Delft University of Technology

Behind the Botnet

Evaluating Avalanche's security controls using a reconstruction of its anatomy from forensic evidence

F.E.G. Miedema

Behind the Botnet

Evaluating Avalanche's security controls using a reconstruction of its anatomy from forensic evidence

by

F.E.G. Miedema

to obtain the degree of Master of Science
Data Science & Technology Track
with a 4TU specialization in Cyber Security

at the Delft University of Technology,
to be defended publicly on Thursday December 22, 2022 at 10:00 AM.

Student number:	4141369	
Thesis committee:	Prof. dr. G. Smaragdakis,	TU Delft, chair
	Dr. R. S. van Wegberg,	TU Delft, first supervisor
	Prof. dr. M. J. G. van Eeten,	TU Delft, second supervisor
	Ir. H. C. Maan,	External, advisor

Cover: Image created with the assistance of DALL·E 2.

Preface

While the proverb *"It is not about the destination, but the journey"* might be so corny because it is true, I can say that I am actually quite relieved to have arrived at this destination (and would not trade it back for the journey). I had envisioned this project quite differently when I started; a different dataset, a different research question, a different project planning, a different deadline, and a different me. I am thankful for the countless growing opportunities this project has brought me. I have learned so much about so many different things; botnet things, forensic things, other nerdy things, but mostly things about myself. I desire the reminiscence of this journey to fuel my future eagerness and confidence when undertaking such a challenging project.

There is one person that I want to thank first and foremost, because I am sure that both me and this project would not have made it to where we are now without him. Rolf, thank you for your unconditional support, your seemingly never-ending positivity and the role model you are for me and other aspiring researchers. Michel and George, thank you for your trust in my capabilities and for the invaluable feedback on my ideas (that often came in all shape and forms). Ir. Maan, you made me feel welcome and at home in our team and you took me under your wing to teach me all sorts of (in your eyes small) bits of knowledge and skills that I needed for this project. Thank you for that!

Then, of course, the larger groups of people that have allowed me to do this research. My two teams of colleagues, who have answered questions, made sure I also had some fun (when I was stressing to make the first, then second and now third deadline) and motivated me to keep going even when I had given up hope. Finally, I want to thank all my lovely friends and family who have supported me throughout this process. A very special thanks to Beryl, for the coffee breaks, the mental support messages and our weekend-working-sessions, and to Lilian, for your role as ’klankbord’ (or ’klaagbord’) and my personal English dictionary.

And, to remind myself, a quote of another wise woman:

"Ik was er al" - M. Baldé

F.E.G. Miedema
Delft, December 2022

Abstract

How did Avalanche, a botnet with an active lifetime of 8 years while serving 20+ malware families, ensure a smooth operation of business? Avalanche had the attention of security researchers and law enforcement, yet it managed to persevere for a long period of time.

In this work, we answer this question by analyzing Avalanche's security controls and its business model based on longitudinal ground truth data from its criminal investigation by German law enforcement. We first analyzed previous botnet research and identified five research challenges: (1) the botnet phenomenon keeps evolving, so continuous research is required, (2) there is not yet a framework to categorize or interpret botnet evasion techniques, (3) botnet research is challenging due to the lack of large real-world datasets, (4) botnet takedowns are challenging and costly, so other avenues for intervening in botnets should be explored, and (5) more research is being done into botnet economics, but it is mostly based on case studies methodologies without access to ground truth data.

We defined the adversarial context of botnets and showed how their responses – evasion techniques – can be interpreted as security controls according to deviant security theory. We created a framework for categorizing these security controls, based on security control types and the type of threat. Turning to our data, we performed an exploratory analysis in which we processed, validated and interpreted the available data based on their different types: server images, network data and databases. Based on the insights from this analysis, we applied the business model canvas and described Avalanche's business model. We describe how Avalanche provides it customers with proxying and domain registration services, generating on average $7,500 of revenue per month from 59 customers. We identified seven security controls, three technical controls and four administrative controls, that were applied to evade detection, to increase resilience against takedowns and to conceal the ownership by the botnet operators.

Our findings show that Avalanche configured itself to adequately respond to the threats in its adversarial context. Its business model – through using different key partners and many replaceable resources – and its application of security controls – such as backups, bot monitoring and proxy architecture – created redundancy in Avalanche's operation, allowing it to detect and resolve threats quickly.

Contents

1
Introduction

How do cybercriminals respond to the adversarial environment they operate in? Law enforcement (LE) operations targeting the infrastructure, actors and facilitators of cyber crime have shifted the current paradigm of cybersecurity. The classical notion of cybersecurity, derived from nation state security and warfare, states that society (the 'defenders') needs to protect itself from threats originating from enemies (the 'attackers'). This axiom has transferred into the mainstream interpretation of cybersecurity: we need to defend ourselves from the criminals attacking our systems, software and data. However, by preventing, disrupting and investigating cybercrime, it is the security of cybercriminals that is now actively targeted [19, 59, 61]. As a result of this, "doing business" as a cybercriminal currently also entails the growing need to ensure operational security.

The concept of "cybercriminal's cybersecurity practices" has been coined by Van De Sandt [60] as "deviant security". To study deviant security, the security practices or security controls of cyber criminals can be studied, in light of their role in the trade-off between security and efficiency. Security controls are defined as "the full range of administrative, physical and technical countermeasures of a preventive, deterrent, detective, corrective, recovery and compensating nature" [60]. By creating an overview of these controls, it provides an overview of how the crime and the criminal is protected.

In general, most research into the security controls of cyber criminals focuses on the technical functionality of the controls and not so much the utilization of the controls in the context of the cybercriminal business model. For example, the anti-forensic capabilities of malware have been studied extensively [20, 30, 36], uncovering different encryption, stenography, wiping and anti-reverse engineering techniques used to obfuscate the working or author of malware. However, why certain techniques are used more often than others, or whether these techniques secure the criminal and/or the crime or can be attributed to the unique working of the malware, is not yet clear. Similarly, evasion techniques of botnets have been studied in different empirical studies, describing techniques such as protocol manipulation (e.g. protocol tunneling) [63] and domain fluxing [31]. These evasion techniques are reported as either a characteristic of the botnet (e.g. DNS-tunneling is seen as a C&C communication pattern), or as a behavioral feature (e.g. domain fluxing is a feature of a fast-fluxing botnet), but not studied to explain the usage of these techniques by botmaster(s).

A contrasting example is the work of Van de Laarschot and Van Wegberg, who studied the different security practices of vendors on the online anonymous market "Hansa", focussing both on which controls were applied and how the use of these controls could be explained from the perspective of different vendors. They found significant differences between vendors selling digital cybercrime items versus vendors selling physical items (e.g. drugs), which could point to a difference in perceived risk and thus a difference in the application of security controls between vendors [59].

For cybercriminals who operate infrastructure that provides a service to other (cyber-)criminals, the consideration for using certain security controls, if any, involves the trade-off between security and efficiency for both its own operation, as for its customers. This is similar to the paradigm in 'standard' cyber security, where security operators need to weigh the impact of applying certain security controls on the usability and costs of the product or service used. In the case of ransomware-as-a-service, the infrastructure might hold private data of both administrators and customers, such as account information, financial information and IP-logging that it would want to protect. At the same time, it wants to make sure that the services it provides to customers – such as their databases with data on compromised targets or back-end servers – are available, reliable and cheap, which means that there is a limit to the amount, type or invasiveness of the controls applied. Switching around the location of the hosted servers means downtime, using more elaborate encryption schemes means better hardware and more complex code is needed and thus higher fixed costs, et cetera. The "you cannot have a secure cake and use it easily and cheaply too" also seems to apply for cybercriminals.

By studying the security controls of these service providers, we do not only understand better how they try to protect themselves from LE, security researchers or other criminals, but also how they operate their business. First, studying security controls can allow us to understand better in which way outside measurements or probes are being evaded or denied by these infrastructure facilitators. This requires empirical research, as this will help to guide future methodology approaches and to understand historical measurement errors. Second, understanding and penetrating the security controls of cybercriminals is the goal of many LE operations, as it is a conditional step before achieving the final goal of operator attribution. Third, studying security controls provides new insights in the economical considerations of these infrastructure operators. Business models, value chains and modus operandi have been studied extensively, but without taking into account the perspective of facilitators ensuring security for both themselves, the infrastructure and their customers.

The main challenge in studying the security controls of cybercriminals, is that it requires a look behind the scenes of these enterprises. Historically, this has only been possible through three approaches: (1) when researchers were able to penetrate (parts of) the backend, (2) when data was leaked by/from the backend, and (3) through law enforcement stings. A notable example of the first approach is the Torpig botnet takeover by Stone-Gross et al. [55]. A recent example of a leak is the Conti-leaks that came from a disgruntled insider that posted the logs of their internal communication channels. Data from police operations has been sparse, but used in several instances. Examples are the analyses of MaxiDed [44] and Hansa Market [21], both marketplace platforms. Although backend access provides researchers with the data needed to study security controls, creating insights is challenging because of the unstructured data sources, which are not gathered through a scientific method but rather created by adversaries in their operations. Moreover, methods of reconstructing and interpreting these backends often heavily depend on manual analyses and have gained little attention in academic research because of their limited use.

To study the security controls of a botnet facilitator, we were granted access to data of the German Law Enforcement operation targeting 'Avalanche'. Avalanche was a botnet that served both as a way to distribute malware and recruit and exploit money mules [24]. First recognized in 2008, it was dismantled after a bit more than eight years in December 2016 by a collaboration of the German Police, FBI, Europol and other global partners. In its lifespan, the Avalanche network was used by more than two hundred cybercriminals [26] and for more than twenty different malware campaigns such as Citadel, Goznym, and Tinba [48]. Avalanche is a compelling case to study for multiple reasons. From the point of studying the controls of infrastructure operators that provide services to others, Avalanche has been used by multiple different groups to spread the aforementioned 20+ campaigns for multiple years. This indicates that Avalanche provided services that had a sustainable demand and was successful in providing their services to customers for a long period of time, meaning that it had both a functioning business model and sufficient security controls to sustain the business during that time. Additionally, data from different hosts was gathered at different points during the investigation, meaning the security controls can be studied from more than a single snapshot of the platform.

This thesis presents the first empirical study of the security controls of service-providing botnet operators, Avalanche, based on longitudinal ground truth data. We answer the question "*How did Avalanche ensure business continuity, given its adversarial context?*". The adversarial context from which risks emerge, is shaped by security practitioners, researchers and law enforcement. From publishing IP-blacklists, creating new detection strategies to criminal investigations and takedowns: there were multiple threats that loomed on the horizon of Avalanche. In order to achieve this continuity, they create ways of dealing with potential threats by taking preventive or mitigative actions: security controls. Because security controls are assets that protect the assets needed to operate the business model, we first answer the question "What was Avalanche's business model?", before analyzing their business operations and infrastructure to answer the question "Which security controls did Avalanche apply". By combining our understanding of Avalanche's business model and their security controls, we can answer our main research question of business continuity.

The contributions that this research seeks to make, are:

- We create a framework to combine current insights on botnet evasion techniques with deviant security concepts (chapter 3);
- We perform an exploratory analysis of the ground truth data of Avalanche (chapter 5);
- We describe Avalanche's business model according to the Business Model Canvas (chapter 6);
- We describe security controls and categorize them according to our framework (chapter 7).

2 Background

Botnets and the resulting botnet attacks have been around for more than twenty years and have been studied extensively. For a detailed definition and description of botnets, we point the reader in the direction of section 2 of the paper of Silva et al. [51] or the introduction of the paper of Thanh Vu et al. [57]. We utilize these two excellent survey studies and the taxonomy paper of Khattak et al. [37] to summarize botnet research from survey studies in section 2.1. These survey studies show that botnet research is mostly focussed on the detection of botnets and, as a result, also focus little on the importance of empirical measurement studies to understand the working of botnets better themselves. To address this, we analyze eight large-scale empirical measurement studies into botnets and summarize the studied features studied and used data sources in section 2.2. We combine these insights with the extensive survey into botnet economics of Georgoulias et al. [29] in section 2.3, to provide a cohesive overview of the current botnet research challenges. Based on these challenges, we describe the research gap we aim to address with this work. Because our analysis is focussed on one specific botnet, Avalanche, we introduce Avalanche in section 2.4 and describe its history, investigation and takedown.

2.1. Botnet survey studies

Silva et al., 2013: In their survey paper from 2013, Silva et al. use 205 botnet research papers until 2011. They provide a comprehensive description of botnets in general, by describing the main components of a botnet, desirable characteristics of a bot, the life cycle of botnets and architectural designs of botnets. Because most of the previous academic literature on botnets had focused on detection techniques, they summarize main findings and structure the techniques into honeypot-based and IDS-based. The latter approach is further divided in signature-based and anomaly based, which in turn can be host-based or network-based through active or passive monitoring. Botnet defense techniques are mainly focussed on preventing bot propagation or hindering bot communication and are either prevent, treat of contain infections. Finally they describe new trends and/or platforms, such as exploiting social network websites for bot communication and/or command & control, and mini-botnets, which are smaller in size and focus on stealth and discretion.

Khattak et al., 2014: Khattak et al. present taxonomies of botnet behaviors, detection mechanisms and defense strategies, derived from their systematic analysis of 162 references to academic and industry papers, as well as news articles. Their taxonomy of botnet behaviors aims to provide a set of features to categorize and describe each botnet. The six main features contain phases from the life-cycle of botnets, as well as components of the botnet, which are propagation, rallying, command & control, purpose, topology and evasion. Botnet detection mechanisms are divided into mechanisms focussed on bot detection, command & control detection or botmaster detection. Each mechanism can also be described as active or passive, where active mechanisms actively participate in the botnet by infiltrating, injecting or marking parts of the botnet, while passive mechanisms detect through observations that are syntactic (e.g. signatures) or semantic (e.g. protocol information or event-related data). While they choose the level of activity or involvement of the detection mechanism (active versus passive) as the main discerning dimension, Khattak et al. also present seven other dimensions by which detection mechanisms can be categorized, such as degree of automation (manual, semi-automated or automated), mode of operation (live or offline), and location of deployment (host or network). More importantly, they describe the interplay between botnet features and detection mechanisms, detailing how the absence or presence of certain features improves or degrades the effectiveness of specific detection mechanisms. The third taxonomy is that of botnet defenses, which is categorized in preventive and remedial mechanisms. Preventive measures are either technical or non-technical, while remedial measures are offensive or defensive.

Thanh Vu et al., 2021: The survey paper of Thanh Vu et al. starts with a summary of the main contributions and reference metrics of 23 literature reviews on botnets, showing the need for an updated literature overview to answer questions relating to botnet incentives, botnet evolution, proposed mitigation strategies and current trends & challenges. Using a search strategy following PICO-criteria[1], they include 224 peer-reviewed English references from 2005 onward related to botnets. Starting with incentives for botnet development, they distinguish benevolent from malevolent botnets. For malevolent incentives, the designated targets and reasons for attack are summarized. Within the evolution of botnets, the authors highlight changes in disguises and subterfuge as well as the intricacies related to botnets with a P2P architecture. Featuring new and established botnet types, extension and browser based, smartphone-based, vehicular, social-network, blockchain-based and IoT-based botnets are described. Botnet detection approaches are divided into neural network detection mechanisms, machine learning and network-based detection mechanisms, and Domain Name System (DNS) based mechanisms. Additionally, mechanisms specifically related to IoT and P2P botnets, mobile botnets, and social network botnets are illustrated. For each of the techniques, the advantages, disadvantage and detection rate (if it was available) is reported. Seven types of mitigation mechanisms, both reactive and proactive are described, which are best practices for end-users and organizations, network-level blocking and packet analysis, honeypots and botnet isolation, attacking P2P botnets, IoT-specific mitigation strategies, community-driven approaches and botnet mitigation with ethical issues (such as spreading anti-botnets or attacking suspected hosts).

The survey studies of Silva et al., Khattak et al. and Thanh Vu et al. provide an overview of topics and findings of botnet research. In general, research into botnets is about specific types of botnets, botnet evolution, attack types, and detection and mitigation techniques. A main insight, that is also mentioned by Thanh Vu et al., is that most botnet literature (as well as other surveys) focusses on botnet detection while a much smaller portion of papers researches mitigation (although the two topics are sometimes also combined) [57]. From the 224 references they found with their survey method, 127 (so 57%) papers describe botnet detection mechanisms.

2.2. Empirical botnet studies

Studying botnets is studying a moving target: botnets evolve and develop themselves based on new technological adoptions (such as IoT- and cloud-based botnets) as well as in response to advancing detection techniques. When you combine this insight with the finding from Khattak et al. that there is an interplay between the presence/absence of certain features of a botnet with the effectiveness of detection mechanisms, it becomes clear that continuous research is needed to better understand the functioning of botnets. However, most survey studies have left one crucial branch of botnet research out of the main picture: empirical botnet (measurement) studies. Empirical measurement studies into Conficker [50], Mirai [7] or Bashlite [41] contribute to a deeper understanding of these botnets, but have not been included in any of the aforementioned survey studies. Besides empirically measuring botnet features, these studies often provide a broader view on a botnet, by researching botnet victims or botnet hosting. Empirical botnet studies are in a way the predecessor of the detection and mitigation studies: they lay the ground work for a deep-dive on specific solutions. This is similar to what Bailey et al. did in their survey paper [10]: they split the existing work into (1) papers focussed on detection techniques and (2) botnet measurement studies. For this reason, we provide an overview of eight empirical measurement studies into botnets.

One of the first large botnets studied extensively was the Storm Worm [34] in 2008, which at its peak was responsible for 8% of all infections worldwide [23]. Since then, many botnets have been the focal point of empirical botnet studies. We provide a small overview of such studies in Table 2.1 below.

While this is by no means a complete overview of all the empirical studies into botnets, nor a complete list of botnets, these eight well-received studies show the general botnet features studied, as well as the data sources used. In short, their findings mainly revolve around ‘size, devices and attacks’. Additionally, none of the studies contained data from internal C&C to C&C server communication, or showed administrative or economical features of these botnets.

[1] PICO: Population, Intervention, Comparison and Outcomes

Table 2.1: Overview of empirical measurement studies of large-scale botnets

Authors	Botnet	Year	Botnet feature(s) studied	Data source(s)
Holz et al. [34]	Storm Worm	2008	Propagation method, bot population size	Botnet infiltration: spam traps, client honeypots, binaries, communication and keys, crawler
Stone-Gross et al. [55]	Torpig	2009	Bot population size	Botnet takeover: bot data, account data, C&C structure
Cho et al. [18]	Mega-D	2010	Attacks: spam operations, C&C architecture labelling	Botnet infiltration: C&C server types, C&C groups, spam template structures
Shin and Gu [50]	Conficker	2010	Distribution over networks, victims	Sinkhole data: victim IP-addresses
Andriesse et al. [6]	P2P Zeus	2013	Network topology, p2p protocol, communication patterns, DGA usage	Sample analysis and reverse engineering: malware samples, communication protocol, announcements and messages, DGA
Antonakakis et al. [7]	Mirai	2017	Size, C&C infrastructure, devices, attacks	Network telescope, scanning, telnet honeypots, DNS, C&C milkers, DDoS IP Addresses
Marzano et al. [41]	Bashlite, Mirai	2018	Malware evolution, attacks, targets, commands, C&C servers	Monitoring infrastructures: honeypots (URL/command logging and payload downloading), 'fake' monitoring bots
Herwig et al. [33]	Hajime	2019	Size, P2P process, geographical dispersion	Backscatter data, binaries, scanning the infrastructure, C&C communication

2.3. Botnet research challenges

Combining the takeaways from previous work, there are some reoccurring challenges. First, research shows that the botnet phenomenon is still continuously evolving. This can be explained from the broad range in designated targets and use cases botnets can be utilized for [57]. Moreover, the amount of vulnerable targets to victimize has not decreased and malware that can be used to create botnets or access to a botnet can be easily bought/rented on online anonymous marketplaces. Botnets also evolve in response to findings from researchers and new detection methods, creating somewhat of an arms-race between the detecters and the malware designers [31, 39, 40, 51, 54].

Second, these responses of botnets to technical advancements have been observed in multiple studies, but there is not yet a taxonomy or framework to describe or interpret them. Besides "evasion techniques" [4, 14, 52, 53, 66], terms such as "obfuscation mechanisms" [5, 7, 12, 29], "deception mechanisms" [12], "defense techniques" [13] and "resilient design" [56] have been used to describe botnet behavior, with little conceptual embedding.

Third, research on botnets is notoriously challenging due to a lack of data. There are not many large real-world datasets that researchers can use for testing their proposed solutions and collecting data from a botnet for empirical measurements often requires both active and passive longitudinal measurements. Moreover, data related to managing the command and control architecture or other back-end infrastructure ran by the botmaster(s) can only be gathered through infiltrating a botnet, through leaks of the botnet, or from law enforcement takedowns.

Fourth, botnet takedowns are challenging and costly. Although there have been successful takedowns of botnet infrastructure, takedowns are an extremely time-consuming (and therefore costly) technique that often can only be done legally by governmental agencies. A few papers have studied the impact and challenges of botnet takedowns, such as Nadji et al. [43], Ife et al. [35] and Le Pochat et al. [39]. Georgoulias et al. [29] performed a survey study and analyzed 28 botnet takedowns since 2008, most of which from industry reports and news articles. They found that takedowns often focus solely on the technical infrastructure of botnets, by domain sinkholing, seizing or shutting down servers (C&C or DNS servers) and that there has been little focus on interventions that target the other, non-technical parts of their operations.

Fifth, research into botnet economics has gained traction, but is often done solely based on case studies because of a lack of ground truth data. Studies into the business models of botnets by Georgoulias et al. [29], Putman et al. [46] and Bottazi & Me [15] have used a case study methodology, in which the authors did not have access to any data related to these cases other than news articles and other research papers.

2.3.1. Research gap

Similar to how empirical measurement botnet studies are the predecessor for better detection and mitigation strategies, we propose an empirical study into the economics of a botnet to aid the design of alternative interventions. We specifically want to look at how botnets interact with the hostile environment they operate in, and interpret these responses through mapping them on deviant security controls. Additionally, we want to provide an in-depth example of these security controls by performing an analysis of ground-truth data of a botnet that had a business model of providing services to customers: Avalanche.

2.4. Avalanche

The operations of Avalanche have been analyzed and studied both during its lifetime and after its takedown. In this section, we paint the picture of the history of avalanche up to the takedown in section 2.4.1. Next, we describe the investigation and takedown based on information from open sources in section 2.4.2.

2.4.1. History of Avalanche

The "Avalanche" group was awarded its name because of the quite literal avalanche of phishing emails and phishing sites it hosted on its infrastructure [1]. It was first seen in December 2008 and received its name from the Anti-Phishing Working Group (APWG). According to APWG, 67% of all phishing attacks[2] in the second half of 2009 could be attributed to the Avalanche infrastructure. In their Global Phishing Survey, APWG hints that Avalanche could be a successor of the threat group "Rock Phish" or "Rock Gang", a hypothesis that is corroborated by the SecureWorks Counter Threat Unit (CTU) [58]. The identifying factors that link Rock Phish and Avalanche together according to SecureWorks are phishing automation factors, such as registering large amounts of domains (instead of hacking of abusing compromised web hosting sites), applying fast-fluxing techniques to ensure longer uptime of phishing domains and using one domain for hosting multiple phishing sites.

APWG notes that phishing domains from Avalanche were often hosted on compromised consumer-level computers, which made Avalanche a 'botnet' of sorts. This ensured that the takedown or suspension of those sites could not be ordered by an ISP or hosting provider, since they were not involved in the hosting. As a result hereof, mitigation efforts at that time concentrated on suspending the domain itself. Avalanche was well aware of this, and thus mostly registered its domains at non-responsive or vulnerable domain name registrars and registries. In 2010 for example, Avalanche abused mainly the .KR and .PL TLDs by registering over 70% of their domains at those ccTLDs [1].

At the end of 2009, APWG reports that the Avalanche infrastructure was used to distribute Zeus, a trojan that, amongst other functionalities, gives the attacker remote access to the victim's computer. Because code of the Zeus malware was only publicly offered for sale in March 2011 and was leaked a couple months later, it is suspected that the author of Zeus closely collaborated with Avalanche [58]. The reason for the change of business model seemed to originate in a combination of improved security of online banking and the success of the Zeus malware. Simply stealing the banking details by phishing and using those details for online payments and transactions proved to be more difficult for Avalanche because of more strict online banking security measures. Zeus malware, that was infecting machines through a combination of spam, drive-by-downloads and fake software upgrade sites, would grant the attacker access to both personal (banking) data and to the computer that was used for that banking. As APWG notes in their report: "*It is simply more profitable to control someone's computer remotely and move large amounts of money than to simply steal victims' online banking credentials.*" [2].

After the switch to distributing Zeus malware, Avalanche started to either collaborate or sell access to its platform as a service to other cybercriminals. From then on, other malware variants have been observed to be spread from Avalanche infrastructure. In total, there were more than 20 different malware families hosted on Avalanche from 2013 onwards [48]. These malware variants include Bolek, Citadel, CoreBot, Gozi2, KINS / VMZeus, Marcher, Matsnu, Nymaim (later GozNym), Pandabanker, Ranbyus, Rovnix, Smart App, Smoke Loader / Dofoil, TeslaCrypt, Tiny Banker / Tinba, Fake Trusteer App, UrlZone, Vawtrak and Xswkit. Andromeda was also hosted on Avalanche, but not mentioned in the public reports in 2016 since there was an ongoing investigation into Andromeda. A year after the takedown of Avalanche, Andromeda was also sinkholed [49]. Public advertisements from the administrators for their services appeared on various forums in 2015 and 2016, advertising a "fast fluxing bulletproof hosting service" [27].

2.4.2. Investigation and takedown

The following information on the investigation and takedown of Avalanche has been written based on multiple public sources: an interview with the German prosecutor Frank Lange by Lukas Heiny [32], the affidavit by FBI special agent Aaron O. Francis [27] and public communications by Europol [24, 25].

The investigation into Avalanche was started officially by the German law enforcement in 2012 [32], after multiple reports were filed about what would later become known as "ransomware": a windows encryption trojan that encrypts data on computers and only releases them after the victim has paid the ransom. This

[2] APWG defines an attack as follows "... a phishing site that targets a specific brand or entity. One domain name can host several discrete attacks against different banks, for example."[3]

malware was sent to possible victims in the form of a spam email with a malicious attachment that, when opened, would download the trojan from a remote server. Retracing the IP-addresses the spam emails originated from, the investigative team found out that there were multiple of these servers at a time and that the servers changed every one or two weeks [32]. Because of this, they suspected that Avalanche might be a botnet: it was misusing the machines of others, and through controlling these, it was spreading malware to the encryption trojan victims. When they monitored connections from these servers, they also observed other types of malware: key-loggers, banking trojans, other encryption trojans et cetera. It seemed like the network was bigger than just the network of one cybercriminal. The investigation itself was challenging: for each server they wanted to monitor, the investigative team needed to obtain warrants, often outside of German jurisdiction which meant a slow international and bureaucratic process. When they did get access to servers, they were often "perfectly encrypted", leaving them with little information to analyze [32]. However, they kept following the connections based on the collected malware sample, identifying first and second-level servers and monitoring them. Through this monitoring of servers, they were able to create insight in the span and size of the Avalanche architecture. In 2014 the investigators found a database with 18 million records of stolen email addresses and passwords and they realized two things: one, that it is not just German citizens that have become targets and two, Avalanche operates 'as a company', providing its services in a professional manner to other cybercriminals [32]. They also found proof for that in the advertisements of the admins – who call themselves "User41", "Firestarter", "flux", "flux2" or "ffhost" – on Russian hacker fora [27, 32]. Additionally, they found out that a second Avalanche network exists, which is not focussed on providing double fast-fluxing services, but on recruiting money mules [27, 32].

In June 2015, the German law enforcements started a collaboration with the FBI, who had also started an investigation into the Avalanche network based on the US-based victims. By working together and combining evidence, they identified 16 suspects and two administrators. The two administrators were seen as the programmers or "masterminds" behind the technical operations of Avalanche and the main goal of the investigative team bacame to arrest the two administrators [32]. To neutralize the threat of Avalanche bots, a botnet sinkhole was created with the help of the ShadowServer organization. On November 2016, in a joint multinational operation facilitated by Europol, the two administrators were arrested and the Avalanche domains were sinkholed. Thirty-nine servers were seized and hosting providers were ordered to shut down 221 servers from the Avalanche network [24]. The infrastructure was hosted in 30 countries and impacted 60 registries worldwide [48]. The sinkhole operation was the largest of its kind, with more than 800,000 domains blocked, seized or sinkholed [24].

3
Conceptual framework

In this chapter we define the concepts *business continuity, security controls* and *adversarial context*, and show how these can be combined into a framework to categorize evasion techniques of botnet operators. We first define the adversarial context of botnets in section 3.1. Next, we interpret the evasion techniques of botnets through the lens of deviant security theory as security controls in section 3.2 and create a framework of threats to botnet operators and their corresponding security controls. We then define business continuity in section 3.3. Because security controls need to be analyzed within their economical context, we describe the business model canvas an its components in section 3.4.

3.1. Adversarial context

The adversarial context in which botnets operate while trying to fulfill their business model, is created by security practitioners, researchers and law enforcement. In the canon security risk model, there are attackers that launch attacks which can lead to incidents at, for example, companies. For botnets, these aforementioned parties are the attackers that launch ‘attacks‘ on their operations. The attacks from researchers come mostly from what we described in our overview of botnet studies: new detection strategies, mitigation mechanisms and empirical studies that unveil their modus operandi. Security practitioners have a broader range of attack types; from security firms publishing white-papers and reports on specific botnets, to firms providing detection and mitigation systems, or publishing blocklists of C&C domains. The most comprehensive attack is however a botnet takedown, often done in a collaboration between law enforcement and security practitioners. While a botnet takedown is often synonym with sinkholing (domain seizure and/or domain preregistering), bot takeovers (peer injection or peerlist poisoning) are also takedown methods, especially in P2P-systems.

We summarize three main types of threats to a botnet operator from this adversarial context:

1. **Blocking:** This can be blocking the infection, blocking the communication (either between bots or between bot and C&C-server), blocking propagation or scanning behavior et cetera. This causes the botmaster to be less effective in its operation.
2. **Takedown of infrastructure:** We interpret this as both the takedown of a part of the infrastructure, or a takedown of the entire infrastructure. Takedown of a part of the infrastructure can for example occur through patching a bot, through quarantining of a bot by an ISP, or through a notice to takedown of a C&C server that leads to a response from the hosting provider. A complete infrastructure takedown can be focussed on different assets of the botnet infrastructure. Domains can be sinkholed through preregistering or seizure, servers can be shutdown or seized, or in the case of a P2P structure, peers can be ’sinkholed’ (through injection and peerlist poisoning). This causes the botmaster to lose access to the assets it uses to operate the botnet and often means the botmaster needs to rebuild its infrastructure and operations.
3. **Attribution of botmaster(s):** Attribution of a botnet operator means that the true identity of the operator can be connected to the botnet. This is necessary for criminal investigations and thus causes the operator to be vulnerable to arrest and prosecution.

3.2. Evasion techniques as deviant security controls

In the previous chapter we described how botnets respond to these external threats with a variety of techniques (see 2.3). Using deviant security theory, we can interpret the aforementioned evasion techniques, obfuscation mechanisms, resilient designs etc. as security controls. Deviant security takes the cyber criminal as the referent object and studies the security policies and mechanisms, e.g. the *security controls*, taken by "natural and legal persons who are criminally liable for the commission of crime, in order to protect the criminal and his/her crimes" [60]. The theoretical framework combines concepts from information technology, social science, legal scholars and microeconomic theory to reason about what security controls are and why they are used. Security controls can be divided into technical, administrative and physical security controls, which can either be defensive or offensive in nature. An example of a technical offensive security control is the anti-analysis capability of Rombertik malware, that was designed in such a way that it would actively wipe and corrupt the computer used for reverse-engineering, decompilation or any other detected malware analysis environment [11]. An administrative defensive control can be to register for services with fake (personal) information, such as registering at cryptocurrency exchanges with stolen credit card information.

The techniques applied by botnet operators in response to threats are thus controls to protect themselves and/or the crime. To provide an overview of these techniques and show how they can be categorized according to deviant security theory and the adversarial context they respond to, we create a framework in Table 3.1. On the vertical axis are the three different types of security controls: technical, administrative and physical. On the horizontal axis are the threats to botnets: blocking, takedown of infrastructure and attribution. To aid the classification of currently known techniques, we mapped the terms used in previous work to the threat they are a response to. Because detection and blocking techniques have improved, botnet operators have developed new techniques for stealth (to go unnoticed) or evasion (to do something different than expected). Because of takedown measures, botnet operators try to make their infrastructure more resilient. Finally, attribution efforts have caused botnets to respond with controls for the concealment of ownership. We fill the framework with the evasion techniques from the eight empirical botnet studies from Table 2.1. Although this does not provide a complete overview of all security controls, the table shows the empirical botnet studies do not provide examples of measures other than technical ones.

Table 3.1: A framework for categorizing botnet security controls

	Blocking		Takedown of infrastructure	Attribution of botmaster(s)
	Stealth	**Evasion**	**Resilience to takedown**	**Concealment of ownership**
Technical	Obfuscation [7]	Polymorphism [18]	P2P-architecture [6, 34]	–
	Deleting binary [7]	Domain Generation Algorithm [7, 50]	Domain-flux [55]	–
	Obfuscate process name [7]	DNS C&C communication [7]	Reputation schemes [6]	–
	Protocol manipulation [51]	P2P DTH for C&C communication [33]	Non-persistent peer list entries [6]	–
Administrative	–	–	–	–
Physical	–	–	–	–

3.3. Business continuity

Business continuity is a term used often in conjunction with "planning", which is the planning done by corporations to prevent disruptive incidents to negatively impact business operations. In that definition, business continuity planning is a subset of general risk management. We however want to use *business continuity* in its definition that is similar to the often-used idiom *organizational resilience*: an evolutionary process in which businesses adapt in response to their environment to continue their business operations [42]. With that definition, business continuity supersedes risk management and can be used to describe the responses of organizations to both disturbance and munificence [42].

For botnet operators, the disturbances in the environment are the types of threats that originate from their adversarial context. Their responses to that – the evasion techniques we interpret as security controls – have the goal to protect their business model, in order to continue their business operations. For explaining why security controls are used, deviant security reasons from the economic concept of *opportunity costs*: a value tradeoff between how much of the objective of commissioning the crime (performing all the activities of the business model) he/she wants to give up for the objective of achieving some level of protection or security. Because of this value tradeoff, security controls should be interpreted in conjunction with the applied business model: security is an asset to defend an asset that is key in the operation of the business model. Thus, through studying (A) the security controls and (B) the business model of botnet operators, we can answer the question of how they achieve business continuity.

We therefore decompose our main research question *"How did Avalanche ensure business continuity, given its adversarial context?"* in two subquestions:

1. What is the business model of Avalanche?
2. Which security controls did Avalanche apply?

3.4. Business model canvas

To understand the business model of botnet operators, we apply the business model canvas. The Business Model Canvas (BMC) [45] is a well-known tool for developing and improving (new) business models. Previous work has used it to describe cybercrime and specifically botnet business models before [29, 46]. Based on nine building blocks, companies can easily describe the core of their business. The building blocks are: value propositions, key partners, key activities, key resources, customer relationship, channels, customer segments, cost structure and revenue stream. We will give a general description of these building blocks here.

The value proposition describes what products and / or services you will provide to your customers. Often the value proposition mentions a problem or an unsatisfied desire experienced by a certain group of people. The key partners, activities and resources give an overview of the infrastructure underlying the business. It lists the main external partners and suppliers, where a trade-off can be made between the benefits (partners can allow for the business to reach its goal faster or contribute to the overall success) and the risks and uncertainties (which necessarily follows from outsourcing responsibilities). The main business activities, which are required by the value proposition, are described, including for instance activities related to distribution channels, customer relations, and production. The resources refer to both tangible (office space, raw materials, computers, financial) and intangible (knowledge, skill, patents) assets. The customer relations, customer segments and channels give a descriptive overview of the target audience: the customer. It describes the target audience (characteristics, needs and desires, size), the relationship with the customer (e.g. the mode and frequency of interaction) and how these customers will be reached (which includes everything from marketing and awareness to purchase, delivery and aftersales). Finally the finances of the business are set out in the cost structure and revenue model building blocks. This last one includes the price of the product / service, as well as the kind of revenue stream (e.g. one-off purchase, subscription, lease).

4 Methodology

Our methodology consists of a tiered approach, in which each next step builds upon the preceding findings. In this chapter, we briefly outline the steps of our approach in section 4.1. Because we use data from Avalanche to answer our research question, we describe the suitability of Avalanche as a case for this research in section 4.2. We conclude this chapter by discussing the ethical considerations that this research raises in section 4.3.

4.1. Approach

We use data that is not often used in academic research (ground truth data from a criminal investigation), to answer a question that has no predefined methods or earlier methodological work we could lend insights or guidance from. Closest to our study is the work of Noroozian et al., who studied the anatomy and economics of bulletproof hoster MaxiDed from ground truth data [44]. To our knowledge, the security controls of a botnet have not yet been studied on empirical ground truth data. Because of this, our approach is of an exploratory nature. We:

- First describe our data through and perform an exploratory analysis, processing, validating and interpreting the data provided to us for this research;
- Next, we analyze Avalanche's business model by applying the Business Model Canvas of Osterwalder & Pigneur [45];
- We can then, based on our understanding of Avalanche's infrastructure and business model, describe security controls and interpreted them in light of our framework.

4.2. Avalanche as a case

Two papers have used Avalanche as their main case. Le Pochat et al. [39] used data from the German investigation of Avalanche, to create a model to classify DGA domains generated by malware that was distributed by the Avalanche infrastructure. The data for this paper consisted of DGA-generated domains in 2017 and 2018, after the takedown of Avalanche, to aid the manual process of correctly sinkholing registered DGA-generated domains. Wainwright and Cilluffo [64] use Avalanche as a case to describe the Crime-as-a-Service model: a model in which the service model is to provide an all-inclusive service from malware to money-laundering. They conclude the following: *"The very essence of the Avalanche network and its CaaS operating model, with its varied portfolio of innovative products and services, pro-active advertising, and customer support features, is a striking example of how cybercriminal groups today work like international businesses"* [64]. Additionally, there are papers that used the case of Avalanche as an example or case in a multi-case study. An example is the paper of Dargahi et al. [22] that create a cyber-kill-chain based taxonomy of crypto-ransomware features and describe the fast-fluxing behavior of Avalanche in relation to the WannaCry ransomware that was hosted on Avalanche.

We want to use avalanche as a case to study the security controls of a botnet operator that had a service-providing business model. We believe the case of Avalanche to be interesting because their platform was used by a broad variety of malware groups and because these variants were successful and had a lot of (negative) impact. Similar to Wainwright and Cilluffo, we argue that Avalanche is a good example of a facilitating platform because it provided a lot of different services for its customers and those services were advertised (semi) publicly. Moreover, because of the longevity of the platform and the multi-year police investigation, there is longitudinal data to study the behavior of Avalanche.

A counterargument to using Avalanche as a case to study security controls, is the remark that Avalanche has been taken down and thus had its security breached. What use is it to study something that got taken down? The main reasoning is that the goal of this study is not to make any qualifying statements about how well or not Avalanche arranged its security, nor is the goal to quantify its security through measurements. There is also a more practical point for studying a platform that has been taken down: it is impossible to get data from a platform that has had no leaks, breaches or takedowns. We therefore use the fact that Avalanche has been active for a long period of time (even while it was being investigated) as a signal that it had at least some functioning security controls.

4.3. Ethical considerations

Using data from criminal backends to study their security raises ethical considerations. The data of Avalanche originates from the police investigation by German law enforcement. All data was seized in accordance with German law. Access to data was provided to us for academic research and this access was only granted to the authors of this work through monitored police systems. The data contained personally identifiable information of both Avalanche administrators, their customers (malware groups), as well as victims. Because it is not possible to retrieve informed consent for these vulnerable parties such as the victims, we took great care in anonymizing their data in our analysis and we only report aggregate statistics. While this data contains evidence of crimes for which the administrators of Avalanche were prosecuted, this work does not seek to provide any legal proof of any criminal conduct whatsoever.

5 Data & exploratory analysis

Laying the foundation for our measurements and analyses, we first provide an in-depth exploration and high-level analysis of data from Avalanche. During the investigation and takedown of the Avalanche platform, German law enforcement collected different sources of forensic evidence related to the platform. From the collected evidence, access has been granted to data related to 19 hosts within the Avalanche network. This data consists of three types: server images, network traffic from wire taps and seized databases (extracted from active hosts, server images or VM snapshots). Table 5.1 provides an overview of the types of data available, the timeframe the available data was from and the size of the data for each host.

Table 5.1: Overview of data types, dates and sizes per host

Host	Server image		Network traffic			Seized database		
	Seize date	Size	First date	Last date	Size	First date	Last date	Size
Host 1	21-09-2017	~3.1 TB	19-10-2015	30-11-2015	~2 TB	-	-	-
Host 2			21-11-2015	27-11-2015	~324 GB	-	-	-
			01-06-2016	07-06-2016	~337 GB	-	-	-
Host 3	28-10-2013	~1 TB	25-02-2013	25-02-2013	~20 GB	-	-	-
Host 4	-	-	24-02-2016	25-02-2016	~38 GB	-	-	-
Host 5	03-03-2017	~149 GB	-	-	-	-	-	-
Host 6	-	-	02-08-2016	30-11-2016	~381 GB	-	-	-
Host 7	07-06-2017	~2 TB	07-10-2015	27-10-2015	~141 GB	-	-	-
Host 8	30-11-2016	~601 GB	-	-	-	-	-	-
Host 9	30-11-2016	~480 GB	-	-	-	-	-	-
Host 10	30-11-2016	~21 GB	-	-	-	-	-	-
Host 11	14-03-2017	~572 GB	-	-	-	-	-	-
Host 12	-	-	03-09-2015	20-10-2022	~34 GB	-	-	-
Host 13	15-03-2017	~297 GB	-	-	-	-	-	-
Host 14	-	-	22-12-2014	21-07-2015	~7 GB	-	-	-
Host 15	-	-	29-02-2016	14-04-2016	~206 GB	-	-	-
Host 16	-	-	20-10-2015	18-12-2015	~568 GB	-	-	-
Host 17	-	-	06-02-2014	29-07-2014	~614 GB	-	-	-
Host 18	-	-	-	-	-	23-03-2016	30-11-2016	~10 GB
Host 19	-	-	-	-	-	21-04-2016	26-04-2016	~23 MB

Criminal backends do not come with an instruction manual, leaving us with the challenge to make sense of the data LE investigators discretely extracted from criminal operations. Moreover, very little to no data descriptions or earlier analyses of the data were provided by the investigative team. We were granted access to in total close to 13 TB of data and it was a challenging feat to understand what was in the data, how it could be interpreted, and how it could be used for our research. Our approach was to explore the data based on its data type and look into the server images, network data and databases separately. For each data source, we followed these steps:

1. Make data readable and searchable through software processing;

2. Validate data;
3. Extract high-level descriptives;
4. Interpret available features.

We describe this process and the resulting insights for each data type in the sections 5.1, 5.2, and 5.3. Last, having understood what our data contains, we interpreted the entirety of the available data, as well as how the data sources are connected, in section 5.4.

5.1. Server images

5.1.1. Processing of server images

The server images of the Avalanche hosts all had the .E01 file format. This is the EnCase Evidence file format, which contains a byte-for-byte depiction of the acquired volumes. Additionally, E01 files hold forensic metadata (such as the time and date of the forensic copy and whether there were read or write errors) and a device-level hash. For our analysis, we used EnCase Forensic (version 21.4.0.109) to load all server images.

5.1.2. Server image validation

All the server images were obtained from the legal seizure of the servers during the Avalanche investigation. This means that procedures relating to forensic evidence have been followed, for example by making full disk images (of all partitions) with write blockers and forensic software and calculating hashes for each image. A forensic report made by the software used to create the forensic copy was provided for each server image. When you load an E01 image in EnCase Forensic, it calculates the acquisition hashes in MD5 and SHA1. To validate that the E01 file we received is similar to the image that was made from the Avalanche host, we compared the MD5 and SHA1 acquisition hashes from EnCase with the hashes from the forensic reports provided to us by German LE. For each image, we observed exactly the same hashes in the forensic report as calculated by EnCase.

We also validated whether EnCase recognized an operating system or file system. This was not the case for the image of host 8, where only data partitions with unallocated space were recognized by EnCase. We hypothesized the image might contain a Linux-based distribution which had a corrupted master boot record (MRB), or a deleted or broken partition table. We tried to recover the data by using the open source program TestDisk [17] but were unsuccessful. We therefore excluded the image of host 8 from our following analyses.

5.1.3. Descriptives

We manually extracted the following three features from each of the server images: OS version, probable install date and time zone. The approach was slightly different for each OS type.

Debian: The OS version for Debian systems was taken from the file `/etc/debian_version`. The probable install date was taken from the first line in the syslog file `/var/log/installer/syslog`. The timezone was taken from the `/etc/timezone` file.

FreeBSD: The OS version for FreeBSD systems was taken from the file `/bin/freebsd-version`. The probable install date was approximated from the access time of the folders in /boot, such as /defaults, /modules, /firmware. We verified this by looking at other files and checking whether these files indeed had the earliest access time. This approximation was necessary because the systems did not have a `/etc/defaults/rc.conf` file. The time zone information was taken from the `/var/db/zoneinfo` file, which saved the name of the timezone file installed last [28].

CentOS: The OS version for CentOS systems was taken from the file `/etc/centos-release`. The probable install date was taken from the creation date of the file `/anaconda-ks.cfg`. This file is created after an installation completes and saves all choices made during the installation [16]. The timezone was taken from the `/etc/localtime` file.

Although there are some similarities between servers in their used time zones, more notable is the amount of different OS versions and install dates. From this we can assume Avalanche either did not have one instance that was deployed to all servers, or that the servers were all installed by different people.

5.1.4. Interpretation of image data

The data pertained in these server images relates to the configuration of each host, their connections to other hosts in the infrastructure and the actions of the administrators. For the configuration and thus role of a host, we can for instance look at the installed packages, the network configuration (e.g. the configured interfaces)

Table 5.2: Descriptives of seized server images

Host	OS version	Install date	Timezone
Host 1	Debian 8.0	26-01-2015	GMT+2
Host 3	Debian 7.2	16-10-2013	GMT-5
Host 5	FreeBSD 10.3	03-08-2016	GMT+2
Host 7	CentOS 6.7	24-01-2015	GMT+2
Host 8	-	-	-
Host 9	CentOS 6.8	16-10-2016	GMT-4
Host 10	CentOS 6.8	25-10-2016	GMT+3
Host 11	CentOS 6.6	31-10-2015	GMT-5
Host 13	FreeBSD 10.3	31-08-2016	GMT-6

and the configured hosts. Especially the installed packages provide insights into the usage of encryption and virtualization. The server images also contain information on their connection to other hosts, which can be extracted from for example the SSH known hosts, the Nginx settings (e.g. proxy parameters). Finally, data related to the action of the administrators can also be extracted, such as the amount of configured users, their passwords and log files of their actions (such as the bash history).

5.2. Network data

5.2.1. Processing network data

We received the network data in the form of .pcap files. PCAPs are packet capture files, which contain packet data that was captured from a host or network. In the case of Avalanche, the PCAP files were gathered from wire taps on the Avalanche hosts during the investigation. A common approach is to analyze .pcap files with the open source program WireShark, but since we received around 4.7 TB of network data we opted for a more automated approach. We processed all PCAPs with the open source program Arkime (formerly known as Moloch) [8], which is created to handle large amounts of packet captures and make them searchable. A high-level overview of the processing pipeline can be seen in Figure 5.1. We used version v3.1.1 (release date 26-01-2022) [9] and processed in total 45,042,359 sessions of network data.

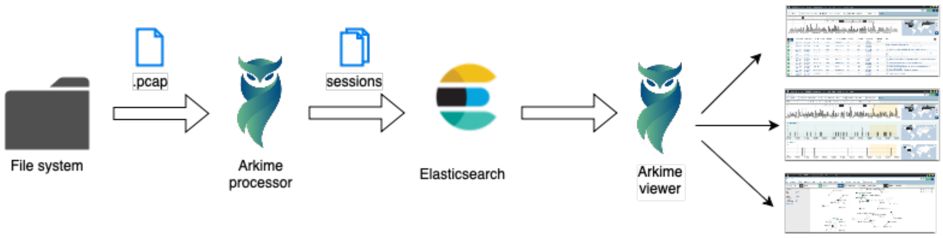

Figure 5.1: PCAP processing pipeline

The PCAP files are processed by Arkime and from the packet data, Arkime groups connections between hosts in sessions. The most obvious example is a TCP session, that is build from different connection segments. For these sessions, metadata relating to the IP-adresses such as GeoIP information is retrieved by Arkime. These resulting sessions and their metadata are stored in a generalized format as Elasticsearch data. To interact with this data, the Arkime Viewer uses Kibana-like features to display, group and visualize session data. Because the original PCAPs are stored separately from the session data, it is possible to download the raw packet data for each session for further analysis in for example WireShark.

5.2.2. Validation

We validated the correct processing of all the PCAP files by verifying that all the files for each host were processed and available in Arkime's file index. Arkime annotates sessions with tags if it encountered any irregularities during its processing, such as "out-of-order-src" or "incomplete-tcp". We provide an overview of the amount of occurrences of these tags and the percentage of sessions that have one or multiple of these tags in Table 5.3.

Table 5.3: Tags of Arkime processing

Tag name	# occurrences	% of total
acked-unseen-segment-dst	5,962,708	13.24
acked-unseen-segment-src	5,742,318	12.75
out-of-order-dst	4,754,217	10.55
out-of-order-src	4,365,668	9.69
incomplete-tcp	561,900	1.25
no-syn-ack	273,084	0.61

The "acked-unseen-segment" tags occured often and originated from Arkime observing ACKs for packets that were not found in the data. Possible causes are packets from sessions that were ongoing when the capture was started, or the capturing device was not able or capable enough to capture all traffic. Both the "out-of-order-dst" and "out-of-order-src" tag indicate that Arkime encountered timestamped packets that were in the incorrect order. This does not influence the reassembling of the TCP-session, but only shows the order differently (true to the timing in the PCAPs). The "incomplete-tcp" and "no-syn-ack" tags stem from the TCP conversation completeness analysis, which calculates the completeness of the TCP conversation based on the observed packet types. The "incomplete-tcp" tag occurs when not all 6 packet types were observed, while "no-syn-ack" only indicates missing the syn-ack packet [65].

Interpreting Table 5.3, it might look like as if there were more tags than sessions. However, manual inspection showed that often a sessions had four to six tags connected to it. We therefore queried to see how many packets had at least one or more tags, and that number was 6,985,607. This means that in total 15.51% of all sessions had at least one (and often multiple) tags. Additionally, we observed that the tags were not distributed equally over time; almost 12% of the sessions with tags (~840,000 sessions) originated from 2.5 hours of recorded network data on a single day, indicating a possible incomplete capturing process at one host for a small period of time.

Our validation showed that the network data is not always complete and that the data captures were not perfect. Intuitively 15.51% of all sessions having at least one tag feels like a lot, but to our knowledge there are no comparable cases to compare this number with. Additionally, there is not much we can do to improve the packet data. Therefore we proceed our analysis with the notion of the presence of missing data.

5.2.3. Descriptives of network data

For each host, we manually extracted the following features from Arkime. The resulting descriptives can be seen in Table 5.4.

- **# days**: the amount of days between the first packet and the last packet, increased by 1 to account for the last day;
- **Total amount of sessions**: amount of sessions processed from the PCAPs by Arkime[1] ;
- **% TCP / UDP / ICMP**: the percentage of sessions with the TCP, UDP or ICMP protocol;
- **% incoming / outgoing**: The percentage of sessions that were incoming (from another client connecting to this host) versus the percentage of sessions that were outgoing (started from this host connecting to another client);
- **Unique source hosts**: The amount of unique IP-adresses that started two or more incoming sessions to this host;
- **Unique destination hosts**: The amount of unique IP-adresses that started this host started outgoing sessions to.

[1] Sessions with more than 10,000 packets are split by Arkime in multiple sessions of 10,000 max (e.g. a session of 33,000 packets will be split into four Arkime-sessions.)

[2] 10,000 is the maximum amount of unique IP's that can be exported from Arkime. The number 10,000 unique source hosts is therefore the amount of hosts with 14 or more incoming connections to host 17.

[3] 10,000 is the maximum amount of unique IP's that can be exported from Arkime. The number 10,000 unique destination hosts is therefore the amount of hosts that host 17 made 31 or more outgoing connections to.

Table 5.4: Descriptives of network data

Host	# days	Total amount of sessions	% TCP / UDP / ICMP	% incoming / outgoing	Unique source hosts	Unique destination hosts
Host 1	43	2,176,450	97.64 / 1.25 / 1.11	98.91 / 1.09	4,014	771
Host 2	202	6,301,962	94.49 / 5.33 / 0.18	91.66 / 8.34	4,523	1,273
Host 3	1	899,196	99.98 / 0.02 / 0.00	77.27 / 22.73	28	8
Host 4	2	951	56.57 / 33.44 / 9.99	96.40 / 3.60	106	5
Host 6	384	16,696,700	98.94 / 0.97 / 0.08	41.59 / 58.41	7,818	2,099
Host 7	21	2,079,635	98.44 / 1.18 / 0.37	99.36 / 0.64	1,955	1,737
Host 12	48	4,902,728	98.86 / 0.96 / 0.17	70.68 / 29.32	2,820	978
Host 15	46	5,038,219	94.29 / 5.36 / 0.35	91.55 / 8.45	9,326	4,240
Host 16	90	2,719,674	95.01/ 4.69 / 0.29	23.00 / 77.00	1,850	1,942
Host 17	174	4,226,844	37.91 / 49.28 / 12.82	82.63 / 17.37	10,000[2]	10,000[3]

5.2.4. Interpretation of network data

The processed network data contains data and metadata about the sessions to and from hosts in the Avalanche infrastructure. For each session, netflow data such as source IP, source port, destination IP, destination port, protocols, packets, databytes and bytes is available. For each IP, metadata such as ASN, country code and RIR is available. Depending on the protocol, additional data fields are available. For example, for HTTP these fields include method, status code, hosts, user agents and several headers (request, response, server), for DNS these are hosts, OpCode, statuscode, query type and query class, and for TLS these are version, cipher, JA3 en JA3s (hashes). We can manually analyze the packet content of each session, provided that the data is not encrypted (which was the case for protocols like TLS and SSH). Overall, the network data relates mostly to the behavior and thus role of the hosts in the network.

5.3. Databases

5.3.1. Processing databases

The in total 44 databases were provided in .sql files. Of those, 42 databases came from host 18 and the other 2 databases from host 19. We imported all databases in MySQL Workbench and saw that the databases were dumps made from a database at different moments in time. This meant that we had 42 snapshots of the database from host 18 and two snapshots of the database from host 19. To combine the database dumps to one aggregated database that contained all information, we wrote a script that merged and then deduplicated all versions of each table. The deduplication was done conservatively: only rows (or database records) that had the exact same data for each variable were deleted. This means that two 'unique' rows can have equal data in each variable except for one. Manual inspection of the resulting aggregated tables showed that in some tables, formats or the encoding of certain columns changed over time. For example, the aggregated table of transactionsBtc has 16 unique rows, while these are 8 rows with a 'normal' datetime format, and 8 identical rows with a 'Z' added to the datetime format. We chose to not to make any manual deduplication changes to the data, to keep our descriptives as close to the original data as possible. For our analyses in later chapters however, we will explicitly describe on which fields we performed our analysis and how these are for example cleaned or deduplicated on a case-by-case basis.

5.3.2. Validation databases

We validated the integrity of the databases both externally and internally. For external comparison, we searched for sessions with mysql data in the network data and observed host 6 making connections with mysql to host 18. In this data, we could observe table and column names and validate that our dumps contained roughly the same tables and columns. We say roughly, because the network data between host 6 and 12 was from a period before we have database dumps. We observed that two tables were removed and that some columns in the tables were changed in name or field type. We manually validated some inserts into the database and verified that we had the same data in our aggregated database. We did however see that there was another database active at host 18, that contained tables and columns different from the databases we were provided with. Manual analysis of these insert statements showed that the other database was used to store (likely scraped) Spamhaus SBL and CBL data. As a second external validation, we verified 7 bitcoin transactions through querying the Bitcoin blockchain, observing that the transaction hash, datetime and amount was correct. For internal validation we cross-referenced keys between tables. For example, we validated that the ID's of the users in tables `balancheChanges`, `domainsRegistration` and `transactionsBtc` also were present in the table `users` and observed 1 ID to be missing (0.7%). Looking at all our validation steps altogether, the data seems both externally and internally consistent. However, we want to stress that while the available data is consistent, the availability of data differs wildly between databases and columns. We will reflect on what this means for our findings based on the technical and economical reconstruction of Avalanche in the discussion.

5.3.3. Descriptives and interpretation

In order to make the descriptives of the databases, we needed to analyze each table and column from each database. Table and column names were mostly non-descriptive, sometimes in Russian and they often seemed to hold duplicated information. We therefore interpreted each table and each column by manually sifting through the data. The result of this process can be found in the 36 tables with descriptives of the analyzed tables in Appendix A. Based on this interpretation, we created the table names and descriptions that can be found in the Tables 5.5 and 5.6 below. We interpreted the database from host 19 as the domain registration database, and the database from host 18 as the infrastructure monitoring database.

Domain registration database: The domain registration database was copied twice: on 03-01-2016 and 05-04-2016. The database contains data on users, their balances, the domains they registered, the accompanying name servers, the certificates used, as well as random personal data that seems to have been used for creating fake WHOIS records. The database only held a login name for each user - that seems to be chosen by the platform itself - and no other information such as email-accounts or other contact data. Mapping accounts to malware strains is thus not possible besides by matching domains to for instance known blacklists or DGA-domains.

Table 5.5: Tables of domain registration database

Table	Unique rows	First date	Last date	Description
balanceChanges	2,236	01-06-2015	04-04-2016	Log of balance charges to the accounts of users (often related to a certain domain)
dns	8	14-08-2015	09-10-2015	Overview of DNS-servers for some specific domains
domainsRegistration	3,633	2010-03-30	2017-04-04	Log of domains, linked to a user, nameservers and other info
firstNames	478	-	-	Overview of first names and their corresponding gender (male/female)
logsRegistrations	12,171	29-05-2015	05-04-2016	Log of different steps in the domain registration phase: checking availability, registering, assigning or updating NICs and DNS-servers, adding certificates etc.
ownNameservers	670	-	-	Mapping of domain ID's to the IP-addresses for their two nameservers
ssls	36	11-06-2015	23-02-2016	Overview of domains, their login information, the registered email and the SSL certificate details
transactionsBtc	16	04-03-2016	30-03-2016	Log of Bitcoin transactions made by uesrs
transactionsWebmoney	20	23-03-2016	04-04-2016	Log of WebMoney transactions made by users
userCommentWebmoney	7	23-03-2016	04-04-2016	Mapping of a WebMoney payment number, timestamp and a user
users	143	02-01-2015	04-04-2016	Overview of users, their balance, their idClient, register date, tariff and BTC address
whoisRandom	7,588	-	-	(Fake) personal data that is used to fill the WHOIS-info for registered domains
whoisRandomUsed	2,786	-	-	(Fake) personal data that is used to fill the WHOIS-info for registered domains
whoisCC	12	-	-	(Stolen) personal data that is used to fill the WHOIS-info for registered domains
words	16,922	-	-	List of random words, possibly used for seeding a DGA

Infrastructure monitoring database: The infrastructure monitoring database was copied 42 times between 23-03-2016 and 30-11-2016. The database contained data on the active servers in the proxy-architecture and their role, domains, IP-addresses, name servers etc. Some tables held historical data, going back to March 2015. There were multiple identifiers that connect the infrastructure monitoring database to the domain registration database, such as account names and account identifiers. Table 5.6 describes the tables, columns and their contents. Some tables held historical data (sometimes from dates preceding the first copy) while other tables only hold data associated with the previous week. This means that although it seems that there is data from March '16 through November '16, some tables only held data from a couple weeks in total. In general, there are four types of tables: *overview* tables that provide an overview of a type of server, ip or domain, *mapping* tables that map one identifier to another, *logging* tables that log certain characteristics with a certain interval in time, and *settings* tables that hold settings for a certain action or check.

5.4. High-level analysis of data source connections

The server images provided an in-depth view into the configuration of the servers, the services that were run and the sub-network the hosts were a part of. The network data allowed us to observe the network data that reaches and leaves hosts in the (managed) network, and, in the case of unencrypted data, study the content of the data that has been received and send. The databases contained longitudinal information on the assets used (servers, IPs, domains, DNS records, etc.) in their network and which customer is related to which asset. By combining these data sources, we had access to information on both the technical and administrative working of the Avalanche platform from March 2010 to November 2016, spanning a large proportion of Avalanche's total lifespan. For our high-level analysis of the connections between data sources, we extracted host IP(s) from server image or network data, queried aggregated domain registration database and aggre-

Table 5.6: Tables of infrastructure monitoring database

Table	Unique rows	First date	Last date	Description
clientLogs	2,719	07-03-2015	29-11-2016	A log that links client ID's and domains to a certain action (1-4)
logs	8,108	08-08-2015	29-11-2016	A total log of which admin user did what actions (1-6) on which domain or IP
bots	3,324	18-07-2015	30-11-2016	An overview table that holds data on the currently active 'bots' (servers) and their settings
botGroups	160	-	-	An overview table of the active bot groups, how they are functioning and if the groups have enough bots
countryIPs	86,016	-	-	A mapping from IP-ranges to 230 country codes
botsNS	8	03-03-2014	29-07-2014	A dated (2014) overview of nameservers that were used for bots
settings	1	-	-	The settings that were used to check whether the proxy-architecture worked as intended
checkSpeedServers	16	-	-	The IDs, IP-addresses and settings of the dedicated/C&C servers
checkSpeedSpeeds	109,156	01-06-2016	30-11-2016	Weekly log of the speed of servers at a certain date and time
dedicatedServers	1976	16-02-2016	30-11-2016	Overview of the dedicated servers and their settings
domainIPs	3,462	-	-	Mapping of domains to IP-addresses
domainIPsWork	3,046	-	-	Another mapping of domains to IP-addresses
domains	1744	03-04-2012	29-11-2016	Overview of domains and accounts that links those to a main identifier, a client identifier and a bot group.
domainsNS	10,960	-	-	Mapping a domain to nameserver.
nameServers	758	11-08-2015	30-11-2016	Overview of the nameservers, their domains and the ip-addresses of the domains.
nameServersAccount	172	-	-	Mapping of nameservers to accounts
nameServersDeleted	612	28-03-2015	25-11-2016	Historical log of the nameserver domains that were deleted.
layerServers	147	11-03-2016	30-11-2016	Overview of specific servers and their layer ID
trafficBots	36,861	16-03-2016	04-05-2016	Weekly log of the requested and sent count and length of the traffic to a certain bot IP-address on a timestamp
trafficDomains	41,451,002	16-03-2016	30-11-2016	Weekly log of the requested and sent count and length of the traffic to a certain domain on a timestamp
webChecks	33	02-12-2015	29-11-2016	The url that is used for the web checks

gated infrastructure management database for host IP(s) and queried network data for host IP(s). In Table 5.7 we provide an overview of our findings of these steps. We did not find any hits of the host IP-adresses in the domain registration database and have therefore excluded the database from the findings.

Table 5.7: Findings of cross-referencing hosts to available data

Host	Infrastructure management database		Network data			
	# hits	column:table	has connections	to host	# connections	Protocols
Host 1	1	redirIP:domains	Yes	-	-	-
Host 2	-	-	Yes	-	-	-
Host 3	-	-	Yes	-	-	-
Host 4	2	domainOrIP:logs	Yes	-	-	-
Host 5	21,327	domainOrIP:trafficDomains	Yes	Host 6	53,383	TCP (HTTP, TLS), ICMP
Host 6	75,261	domainOrIP:trafficDomains	Yes	Host 5	8,299,845	TCP (HTTP, TLS), UDP (Syslog)
	1	ip:checkSpeedServers		Host 7	3,513	TCP (HTTP)
	18	ip:layers		Host 15	1	TCP
	320	proxyIP:dedicatedServers		Host 18	210,620	TCP (HTTP, TLS, SMTP, MySQL, POP3)
	635	proxyIP:bots				
Host 7	20	redirIP:domains	Yes	-	-	-
Host 9	5	redirIP:domains	No	-	-	-
Host 10	3	redirIP:domains	No	-	-	-
Host 11	19	redirIP:domains	No	-	-	-
Host 12	-	-	Yes	Host 16	7	TCP (HTTP)
Host 13	-	-	No	-	-	-
Host 14	-	-	Yes	-	-	-
Host 15	2	domainOrIP:logs	Yes	Host 6	4,603	TCP (HTTP, TLS)
	4,169	domainOrIP:trafficDomains				
Host 16	-	-	Yes	Host 6	95	TCP (SSH)
				Host 12	283 [4]	TCP (HTTP, SSH)
				Host 18	176,275	TCP (HTTP, SSH, TLS, SMTP, POP3, MySQL)
				Host 19	322	TCP (SSH)
Host 17	-	-	Yes	-	-	-
Host 18	-	-	Yes	Host 6	614	TCP (HTTP)
				Host 12	4	TCP (HTTP)
				Host 16	2,142	TCP (HTTP, POP3, SMTP)
Host 19	-	-	Yes	-	-	-

6 Avalanche's business model

To understand Avalanche's business model, we describe the nine building blocks of the business model canvas in this chapter.

6.1. Value proposition

The value proposition of Avalanche was to provide the service of a stable proxy-infrastructure to other cybercriminals. Their customers could use Avalanche bots as a virtual front for the malware backends, botnet controllers or phishing websites they wanted to operate. Avalanche also added related services to make their offering more appealing, like the (automated) domain registration system and even setting up servers for their customers [27]. The unique selling point was most likely Avalanches reliability: its fast-fluxing architecture had been running since 2009 and had weathered through years of advancement in IP- and domain-blocking techniques.

6.2. Key partners

In order to deliver this value proposition, Avalanche relied on multiple partners. The main one of course being the hosting providers. Avalanche rented mainly VPS and one dedicated server from different hosting providers worldwide for their second-layer servers. Of the 16 servers that had the role of second-layer, only two were rented from the same hosting provider. All 14 other servers were each purchased from different providers, even if two servers were hosted in the same country.

The second important type of partners is domain registrars: based on the logs table of the domain registration database, Avalanche seemed to have used nine different domain registrars. These registrars are somewhat geographically spread over the world, with a registrar in HongKong, China, Russia, Germany, Pakistan, Singapore, The Bahamas and two in the USA. Besides hosting providers and domain registrars, there was a broad range of suppliers that Avalanche relied on, such as certificate providers, providers of stolen identifies (which could have come from one of their customers, or for example from an online anonymous marketplace), VPN-provider(s) to SSH to the infrastructure via a safe connection, a cryptocurrency-platform to generate addresses for customers to send Bitcoin to and WebMoney and other online payment services to receive money from customers.

Although we have not analyzed this, the FBI affidavit mentions that the administrators collaborated with cybercriminals who ran money mule schemes [27]. We can also consider this party as a key partner, if Avalanche used these money mule schemes themselves or if they offered this as a service to their customers.

6.3. Key activities

We characterize Avalanche's key activities as follows:

- **Building and maintaining a stable fast-fluxing infrastructure:** this was the main activity for Avalanche. Multiple sub-activities were part of this, such as (1) compromising servers so they can be used as first layer bots, (2) configuring servers according to different roles and their requirements, and (3) monitoring the servers and the correct working of the proxy architecture.

- **Building and maintaining a domain registration system:** Avalanche built a domain registration system, that was essentially a top-layer built onto several API's of domain registrars. They needed to connect this system with their architecture, in order for the automated processing of changes in domains, nameservers and IPs to facilitate double fast fluxing.
- **Customer service:** An important part of Avalanche's offering was providing customer service. There is no such thing as a standard working of malware, which meant that Avalanche needed to work with its customers to make the infrastructure work for them. Based on the variety of malware strains that were a customer of Avalanche, Avalanche needed to facilitate multiple different protocols, domain generation algorithms and service requests (such as databases to store victim data).

6.4. Channels

There were two main communication channels: online fora for marketing communication and jabber for business and customer service communication. In the affidavit of the FBI, the fora *Verified* and *Mazafaka* are named as fora where the administrators of Avalanche posted advertisements to potential customers [27]. For business and service communication, the administrators of Avalanche had hosted their own jabber service, through which they could be contacted.

6.5. Customer relationship

Maintaining good customer relationships is key to successfully operating a business in which both parties are anonymous. This is similar to online anonymous markets, where reputation and customer service have shown to positively correlate with vendor performance [62]. If we observe the average lifetime of a customer based on the creation date of the account and the last time an update was made to one of its domains, the in total 59 customers purchased services from Avalanche for 168 days. Although we have little to compare it with, we would argue this does show that the average customer was satisfied enough with Avalanche's services to keep purchasing them for a non-negligible period of time.

6.6. Customer segments

In the business model canvas, there are different segments of customers if they want or need different services, or have the incentive to pay for a differentiated service. For Avalanche, we hypothesize that the customers that solely used Avalanche for fluxing, the customers that used their fluxing and registered domains, and the customers that (also) purchased configured servers can belong to different customer segments. While we do have the data to distinguish between customers that only registered domains and customers that also purchased configured servers, the other data available does not allow us to study if the observed differences can be explained from the type of malware used, the knowledge of the customer, its trust in the infrastructure et cetera. Additionally, it is interesting to note that Avalanche did differentiate between customers in the price they asked for services: some customers paid $25 for a domain registration, while others paid $35.

6.7. Key resources

The key resources that were needed to operate Avalanche's business, were:

- **Compromised servers:** because these severs would take on the role of the first layer in the proxy architecture, they needed to be able to handle large streams of traffic and thus have a stable internet connection;
- **Servers for second and third layer proxy services:** these servers should be even more powerful and well-connected, since the multitude of first-layer bots would all connect to these servers;
- **Malware backend servers:** all-round servers that could be configured to serve the differing needs of Avalanche's customers;
- **Domains:** domains were a resource necessary for running the proxy architecture;
- **Payment services:** payment services, preferably anonymous services, were needed to receive payments from customers;
- **Software:** to configure the proxy architecture, to monitor the infrastructure, to register domains, to automate tasks such as backups and synchronization etc.;
- **Certificates:** SSL certificates were needed to encrypt traffic;

- **Administrator(s):** one or multiple skilled persons were needed to build the infrastructure, manage it, provide service to customers etc.;
- **Fake/stolen identities:** these were needed of the domain registration and like also for the server registration;
- **VPN-services:** VPN-services were needed for the administrators to anonymously connect to the infrastructure.

6.8. Cost structure

Unfortunately, our data does not contain much information that relates to the actual costs Avalanche experienced for their operation. We hypothesize Avalanche faced somewhat fixed costs for maintaining the infrastructure, based on the observation that the amount of second layer servers and identified management servers did not fluctuate much over time. Additionally, initial investments might have been made to create or procure the software used for configuration, management and domain registration, as well as purchasing access to payment and VPN services. Other than that, we would argue that Avalanche only had variable costs. When it would want to expand its architecture to serve more customers, the amount of compromised servers, malware backends, domains, certificates, fake identities and administrator time would scale with the amount of customers.

6.9. Revenue stream

Although we only have access to revenue related data in the form of the balanceChanges table in the domain registration database, we estimated a monthly revenue for Avalanche. The balanceChanges table contains data from the 8 month period between June 2015 and April 2016. In that period, the aggregate of all the spent funds was nearly $60,000, or $7,500/month. This is the revenue that was generated through selling registered domains to customers ($25-$35) and providing them with a server in the infrastructure ($100-$150) [27].

7

Security controls

In this chapter, we describe Avalanche's security controls based on our explorative analysis and application of the business model canvas. We use the concepts of the framework we presented in chapter 3: stealth and evasion, resilience to takedown and concealment of ownership.

7.1. Stealth and evasion

7.1.1. Proxy architecture

Avalanche used a three-layered proxy architecture, to create a front of inconspicuous servers that hid the location, identifying characteristics and other behavior of the servers that were the initiators of the actual attacks. This meant that security researchers trying to study the malicious behavior of Avalanche, based on the network connections of the different types of malware, would only be able to follow the trail until the outer perimeters of the Avalanche infrastructure: the fist layer bots. Moreover, there were many first layer bots active at the same time, which gave Avalanche's customers the opportunity to spread their connections over multiple proxies. This meant that when researchers would observe two different infections, they might connect to two seemingly completely different servers.

As a result of this, blocking strategies based on the IP and/or domain the malware contacted would have been less effective, because the servers that launched the attacks and collected data from victims evaded detection and could continue their practices even when one or multiple fist layer bots were taken down. This means that using a proxy architecture increases the resilience of takedown, since it protects the takedown of the more valuable and difficult to replace resources of the botnet.

7.2. Resilience to takedown

7.2.1. Double fast-flux

While employing proxies did make blocking strategies less effective, it was still the case that when a bot IP or domain was blocked, the bot would need to be taken out of circulation. To combat that, Avalanche employed a double fast-flux of the nameservers of domains and the IPs the nameservers replied. This fluxing happened in the connections between victims and the first layer bots. Because we do not have a server image nor network data from a first layer server, we observed the process of fluxing from the databases.

The process works as follows. The customer of Avalanche requests to register a domain via their domain registration service, for example `thisisnotamalwaredomain.com`. Avalanche checks the availability of that domain with one of its registrars, and then registers the domain and subtracts money from the customers balance. Either a generic nameserver is set (like ns1.yahoo.com), or the domain receives an already registered domain as nameserver. In this example, we assume the latter: the two nameservers registered for this domain are `ns1.thisisreallysafe.com` and `ns2.thisisreallysafe.com`. The registration process is logged in the log table of the domain registration database. The domain is then added to the domainRegistration table. In the infrastructure monitoring database, the domain is linked to the account of the customer and to a domainID. In our example this domainID is linked to four different IPs, which are first layer bots. In the domainsNS table, the domainID is linked to the ID of the nameserver. In the nameServers table we are able to observe that our domain `thisisreallysafe.com` had in total five different IPs associated to it over the course of its lifetime. These IPs were also first layer bots, but then from the dedicatedServers table.

From the point of the victim, its system would try to connect to Avalanche based on the domain name `randomstring.thisisnotamalwaredomain.com`. To retrieve the IP of this domain, the victim is referred to its nameserver: ns1.thisisreallysafe.com. Querying this nameserver, the query would be forwarded to the IP of the nameserver, which was a compromised first-layer bot. This first layer bot would most likely forward the query to its second layer server, who would query the database, to then respond with one of the four IPs connected to this domain as the location the victim should connect to. If the victim were to repeat this process, it would observe different IPs for the nameserver as different IPs that the nameserver responded with.

The goal of double fast-fluxing is to prevent other parties being able to observe patterns in domains, nameservers and IPs, to avert attempts of creating blocklists and thus the takedown of the domain or IP. Because this can also be seen as evading detection [55], we also categorize it as such.

7.2.2. Bot monitoring

Even with controls such as the proxy architecture and fast fluxing, a domain could get blocked and a server could get taken down by the hosting provider or by the original owner of the server. To be able to quickly respond to this, Avalanche implemented a variety of bot monitoring techniques. First, they monitored the availability of the bots through monitoring the proxy architecture. Every x minutes, the central c&c server would send a GET request to first and second layer servers of a test website (e.g. thisisatest.com) and observe whether this request would be correctly forwarded. For both types of servers, it kept track of this in the infrastructure monitoring server. Second, they monitored the traffic to domains the and speed of the servers extensively: the table trafficDomains has 41,451,002 rows and holds data on the observed traffic to almost 60,000 domains. The table checkSpeed verified the speeds of the second layer servers.

While this allowed Avalanche to respond swiftly to misconfigurations, blockings and takedowns of servers, it did make their whole operation more vulnerable to a whole infrastructure takedown: through wire tap monitoring of each first, second or third layer server of their infrastructure and observing the requests to the testing website, it was possible to infer the IPs, roles and locations of other servers in the proxy architecture.

7.2.3. Blacklist checking

Besides monitoring internally, Avalanche monitored its domains and IPs externally. It did that through scraping the publicly available Spamhaus block lists (SBL). The result of this monitoring can be found in multiple of their tables of the infrastructure monitoring database: the inBlacklist column in the tables bots, domains and nameServers, the notified column of the botGroups and dedicatedServers table, and the notifiedProblem column of the layerServers and webChecks tables. Interestingly enough, there were no first-layer bots with a inBlacklist value. Of the 916 domains in the infrastructure management server had 144 (or round 16%) a inBlacklist value, provings its value of this monitoring of the records (administration) of Spamhaus.

7.2.4. Backups

We observed a two-hourly process between the central C&C server at that time, and a different host. Given the automated nature of this communication (9 minutes before every odd hour), the used protocol (SSH) and the size of the sessions (on average 20,000 databytes), we hypothesized the other host was probably used as a backup for the central C&C server. The connection is initiated by this other host, but the majority of the data is send from the central C&C to this host. Additionally, there is an automated SSH process from another IP to this C&C server, smaller in databytes size but occurring more often (every 3-5 seconds), which could indicate a current syncing of settings or data. Having backup and/or fallback servers that can be easily spun up to work in case of a takedown or malfunctioning positively influences the respond time. Similar to the bot monitoring process: an automated process from and to the central C&C server can be observed when analyzing the network traffic of that server.

7.3. Concealment of ownership

7.3.1. VPN usage

One of the most important steps in the concealment of ownership, is to conceal the connections that reveal the location and thus possible identity of the administrators. To analyze this, we extracted all unique source IPs that used the SSH protocol in all the network data we had available and queried the MaxMind Anonymous

GeoIP database. We observed 3,304 unique IPs using the SSH protocol, of which 3,238 could be found in the MaxMind database. Of these, we excluded IPs that were classified by MaxMind with the anonymous status of Anonymous VPN, Hosting Provider, Tor Exit Node and Public proxy. This left 2446 IPs, of which we only selected the IPs with a Cable/DSL or Cellular connection type (excluding Corporate and Unknown), resulting in 2156 IPs. To retrieve the identify behind an IP address, a court order is needed. This is historically a bureaucratic and lengthy process, especially if the court order targets a legal entity in a different country. This was mentioned by the German prosecutor in his interview about Avalanche: *"Die Amerikaner lachen nur, wenn sie Post von einem deutschen Richter erhalten!"* [32]. Because of this, we focused only on IPs from EU countries, arguing that those IPs were realistically vulnerable to deanonymization. Ultimately, these were 339 IPs of the 3,304, meaning that around 10% of the SSH connections made to the Avalanche infrastructure were possibly not done via anonymizing services and from an IP that was vulnerable to an EU court order.

7.3.2. Registering domains with fake data

Before the implementation of GDPR policies that caused registrars to redact WHOIS data in 2018, data such as the name, phone number and even address of the person who registered a domain would be published by the registrar. When it was possible, Avalanche would activate the WHOIS-privacy settings. However, for the registrars that did not have that as a possibility, Avalanche had multiple tables filled with data to be used for these registrations. Because one of the table names contains 'CC', we find it likely that these tables hold stolen credit-card data. We queried the historical WHOIS data for some domains and saw the names and addresses that matched data from the WHOIS tables of the domain registration database. By not only using not their own data, but using a different 'persona' for each domain that they registered, they made it difficult to observe patterns in their registration and thus ownership of the domains.

7.4. Framework of Avalanche's security controls

We add the observed security controls of Avalanche to the framework, to create the overview in Table 7.1.

Table 7.1: A framework of botnet security controls from previous work and Avalanche (A)

	Blocking		**Takedown of infrastructure**	**Attribution of botmaster(s)**
	Stealth	**Evasion**	**Resilience to takedown**	**Concealment of ownership**
Technical	Obfuscation [7]	Polymorphism [18]	P2P-architecture [6, 34]	
	Deleting binary [7]	Domain Generation Algorithm [7, 50]	Domain-flux [55]	
	Obfuscate process name [7]	DNS C&C communication [7]	Reputation schemes [6]	
	Protocol manipulation [51]	P2P DTH for C&C communication [33]	Non-persistent peer list entries [6]	
	-	Proxy architecture (A)		VPN usage (A)
		Double fast-flux (A)		
Administrative			Bot monitoring (A)	Registering domains with fake data (A)
			Blacklist checking (A)	
			Backups (A)	
Physical				

8

Discussion and conclusion

Botnets and their plethora of resulting attacks continue to persist as an influential cybercrime threat. Analogous to increased outsourcing and the development of service-providing business models in all of cybercrime, botnet-as-a-service models have developed. These range from buying software to create botnets, to renting access to a botnet, or specifically purchasing services performed by a botnet (such as attacks). Previous work on botnet has mostly focussed on creating detection and mitigation strategies, often at the bot or bot-communication level. Other works have studied the effects of large-scale interventions like takedowns, but have found these to be notoriously challenging and sometimes even ineffective. Recently, a shift to a more economically focused analysis of botnets has occurred. These studies focus on the financial aspects of operating a botnet business model, in order to find directions for alternative interventions or choke points. Because this requires a behind-the-scenes look that is often unavailable, most studies have settled for surveys or case studies. In this work, we have tried to fill this gap by performing an analysis of the business model and security controls of Avalanche, through the analysis of ground truth data. The main research question of this work was as follows: **How did Avalanche ensure business continuity, given its adversarial context**?

We reflect on our findings and discuss the limitations of our data and approach in section 8.1. We then conclude by answering our research questions in section 8.2.

8.1. Discussion

8.1.1. Forensic investigation data as a source of empirical evidence

As the data for this research is not gathered through a scientific method designed by the authors, its completeness, accuracy and authenticity needed to be evaluated. The approach for this needed to be different for every data source: data in the seized databases could have been manipulated by Avalanche's administrators, while wire tap data contains all network data measured by an external party (e.g., hosting provider or internet service provider) and could thus be noisy or incomplete. Validation is especially difficult for server images: a validation of the acquisition hash will validate the copy of the image, but does not provide any proof of the validity of the data that was copied. We handled these challenges by employing multiple internal and external validation approaches, described in chapter 5. Mainly, we validated the different data sources by comparing them to one another, trying to substantiate conclusions based on findings from different sources. Another challenge of using investigation data was that the data we received was from different hosts and from different moments in time. This made it difficult to attribute findings at different moments in time to their correct origin: did we observe something we saw before that has changed, or is this something new? For example, there were plenty of domains from the domain registration database that we could not observe in any network data. But because the network data stems from different hosts over time, we do not know why we could observe some, and not others. Because of this, we limit our findings to observed patterns and abstain from making statements of concepts we have not found to be present. All in all, we believe that the provided data had the size, diversity and consistency to study Avalanche's business continuity. A more in-depth understanding of for example the fast-fluxing behavior of Avalanche may require additional analysis, for example from outside measurements.

8.1.2. Avalanche in comparison to other botnets

We studied Avalanche and its business continuity in the context of Avalanche being a botnet. In other articles and studies, however, Avalanche is described as a "bulletproof hosting service" [39], "an infrastructure platform, used to deliver malware and spam" [24], a "delivery and management platform" [48], "essentially a cloud-computing platform designed for cybercriminals" [38] or a "cybercrime-as-a-service malware attack network" [47]. While we believe Avalanche fits most of these descriptions, we do argue that Avalanches infrastructure fits the technical description of a botnet best. The first layer servers were compromised machines, 'bots', of which the weak SSH password was exploited in order to install unwanted software (malware). These first-layer bots were controlled from a command and control infrastructure and had (continuing) communication between bot and C&C. All this means the Avalanche infrastructure had all the main components that make up a botnet. Different from a more standard botnet, was Avalanche's behavior. The bots did for example not propagate and infect other machines with the same malware they had. One could also argue that Avalanche's first layer bots technically did not perform any attacks directly, but only tunneled data, especially in the case of the different other botnets that used Avalanche (like Rovnix and Andromeda). However, from the point of view of victims infected with malware from Avalanche's infrastructure, Avalanche bots were the C&C servers that their malware-infected systems made connections to, making Avalanche something akin a C&C-botnet. Additionally, our analysis of Avalanche's security controls showed overlap with known stealth, evasion, resilience and concealment techniques of botnets, strengthening the comparison. So, while Avalanche can be described as a cybercrime-as-a-service, cloud-computing, malware attack, delivery and management bulletproof hosting platform, we believe its similarity in assets and evasion techniques supports our classification of Avalanche as a botnet.

An important distinction between Avalanche and other botnets, is that of the services it supplied and thus its business model. Because the foundation of our analysis lies in the application of the concept op business continuity and the business model canvas, we believe our findings should only be compared with botnets that have a customer-serving service-oriented business model. For botnets where the customer is the administrator itself, many concepts of the business model change. In Avalanche, the key activities, channels and revenue streams are intertwined with having customers. Similarly, from deviant theory we can argue that because there are other assets to defend, the value trade-off differs for botnet administrators without customers. We therefore think the application of these findings to a non-customer-serving botnet merits a new analysis.

8.2. Conclusion

In order to answer our main research question, we first answer our two sub-questions:

1. What was Avalanche's business model?

Avalanche's business model was built on the value proposition of providing customers with a stable fast-fluxing proxy-infrastructure through which they could proxy their traffic to and from potential victims. The three key activities that Avalanche performed to fulfill this value proposition were: building and maintaining a stable fast-fluxing infrastructure, building and maintaining a domain registration system and customer service. The necessary resources for this were compromised servers (servers with compromised SSH passwords), servers for second and third layer proxy services (rented themselves at 14 different geographically spread hosting providers), malware backend servers (rented themselves, accessed by customers), domains (bought via nine different registrars), payment services (Bitcoin, WebMoney and other), software (e.g. monitoring), certificates (SSL), administrators, fake/stolen identities and VPN-services. The key partners they relied on were hosting providers, domain registrars, certificate providers, providers of stolen identifies, VPN-providers, cryptocurrency-platform and other online payment service providers. Avalanche communicated via two types of channels: online fora for marketing purposes and its own Jabber service for customer service purposes. They created relationships with customers that lasted on average 5,5 months and differentiating factors for customer segments other than types of services they bought could not be observed. There was no data to estimate any costs, but the revenue estimate based on 8 months of data showed an average revenue of $7,500 per month.

2. What were Avalanche's security controls?

We observed seven security controls, of which three were technical controls and four were administrative controls. The technical controls were the proxy architecture, double fast flux and VPN usage. The proxy architecture of Avalanche consisted of three layers: a first layer of compromised server ('bots'), a second layer of proxying servers, and a third layer of a central C&C server and a variety of malware backends. The first layers were grouped in 'bot groups' and different customers of Avalanche made use of different of these bot groups. The proxy architecture was a technical control to prevent the reconnaissance and takedown of the following tiers of servers, by shielding them with compromised and easily replaceable servers. The double fast-fluxing behavior created through cycling the IPs and nameservers of domains mainly evaded the detection of patterns in domain-IP combinations. This prevented (or at least, hindered) the takedown of these domains and IPs through detection algorithms and blacklists. VPN usage was necessary to let the administrators connect anonymously to the infrastructure. Our analysis showed that around 10% of the SSH connections to the infrastructure were done from an IP vulnerable to deanonymization through an EU court order.

The four administrative controls were bot monitoring, blacklist checking, backups and registering domains with fake data. The correct working of the infrastructure was checked automatically in multiple different ways. Through this monitoring, the administrators had an administrative safeguard to spot misconfigurations, notice takedowns and debug errors. The administration of another party, Spamhaus, was (mis)used to check for blocked domains and IPs. To be resilient if a takedown of the central C&C server were to happen, backup processes to other hosts were setup. Finally, Avalanche used fake data for registering domains, creating a layer of administrative deception.

We can now answer our main research question: **How did Avalanche ensure business continuity, given its adversarial context?**
The adversarial context in which Avalanche operated, generated different threats to its business model: Avalanche had the attention of the industry working group APWG from 2009 until 2011 and survived their coordinated actions to limit the uptime of Avalanche domains. Avalanche learned from their experience of hosting phishing sites and used this knowledge to create a service of reliably and complaint-free proxying of malicious traffic. Based on our findings, we can conclude that Avalanche responded to its adversarial environment through employing different technical and administrative controls and creating a business model to achieve technical and administrative redundancy.

Avalanche's security controls achieved business continuity through (1) evasion or detection and (2) the adaptability and quick response in case a threat did materialize. The proxy architecture and double fast fluxed worked in shielding and hiding the malware servers, and Avalanche was as a result able to serve customers with this service. The Spamhaus blocklist notified them of blocked domains and their monitoring & backup services allowed them to quickly identify incidents and resolve them through the replacement of servers if needed. Similar to their technical capabilities of switching IPs and domains, Avalanche used multiple different hosting providers and registrars, all geographically dispersed. Of the 10 key resources needed to run the business model, only the administrators themselves and their software are cumbersome to replace, making their business model also redundant to changes in partners, resources or activities.

Bibliography

[1] Greg Aaron and Rod Rasmussen. Global Phishing Survey 2009 2H. Techreport, May 2010.

[2] Greg Aaron and Rod Rasmussen. Gobal Phishing Survey 2010 1H. Techreport, 2010.

[3] Greg Aaron and Rod Rasmussen. Gobal Phishing Survey 2010 2H. Techreport, 2010.

[4] Ahmad Al-Nawasrah, Ammar Ali Almomani, Samer Atawneh, and Mohammad Alauthman. A Survey of Fast Flux Botnet Detection With Fast Flux Cloud Computing:. *International Journal of Cloud Applications and Computing*, 10(3):17–53, July 2020. ISSN 2156-1834, 2156-1826. doi: 10.4018/IJCAC.2020070102. https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJCAC.2020070102.

[5] Syed Taha Ali, Patrick McCorry, Peter Hyun-Jeen Lee, and Feng Hao. ZombieCoin 2.0: Managing next-generation botnets using Bitcoin. *International Journal of Information Security*, 17(4):411–422, August 2018. ISSN 1615-5262, 1615-5270. doi: 10.1007/s10207-017-0379-8. http://link.springer.com/10.1007/s10207-017-0379-8.

[6] Dennis Andriesse, Christian Rossow, and Herbert Bos. Reliable Recon in Adversarial Peer-to-Peer Botnets. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference - IMC '15*, pages 129–140, Tokyo, Japan, 2015. ACM Press. ISBN 978-1-4503-3848-6. doi: 10.1145/2815675.2815682. http://dl.acm.org/citation.cfm?doid=2815675.2815682.

[7] Manos Antonakakis, Tim April, Michael Bailey, Matthew Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. page 19, 2017.

[8] Arkime. Arkime. https://github.com/arkime/arkime, 2022.

[9] Arkime. Changelog. https://github.com/arkime/arkime/blob/main/CHANGELOG, 2022.

[10] Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu, and Manish Karir. A Survey of Botnet Technology and Defenses. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 299–304, Washington, DC, USA, March 2009. IEEE. ISBN 978-0-7695-3568-5. doi: 10.1109/CATCH.2009.40. http://ieeexplore.ieee.org/document/4804459/.

[11] Ben Baker and Alex Chiu. Threat Spotlight: Rombertik – Gazing Past the Smoke, Mirrors, and Trapdoors. https://blogs.cisco.com/security/talos/rombertik, 2015.

[12] Paul Barford and Vinod Yegneswaran. An Inside Look at Botnets. In *Malware Analysis*. 2007.

[13] Anchit Bijalwan, Meenakshi Thapaliyal, Emmanuel S Piili, and R. C. Joshi. Survey and Research Challenges of Botnet Forensics. *International Journal of Computer Applications*, 75(7):43–50, August 2013. ISSN 09758887. doi: 10.5120/13127-0483. http://research.ijcaonline.org/volume75/number7/pxc3890483.pdf.

[14] Paul Black, Iqbal Gondal, and Robert Layton. A survey of similarities in banking malware behaviours. *Computers & Security*, 77:756–772, August 2018. ISSN 01674048. doi: 10.1016/j.cose.2017.09.013. https://linkinghub.elsevier.com/retrieve/pii/S016740481730202X.

[15] Giovanni Bottazzi and Gianluigi Me. The Botnet Revenue Model. In *Proceedings of the 7th International Conference on Security of Information and Networks - SIN '14*, pages 459–465, Glasgow, Scotland, UK, 2014. ACM Press. ISBN 978-1-4503-3033-6. doi: 10.1145/2659651.2659673. http://dl.acm.org/citation.cfm?doid=2659651.2659673.

[16] CentOS. CentOS Installation Guide - Kickstart Installations. https://docs.centos.org/en-US/centos/install-guide/Kickstart2/, 2022.

[17] cgsecurity.org. TestDisk. https://github.com/cgsecurity/testdisk, 2022.

[18] Chia Yuan Cho, Juan Caballero, Chris Grier, Vern Paxson, and Dawn Song. Insights from the Inside: A View of Botnet Management from Infiltration. 2009.

[19] Ben Collier, Daniel R Thomas, Richard Clayton, Alice Hutchings, and Yi Ting Chua. Nfluence, infrastructure, and recentering cybercrime policing: Evaluating emerging approaches to online law enforcement through a market for cybercrime services. *Policing and Society*, 2022.

[20] Kevin Conlan, Ibrahim Baggili, and Frank Breitinger. Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy. *Digital Investigation*, 18:S66–S75, August 2016. ISSN 17422876. doi: 10.1016/j.diin.2016.04.006. https://linkinghub.elsevier.com/retrieve/pii/S1742287616300378.

[21] Alejandro Cuevas, Fieke Miedema, Kyle Soska, Nicolas Christin, and Rolf van Wegberg. Measurement by Proxy: On the Accuracy of Online Marketplace Measurements. 2022.

[22] Tooska Dargahi, Ali Dehghantanha, Pooneh Nikkhah Bahrami, Mauro Conti, Giuseppe Bianchi, and Loris Benedetto. A Cyber-Kill-Chain based taxonomy of crypto-ransomware features. *Journal of Computer Virology and Hacking Techniques*, 15(4):277–305, December 2019. ISSN 2263-8733. doi: 10.1007/s11416-019-00338-7. http://link.springer.com/10.1007/s11416-019-00338-7.

[23] George Dvorsky. Storm Botnet storms the Net. https://ieet.org/index.php/IEET2/more/dvorsky20070927/, June 2019.

[24] Europol. 'Avalanche' network dismantled in international cyber operation. https://www.europol.europa.eu/newsroom/news/%E2%80%98avalanche%E2%80%99-network-dismantled-in-international-cyber-operation, December 2016.

[25] Europol. Operation Avalanche infographic, 2016.

[26] Europol. GOZNYM MALWARE: CYBERCRIMINAL NETWORK DISMANTLED IN INTERNATIONAL OPERATION. https://www.europol.europa.eu/media-press/newsroom/news/goznym-malware-cybercriminal-network-dismantled-in-international-operation, 2019.

[27] Aaron O. Francis. DECLARATION OF SPECIAL AGENT AARON 0 . FRANCIS IN SUPPORT OF APPLICATION FOR AN EMERGENCY TEMPORARY RESTRAINING ORDER AND ORDER TO SHOW CAUSE RE PRELIMINARY INJUNCTION. https://www.justice.gov/archives/opa/page/file/915231/download, 2016.

[28] FreeBSD. FreeBSD System Manager's Manual. https://www.freebsd.org/cgi/man.cgi?query=tzsetup&sektion=8&format=html, 2022.

[29] Dimitrios Georgoulias, Jens Myrup Pedersen, Morten Falch, and Emmanouil Vasilomanolakis. Botnet businessmodels, takedown attempts, and the darkweb market: A survey. *ACM Computing Surveys*, page 1122445.1122456, March 2022. ISSN 0360-0300, 1557-7341. doi: 10.1145/1122445.1122456. https://dl.acm.org/doi/10.1145/1122445.1122456.

[30] Murat Gul and Emin Kugu. A survey on anti-forensics techniques. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–6, Malatya, September 2017. IEEE. ISBN 978-1-5386-1880-6. doi: 10.1109/IDAP.2017.8090341. http://ieeexplore.ieee.org/document/8090341/.

[31] Nicole M. Hands, Baijian Yang, and Raymond A. Hansen. A Study on Botnets Utilizing DNS. In *Proceedings of the 4th Annual ACM Conference on Research in Information Technology - RIIT '15*, pages 23–28, Chicago, Illinois, USA, 2015. ACM Press. ISBN 978-1-4503-3836-3. doi: 10.1145/2808062.2808070. http://dl.acm.org/citation.cfm?doid=2808062.2808070.

[32] Lukas Heiny. Die Jagd auf Avalanche. https://www.stern.de/digital/online/cyberkriminalitaet--die-jagd-auf-avalanche-7338648.html, 2017.

[33] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet. In *Proceedings 2019 Network and Distributed System Security Symposium*, San Diego, CA, 2019. Internet Society. ISBN 978-1-891562-55-6. doi: 10.14722/ndss.2019.23488. https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_02B-3_Herwig_paper.pdf.

[34] Thorsten Holz, Moritz Steiner, Frederic Dahl, Ernst Biersack, and Felix Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. page 9, 2008.

[35] Colin C. Ife, Yun Shen, Steven J. Murdoch, and Gianluca Stringhini. Marked for Disruption: Tracing the Evolution of Malware Delivery Operations Targeted for Takedown. In *24th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 340–353. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 978-1-4503-9058-3. https://doi.org/10.1145/3471621.3471844.

[36] Anu Jain and Gurpal Singh Chhabra. Anti-forensics techniques: An analytical review. In *2014 Seventh International Conference on Contemporary Computing (IC3)*, pages 412–418, Noida, India, August 2014. IEEE. ISBN 978-1-4799-5173-4 978-1-4799-5172-7 978-1-4799-5171-0. doi: 10.1109/IC3.2014.6897209. https://ieeexplore.ieee.org/document/6897209.

[37] Sheharbano Khattak, Naurin Rasheed Ramay, Kamran Riaz Khan, Affan A. Syed, and Syed Ali Khayam. A Taxonomy of Botnet Behavior, Detection, and Defense. *IEEE Communications Surveys & Tutorials*, 16(2):898–924, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2013.091213.00134. http://ieeexplore.ieee.org/document/6616686/.

[38] Jeremy Kirk. Police Shut Down Global Cybercriminal Fraud Service. https://www.databreachtoday.com/police-shut-down-global-cybercriminal-fraud-service-a-9572, December 2016.

[39] Victor Le Pochat, Tim Van hamme, Sourena Maroofi, Tom Van Goethem, Davy Preuveneers, Andrzej Duda, Wouter Joosen, and Maciej Korczynski. A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints. In *Proceedings 2020 Network and Distributed System Security Symposium*, San Diego, CA, 2020. Internet Society. ISBN 978-1-891562-61-7. doi: 10.14722/ndss.2020.24161. https://www.ndss-symposium.org/wp-content/uploads/2020/02/24161.pdf.

[40] Steve Mansfield-Devine. Battle of the botnets. *Network Security*, 2010(5):4–6, May 2010. ISSN 13534858. doi: 10.1016/S1353-4858(10)70054-4. https://linkinghub.elsevier.com/retrieve/pii/S1353485810700544.

[41] Artur Marzano, David Alexander, Osvaldo Fonseca, Elverton Fazzion, Cristine Hoepers, Klaus Steding-Jessen, Marcelo H. P. C. Chaves, Italo Cunha, Dorgival Guedes, and Wagner Jr. Meira. The Evolution of Bashlite and Mirai IoT Botnets. https://homepages.dcc.ufmg.br/cunha/papers/marzano18iscc-botnets.pdf, 2018.

[42] Ian P McCarthy, Mark Collard, and Michael Johnson. Adaptive organizational resilience: An evolutionary perspective. *Current Opinion in Environmental Sustainability*, 28:33–40, October 2017. ISSN 18773435. doi: 10.1016/j.cosust.2017.07.005. https://linkinghub.elsevier.com/retrieve/pii/S1877343517300283.

[43] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, David Dagon, and Wenke Lee. Beheading hydras: Performing effective botnet takedowns. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13*, pages 121–132, Berlin, Germany, 2013. ACM Press. ISBN 978-1-4503-2477-9. doi: 10.1145/2508859.2516749. http://dl.acm.org/citation.cfm?doid=2508859.2516749.

[44] Arman Noroozian, Jan Koenders, Eelco van Veldhuizen, Carlos H Ganan, Sumayah Alrwais, Damon McCoy, and Michel van Eeten. Platforms in Everything: Analyzing Ground-Truth Data on the Anatomy and Economics of Bullet-Proof Hosting. page 17, 2019.

[45] Alexander Osterwalder, Yves Pigneur, and Tim Clark. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers.* Wiley, Hoboken, NJ, 2010. ISBN 978-0-470-87641-1.

[46] C. G. J. Putman, Abhishta, and Lambert J. M. Nieuwenhuis. Business Model of a Botnet. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 441–445, March 2018. doi: 10.1109/PDP2018.2018.00077. http://arxiv.org/abs/1804.10848.

[47] Mathew Schwartz. FBI and Europol Disrupt GozNym Malware Attack Network. https://www.bankinfosecurity.com/fbi-europol-disrupt-goznym-malware-attack-network-a-12493?highlight=true, May 2019.

[48] ShadowServer. Avalanche - Law Enforcement Take Down. https://www.shadowserver.org/news/avalanche/, December 2016.

[49] ShadowServer. Avalanche year two, this time with Andromeda. https://www.shadowserver.org/news/avalanche-year-two-this-time-with-andromeda/, December 2017.

[50] Seungwon Shin and Guofei Gu. Conficker and beyond: A large-scale empirical study. In *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, page 151, Austin, Texas, 2010. ACM Press. ISBN 978-1-4503-0133-6. doi: 10.1145/1920261.1920285. http://portal.acm.org/citation.cfm?doid=1920261.1920285.

[51] Sérgio S.C. Silva, Rodrigo M.P. Silva, Raquel C.G. Pinto, and Ronaldo M. Salles. Botnets: A survey. *Computer Networks*, 57(2):378–403, 2013. ISSN 13891286. doi: 10.1016/j.comnet.2012.07.021. https://linkinghub.elsevier.com/retrieve/pii/S1389128612003568.

[52] Manmeet Singh, Maninder Singh, and Sanmeet Kaur. Issues and challenges in DNS based botnet detection: A survey. *Computers & Security*, 86:28–52, September 2019. ISSN 01674048. doi: 10.1016/j.cose.2019.05.019. https://linkinghub.elsevier.com/retrieve/pii/S0167404819301117.

[53] Somayeh Soltani, Seyed Amin Hosseini Seno, Maryam Nezhadkamali, and Rahmat Budirato. A Survey On Real World Botnets And Detection Mechanisms. 2014.

[54] Jan Spooren, Davy Preuveneers, Lieven Desmet, Peter Janssen, and Wouter Joosen. Detection of algorithmically generated domain names used by botnets: A dual arms race. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1916–1923, Limassol Cyprus, April 2019. ACM. ISBN 978-1-4503-5933-7. doi: 10.1145/3297280.3297467. https://dl.acm.org/doi/10.1145/3297280.3297467.

[55] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 635–647, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-894-0. doi: 10.1145/1653662.1653738. http://doi.acm.org/10.1145/1653662.1653738.

[56] Tsuyoshi Taniguchi, Harm Griffioen, and Christian Doerr. Analysis and Takeover of the Bitcoin-Coordinated Pony Malware. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 916–930, Virtual Event Hong Kong, May 2021. ACM. ISBN 978-1-4503-8287-8. doi: 10.1145/3433210.3437520. https://dl.acm.org/doi/10.1145/3433210.3437520.

[57] Simon Nam Thanh Vu, Mads Stege, Peter Issam El-Habr, Jesper Bang, and Nicola Dragoni. A Survey on Botnets: Incentives, Evolution, Detection and Current Trends. *Future Internet*, 13(8):198, July 2021. ISSN 1999-5903. doi: 10.3390/fi13080198. https://www.mdpi.com/1999-5903/13/8/198.

[58] Secureworks Counter Threat Unit. Evolution of the GOLD EVERGREEN Threat Group. https://www.secureworks.com/research/evolution-of-the-gold-evergreen-threat-group, 2017.

[59] Jochem van de Laarschot and Rolf van Wegberg. Risky Business? Investigating the Security Practices of Vendors on an Online Anonymous Market using Ground-Truth Data. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[60] Erik Van De Sandt. *Deviant Security: The Technical Computer Security Practices of Cyber Criminals.* PhD thesis, University of Bristol, 2019.

[61] Rolf Van Wegberg and Thijmen Verburgh. Lost in the Dream? Measuring the effects of Operation Bayonet on vendors migrating to Dream Market. In *Proceedings of the Evolution of the Darknet Workshop*, 2018.

[62] Rolf van Wegberg, Fieke Miedema, Ugur Akyazi, Arman Noroozian, Bram Klievink, and Michel van Eeten. Go See a Specialist? Predicting Cybercrime Sales on Online Anonymous Markets from Vendor and Product Characteristics. In *Proceedings of The Web Conference 2020*, pages 816–826, Taipei Taiwan, April 2020. ACM. ISBN 978-1-4503-7023-3. doi: 10.1145/3366423.3380162. https://dl.acm.org/doi/10.1145/3366423.3380162.

[63] Gernot Vormayr, Tanja Zseby, and Joachim Fabini. Botnet Communication Patterns. *IEEE Communications Surveys & Tutorials*, 19(4):2768–2796, 2017. ISSN 1553-877X. doi: 10.1109/COMST.2017.2749442. http://ieeexplore.ieee.org/document/8026031/.

[64] Robert Wainwright and Frank J Cilluffo. Responding to Cybercrime at Scale: Operation Avalanche – A Case Study. 2017.

[65] Wireshark. TCP Analysis. https://www.wireshark.org/docs/wsug_html_chunked/ChAdvTCPAnalysis.html, 2022.

[66] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, and Dan Garant. Botnet detection based on traffic behavior analysis and flow intervals. *Computers & Security*, 39:2–16, November 2013. ISSN 01674048. doi: 10.1016/j.cose.2013.04.007. https://linkinghub.elsevier.com/retrieve/pii/S0167404813000837.

A

Descriptives of tables and columns of the infrastructure monitoring database

Table A.1: High-level descriptives of balanceChanges table (rows = 2,236)

Column	Type	Mean	Min	Max	NumUnique
idUser	Integer	18.69	2	58	49
idAdmin	Integer	0.35	0	2	3
action	Integer	0.96	0	5	6
domain	String	-	-	-	1040
amount	Float	2.46	-150.0	2000.0	81
datetime	Datetime	-	01-06-2015	04-04-2016	-
idTransaction		0.04	0	10	7

Table A.2: High-level descriptives of dns table (rows = 8)

Column	Type	Mean	Min	Max	NumUnique
idDns	Integer	12.38	8	16	8
idDomain	Integer	2371.0	2158	2546	4
recordType	String	-	-	-	3
ip	IP-address	-	-	-	5
mxPriorityValue	8.75	0	10	2	
datetime	Datetime	-	14-08-2015	09-10-2015	-

Table A.3: High-level descriptives of domainsRegistration table (rows = 3,633)

Column	Type	Mean	Min	Max	NumUnique
idDomain	Integer	1799.24	1	3540	3513
datetimeInserted	Datetime	-	29-05-2015	04-04-2016	-
domain	String	-	-	-	3460
idUser	Integer	9.14	1	58	45
dateBeg	Date	-	2010-03-30	2016-04-04	-
dateExp	Date	-	2011-03-30	2017-04-04	-
idRegistrar	Integer	2.41	1	9	9
adminComment	String	-	-	-	47
nameservers	String	-	-	-	531
NIChandle	String	-	-	-	414
isHidden	Binary	0.19	0	1	2
isLocked	Binary	0.68	0	1	2
datetimeLockedChanged	Datetime	-	29-07-2015	04-04-2016	-

Table A.4: High-level descriptives of firstNames table (rows = 478)

Column	Type	Mean	Min	Max	NumUnique
idName	Integer	239.5	1	478	478
firstName	String	-	-	-	472
gender	Binary	1.64	1	2	2

Table A.5: High-level descriptives of logsRegistrations table (rows = 12,171)

Column	Type	Mean	Min	Max	NumUnique
idLog	Integer	6771.69	35	12897	12171
idUser	Integer	12.52	0	59	52
action	Integer	27.13	0	102	40
isError	Binary	0.07	0	1	2
comment	String	-	-	-	3904
domain	String	-	-	-	1935
request	String	-	-	-	8958
response	String	-	-	-	8366
datetime	Datetime	-	29-05-2015	05-04-2016	-
registrar	Integer	3.49	0	9	10

Table A.6: High-level descriptives of ownNameservers table (rows = 670)

Column	Type	Mean	Min	Max	NumUnique
idNS	Integer	308.26	1	636	630
idDomain	Integer	3056.47	2468	3536	311
host	String	-	-	-	10
ip	IP-Address	-	-	-	332

Table A.7: High-level descriptives of ssls table (rows = 36)

Column	Type	Mean	Min	Max	NumUnique
idSSL	Integer	21.42	2	43	36
datetimeInsert	Datetime	-	11-06-2015	23-02-2016	-
domain	String	-	-	-	36
idUser	Integer	1.86	1	2	2
idRegistrar	Integer	0.44	0	4	2
regName	String	-	-	-	2
adminComment	String	-	-	-	26
email	Email-Address	-	-	-	36
sslCsr	String	-	-	-	36
sslKey	String	-	-	-	36
sslCrt	String	-	-	-	36

Table A.8: High-level descriptives of transactionsBtc table (rows = 16)

Column	Type	Mean	Min	Max	NumUnique
idTransaction	Integer	4.5	1	8	8
idUser	Integer	25.62	1	58	6
datetimeAdd	Datetime	-	04-03-2016	30-03-2016	-
amount	Float	437.55	-0.27000001	5747.92274689	10
recipientAddress	String	-	-	-	11
hash	String	-	-	-	15

Table A.9: High-level descriptives of transactionsWebmoney table (rows = 20)

Column	Type	Mean	Min	Max	NumUnique
idTransaction	Integer	5.5	1	10	10
datetimeAdd	Datetime	-	23-03-2016	04-04-2016	-
amount	Float	5746531.7	0.0	68921216.0	12
paymentComment	String	-	-	-	10
transactionIdentifier	Integer	-	-	-	19

Table A.10: High-level descriptives of userCommentWebmoney table (rows = 7)

Column	Type	Mean	Min	Max	NumUnique
idComment	Integer	4.0	1	7	7
userComment	String	-	-	-	7
datetimeAdd	Datetime	-	23-03-2016	04-04-2016	-
idUser	Integer	46.43	38	58	5

Table A.11: High-level descriptives of users table (rows = 143)

Column	Type	Mean	Min	Max	NumUnique
idUser	Integer	29.6	1	59	52
login	String	-	-	-	52
passwdHash	String	-	-	-	52
balance	Float	63.18	-125.0	1934.0	31
minBalance	Float	-35.26	-350.0	0.0	15
enabled	Binary	0.98	0	1	2
adminComment	String	-	-	-	49
datetimeRegister	Datetime	-	02-01-2015	04-04-2016	-
tariff	object	69702813.29	-2092748416	2085736576	56
showUrl	Interger	-46.28	-128	2	4
btcAddress	String	-	-	-	54

Table A.12: High-level descriptives of whoisRandom table (rows = 7,588)

Column	Type	Mean	Min	Max	NumUnique
idWhois	Integer	2404.72	118	5263	5146
fullName	String	-	-	-	6095
firstName	String	-	-	-	577
lastName	String	-	-	-	4236
address	String	-	-	-	4789
city	String	-	-	-	3308
state	String	-	-	-	50
country	String	-	-	-	2
postalCode	String	-	-	-	4019
phone	String	-	-	-	4795

Table A.13: High-level descriptives of whoisRandomUsed table (rows = 2,786)

Column	Type	Mean	Min	Max	NumUnique
idWhois	Integer	1621.63	216	3023	2786
fullName	String	-	-	-	1410
firstName	String	-	-	-	144
lastName	String	-	-	-	1054
address	String	-	-	-	112
city	String	-	-	-	55
state	String	-	-	-	4
country	String	-	-	-	1
postalCode	String	-	-	-	83
phone	String	-	-	-	112

Table A.14: High-level descriptives of whoisCC table (rows = 12)

Column	Type	Mean	Min	Max	NumUnique
idWhois	Integer	473.08	468	478	11
fullName	String	-	-	-	11
firstName	String	-	-	-	11
lastName	String	-	-	-	11
address	String	-	-	-	11
city	String	-	-	-	10
state	String	-	-	-	8
country	String	-	-	-	2
postalCode	String	-	-	-	11
phone	String	-	-	-	11

Table A.15: High-level descriptives of words table (rows = 16,922)

Column	Type	Mean	Min	Max	NumUnique
idWord	Integer	8461.5	1	16922	16922
word	String	-	-	-	16260

B

Descriptives of tables and columns in the domain registration database

Table B.1: High-level descriptives of clientLogs table (rows = 2,719)

Column	Type	Mean	Min	Max	NumUnique
idClient	Integer	211.88	8	264	39
action	Integer	1.52	0	3	4
domain	String	-	-	-	2273
datetime	Datetime	-	07-03-2015	29-11-2016	-

Table B.2: High-level descriptives of logs table (rows = 8,108)

Column	Type	Mean	Min	Max	NumUnique
idUser	Integer	1.59	1	3	3
action	Integer	3.51	1	6	6
domainOrIP	String	-	-	-	6529
datetime	Datetime	-	08-08-2015	29-11-2016	-

Table B.3: High-level descriptives of bots table (rows = 3,324)

Column	Type	Mean	Min	Max	NumUnique
idBot	Integer	122,333.03	121,032	122,861	939
botGroupID	Integer	253.55	100	462	56
inBlacklist	Binary	0.0	0	0	1
loginDetails	String	-	-	-	981
ip	IP-address	-	-	-	936
proxyIP	IP-address	-	-	-	20
ipAlias	IP-address	-	-	-	65
addDate	Datetime	-	18-07-2015	29-11-2016	-
checkDatetime	Datetime	-	23-03-2016	30-11-2016	-
lastCheckDatetime	Datetime	-	14-03-2016	30-11-2016	-
nginxConfigUpdateDatetime	Datetime	-	05-02-2016	29-11-2016	-
mainScriptUpdateDatetime	Datetime	-	06-02-2016	30-11-2016	-

Table B.4: High-level descriptives of botGroups table (rows = 160)

Column	Type	Mean	Min	Max	NumUnique
idBotGroup	Integer	307.66	100	462	61
idLayer	Integer	3.46	1	6	5
countYellowAlert	Integer	0.18	0	2	3
countRedAlert	Integer	-0.09	-1	1	3
botsEnough	Integer	0.63	-1	1	3
notified	Binary	0.34	0	1	2

Table B.5: High-level descriptives of countryIP table (rows = 86,016)

Column	Type	Mean	Min	Max	NumUnique
startIP	IP-address	-	-	-	86,016
endIP	IP-address	-	-	-	86,016
countryCode	String	-	-	-	230

Table B.6: High-level descriptives of botsNS table (rows = 8)

Column	Type	Mean	Min	Max	NumUnique
idBotNS	Integer	49.0	22	58	8
working	Binary	0.38	0	1	2
ip	IP-address	-	-	-	8
addDatetime	Datetime	-	03-03-2014	14-07-2014	-
checkDateTime	Datetime	-	29-07-2014	29-07-2014	-
lastCheckDatetime	Datetime	-	20-07-2014	29-07-2014	-

Table B.7: High-level descriptives of settings table (rows = 1)

Column	Type	Mean	Min	Max	NumUnique
intervalRefresh	Integer	30.0	30	30	1
intervalRefreshSpeed	Integer	5760.0	5760	5760	1
intervalRefreshOnline	Integer	15.0	15	15	1
remotePort	Integer	80.0	80	80	1
localPort	Integer	80.0	80	80	1
minSpeedRedir	Integer	5.0	5	5	1
configURL	URL	-	-	-	1

Table B.8: High-level descriptives of checkSpeedServers table (rows = 16)

Column	Type	Mean	Min	Max	NumUnique
idServer	Integer	44.56	23	53	16
alertSpeed	Integer	100.0	100	100	1
active	Binary	1.0	1	1	1
ip	IP-address	-	-	-	16
checkURL	URL	-	-	-	16

Table B.9: High-level descriptives of checkSpeedSpeeds table (rows = 109,156)

Column	Type	Mean	Min	Max	NumUnique
idServer	Integer	42.7	23	53	18
speed	Integer	903.79	0	1537	247
datetime	Datetime	-	06-01-2016	30-11-2016	-

Table B.10: High-level descriptives of dedicatedServers table (rows = 1,976)

Column	Type	Mean	Min	Max	NumUnique
idServer	Integer	4048.23	2698	4561	526
works	Binary	1.22	0	2	3
needDNS	Binary	1.0	1	1	1
worksDNS	Binary	1.25	0	2	3
gotDNS	Binary	0.1	0	1	2
needMail	Binary	0.0	0	0	1
worksMail	Binary	0.0	0	0	1
inArchive	Binary	0.32	0	1	2
notified	Binary	0.09	0	1	2
ip	IP-address	-	-	-	524
ipAlias	IP-address	-	-	-	20
proxyIP	IP-address	-	-	-	16
loginDetails	String	-	-	-	733
addDatetime	Datetime	-	27-06-2015	29-11-2016	-
DNSconfigUpdate	Datetime	-	16-02-2016	30-11-2016	-
mainScriptUpdate	Datetime	-	16-02-2016	30-11-2016	-

Table B.11: High-level descriptives of domainIPs table (rows = 3,462)

Column	Type	Mean	Min	Max	NumUnique
idDomain	Integer	6330.13	5637	6924	757
ipServer	IP-address	-	-	-	357

Table B.12: High-level descriptives of domainIPsWork table (rows = 3,046)

Column	Type	Mean	Min	Max	NumUnique
idDomain	Integer	6351.48	5637	6924	757
ipServer	IP-address	-	-	-	358

Table B.13: High-level descriptives of domains table (rows = 1,744)

Column	Type	Mean	Min	Max	NumUnique
idDomainOrAccount	Integer	6204.23	598	6924	916
domainOrAccount	Integer	-	-	-	891
idMain	Integer	5596.44	598	6918	59
idClient	Integer	212.24	8	264	43
numBots	Integer	4.97	0	10	3
idBotGroup	Integer	84.35	0	461	43
idBotGroupReserve	Integer	129.63	0	462	28
maxDomainsAccount	Integer	3.21	0	35	12
needChange	Binary	0.27	0	1	2
isMain	Binary	0.18	0	1	2
inArchive	Binary	0.27	0	1	2
work	Binary	1.39	0	2	3
inBlacklist	Binary	2.86	0	3	4
doRedir	Binary	0.17	0	1	2
useOwnNS	Binary	0.16	0	1	2
ftpLogin	String	-	-	-	60
ftpPassword	String	-	-	-	67
homedir	String	-	-	-	891
info	String	-	-	-	164
redirIP	IP-address	-	-	-	76
createDatetime	Datetime	-	03-04-2012	29-11-2016	-
expireDatetime	Datetime	-	06-02-2015	23-07-2115	-
blacklistDatetime	Datetime	-	13-11-2014	29-11-2016	-
insertDatetime	Datetime	-	16-07-2015	29-11-2016	-

Table B.14: High-level descriptives of domainsNS table (rows = 10,960)

Column	Type	Mean	Min	Max	NumUnique
idDomain	Integer	3233.46	0	6924	6112
idNS	Integer	1424.05	1	2531	1062

Table B.15: High-level descriptives of nameServers table (rows = 758)

Column	Type	Mean	Min	Max	NumUnique
idNS	Integer	2401.86	2068	2535	173
needChange	Binary	1.0	1	1	1
work	Binary	0.91	0	2	3
inBlacklist	Binary	2.93	2	3	2
notified	Binary	0.02	0	1	2
ownNSforDomain	Binary	0.11	0	1	2
domain	String	-	-	-	172
registrar	String	-	-	-	24
commentForUser	String	-	-	-	3
ns1	IP-address	-	-	-	239
ns2	IP-address	-	-	-	242
lastChangeDatetime	Datetime -	22-02-2016	30-11-2016	-	
blacklistDatetime	Datetime	-	20-01-2016	29-11-2016	-
addDatetime	Datetime	-	11-08-2015	29-11-2016	-

Table B.16: High-level descriptives of nameServersAccount table (rows = 172)

Column	Type	Mean	Min	Max	NumUnique
idNS	Integer	2427.87	2068	2531	159
idDomainOrAccount	Integer	5318.49	0	6918	49

Table B.17: High-level descriptives of nameServersDeleted table (rows = 612)

Column	Type	Mean	Min	Max	NumUnique
idNS	Integer	2352.28	2040	2659	612
domain	String	-	-	-	606
deleteDatetime	Datetime	-	28-03-2015	25-11-2016	-

Table B.18: High-level descriptives of layerServers table (rows = 147)

Column	Type	Mean	Min	Max	NumUnique
idLayerServer	Integer	16.63	5	26	16
isActive	Binary	1.0	1	1	1
works	Binary	1.03	1	2	2
notifiedProblem	Binary	0.02	0	1	2
ip	IP-address	-	-	-	16
comment	String	-	-	-	3
lastChangeDatetime	Datetime	-	11-03-2016	30-11-2016	-

Table B.19: High-level descriptives of trafficBots table (rows = 36,861)

Column	Type	Mean	Min	Max	NumUnique
requestCount	Integer	2.02	1	49	27
requestLength	Integer	0.41	0	13	13
sentLength	Integer	560.29	0	15347	2307
sentLengthBody	Integer	559.89	0	15336	2288
trafficType	Binary	2.0	2	2	1
ip	IP-address	-	-	-	9
logDatetime	Datetime	-	16-03-2016	04-05-2016	-

Table B.20: High-level descriptives of trafficDomains table (rows = 41,451,002)

Column	Type	Mean	Min	Max	NumUnique
idDomainOrAccount	Integer	3010.64	0	6918	45
requestCount	Integer	35.19	1	65778	6175
requestLength	Integer	70.83	0	192240	20846
sentLength	Integer	720.31	0	8754200	106613
sentLengthBody	Integer	714.44	0	8754200	104679
idLayerServer	Integer	16.2	0	26	17
trafficType	Binary	0.14	0	2	3
domainOrIP	String	-	-	-	59849
logDatetime	Datetime	-	16-03-2016	30-11-2016	-

Table B.21: High-level descriptives of webChecks table (rows = 33)

Column	**Type**	**Mean**	**Min**	**Max**	**NumUnique**
works	Binary	1.0	1	1	1
notifiedProblem	Binary	0.0	0	0	1
url	URL	-	-	-	5
insertDatetime	Datetime	-	02-12-2015	05-11-2016	-
lastChangeDatetime	Datetime	-	06-03-2016	29-11-2016	-