12937*, 2022.

[19] Deciphering Russia's "sovereign Internet law" | DGAP. https://dgap.org/en/research/publications/deciphering-russias-sovereign-internet-law. (Accessed on 01/19/2023).

[20] Mykhailo Fedorov. Letter to Goran Marby, President and CEO of ICANN. https://eump.org/media/2022/Goran-Marby.pdf, 2022. Accessed: 2022-06.

[21] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[22] Glenn Carl, George Kesidis, Richard R Brooks, and Suresh Rai. Denial-of-Service attack-detection techniques. *IEEE Internet computing*, 10(1):82–89, 2006.

[23] K Munivara Prasad, A Rama Mohan Reddy, and K Venugopal Rao. Discriminating DDoS attack traffic from flash crowds on Internet threat monitors (itm) using entropy variations. *African Journal of Computing & ICT*, 6(2):53, 2013.

[24] Marco De Vivo, Eddy Carrasco, Germinal Isern, and Gabriela O De Vivo. A review of port scanning techniques. *ACM SIGCOMM Computer Communication Review*, 29(2):41–48, 1999.

[25] Kris Katterjohn. Port scanning techniques, 2007.

[26] Sérgio SC Silva, Rodrigo MP Silva, Raquel CG Pinto, and Ronaldo M Salles. Botnets: A survey. *Computer Networks*, 57(2):378–403, 2013.

[27] Ibrahim Ghafir, Mohammad Hammoudeh, and Vaclav Prenosil. Botnet command and control traffic detection challenges a correlation based solution. pages 1–5, 12 2016.

[28] Khlood Shinan, Khalid Alsubhi, Ahmed Alzahrani, and Muhammad Usman Ashraf. Machine Learning-based botnet detection in software-defined network: A systematic review. *Symmetry*, 13(5), 2021.

[29] Maryam Feily, Alireza Shahrestani, and Sureswaran Ramadass. A survey of botnet and botnet detection. In *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pages 268–273. IEEE, 2009.

[30] Noor Zuraidin Mohd Safar, Noryusliza Abdullah, Hazalila Kamaludin, Suhaimi Abd Ishak, and Mohd Rizal Mohd Isa. Characterising and detection of botnet in P2P network for UDP protocol. *Indonesian Journal of Electrical Engineering and Computer Science*, 18(3):1584–1595, 2020.

[31] Shahid Anwar, Jasni Mohamad Zain, Mohamad Zolkipli, and Zakira Inayat. A review paper on botnet and botnet detection techniques in cloud computing. 09 2014.

[32] Ping Wang, Sherri Sparks, and Cliff C Zou. An advanced hybrid Peer-to-Peer botnet. *IEEE Transactions on Dependable and Secure Computing*, 7(2):113–127, 2008.

[33] Aditya K Sood, Richard J Enbody, and Rohit Bansal. Dissecting spyeye–understanding the design of third generation botnets. *Computer Networks*, 57(2):436–450, 2013.

[34] Hamad Binsalleeh, Thomas Ormerod, Amine Boukhtouta, Prosenjit Sinha, Amr Youssef, Mourad Debbabi, and Lingyu Wang. On the analysis of the zeus botnet crimeware toolkit. In *2010 eighth international conference on privacy, security and trust*, pages 31–38. IEEE, 2010.

[35] Dafan Dong, Ying Wu, Liang He, Guowei Huang, and Gongyi Wu. Deep analysis of intending Peer-to-Peer botnet. In *2008 Seventh International Conference on Grid and Cooperative Computing*, pages 407–411. IEEE, 2008.

[36] Dae-il Jang, Minsoo Kim, Hyun-chul Jung, and Bong-Nam Noh. Analysis of HTTP2P botnet: case study waledac. In *2009 IEEE 9th Malaysia International Conference on Communications (Micc)*, pages 409–412. IEEE, 2009.

[37] Knud Lasse Lueth. IoT market analysis: Sizing the opportunity. *IoT Analytic Report, March*, 2015.

[38] MMD-0056-2016—Linux/Mirai, how an old ELF malcode is recycled, author=Die, Malware Must, 2016.

[39] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the Mirai botnet. In *26th {USENIX} security symposium ({USENIX} Security 17)*, pages 1093–1110, 2017.

[40] Georgios Kambourakis, Constantinos Kolias, and Angelos Stavrou. The Mirai botnet and the IoT zombie armies. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pages 267–272. IEEE, 2017.

[41] Joel Margolis, Tae Tom Oh, Suyash Jadhav, Young Ho Kim, and Jeong Neyo Kim. An in-depth analysis of the Mirai botnet. In *2017 International Conference on Software Security and Assurance (ICSSA)*, pages 6–12. IEEE, 2017.

[42] Roger Hallman, Josiah Bryan, Geancarlo Palavicini Jr, Joseph Divita, and Jose Romero-Mariona. IoDDoS — The Internet of Distributed Denial of Service attacks: A case study of the Mirai malware and IoT-based botnets. 04 2017.

[43] Initial sequence number selection. https://tools.ietf.org/html/rfc793, 1981. (Accessed on 03/18/2022).

[44] Github - jgamblin/mirai-source-code: Leaked mirai source code for research/ioc development purposes. https://github.com/jgamblin/Mirai-Source-Code. (Accessed on 12/24/2022).

[45] An extended analysis of an IoT malware from a blackhole network. https://www.eunis.org/download/TNC2017/Fullpaper-IoTBlackhole_CW.pdf, 06 2017. (Accessed on 12/24/2022).

[46] Harm Griffioen and Christian Doerr. Examining Mirai's battle over the Internet of Things. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 743–756, New York, NY, USA, 2020. Association for Computing Machinery.

[47] B. krebs. krebsonsecurity hit with record DDoS. https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

[48] Brace yourselves—source code powering potent IoT DDoSes just went public | Ars Technica. https://arstechnica.com/information-technology/2016/10/brace-yourselves-source-code-powering-potent-iot-ddoses-just-went-public/. (Accessed on 05/24/2022).

[49] Major DDoS attack on Dyn disrupts AWS, Twitter, Spotify and more - DCD. https://www.datacenterdynamics.com/en/news/major-ddos-attack-on-dyn-disrupts-aws-twitter-spotify-and-more/. (Accessed on 05/24/2022).

[50] Talktalk and post office customers hit by Mirai worm attack. https://www.wired.co.uk/article/deutsche-telekom-cyber-attack-mirai, 2016.

[51] https://www.radware.com/security/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/. (Accessed on 05/24/2022).

[52] Reaper botnet. https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/reaper-botnet/. (Accessed on 06/01/2022).

[53] Reaper madness | netscout. https://www.netscout.com/blog/asert/reaper-madness. (Accessed on 05/24/2022).

[54] Repear botnet. https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/reaper-botnet/#:~:text=Reaper%20scans%20on%20TCP%20Ports,exploits%20included%20in%20the%20botnet, 2017.

[55] Millions of networks compromised by new reaper botnet. https://www.trendmicro.com/vinfo/pl/security/news/cybercrime-and-digital-threats/millions-of-networks-compromised-by-new-reaper-botnet, 2017.

[56] Botnets never die, Satori refuses to fade away. https://blog.netlab.360.com/botnets-never-die-satori-refuses-to-fade-away-en/, 2018.

[57] Warning: Satori, a Mirai branch is spreading in worm style on port 37215 and 52869. -https://blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/, 2017.

[58] When cameras and routers attack phones. spike in cve-2014-8361 exploits against port 52869. https://isc.sans.edu/forums/diary/When+Cameras+and+Routers+attack+Phones+Spike+in+CVE20148361+Exploits+Against+Port+52869/23942/, 2018.

[59] Security notice - statement on remote code execu- tion vulnerability in huawei hg532 product. https://www.huawei.com/en/psirt/security-notices/huawei-sn-20171130-01-hg532-en, 2018.

[60] Sam Edwards and Ioannis Profetis. Hajime: Analysis of a decentralized Internet worm for IoT devices. *Rapidity Networks*, 16:1–18, 2016.

[61] Critical rce vulnerability found in over a million gpon home routers. https://www.vpnmentor.com/blog/critical-vulnerability-gpon-router/, 2016.

[62] A wicked family of bots. https://www.fortinet.com/blog/threat-research/a-wicked-family-of-bots, 2018.

[63] Gpon exploit in the wild (i) - muhstik botnet among others. https://blog.netlab.360.com/gpon-exploit-in-the-wild-i-muhstik-botnet-among-others-en/, 2018.

[64] The botnet cluster on the 185.244.25.0/24. https://blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/. (Accessed on 07/19/2022).

[65] An update for a very active DDoS botnet: Moobot. https://blog.netlab.360.com/ddos-botnet-moobot-en/, 2020.

[66] The botnet cluster on the 185.244.25.0/24. https://blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/, 2019.

[67] What is a DNS zone? | Cloudflare. https://www.cloudflare.com/en-gb/learning/dns/glossary/dns-zone/. (Accessed on 01/08/2023).

[68] P. Mockapetris. RFC 1035 - Domain names - implementation and specification. 1987.

[69] P. Mockapetris. RFC 1034 - Domain names - concepts and facilities. 11 1987.

[70] DNS zones and zone files explained. http://www.steves-internet-guide.com/dns-zones-explained/#:~:text=A%20zone%20file%20is%20a,to%20the%20other%20DNS%20servers. (Accessed on 01/08/2023).

[71] Yo-Der Song, Aniket Mahanti, and Soorya Charan Ravichandran. Understanding evolution and adoption of Top Level Domains and DNSSEC. In *2019 IEEE International Symposium on Measurements Networking (M N)*. IEEE, 2019.

[72] Tristan Halvorson, Matthew F. Der, Ian Foster, Stefan Savage, Lawrence K. Saul, and Geoffrey M. Voelker. From .Academy to .Zone: An analysis of the new TLD land rush. In *Proceedings of the 2015 Internet Measurement Conference*, 2015.

[73] Maciej Korczynski, Maarten Wullink, Samaneh Tajalizadehkhoob, Giovane CM Moura, Arman Noroozian, Drew Bagley, and Cristian Hesselman. Cybercrime after the sunrise: A statistical analysis of DNS abuse in new gTLDs. In *ASIACCS 2018 - Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*, 2018.

[74] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. Comparing the effects of DNS, DoT, and DoH on web performance. In *Proceedings of The Web Conference 2020*, WWW '20, page 562–572, New York, NY, USA, 2020. Association for Computing Machinery.

[75] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going wild: Large-scale classification of open DNS resolvers. In *Proceedings of the 2015 Internet Measurement Conference*, pages 355–368, 2015.

[76] ICANN. Generic Top-Level Domain (gTLD) Registry Agreements. https://www.icann.org/en/registry-agreements. (Accessed on 2022/03/03).

[77] 2013 Registrar Accreditation Agreement. https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en. (Accessed on 2022/03/03).

[78] ICANN. Life cycle of a typical gTLD domain name. https://www.icann.org/resources/pages/gtld-lifecycle-2012-02-25-en. (Accessed on 2022/03/03).

[79] Domain name life cycle: Life of a typical top-level domain. - connectreseller. https://www.connectreseller.com/blog/domain-name-life-cycle-life-of-a-typical-top-level-domain/. (Accessed on 12/14/2021).

[80] ICANN. AGP (Add Grace Period) Limits Policy. https://www.icann.org/resources/pages/agp-policy-2008-12-17-en. (Accessed on 2022/03/03).

[81] DNSimple. ICANN 60-Day Lock After Change of Registrant. https://support.dnsimple.com/articles/icann-60-day-lock-registrant-change/. (Accessed on 2022/03/03).

[82] ICANN. Registrar abuse reports. https://www.icann.org/resources/pages/abuse-2014-01-29-en. (Accessed on 2022/03/03).

[83] Namecheap. Our fight against fraud is just getting started. https://www.namecheap.com/blog/namecheaps-fight-against-fraud-is-just-getting-started/. (Accessed on 2022/03/03).

[84] Wikipedia. Coreflood. https://en.wikipedia.org/wiki/Coreflood. (Accessed on 2022/03/03).

[85] US District Court for the District of Connecticut. Government's Supplemental Memorandum in Support of Preliminary Injunction. https://www.justice.gov/archive/opa/documents/coreflood-govt-supp.pdf. (Accessed on 2022/03/03).

[86] ICANN. Expedited Registry Security Request Process. https://www.icann.org/resources/pages/ersr-2012-02-25-en. (Accessed on 2022/03/03).

[87] Domain Incite. Registrars to get more domain takedown powers. https://domainincite.com/26917-registrars-to-get-more-domain-takedown-powers. (Accessed on 2022/03/03).

[88] Valerio Persico, Alessio Botta, Pietro Marchetta, Antonio Montieri, and Antonio Pescapé. On the performance of the wide-area networks interconnecting public-cloud datacenters around the globe. *Computer Networks*, 112:67–83, 2017.

[89] Giuseppe Aceto, Alessio Botta, Antonio Pescapé, and Cedric Westphal. Efficient storage and processing of high-volume network monitoring data. *IEEE Transactions on Network and Service Management*, 10(2):162–175, 2013.

[90] Alessio Botta, Walter de Donato, Antonio Pescapé, and Giorgio Ventre. Discovering topologies at router level: Part ii. In *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, pages 2696–2701. IEEE, 2007.

[91] Luigi Gallo, Alessio Botta, and Giorgio Ventre. Identifying threats in a large company's inbox. In *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, pages 1–7. ACM, 2019.

[92] Tanja Zseby, Benoît Claise, Juergen Quittek, and Sebastian Zander. Requirements for IP Flow Information Export (IPFIX). RFC 3917, October 2004.

[93] Cliff C Zou, Don Towsley, and Weibo Gong. On the performance of Internet worm scanning strategies. *Performance Evaluation*, 63(7):700–723, 2006.

[94] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu, Aiko Pras, and Burkhard Stiller. An overview of IP flow-based intrusion detection. *IEEE communications surveys & tutorials*, 12(3):343–356, 2010.

[95] Alberto Dainotti, Pescapé Antonio, and Ventre Giorgio. Nis04-1: Wavelet-based detection of dos attacks. In *IEEE globecom 2006*, pages 1–6. IEEE, 2006.

[96] Soniya Balram and M Wiscy. Detection of TCP SYN scanning using packet counts and neural network. In *2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pages 646–649. IEEE, 2008.

[97] Avinash Sridharan, Tao Ye, and Supratik Bhattacharyya. Connectionless port scan detection on the backbone. In *2006 IEEE International Performance Computing and Communications Conference*, pages 10–pp. IEEE, 2006.

[98] Arno Wagner and Bernhard Plattner. Entropy based worm and anomaly detection in fast IP networks. In *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WET-ICE'05)*, pages 172–177. IEEE, 2005.

[99] Maciej Korczynski, Lucjan Janowski, and Andrzej Duda. An accurate sampling scheme for detecting SYN flooding attacks and portscans. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.

[100] Abdul Ghaffar Shoro and Tariq Rahim Soomro. View of Big Data analysis: Apache Spark perspective. Global Journals Inc. (USA), 2015. (Accessed on 08/2019).

[101] Romain Fontugne, Patrice Abry, Kensuke Fukuda, Darryl Veitch, Kenjiro Cho, Pierre Borgnat, and Herwig Wendt. Scaling in internet traffic: A 14 year and 3 day longitudinal study, with multiscale analyses and random projections. *IEEE/ACM Transactions on Networking*, 25(4):2152–2165, 2017.

[102] Pierre Borgnat, Guillaume Dewaele, Kensuke Fukuda, Patrice Abry, and Kenjiro Cho. Seven years and one day: Sketching the evolution of Internet traffic. In *IEEE INFOCOM 2009*, pages 711–719. IEEE, 2009.

[103] Johan Mazel, Romain Fontugne, and Kensuke Fukuda. A taxonomy of anomalies in backbone network traffic. In *2014 international wireless communications and mobile computing conference (IWCMC)*, pages 30–36. IEEE, 2014.

[104] Romain Fontugne, Pierre Borgnat, Patrice Abry, and Kensuke Fukuda. Mawilab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *Proceedings of the 6th International Conference*, New York, USA, 2010. ACM Conext 2010.

[105] Jurgen A H R Claassen. The gold standard: not a golden standard. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC557893/, August 2018. (Accessed on 01/2019).

[106] Antonia Affinito, Alessio Botta, Luigi Gallo, Mauro Garofalo, and Giorgio Ventre. Spark-based port and net scan detection. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 1172–1179, New York, NY, USA, 2020. Association for Computing Machinery.

[107] Cisco annual internet report (2018 - 2023). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[108] 360netlab. (2018, february) adb.miner: More information[blog post]. https://blog.netlab.360.com/adb-miner-more-information-en/.

[109] CGJ Putman, Lambert JM Nieuwenhuis, et al. Business model of a botnet. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 441–445. IEEE, 2018.

[110] Ya Liu and Hui Wang. Tracking Mirai variants. *Virus Bulletin*, pages 1–18, 2018.

[111] B. krebs. new mirai worm knocks 900k germans offline. https://krebsonsecurity.com/2016/11/new-mirai-worm-knocks-900k-germans-offline.

[112] Yimu Ji, Lu Yao, Shangdong Liu, Haichang Yao, Qing Ye, and Ruchuan Wang. The study on the botnet and its prevention policies in the Internet of Things. In *2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD))*, pages 837–842. IEEE, 2018.

[113] Sadegh Torabi, Elias Bou-Harb, Chadi Assi, Mario Galluscio, Amine Boukhtouta, and Mourad Debbabi. Inferring, characterizing, and investigating Internet-scale malicious IoT device activities: A network telescope perspective. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 562–573. IEEE, 2018.

[114] Shohei Araki, Kneji Takahashi, Bo Hu, Kazunori Kamiya, and Masaki Tanikawa. Subspace clustering for interpretable botnet traffic analysis. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[115] Morteza Safaei Pour, Antonio Mangino, Kurt Friday, Matthias Rathbun, Elias Bou-Harb, Farkhund Iqbal, Sagar Samtani, Jorge Crichigno, and Nasir Ghani. On data-driven curation, learning, and analysis for inferring evolving Internet-of-Things (IoT) botnets in the wild. *Computers & Security*, 91:101707, 2020.

[116] Mert Nakip and Erol Gelenbe. Mirai botnet attack detection with auto-associative dense random neural network. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 01–06. IEEE, 2021.

[117] Antonia Raiane S Araujo Cruz, Rafael L Gomes, and Marcial P Fernandez. An intelligent mechanism to detect cyberattacks of Mirai botnet in IoT networks. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 236–243. IEEE, 2021.

[118] Suzan Almutairi, Saoucene Mahfoudh, Sultan Almutairi, and Jalal S Alowibdi. Hybrid botnet detection based on host and network analysis. *Journal of Computer Networks and Communications*, 2020, 2020.

[119] Basil AsSadhan, José MF Moura, David Lapsley, Christine Jones, and W Timothy Strayer. Detecting botnets using command and control traffic. In *2009 Eighth IEEE International Symposium on Network Computing and Applications*, pages 156–162. IEEE, 2009.

[120] Bbc news | business | criminals 'may overwhelm the web'. http://news.bbc.co.uk/2/hi/business/6298641.stm. (Accessed on 11/12/2022).

[121] White ops | 9 of history's notable botnet attacks. https://www.humansecurity.com/learn/blog/9-of-the-most-notable-botnets#:~:text=The%20first%20botnet%20to%20gain,a%20little%20over%20a%20year. (Accessed on 04/04/2022).

[122] Infodox. Hydra irc bot, the 25 minute overview of the kit. http://insecurety.net/?p=90, 2011. (Accessed on 04/19/2022).

[123] Heads of the hydra. malware for network devices | securelist. https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/. (Accessed on 11/13/2022).

[124] F. Fazzi. Lightaidra – irc-based mass router scanner/exploiter. http://packetstormsecurity.org/files/109244. (Accessed on 04/19/2022).

[125] Artur Marzano, David Alexander, Osvaldo Fonseca, Elverton Fazzion, Cristine Hoepers, Klaus Steding-Jessen, Marcelo HPC Chaves, Ítalo Cunha, Dorgival Guedes, and Wagner Meira. The evolution of Bashlite and Mirai IoT botnets. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00813–00818. IEEE, 2018.

[126] P. Paganini. The Linux Remaiten malware is building a botnet of IoT devices. http://securityaffairs.co/wordpress/45820/iot/linux-remaiten-iot-botnet.html, March 2016. (Accessed on 04/19/2022).

[127] Malware Must Die. MMD-0059-2016 - Linux/IRCTelnet (new Aidra) - A DDoS botnet aims IoT w/ IPv6 ready. https://blog.malwaremustdie.org/2016/10/mmd-0059-2016-linuxirctelnet-new-ddos.html, October 2016. (Accessed on 04/19/2022).

[128] D. Chiu T. Yeh and K. Lu. Persirai: New Internet of Things (IoT) botnet targets IP cameras. blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras, May 2017. (Accessed on 04/19/2022).

[129] Claude Fachkha and Mourad Debbabi. Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization. *IEEE Communications Surveys & Tutorials*, 18(2):1197–1227, 2016.

[130] John Matherly. Complete guide to shodan. *Shodan, LLC (2016-02-25)*, 1, 2015.

[131] Alberto Dainotti, Alistair King, Kimberly Claffy, Ferdinando Papale, and Antonio Pescapé. Analysis of a "/0" stealth scan from a botnet. *IEEE/ACM Transactions on Networking*, 23(2):341–354, 2014.

[132] Claude Fachkha, Elias Bou-Harb, and Mourad Debbabi. On the inference and prediction of DDoS campaigns. *Wireless Communications and Mobile Computing*, 15(6):1066–1078, 2015.

[133] Luca Gioacchini, Luca Vassio, Marco Mellia, Idilio Drago, Zied Ben Houidi, and Dario Rossi. Darkvec: automatic analysis of darknet traffic with word embeddings. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 76–89, 2021.

[134] Olivier Cabana, Amr M Youssef, Mourad Debbabi, Bernard Lebel, Marthe Kassouf, Ribal Atallah, and Basile L Agba. Threat intelligence generation using network telescope data for industrial control systems. *IEEE Transactions on Information Forensics and Security*, 16:3355–3370, 2021.

[135] Zhou Shao, Sha Yuan, and Yongli Wang. Adaptive online learning for IoT botnet detection. *Information Sciences*, 574:84–95, 2021.

[136] Mohammad Alauthman, Nauman Aslam, Mouhammd Al-Kasassbeh, Suleman Khan, Ahmad Al-Qerem, and Kim-Kwang Raymond Choo. An efficient reinforcement learning-based botnet detection approach. *Journal of Network and Computer Applications*, 150:102479, 2020.

[137] Wei Wang, Yaoyao Shang, Yongzhong He, Yidong Li, and Jiqiang Liu. Botmark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. *Information Sciences*, 511:284–296, 2020.

[138] M Ganesh Karthik and MB Mukesh Krishnan. Securing an Internet of Things from Distributed Denial of Service and Mirai botnet attacks using a novel hybrid detection and mitigation mechanism. *Int. J. Intell. Eng. Syst*, 14:113–123, 2021.

[139] Fehmi Jaafar, Darine Ameyed, Amine Barrak, and Mohamed Cheriet. Identification of compromised IoT devices: Combined approach based on energy consumption and network traffic analysis. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*, pages 514–523. IEEE, 2021.

[140] Joao Ceron, Klaus Steding-Jessen, Cristine Hoepers, Lisandro Zambenedetti, and Cintia Margi. Improving IoT Botnet Investigation Using an Adaptive Network Layer. 2019.

[141] Jun Xu, Jinliang Fan, Mostafa Ammar, and Sue B. Moon. On the design and performance of prefix-preserving IP traffic trace anonymization. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, page 263–266, New York, NY, USA, 2001. Association for Computing Machinery.

[142] Agathe Blaise, Mathieu Bouet, Stefano Secci, and Vania Conan. Split-and-merge: detecting unknown botnets. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 153–161. IEEE, 2019.

[143] Heightened DDoS Threat Posed by Mirai and Other Botnets | CISA. https://www.cisa.gov/uscert/ncas/alerts/TA16-288A. (Accessed on 08/19/2022).

[144] Election day: Tracking the Mirai botnet | rapid7 blog. https://www.rapid7.com/blog/post/2016/11/08/election-day-tracking-the-mirai-botnet/. (Accessed on 08/19/2022).

[145] Adb.mirai: New mirai botnet variant spreading via the adb debug port - nsfocus, inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks. https://nsfocusglobal.com/adb-mirai-new-mirai-botnet-variant-spreading-via-the-adb-debug-port/. (Accessed on 01/02/2023).

[146] Dr. John C. Klensin. Role of the Domain Name System (DNS). RFC 3467, March 2003.

[147] Kyle Schomp, Mark Allman, and Michael Rabinovich. DNS resolvers considered harmful. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, page 1–7, New York, NY, USA, 2014. Association for Computing Machinery.

[148] Craig A Shue and Andrew J Kalafut. Resolvers revealed: Characterizing DNS resolvers and their clients. *ACM Transactions on Internet Technology (TOIT)*, 12(4):1–17, 2013.

[149] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. Comparing DNS resolvers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, page 15–21, New York, NY, USA, 2010. Association for Computing Machinery.

[150] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. Where are you taking me? Behavioral analysis of open DNS resolvers. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 493–504. IEEE, 2019.

[151] Yuanchen He, Zhenyu Zhong, Sven Krasser, and Yuchun Tang. Mining DNS for malicious domain registrations. In *6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*, pages 1–6. IEEE, 2010.

[152] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. A survey on malicious domains detection through DNS data analysis. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

[153] Alessio Botta, Gennaro Esposito Mocerino, Stefano Cilio, and Giorgio Ventre. A Machine Learning approach for dynamic selection of available bandwidth measurement tools. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE, 2021.

[154] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig. An empirical study of the cost of DNS-over-HTTPS. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 15–21, NY, USA, 2019. Association for Computing Machinery.

[155] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. An End-to-End, large-scale measurement of DNS-over-Encryption: How far have we come? In *Proceedings of the Internet Measurement Conference*, IMC '19, 2019.

[156] Ranking the performance of public DNS providers. https://blog.thousandeyes.com/ranking-performance-public-dns-providers-2018/. (Accessed on 02/17/2021).

[157] Why You Shouldn't Use Your ISP's Default DNS Server. https://www.howtogeek.com/664608/why-you-shouldnt-be-using-your-isps-default-dns-server/. (Accessed on 02/17/2021).

[158] David Dagon, Chris Lee, Wenke Lee, and Niels Provos. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In *Proc. 15th Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2008.

[159] DNS Performance - Compare the speed and uptime of enterprise and commercial DNS services | DNSPerf. https://www.dnsperf.com/. (Accessed on 02/17/2021).

[160] Austin Hounsel, Paul Schmitt, Kevin Borgolte, and Nick Feamster. Can encrypted DNS be fast? In *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29–April 1, 2021, Proceedings 22*, pages 444–459. Springer, 2021.

[161] GitHub - shuque/pydig: pydig: a DNS query tool written in Python. https://github.com/shuque/pydig. (Accessed on 07/28/2020).

[162] Antonia Affinito, Alessio Botta, and Giorgio Ventre. The impact of covid on network utilization: an analysis on domain popularity. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2020.

[163] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D Strowes, and Narseo Vallina-Rodriguez. A long way to the top: Significance, structure, and stability of Internet top lists. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, page 478–493, New York, NY, USA, 2018. Association for Computing Machinery.

[164] Matthew R McNiece, Ruidan Li, and Bradley Reaves. Characterizing the security of endogenous and exogenous desktop application network flows. In *Passive and Active Measurement: 22nd International Conference, PAM 2021, Virtual Event, March 29–April 1, 2021, Proceedings 22*, pages 531–546. Springer, 2021.

[165] DNS-over-HTTPS (DoH) | Public DNS | Google Developers. https://developers.google.com/speed/public-dns/docs/doh. (Accessed on 01/21/2021).

[166] Using DNS over HTTPS (DoH) with OpenDNS – OpenDNS. https://support.opendns.com/hc/en-us/articles/360038086532-Using-DNS-over-HTTPS-DoH-with-OpenDNS. (Accessed on 01/21/2021).

[167] A. Kumar. RFC 1536 - Common DNS implementation errors and suggested fixes. 1993.

[168] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. Where are you taking me? behavioral analysis of open DNS resolvers. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 493–504. IEEE, 2019.

[169] A simple guide to building a confusion matrix. https://blogs.oracle.com/ai-and-datascience/post/a-simple-guide-to-building-a-confusion-matrix#:~:text=The%20confusion%20matrix%20is%20represented,normality%20or%20a%20normal%20behavior. (Accessed on 02/11/2022).

[170] Operation poisoned hurricane | fireeye inc. https://www.fireeye.com/blog/threat-research/2014/08/operation-poisoned-hurricane.html. (Accessed on 01/28/2021).

[171] Understanding DNS Sinkholes - A weapon against malware - Infosec resources. https://resources.infosecinstitute.com/topic/dns-sinkhole/. (Accessed on 02/11/2021).

[172] Googleusercontent.com can trip you up, if you disable third-party cookies | get more done, with kerika. https://blog.kerika.com/googleusercontent-com-can-trip-you-up-if-you-disable-third-party-cookies/. (Accessed on 02/12/2021).

[173] Hiding malware inside images on googleusercontent. https://blog.sucuri.net/2018/07/hiding-malware-inside-images-on-googleusercontent.html.

[174] CAIDA. DZDB. https://catalog.caida.org/details/software/dzdb_api. Accessed: 2022/03/03.

[175] Xiaoqing Sun, Mingkai Tong, Jiahai Yang, Liu Xinran, and Liu Heng. HinDom: A robust malicious domain detection system based on heterogeneous information network with transductive classification. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, 2019.

[176] Yong Shi, Gong Chen, and Juntao Li. Malicious domain name detection based on extreme Machine Learning. *Neural Processing Letters*, 48:1347–1357, 2018.

[177] Nitay Hason, Amit Dvir, and Chen Hajaj. Robust malicious domain detection. In *Cyber Security Cryptography and Machine Learning*, pages 45–61, 2020.

[178] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. Exposure: A passive DNS analysis service to detect and report malicious domains. page 28. ACM Trans. Inf. Syst. Secur., 2014.

[179] Vinayakumar Ravi, Soman Kp, and Prabaharan Poornachandran. Detecting malicious domain names using deep learning approaches at scale. *Journal of Intelligent and Fuzzy Systems*, pages 1355–1367, 2018.

[180] Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. Understanding the domain registration behavior of spammers. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, 2013.

[181] Spamhaus. Weaponizing Domain Names: how bulk registration aids global spam campaigns. https://www.spamhaus.org/news/article/795/weaponizing-domain-names-how-bulk-registration-aids-global-spam-campaigns. (Accessed on 2022/03/03).

[182] Yury Zhauniarovich, Issa Khalil, Ting Yu, and Marc Dacier. A survey on malicious domains detection through DNS data analysis. *ACM Comput. Surv.*, page 36, 2018.

[183] Greg Aaron, Lyman Chapin, David Piscitello, and Dr. Colin Strutt. Phishing landscape 2021 an annual study of the scope and distribution of phishing. *Interisle Consulting Group, LLC*, 2021.

[184] Thomas Vissers, Jan Spooren, Pieter Agten, Dirk Jumpertz, Peter Janssen, Marc Van Wesemael, Frank Piessens, Wouter Joosen, and Lieven Desmet. Exploring the ecosystem of malicious domain registrations in the .eu TLD. In *Research in Attacks, Intrusions, and Defenses*, pages 472–493, 2017.

[185] Pawel Foremski and Paul Vixie. The modality of mortality in domain names. *Virus*, page 1, 2018.

[186] Timothy Barron, Najmeh Miramirkhani, and Nick Nikiforakis. Now you see it, now you Don't: A large-scale analysis of early domain deletions. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, 2019.

[187] Tobias Lauinger, Kaan Onarlioglu, Abdelberi Chaabane, William Robertson, and Engin Kirda. Whois lost in translation: (mis)understanding domain name expiration and re-registration. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, page 247–253, New York, NY, USA, 2016. Association for Computing Machinery.

[188] An 18 month summary of ICANN's DNSTICR project. https://www.icann.org/en/blogs/details/an-18-month-summary-of-icanns-dnsticr-project-2-9-2021-en. (Accessed on 05/12/2022).

[189] Cisco. Umbrella Investigate API: Domain Status, Risk Score. https://developer.cisco.com/docs/cloud-security/#!investigate-getting-started/getting-started. (Accessed on 2022/03/03).

[190] Cisco Umbrella Investigate. https://umbrella.cisco.com/products/umbrella-investigate. (Accessed on 2022/03/03).

[191] ZookNIC. Domain name counts. http://www.zooknic.com/Domains/counts.html. (Accessed on 2022/03/03).

[192] Etienne Roser. On the podium of the new gTLDs: .top, .xyz and .club. https://www.internetx.com/en/news-detailview/on-the-podium-of-the-new-gtlds-top-xyz-and-club-1/. (Accessed on 2022/03/03).

[193] Jonathan M. Spring. Modeling malicious domain name take-down dynamics: Why ecrime pays. In *2013 APWG eCrime Researchers Summit*, pages 1–9. IEEE, 2013.

[194] Spamhaus. The most abused top-level domains in 2018. https://www.spamhaus.com/resource-center/the-most-abused-top-level-domains-in-2018/. (Accessed on 2022/03/03).

[195] The last four years in retrospect: A brief review of DNS abuse trends. https://www.icann.org/en/system/files/files/last-four-years-retrospect-brief-review-dns-abuse-trends-22mar22-en.pdf. (Accessed on 05/20/2022).

[196] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, page 6, 2010.

[197] Video conferencing apps surge from coronavirus impact. https://www.appannie.com/en/insights/market-data/video-conferencing-apps-surge-coronavirus.

[198] The virus changed the way we Internet - The New York Times. https://www.nytimes.com/interactive/2020/04/07/technology/coronavirus-internet-use.html. (Accessed on 06/10/2020).

[199] Coronavirus: Most downloaded apps google play store during confinement france | statista. https://www.statista.com/statistics/1106847/most-downloaded-apps-google-play-store-covid-19-france/. (Accessed on 06/10/2020).

[200] M. Trevisan, D. Giordano, I. Drago, M. Munafò, and M. Mellia. Five years at the edge: Watching internet from the isp network. *IEEE/ACM Transactions on Networking*, 28(2):561–574, 2020.

[201] J. L. Garcia-Dorado, A. Finamore, M. Mellia, M. Meo, and M. Munafo. Characterization of isp traffic: Trends, user habits, and access technology impact. *IEEE Transactions on Network and Service Management*, 9(2):142–155, 2012.

[202] Thomas Favale, Francesca Soro, Martino Trevisan, Idilio Drago, and Marco Mellia. Campus traffic and e-learning during covid-19 pandemic. *Computer Networks*, 176:107290, 2020.

[203] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez. A long way to the top: Significance, structure, and stability of internet top lists. *Proceedings of the Internet Measurement Conference 2018*, October.

[204] Cisco umbrella 1 million - cisco umbrella. https://umbrella.cisco.com/blog/cisco-umbrella-1-million. (Accessed on 06/08/2020).

[205] Aws | alexa top sites - elenchi aggiornati dei principali siti sul web. https://aws.amazon.com/it/alexa-top-sites/. (Accessed on 06/08/2020).

[206] Top 6 myths about the alexa traffic rank - alexa blog. https://blog.alexa.com/top-6-myths-about-the-alexa-traffic-rank/. (Accessed on 06/12/2020).

[207] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.

[208] How are alexa's traffic rankings determined? – alexa support. https://support.alexa.com/hc/en-us/articles/200449744. (Accessed on 06/08/2020).

[209] Domain whitelist benchmark: Alexa vs umbrella - netresec blog. https://www.netresec.com/?page=Blog&month=2017-04&post=Domain-Whitelist-Benchmark%3A-Alexa-vs-Umbrella. (Accessed on 06/12/2020).

[210] Coronavirus: Tiktok reports record growth as mobile use soars under lockdown | al arabiya english. https://english.alarabiya.net/en/coronavirus/2020/05/07/Coronavirus-TikTok-records-record-growth-as-mobile-use-soars-under-lockdown. (Accessed on 07/08/2020).

[211] Alessio Botta, Donato Emma, Antonio Pescapé, and Giorgio Ventre. Systematic performance modeling and characterization of heterogeneous IP networks. *11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, 72(7):1134–1143, November 2006.

[212] Yale School of Management. Almost 1,000 Companies Have Curtailed Operations in Russia — But Some Remain. https://som.yale.edu/story/2022/almost-1000-companies-have-curtailed-operations-russia-some-remain, 2022. Accessed: 2022-05.

[213] US Department of Treasury. Specially designated nationals and blocked persons list (sdn) human readable lists. https://home.treasury.gov/policy-issues/financial-sanctions/specially-designated-nationals-and-blocked-persons-list-sdn-human-readable-lists, apr 2022. Accessed April 14th.

[214] Ron Miller. Amazon, Microsoft and Google Have Suspended Cloud Sales in Russia. *TechCrunch*, 2022. Accessed: 2022-05.

[215] GoDaddy. How GoDaddy is Supporting Ukrainian Customers. https://aboutus.godaddy.net/newsroom/company-news/news-details/2022/How-GoDaddy-is-Supporting-Ukrainian-Customers/default.aspx, mar 2022. Accessed: 2022-05.

[216] Reuters. U.S. firm Cogent cutting internet service to Russia. https://www.reuters.com/technology/us-firm-cogent-cutting-internet-service-russia-2022-03-04/, mar 2022. Accessed: 2022-05.

[217] Alena Epifanova and Philipp Dietrich. Russia's Quest for Digital Sovereignty: Ambitions, Realities and Its Place in the World. *German Council on Foreign Relations*, (1), 2022.

[218] RU-Center. Recommended SSL certificates. https://www.nic.ru/en/catalog/ssl/recommended-ssl, may 2022. Accessed May 14th.

[219] Barney Warf. Geographies of global internet censorship. *GeoJournal*, 76(1):1–23, 2011.

[220] James Griffiths. *The great firewall of China: How to build and control an alternative version of the internet.* Bloomsbury Publishing, 2021.

[221] Lukas Kawerau, Nils B. Weidmann, and Alberto Dainotti. Attack or block? repertoires of digital censorship in autocracies. *Journal of Information Technology & Politics*, 0(0):1–14, 2022.

[222] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C. Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. Analysis of Country-Wide Internet Outages Caused by Censorship. *IEEE/ACM Transactions on Networking*, 22(6):1964–1977, dec 2014.

[223] E. Moyakine and A. Tabachnik. Struggling to strike the right balance between interests at stake: The 'yarovaya', 'fake news' and 'disrespect' laws as examples of ill-conceived legislation in the age of modern technology. *Computer Law & Security Review*, 40:105512, 2021.

[224] Luciano Zembruzki, Raffaele Sommese, Lisandro Zambenedetti Granville, Arthur Selle Jacobs, Mattijs Jonker, and Giovane C. M. Moura. Hosting industry centralization and consolidation. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9. IEEE, 2022.

[225] Enze Liu, Gautam Akiwate, Mattijs Jonker, Ariana Mirian, Stefan Savage, and Geoffrey M. Voelker. Who's got your mail? characterizing mail service provider usage. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, pages 122–136, New York, NY, USA, 2021. Association for Computing Machinery.

[226] Reethika Ramesh, Ram Sundara Raman, Matthew Bernhard, Victor Ongkowijaya, Leonid Evdokimov, Anne Edmundson, Steven Sprecher, Muhammad Ikram, and Roya Ensafi. Decentralized control: A case study of Russia. In *Network and Distributed System Security*. The Internet Society, 2020.

[227] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE journal on selected areas in communications (JSAC)*, 34(6):1877–1888, June 2016.

[228] IP2Location. IP2Location IP Address Geolocation Database. https://www.ip2location.com/database/ip2location/, n.d. Accessed: 2022-05.

[229] Censys. Censys Bulk Data Access. https://censys.io/data, May 2022.

[230] UK Government. The UK Sanctions List. https://www.gov.uk/government/publications/the-uk-sanctions-list, July 2020. Accessed 2022-04-14.

[231] Department Of The Treasury. GENERAL LICENSE NO. 25 Authorizing Transactions Related to Telecommunications and Certain Internet-Based Communications. https://home.treasury.gov/system/files/126/russia_gl25.pdf, apr 2022.

[232] Amazon. Updates to Amazon's retail, entertainment, and AWS businesses in Russia and Belarus. *aboutamazon*, March 2022. Accessed: 2022-05.

[233] Kevin Murphy. Now Sedo Pulls the Plug on Russians. *Domain Incite*, 2022. Accessed: 2022-05.

[234] Matthew Prince. Steps We've Taken Around Cloudflare's Services in Ukraine, Belarus and Russia. https://blog.cloudflare.com/steps-taken-around-cloudflares-services-in-ukraine-belarus-and-russia/, 2022. Accessed: 2022-05.

[235] Rebecca Klar. Google Cloud to stop accepting new customers in Russia. *msn*, 2022. Accessed: 2022-05.

Author's Publications

- Antonia Affinito, Alessio Botta, Luigi Gallo, Mauro Garofalo, and Giorgio Ventre; "Spark-based port and net scan detection". Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 1172-1179, Association for Computing Machinery (ACM).
- Antonia Affinito, Alessio Botta, and Giorgio Ventre. "The impact of covid on network utilization: an analysis on domain popularity". IEEE International Workshop on Computer-Aided Modeling, Analysis, and Design of Communication Links and Networks, CAMAD, 2020, pp. 1-6, IEEE.
- Antonia Affinito, Alessio Botta, and Giorgio Ventre. "Local and Public DNS Resolvers: do you trade off performance against security?". IFIP Networking Conference, 2022, pp. 1-9, IEEE.
- Antonia Affinito, Raffaele Sommese, Gautam Awikate, Stefan Savage, KC Claffy, Geoffrey M. Voelker, Alessio Botta, and Mattijs Jonker. "Domain Name Lifetimes: Baseline and Threats". Proceedings on the 6th edition of the Network Traffic Measurement and Analysis (TMA) Conference, 2022, IFIP.
- Mattijs Jonker, Gautam Akiwate, Antonia Affinito, KC Claffy, Alessio Botta, Geoffrey M. Voelker, Roland van Rijswijk-Deij, and Stefan Savage. "Where .ru? Assessing the Impact of Conflict on Russian Domain Infrastructure". Proceedings of the 22nd ACM Internet Measurement Conference, 2022, pp. 159-165, Association for Computing Machinery (ACM).