Analysis and Takeover of the Bitcoin-Coordinated Pony Malware

Tsuyoshi Taniguchi
Fujitsu System Integration Laboratories LTD.
Tokyo, Japan
taniguchi.tsuyo@fujitsu.com

Harm Griffioen
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany
harm.griffioen@hpi.de

Christian Doerr
Hasso Plattner Institute for Digital Engineering
University of Potsdam, Germany
christian.doerr@hpi.de

ABSTRACT

Malware, like all products and services, evolves with bursts of innovation. These advances usually happen whenever security controls get "good enough" to significantly impact the revenue stream of malicious actors, and in the past we have seen the malware ecosystem to adopt concepts such as code obfuscation, polymorphism, domain-generation algorithms (DGAs), as well as virtual machine and sandbox evasion whenever defenses were able to perform consistent and pervasive suppression of these threats.

The latest innovation step addresses one of the main Archilles' heels in malware operations: the resilient addressing of the command & control (C&C) server. As domain blacklisting and DGA reversing have become mature security practices, malware authors are now turning to the Bitcoin blockchain, and use its resilient design principle to disseminate control information that cannot be removed by defenders. In this paper, we report on the adoption of Bitcoin-based C&C addressing in the Pony malware, one of the most widely occurring malware platforms on Windows. We forensically analyze the blockchain-based C&C mechanism of the Pony malware, track the malicious operations over a period of 12 months, and report how the adversaries experimented and optimized their deployment over time. We identify a security flaw in the C&C addressing, which is used to perform a takeover of the malware's loading mechanism to quantify the volume and origin of the incoming infections.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

malware, cyber threat intelligence, blockchain

ACM Reference Format:
Tsuyoshi Taniguchi, Harm Griffioen, and Christian Doerr. 2021. Analysis and Takeover of the Bitcoin-Coordinated Pony Malware. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3433210.3437520

ASIA CCS '21, June 7–11, 2021, Hong Kong, Hong Kong

ACM ISBN 978-1-4503-8287-8/21/06…$15.00
https://doi.org/10.1145/3433210.3437520

1 INTRODUCTION

Having freshly been infected with malware, a victim PC reaches out to the bot master for instructions and additional downloads. While in the early years, the address of this so-called command & control (C&C) server was typically hardcoded in the malware binary itself, this mechanism turned out to be ineffective for the malicious actors to maintain control over a botnet, as defenders could simply blacklist the IP address or domain name associated with known C&C servers and thereby render the entire botnet inoperable. In response to this, malware owners introduced the concept of domain-generation algorithms (DGAs), which programmatically compute an Internet domain name that is temporarily used to make contact with the C&C server. By creating DGAs that generate thousands of candidate domains per day, the actors hence try to disturb the economic balance between attack and defense: while a defender now needs to buy, blacklist or take-down all of the output domains that the DGA could generate on a given day to fully suppress the malware during that interval, the bot master only needs to register one domain name to maintain control over the installation.

DGAs have however the major disadvantage that the algorithm evidently has to be distributed to the victims as part of the malware itself. By reverse engineering the binary, security researchers can hence extract the algorithm and precompute all future candidate domains to disseminate them in blacklists like DGArchive[1]. This entire predictability of the algorithm allows for efficient suppression of malware C&C communication today, for example by loading the current list of domain names into intrusion detection systems or null-routing the DGA domains in an organization's DNS servers. In result, reverse-engineering of malware and blockage of centralized C&C communication is considered a standard operating procedure in industry today.

As all existing mitigation operations for C&C channels rely on the fact that defenders can study and prepare in advance, the Cerber botnet owners introduced the next step in the sequence of malware evolution in 2017: C&C via the Bitcoin blockchain [26]. Instead of shipping a DGA with their malware, the current address of the C&C server would be announced through a transaction in the Bitcoin blockchain. Infected clients would monitor the blockchain for payments, identify the bot master's transaction, and contact the address recorded in the payment information. As Bitcoin transactions cannot be predicted, access to the blockchain cannot be universally blacklisted without collateral damage, and transactions cannot be selectively removed from the blockchain by design, this introduced a seemingly resilient mechanism for botnet coordination. Cerber was however detected and dismantled by law enforcement based on

[1] https://dgarchive.caad.fkie.fraunhofer.de/welcome/

operational mistakes [13], but provided a glimpse into a potential future of malware coordination.

This new development of coordination has now become mainstream, when the Pony malware adopted the principle of Bitcoin-based C&C addressing into its bootstrapping mechanism in 2019. In this paper, we report on the second major malware family performing this evolutionary leap, and track the adversarial tactics, techniques, and procedures (TTPs) for a period of the one year in maintaining the infrastructure in response to take-down attempts from Internet Service Providers. While the adversaries setup a large infrastructure and spend over 2,400 USD to signal new IP addresses, the algorithm design itself contained flaws allowing a hostile takeovers of the malware. We use this to take over the malware to get insights into the infection behavior of this particular malware strain.

In this paper, we make the following 4 main contributions:

- We provide a forensic investigation of Pony's blockchain-based C&C mechanism to show how the malware authors have evolved to counteract existing blacklisting approaches against DGA pre-computation and domain take-downs. We find that adversaries succeed in hiding their infrastructure as they are not listed in common threat intelligence feeds.
- We are the first to perform an analysis of the actors' evolution in TTPs, and demonstrate how they go through sequences of innovations to determine a cost efficient setup that will serve their needs.
- We perform the first takeover of a Bitcoin-based C&C mechanism, as we show that a flaw in the mechanism makes it possible to realize temporary takeover of the Pony C&C address. We redirect connections from new Pony infections and get unique insights into the infection behavior of the malware.
- We show that the victim population is highly homogeneous and targeted, by far the most infections occur in Russia, which is highly surprising as previous malware often stays clear of Russia to avoid interactions with the legal system there [36].

The remainder of this paper is structured as follows: Section 2 provides a brief overview into the evolution of the malware ecosystem and specifically mechanisms bot masters have used for addressing C&C infrastructure. Section 3 summarizes previous research. Section 4 describes the inner workings of the Pony ecosystem, and the Bitcoin-based C&C addressing of the Pony malware. Section 5 shows insights into the orchestration of payments on the Bitcoin blockchain. Section 6 shows the evolution of hosting providers used by the malware owners, which started out on single hosting providers, but transitioned into using a diverse set of providers. Section 7 explores the malware itself. Section 8 describes an intervention made to intercept newly infected clients, and analyzes user participation in this temporary take-down. Section 9 reflects on our study and discusses the implications for future botnet defense. Section 10 summarizes our findings and concludes our work.

2 EVOLUTION OF C&C PARADIGMS

Innovation and evolution in malicious software comes in waves. Whenever the current state of the art in detection and mitigation of for example botnets becomes "good enough" to seriously impact their proliferation and the financial bottom line of the perpetrators, the actors introduce new technical innovations to restore the status quo and continue their activities as usual. This has essentially led to a cat-and-mouse game in distinct phases over the past 25 years.

This co-evolution began in the 1990s with the innovation such as packing and polymorphism, which made the detection of malware based on signature databases more difficult. In the early 2000, malware authors improved the way they communicated with the malware, switching from control protocols based on Internet Relay Chat (IRC) – which drastically stood out from Internet background traffic – to the more ubiquitous HTTP. Around the same time, we saw a trend to move over from centralized C&C schemes to distributed solutions, which were more difficult to take down, to the now very commonly used hybrid infrastructures.

After malware authors have changed how the malware looks like and how it communicates, over the past decade, we have now seen major developments in the way individual bots locate and address their C&C server. When the destination IP address would be hardcoded into the malware binary itself, it is relatively easy to take down the installation, as law enforcement or the service provider would shut down the server due to malicious usage. With malware binaries already having been distributed, the botmaster would immediately lose control over the installation, and as the binaries linger in people's inboxes in phishing emails or are already installed on victim's computers, has no easy way of updating the software. Malware therefore moved to addressing by domain names, and while the destination IP addresses could now be updated to account for takedown of the servers, this only shifted the problem as law enforcement would work with domain registries to take down domains used for malicious purposes. As the domain evidently would need to be encoded in the malware binary itself, reverse-engineering or observing the interaction of a single malware copy would be enough to localize and begin mitigating the botnet communication.

For this reason, a lot of malware today has adopted the use of DGAs. In a DGA scenario, the malware binary uses an algorithm to create a large amount of candidate domain names through which it would try to connect the C&C server. This list of potential domains is only valid for a limited time span and regularly rotated. The malware would walk through the list one by one to test which domain name would at the moment lead to the C&C server. Although in a forensic evaluation of the malware the algorithm could still be easily extracted from the malware, the use of the DGA changes the economics of the mitigation activity. To take down the botnet, defenders would need to register or block all candidate domains that would be generated by the algorithm, while the adversary would need to register only one. This asymmetry of effort is further aggravated by the fact that the list frequently changes, and after each update, a new set of domain names would need to be blocked.

Combating state-of-the-art malware C&C. There exist however a number of angles to detect and combat malware infections that leverage a DGA today:

- First, while searching for its C&C server, an infected machine performs a large number of queries to domain names that

do not currently exist. These floods of so-called NXDomain responses drastically stand out in a network, as users may occasionally mistype domain names but not hundreds of times in a row in regular intervals. Many solutions leverage periodic or bulk lookups, combined with large number of NXDomains, to localize infected machines [6, 33, 40].
- Second, due to their design, many DGAs often create high-entropy domain names of random letters, numbers, or morphemes. These domain names are drastically different from the ones where normal services would be reachable at. Other lines of work leverage the different entropy signatures of normal and botnet domain lookups to identify bots [9, 21, 34, 35, 39].
- Third, as the DGA has to be embedded in the code, it can be easily extracted by reverse-engineering the malware, and then precomputing all domain names that would be used at any point in the future. These pre-computed lists are then distributed to network owners who can null route them in their local DNS servers, and become notified if a local host attempted a connection [27].

These three angles have proved to be highly successful against malware today, this put pressure on the ecosystem, triggering malware owners to their latest innovation: the use of the Bitcoin blockchain for addressing the C&C server. With the information about the current C&C address being somehow encoded in the blockchain, clients would directly obtain it from the blockchain and could make a valid connection on the first try, thereby eliminating the treacherous train of NXDomain lookups. Furthermore, any kind of domain name, subdomain or IP address could be distributed, thereby avoiding the usage of a high entropy name that is a typical byproduct from the algorithmic generation of domains. Lastly, as the malware owner could place any information into the blockchain at any point in time, there remains no opportunity for precomputation of domain names, as updates are placed and distributed whenever needed and there is no way of predicting what exactly the botmaster would send to the blockchain. Bitcoin-based coordination thus strikes at the heart of existing mitigation schemes and will be a serious disruption to current mitigation efforts.

3 RELATED WORK

To weaponize a botnet, an adversary needs to be able to send commands to infected devices that are part of the net, telling them what to do or where to send new data. These so-called C&C messages generate traffic, which has consequently been used by defenders to identify the presence of bots in a network [7, 17]. The first botnets such as EggDrop and SDbot used IRC channels to communicate [30], but as IRC stopped seeing common use, the traffic generated by such botnets readily stood out. Malware developers therefore turned to protocols that are more commonly used in the current Internet.

After identifying C&C traffic towards a server, defenders will generally block these servers from communicating with their network in order to mitigate further C&C. Some malicious servers also end up in so-called blocklists [11], mitigating the malicious activity in all networks using the list. While this affects botnets which have a static IP address or domain, the timeliness of these lists is low enough for an adversary to change these indicators of compromise and continue the operation of the malware [16].

A common method to continuously change the location of the C&C infrastructure is using a DGA. The first malware to pick up a DGA was the Sality botnet back in 2006, which combined static and dynamic methods to generate new domain names [1]. Soon after, botnets implemented algorithms to generate entire fully qualified domain names, documented by Stone-Gross et al. [32]. Antonakakis et al. proposed the first detection system, called Pleiades, to identify DGA-based bots within a monitored network without reverse engineering the bot malware [6]. Pleiades was placed below the local recursive DNS server or at the edge of a network to monitor DNS query/response messages from/to the machines within the network, then analyzed NXDomain responses. Plohmann et al. [27] showed dynamic name generation has since become standard practice, identifying 43 different DGA families and reverse engineering their inner working. Lever et al. analyzed the network communications of 26.8 million samples that were collected over a period of five years [19]. Based on DGArchive by Plohmann as one of data sources, they measured the prevalence of domains created by DGAs in network communication from malware samples, and found that at least 44% of the domains from dynamic malware traces were generated by 42 DGA families.

The surge of these DGAs in malware families spurred researchers to come up with solutions against this new type of coordination, and has led to a large body of defensive mechanisms aimed at detecting and mitigating specifically this coordination. Some research aimed to identify these algorithmically generated domains by leveraging the difference between normal domains and generated domains [9, 21, 34, 35, 39]. These methods did not work against adversaries that created believable domain names using word lists in the generation, allowing adversaries to circumvent this detection. New methods however aimed to identify these domains as well [2, 37].

Next to detection research into C&C traffic and infrastructure, researchers devised strategies which would circumvent these detection methods and would make it hard for defenders to actively detect and mitigate the C&C infrastructure on a network level. Recently, research has focused on the feasibility of using the blockchain to signal the location of C&C servers or even the commands themselves, which offers anonymity for the botmaster, and makes it very hard to fully take down the infrastructure. ZombieCoin [3] inserted C&C data in the `OP_RETURN` output script function, sending the commands directly to the bots, eliminating the need for a C&C server alltogether. The authors extended on this paper by adding new functionality such as larger payloads and the ability to partition botnets through the blockchain [4]. ChainChannels [14] devised an even stealthier way of sending messages, based on the cryptographic operations in the blockchain. Baden et al. [8] devised a method to send botnet instructions over the cheaper Ethereum network.

Malware authors have started to pick up these C&C methods of using the blockchain. The first fully blockchain powered malware, named Cerber, was identified and analyzed in 2018 by Pletinx et al. [26], showing how adversaries carefully test and roll out their new infrastructure on the blockchain. Since the analysis of Cerber, new botnets have started experimenting with the blockchain, one of which is the Pony malware which was also reported as a concept but

not further analyzed by Check Point [12]. How blockchain-based C&C is used by malware authors in practice is a new and largely unstudied concept. This paper is the first work to study the TTPs used by the adversaries in Bitcoin-based addressing the Pony malware, through which we aim to set the first steps in understanding this new phenomenon to be used future detection methods.

Finally, our work leverages research on the traceability of Bitcoin for cyber incidents or ransomware payments. Satoshi Nakamoto (presumed to be a pseudonym) who first designed a peer-to-peer electronic cash system stated in the original Bitcoin paper [23] the risk of privacy as follows: *Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.* Based on the so-called multi-input heuristic, several researchers started evaluating and analyzing the privacy and anonymity of Bitcoin [5, 18, 28, 29]. When it had become clear that perpetrators abused Bitcoin for purposes such as Silk Road or ransomware payments like WannaCry, other researchers paid attention to the traceability of Bitcoin [22] based on the multi-input heuristic and each original heuristic for tracing incidents such as Silk Road [31], ransomware payments regarding CryptoLocker [20], 35 ransomware families [25], and transactions across cryptocurrency ledgers [38]. On the other hands, as we will describe in the following section, the Bitcoin operation for hiding the location of C&C servers related to the Pony malware is different from the behavior observed in these previous malicious uses. That is, the adversaries repeatedly sent Bitcoin from a single sending Bitcoin address to the same receiving Bitcoin address, and pool all funds into one large wallet. While the behavior is slightly different, we leverage these previous works to trace the extent of the operation and find all Bitcoin addresses owned by the actors.

4 BACKGROUND

The previous sections described how C&C coordination mechanisms evolved and why Bitcoin-based control would provide a competitive advantage over the state-of-the-art in detection and mitigation. In this section, we provide an overview of the Pony malware ecosystem, its use cases as well as the general functionality of its Bitcoin-based C&C addressing mechanism and the datasets used for this study. This provides the necessary background to discuss the concrete tactics pursued by the malware owners in the next sections.

4.1 The Pony Malware Ecosystem

The Pony malware family has been active since 2011 and is one of the most widely occurring malware components on the Windows platform. It can serve a variety of purposes, and act as a loader platform to download additional malware to install (the pay-per-install business model), turn the machine into a zombie client as part of a botnet, or perform credential stealing on the victim PC. Its credential theft capabilities are relatively advanced, targeting more than 100 different applications, ranging from email clients, web browsers, remote access such as SSH and VPN, and cryptocurrencies wallets, and even has the capability of trying to decrypt local password storage. According to recent malware surveys, Pony is the currently largest player in the credential theft business with a market share of 39% [10].

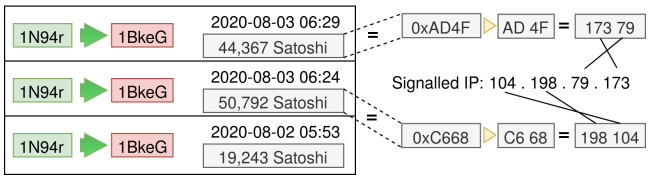

Figure 1: The C&C server address is hidden in the transaction amount of two consecutive payments

In 2013, the source code was leaked to the public and shortly after new iterations of the malware emerged, some of them offered for sale or marketed as a platform for conducting cybercrime. With the publicly available source, a number of variants with different functionality exist now, leading to a widely fractured de-centralized ecosystem under the control of different actors. Pony is by now also adopted as a module into the tooling of advanced persistent threat groups [10].

4.2 Bitcoin-based C&C Addressing in Pony

In mid 2019, a strain of the Pony began a transition from the established addressing of C&C infrastructure by a DGA to signaling the location of the C&C server through the Bitcoin blockchain. For this, every Pony malware loader has a particular Bitcoin address hardcoded in the source code, for which it monitors payment transactions via a handful of blockchain sites such as api.blockcypher.com or blockchain.info. It retrieves from these public services a record of recent transactions related to this address, and decodes the currently used C&C IP address from the payment records using a very simple mechanism.

Figure 1 depicts how this data is hidden in the transaction. Over time, the Bitcoin address hardcoded in the malware – in this example the address starting with *1BkeG*[2] – receives payments from one or more other Bitcoin addresses with varying amounts of Bitcoin between 0.0000101 BTC and 0.00065278 BTC, or 0.12 USD and 7.75 USD respectively at the time of writing. The payment amount encodes two octets of the C&C IP, and can be retrieved by converting the transaction value in satoshi (thus, 0.000xxxx BTC) in hexadecimal format, and swapping the two bytes. As shown in the figure, the decimal value of 44,367 satoshi has the hexadecimal representation of 0xAD4F, and the bytes of 0xAD = 173 and 0x4F = 79 as placed in little endian format as the last two bytes of the IP address. The process is repeated with the second-to-last transaction to obtain the most significant parts of the IP address. In the remainder of the paper, we will refer to the person making the payment as the "sender" and the receiving address as the "IP signal", as this is the one monitored by the malware and whose transactions communicate the current C&C address.

The split of the signaled information into two separate transactions is meaningful, because it reduces the overall volume of money

[2]In the remainder of this paper we will refer to addresses only by the first 5 letters for compactness and readability, and refer the reader to the appendix for the complete addresses.

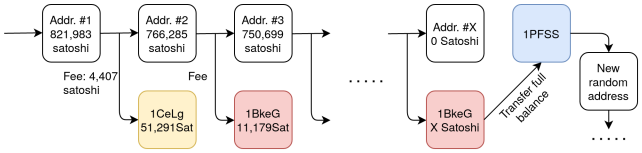

Figure 2: First transaction format: Temporary one-time use addresses transfer the correct balance to the main addresses in a chain. When depleted, 1PFSS replenishes the balance in the temporary addresses

that the owners have floating through the system. While the maximum payment needed to signal an IPv4 address using the method described above would be 0.00130556 BTC, encoding a routable IP in a single transaction would using this mechanism require as much as 42.8 BTC. The swapping of the two bytes has however no clear purpose. First, it does not reduce the transaction amount statistically, second, it does not really increase the covertness of this transmission channel, and third it is not a programming artifact as IP addresses are actually formatted in big endian format.

4.3 Datasets Used

For this study, we rely on three datasets to unravel and track the C&C infrastructure of Pony:

First, we used the BlockCypher API v1[3] for collecting transaction information signaling the malicious IP addresses. In particular, we made use of the fields block_height, fees, and confirmed. The 'block_height' provides the height of the block that contains a corresponding transaction. The 'fees' is the total number of fees collected by miners in a corresponding block. The 'confirmed' is the time at which transaction was included in a corresponding block. We implemented a C&C IP decoder from the transactions based on the way of encoding C&C IPs in Figure 1. Every time we detect newly added transactions related to IP signaling addresses in the Bitcoin blockchain, we decode the C&C IP from the transactions. This IP is used to identify malware samples which communicated to the decoded C&C IPs based on dynamic analysis reports by Dr.Web, Lastline, Tencent HABO, and QiAnXin RedDrip from VirusTotal[4]. We collected 748 samples from August 30, 2019, to August 14, 2020. It should be noted here that IP addresses were decoded from the most recent transactions at the time a sample was executed in the vendor's sandbox. Although we regularly tried to find other strains of malware which abused other IP signaling addresses by searching malware samples which abused blockchain APIs, we have not found any new malware at the time of our study.

Second, we obtained malware samples from Hybrid Analysis[5] that were identified as Pony samples in the above analysis. For every sample, we made behavioral runtime analyses with the CrowdStrike Falcon sandbox. This yielded a total of 287 (of 748) valid Pony samples. In Section 7, we will describe the analysis results in detail.

Third, based on the address identifiers included in the obtained samples, we expanded our scope to all transactions made from and to these addresses. To analyze the transaction structure on the blockchain we used Orbit[6]. In total, the dataset contained 2,594 unique transactions spanning over 40 Bitcoin addresses (Appendix table 3). The full address information of the main addresses used in the operation is included in Appendix A. IP addresses were mapped to country and autonomous systems (ASes) using the GeoLite2 database provided by Maxmind[7].

5 THE EVOLUTION OF THE BITCOIN-BASED C&C COORDINATION MECHANISM

Although one earlier malware had already pioneered the use of Bitcoin as C&C mechanism in 2018 [26], no other malware had followed on this innovation step until the Pony malware described in this paper. This could be due to the fact that Bitcoin-based coordination is still highly novel and largely uncharted territory in the sense that there are no established practices yet on how to best realize it, as for example would exist in case of malware polymorphism or DGAs, where ready-made software and kits exist that authors can just use. Thus, malware authors essentially need to develop their own strategy, experiment over time with advantages and disadvantages of a chosen solution, and potentially adapt over time based on lessons learned. We found that such innovations and learning cycles were the case with the Pony malware, which underwent multiple cycles of different implementations and usage patterns. In this section we will analyze the TTPs in use by the malware owners and how these were adapted over time.

5.1 Payment Orchestration

For the Bitcoin-based coordination mechanism as designed by the malware owners, in principle only two Bitcoin addresses are needed. First, there needs to be an address that is monitored by the malware and which signals the information the malware owners would like to communicate. Second, as funds need to be transferred to this signaling address, another account is obviously needed which originates this transaction. For evident reasons, we will refer to these as the IP signaling and sending address in our discussion. In the Bitcoin blockchain, no concept of an "account balance" exists. Instead, money is sent by "spending" a previous transaction. In other words, if an account has an earlier deposit of 1 BTC in a transaction with the ID X and this account would like to pay 0.7 BTC to someone else, a new transaction Y is made that spends X, sending 0.7 BTC to the recipient and directing the leftover 0.3 BTC in a separate transaction Z to somewhere else. This address which collects the remainder of each transaction could either be the sending address, or a new one.

Bitcoin-based malware coordination is still a novel concept, and also the authors behind the Pony malware went through cycles of experimentation and optimization of how to operate such a mechanism, starting from the moment it was first rolled out in August 2019. As discussed above, in the scheme devised by the malware owners, the current location of the C&C server is signaled by the value of two transactions sent to a signaling address, which is hard-coded in the malware binary. This signaling address receiving the transaction remained identical for the entire time this scheme has been in operation, except for a brief intermezzo where for one

[3]https://www.blockcypher.com/dev/bitcoin/
[4]https://www.virustotal.com/gui/
[5]https://www.hybrid-analysis.com/
[6]https://github.com/s0md3v/Orbit
[7]https://dev.maxmind.com/geoip/geoip2/geolite2/

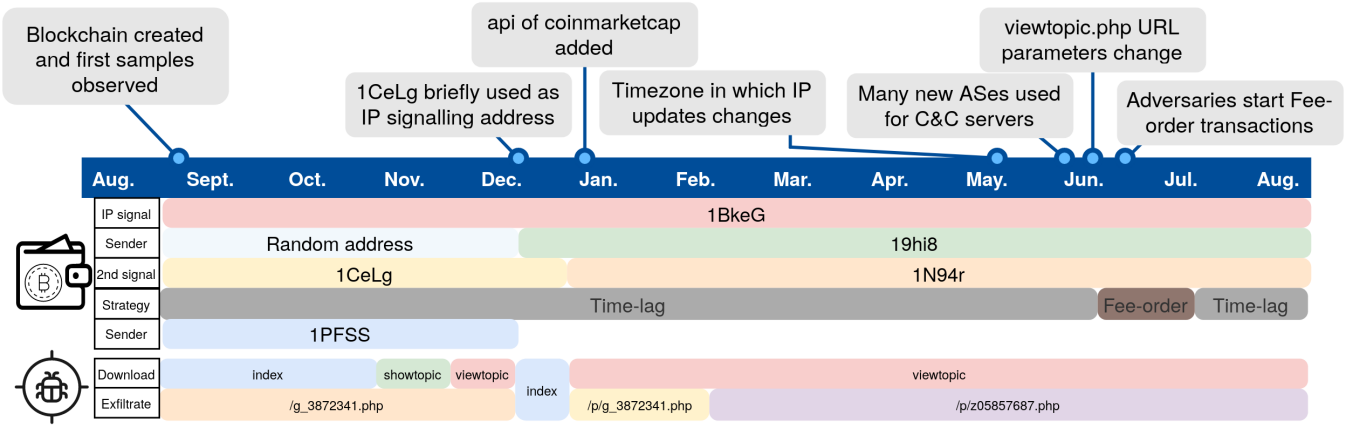

Figure 3: Timeline of updates on the Pony malware wallets and contacted endpoints

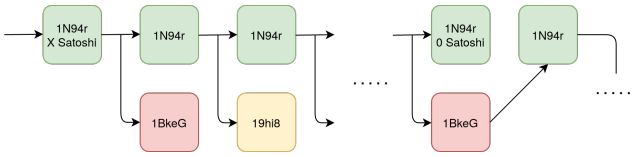

Figure 4: Transaction format after Dec. 10 2019: 1N94r is used to transfer the correct balance to the main addresses, and to aggregate the balance

week in December 2019 new malware was distributed switching to a different address that immediately disappeared again.

How exactly the authors orchestrated the payments to the signaling address changed significantly over this year's time. For the first four months of the operation, payments were made to the signaling address from random Bitcoin addresses. As shown in Figure 2, each sending address would make two payments that contained the C&C IP address to 1BkeG, and after subtraction of the fees deposit the remainder in a new Bitcoin address. The original sending address would be subsequently abandoned. With each round, the funds would diminish due to the signaling transaction and the related transaction fees. Once the balance in the chain of random addresses was depleted, all the funds that accumulated in the signaling address would be transferred to a separate long-term address, 1PFSS, which would then make a seeding deposit to a new random address from which the cycle would continue.

As shown in Figure 2, besides signaling transactions to 1BkeG, occasionally other payments were made from the random sending addresses to a separate long-term address 1CeLg. We refer to this as the second signaling address, as the nature of the transactions would match the C&C encoding also followed by 1BkeG. This secondary signaling address might have potentially been used for infrastructure testing, as this address was not monitored by any Pony malware except for a brief period at the beginning of December 2019, when suddenly signaling shifted from 1BkeG to 1CeLg for 6 days, before restoring operation back to 1BkeG and abandoning 1CeLg entirely from now on.

Figure 3 shows a timeline of the Bitcoin addresses used for sending and receiving funds. The temporary shift of operations in December 2019 also saw a major disruption of how the malware operated in general. At this moment, we observed a significant change in how the binary worked, such as a shift in which registry entries were made and checked, and how the loader pulled new components and exfiltrated data.

Curiously though, after this week-long transitionary period, most of the updates were rolled back and resumed to the previous state, with the major exception of how the addresses were orchestrated and payments were done. From this moment on, payments were again made to 1BkeG, but instead of using random addresses all funds came from a new single long-term sending address, 1N94r. As shown in Figure 4, this address would continuously make the two signaling payments to the receiving address, and as soon as the sending address would be emptied out, funds were shifted back from 1BkeG to 1N94r, and the procedure would continue. This essentially resulted in a ping-pong of funds between only two Bitcoin addresses. Again, occasional transactions were made to a newly created second signaling address, which was never used in malware deployed in the wild.

Although apparently an operational detail, the shift in operations can be associated with a change in procedures on account of the adversary. The usage of random, disposable addresses is the default behavior for web-based Bitcoin exchanges such as blockchain.com, where customer funds are under the control of the operator and customer transaction take the format of transactions to one-time address accounts. This means that until December 2019, the perpetrators were using exchange portals, which would due to KYC policies typically allow a linkage between funds and owners [24]. The continuous transactions from the same address mean that the actors now took the payments into their own hands, as this type of transaction is only possible if the owner would have access to the address' private key, which is not the case of the online exchange platforms [24]. This would provide extra anonymity and seem a logical step to make, except that it would introduce some complexity in operating a wallet and storing the keys using either software

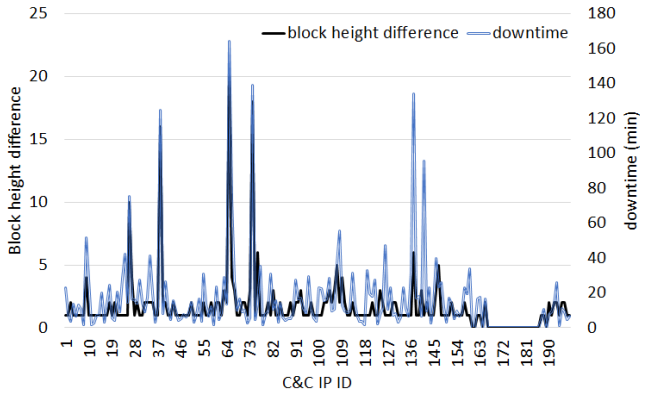

Figure 5: Block height difference between two transactions and the resulting downtime

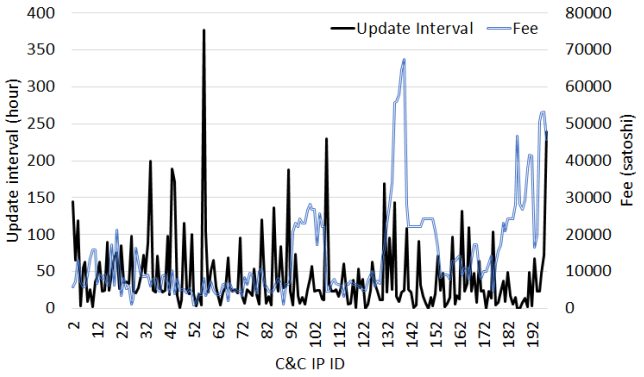

Figure 6: Update interval of C&C server IP addresses and the fee paid for the transactions

on their computer or a hardware implementation. The Pony authors had however this capability all along – as the static long-term address 1PFSS had already replenished the random addresses for months –, yet did not bother to use this capability consistently throughout their operation until the major overhaul in December 2019.

Pony's C&C mechanism signaled the server IP address in two separate transactions, as otherwise the necessary payments would have been astronomical. When two separate transactions are used, this creates an operational problem. Payments sent through the Bitcoin system are only final if the transaction is included and confirmed by the person who mined the next block in the blockchain. The propagation through the blockchain network creates some delay, furthermore miners will independently determine which transactions to include into their block. Thus, if the malware owner would send 50,792 satoshi followed by 44,367 satoshi to signal 104.198.79.173 (see Figure 1) but the two transactions had been included in a different order, bots would be contacting the wrong IP address.

During the one year in which we observed Pony's use of Bitcoin, we saw that the adversaries were experimenting and refining two strategies to avoid and mitigate these operational problems for their setup. We refer to the first strategy in use as *time lag*, where transaction were sent with a delay to end up in different blocks. Later, the adversaries attempted to control the behavior of the blockchain by paying higher incentives to the miners, to which we refer as *fee order*.

5.2 Time Lag

A simple and evidently effective strategy to make sure that one transaction is recorded before the other is to create an artificial delay in between them, and only fire the second one once the first transaction has been confirmed in a block and spread throughout the network. Indeed, this approach was also at first followed by the Pony authors, but comes at the price that there is an intermittent period during which bots would contact the wrong C&C address by decoding the new and an old transaction value. While the block interval in the blockchain is currently on average 10 minutes, we see that in practice the time lag between transaction was significantly longer. Figure 5 shows the time in between first and second transaction in minutes for the progression of 198 C&C IPs signaled in the blockchain by Pony. We calculate downtime as the differences of confirmed time in the JSON file by BlockCypher between the first and second transactions. The dates and IP addresses used as C&C over time are tabulated in the appendix. As can be seen in the graph, the follow-up transaction was frequently not included in the subsequent block and the malware owner often experienced significant delays and in result outages of their system. Interruptions of half an hour or longer occur regularly, the maximum disruption we observed tallied to 164 minutes. The resulting misdirection of bots is not merely a hypothetical issue, as we observed malware samples executed in antivirus vendors' sandboxes to contact malformed IPs that would be decoded in these intermittent periods.

In addition to the time of disruption, Figure 5 also displays the block height difference of the two signaling transactions – in other words how many blocks the two were apart – as a function of the C&C IP. As we see in the figure, especially in the beginning, massive downtimes coincided with major block height differences. The peak outage of 164 minutes was for example caused as the second payment was only included 20 blocks after the first transaction. This can be attributed to one of two reasons, either they did not send the second transaction fast enough, or the transaction fees advertised for the second payment were not attractive enough for miners to include them in their block.

5.3 Fee Order

The manner in which payments are included in the blockchain is not only a question at which time they were sent, but also how they were paid for. To include a transaction in the blockchain, the sender has to pay a fee. This fee is dependent on the size of the transaction in bytes, but the sender can choose to offer a higher fee to increase the incentive that a miner would include the transaction in the currently mined block. Starting from March 2020, the malware owners discovered this alternative strategy to exert control over how transactions were included into the blockchain. By setting the right amount of fees, the malware owners could influence that

both transactions would be prioritized and become part of the same block, while still maintaining the right order.

This new concept was put into place from C&C IP ID 165 in Figure 5. When we look at the block height difference and resulting downtime, we see that large peaks for the block height essentially disappear. Using higher fees to elevate incentives however also increases the expenditure of operating the system, and poses the question what is the right fee to use to push the block height difference down to 0 while not overpaying the miners. While the perpetrators found a strategy that proofed effective against downtimes, the operating costs seemed to have been too substantial to maintain, as soon after the malware authors deviated from it in an attempt to optimize this relationship.

While high fees provide a larger incentive to include the two payments into the blockchain in a timely manner, there is much uncertainty which fees would be the right one to use at a particular moment due to a number of external factors. What is the current market price and hence constitutes a good fee that would prioritize a transaction depends for example on the number of transactions in a block and the capability of the miner. These aspects would not be known to the person making the transaction. After the switch to the fee-order strategy, the Pony authors went through a series of trials, experimenting with fees to optimize their operation.

Fee increases. From March 12, 2020, higher fees were set by the adversaries, which was abandoned at March 26. While it was quickly abandoned, the higher fee considerably sped up the confirmation time for a transaction to reach the blockchain, minimizing the downtime of the system. Figure 6 shows the update interval of the C&C address and well as the fee used per update, March 12 corresponds with C&C IP ID 95, and the experimentation period ends at ID 107.

Result of Bitcoin Halving. Every four years, the reward for mining a Bitcoin block is cut in half. The increasing difficulty and dropping revenues over time have caused significant changes in the Bitcoin ecosystem, while at first mining is lucrative to do even on a commodity hardware PC, mining has first moved to GPUs and now requires hardware components deployed in countries with low electricity costs to still turn a profit. The resulting disruptions from Bitcoin halving has caused fee soars.

This had also a major influence on the Pony owners. While in the week of May 11, 2020, the adversaries only paid 30,000 satoshi in fees, the fees more than double in the week afterwards, to 70,000 satoshi in the week of May 18. Figure 6 shows the result of the Bitcoin halving at C&C IP ID 132. The solution to this was simple but effective, the adversaries simply shifted when they were making transactions in the blockchain. While until now, all signaling transactions had been done in European business hours, the malware owners shifted from May 26 onwards their activities to midnight in Central European Time, when apparently less transactions are made. This dropped the required fees by more than two thirds (C&C IP ID 140), and also came with the added benefit that transactions were confirmed quicker now. Figure 7 shows the time in UTC where updates were made before and after May 26.

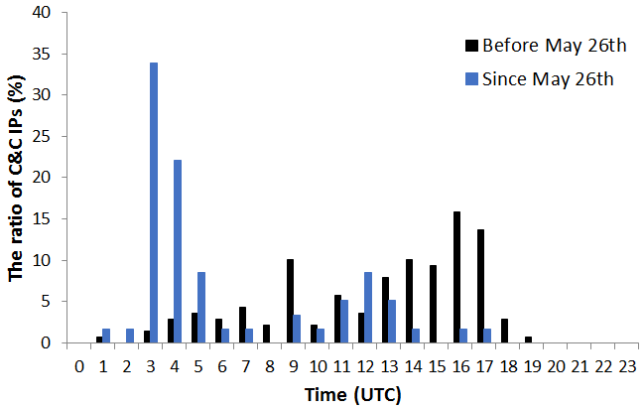

Figure 7: The ratio of update time of C&C IPs before and since May 26 2020

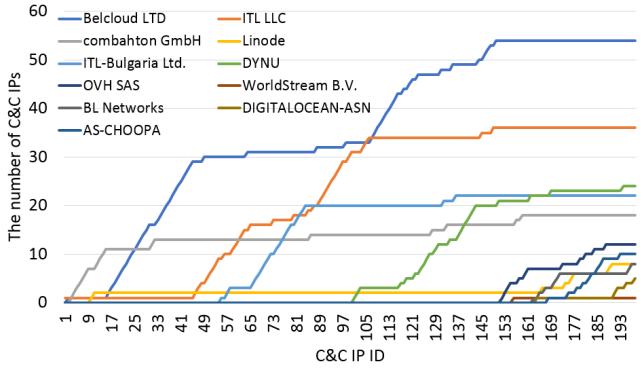

Figure 8: Autonomous systems used for C&C servers

6 INFRASTRUCTURE

The Pony malware relies on the blockchain to signal the C&C IP address, and does not use the blockchain as a C&C instance on its own. Like is the case when using a DGA, the adversaries have to signal a server that is under their control, and have to register and set up systems for C&C. In total, the adversaries have signaled 198 C&C servers through the blockchain during the study (For the full list, see Appendix B). This section identifies the infrastructure used by the adversaries, and discusses the replacement speed of these C&C servers.

6.1 Infrastructure Location

IP addresses signaled through the blockchain mainly belong to cloud and hosting providers, where the adversaries rent servers for their malicious activity. Figure 8 shows the cumulative count of ASes used by the adversaries as function of the C&C IP address index – in other words the progression over time –. Initially, the actors concentrated on a single provider at a time, which are heavily used, but then abandoned in favor of another. Occasionally, this circle is repeated. The figure shows multiple shifts, where Belcloud LTD. was almost exclusively used from C&C IP ID 10 up to 46, ITL-Bulgaria was used from ID 73 up to 82. From ID 154, we see

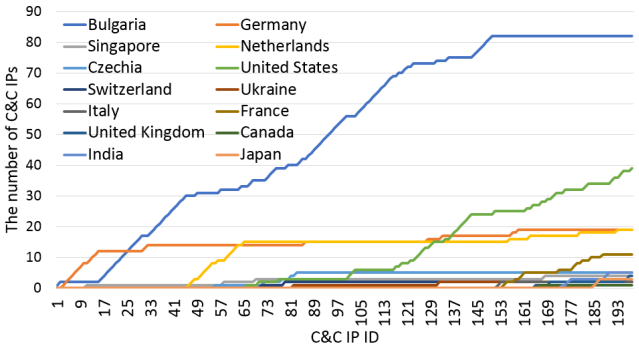

Figure 9: Country location of C&C servers

actors to diversify significantly with the inclusion of OVH SAS, WorldStream and DigitalOcean, and subsequently the adversaries always change different providers instead of only using one.

Figure 9 shows a cumulative count of the countries the C&C IP addresses were located in. Initially the operation was 50% based in Bulgaria up until C&C IP ID 154 at June 4, 2020, after which the providers located in Bulgaria were completely abandoned in favor of other larger cloud providers such as DigitalOcean and OVH (detailed data can be found in Appendix C). While these providers have a strict policy against malicious use and will take a server down after it has been identified as being malicious, there have not been any server takedowns while the malware was active on this IP, as the C&C program always remained briefly active after the adversaries had already signaled a new IP address in the blockchain.

6.2 Infrastructure Changes

C&C infrastructure is often included in blacklists after detection, as the traffic from these servers is malicious and unsolicited by network operators. Moreover, domains based on a DGA are even preemptively blocked to also stop an adversary in the future. While blocking future domains is no longer possible with adversaries using the blockchain, it is trivial to detect the blockchain update and block the traffic to this server immediately, or to block the C&C server after it has been identified by a SOC or security vendor. Of all 198 IP addresses, not a single one has been listed in AlienVault[8] or AbuseIPDB[9] as a known IP address related to Pony C&C servers at the time it was used.

Figure 6 shows the interval in which the adversaries update their infrastructure, and the fee that is paid for these updates. On average the adversaries update the infrastructure every 42 hours, but 38% of the transitions were made after less than a day since signaling an IP address. In 3.6% the transition was done even within an hour. This shows the versatility of using the blockchain as an instrument of C&C IP signaling, where malware authors can change the location of the infrastructure at any time instead of relying on an algorithm that changes in set intervals. After changing the IP address in the blockchain, the adversaries quickly take down the previous server.

While most of the transitions are fast, there are some large outliers. Most notably is C&C IP ID 55 with a downtime of 15 days, from December 24, 2019, to January 9, 2020. Coincidentally this downtime occurs exactly during Christmas and New Years, which could occur due to personal holidays from the malware author or the lack of people at workplaces who would click on the phishing link, which is economically nonviable.

6.3 Infrastructure Cost

The amount of Bitcoin used by the adversaries to signal the IP address is added as wallet balance, and is not lost for the adversaries. The fees paid to make sure the transaction is added to a block and included in the chain is however lost. Figure 6 shows the fee used to signal every IP address to the blockchain. The adversaries can set these fees themselves, and generally it holds that the higher the fee, the faster the transaction is confirmed in the blockchain. The figure shows a large increase in fees since C&C IP ID 132, with the fee rising with over 500%. This increase in fee is a result of the Bitcoin halving, making transactions on the blockchain much more expensive. After the halving, the adversaries paid over 7 USD to signal one IP address to the blockchain. To drive these fees down, the update time of the IP addresses shifted to a less busy time on the blockchain, as shown in Figure 7. This allowed for the decrease in fees while still confirming the transactions in a reasonable time.

The total fee paid by the adversaries on the Bitcoin network is 21,732,015 satoshi, which is approximately 2,421.21 USD only for signaling the IP address to the malware. As since the Bitcoin halving the fee was much more expensive, over 60% of total fee costs were in the last three months: 1,496.74 USD from May 11 to July 31, 2020.

After the trading time shift, the fees were from 1 to 2 USD for a while. Since C&C IP ID 182, the fee set by the adversaries has gone up again, hitting over 5 USD for a single IP address. While the blockchain requires slightly higher fees to consistently confirm transactions than the IP addresses signaled before, the adversaries set their fee approximately 4 times higher than the "priority-fee" defined by blockchain.com. While this makes sure the transaction goes through in a matter of minutes, the use-case of the adversaries and the incurred extra fees raise the question whether this would be necessary.

7 MALWARE ANALYSIS

To further understand the malware behavior and evolution, we analyze 287 malware samples in a time frame from September 1, 2019, up to August 5, 2020. These malware samples are obtained out of the 748 malware hashes we have found previously. We do not have access to the binaries for the other hashes. In this section, we report on the evolution of the malware itself, and on how it interacts with the loader server. To analyze the behavior of the malware, we set up a Cuckoo environment using a freshly installed Windows 10 system on a dedicated machine as sandbox for every sample.

7.1 Malware Behavior

The 287 different malware samples analyzed in this study all behave in the same way, shown in Figure 10. First, the adversary signals an IP address to the blockchain that will be used as C&C server. Second, malware distribution happens through phishing

[8]https://otx.alienvault.com/
[9]https://www.abuseipdb.com/

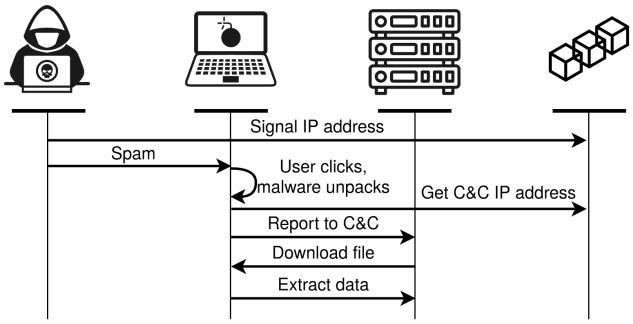

Figure 10: Pony malware behavior

and spam emails, and triggers when a user clicks and thereby runs the malware. The malware will unpack and obtain the C&C address from the blockchain. Subsequently, the malware reports to the C&C server to download an encrypted dynamic link library (DLL), which after unpacking is used to extract sensitive data of the user to the adversaries by sending encrypted HTTP POST messages.

After the DLL is obtained from the server the malware adds itself to the startup tab of the infected system, after which the command: `cmd.exe /c ping127.0.0.1 & del /F /Q <MALWARE FILE>` is executed, deleting the original file.

7.2 Failsafe Design

While signaling the IP address through the blockchain is a rigid and hard to block method of communicating the address of the C&C server, a blockchain website being temporarily unavailable or blocked in a certain network would hamper the operation of the malware. To counteract this, the authors of the Pony malware have added failsafe mechanisms to ensure that the malware is able to find the C&C server address by implementing three different blockchain APIs: (1) blockchain.info, (2) api.coinmarketcap.com and (3) api.blockcypher.com.

These APIs are always contacted in order, and the next site will be requested on either a timeout, an invalid response or an invalid SSL certificate that does not trace back to a root certificate installed on the system.

If the malware is unable to contact any of the blockchain APIs, or receives an invalid file from the loader server, it still deletes the original file and does not add anything to the startup tab, hiding the infectious file, but also effectively mitigating the infection.

7.3 Malware Evolution

After some time, malware evolves to be either more efficient, or to remove artifacts that are being detected by anti-virus products. The different stages of the malware evolution for the Pony malware are shown in Figure 3.

In the first few months of operation, the malware contacted api.blockcypher.com as the first choice for contacting the blockchain. After 2 months of operation, the adversaries switched the order of blockchain websites in favor of blockchain.info. While these APIs are exactly the same for the use-case of signaling the IP address, the request limit of blockchain.info is higher, which might be the reason

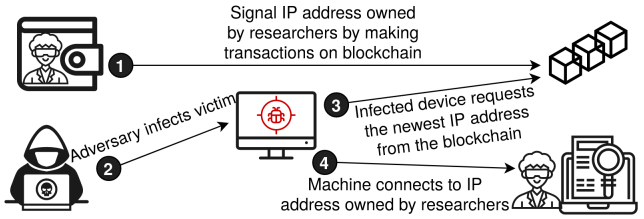

Figure 11: Methodology for malware takeover, redirecting infected devices to our server

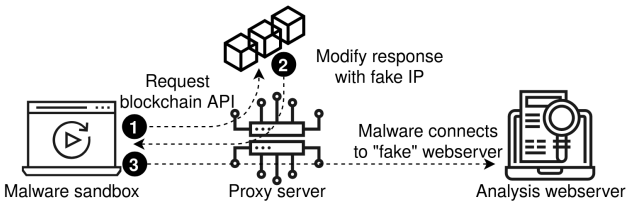

Figure 12: Test setup to identify the feasibility of a takeover, testing various blockchain responses to identify whether the adversaries do additional checks preventing a takeover

this changed in the malware. In January 2020, the coinmarketcap API was implemented as the second choice for more redundancy.

The files contacted on the C&C server to download the malware file and exfiltrate the data also underwent changes during the measurement period, shown in the bottom two timelines in Figure 3. While it would be expected that the malware authors support previous malware configurations for at least a while after an update, and thereby not missing any revenue, support for older samples is immediately stopped when an update occurs.

8 C&C TAKEOVER

Although coordination via the blockchain is a highly resilient mechanism against takedown, the design of the Pony implementors contained some weaknesses that would make this possible regardless. While Cerber for example signaled C&C information in the outgoing transactions which only the wallet owner could trigger, Pony encoded data in incoming transactions - however in the blockchain everyone can transfer funds to arbitrary addresses and thus effectively signal new C&C information. In this section, we report on a takeover of Pony and what we learned about the victims and the behavior of the adversaries after the takeover.

8.1 Takeover Feasibility

Before attempting and possibly alerting the adversaries to the possibility of a temporary malware takeover, we test whether Pony malware samples would actually be vulnerable to this attack, and do not perform any additional checks on the transactions made towards the 1BkeG address. The takeover works as depicted in Figure 11, starting with signaling a new IP address to the blockchain by a freshly created address. When an infected device consecutively queries the newest transactions from the blockchain, the C&C server will be located at the newly signaled IP address. To

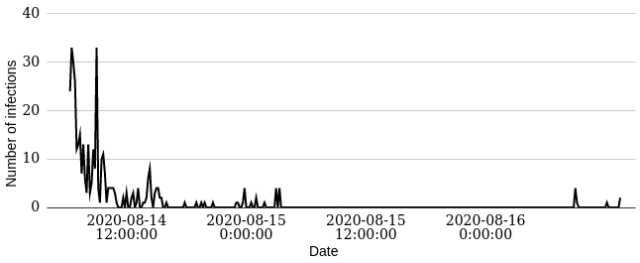

Figure 13: Infections per 10 minutes identified during the initial takeover

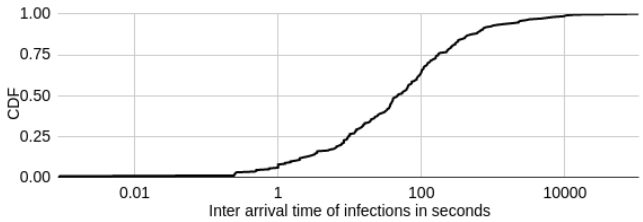

Figure 14: Inter-arrival time of infections

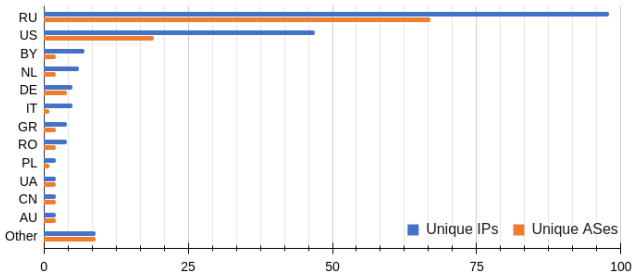

Figure 15: Spread of infections for unique IP addresses and ASes per country

make sure the adversaries do not do any additional checks, such as verifying that the transactions originate from the aggregation address 1N94r as shown in Figure 4, we test 287 malware samples for their behavior when presented a fake IP address through the blockchain. Figure 12 shows our test setup, in which we run the malware through a proxy, and overwrite all responses from blockchain.info to contain "fake" transactions. All 287 malware samples were vulnerable to this attack, giving a full view of the spread of infections after the takeover. Note that presenting a fake server which does not respond with a correct encrypted DLL file ensures that the malware does not trigger any exfiltration and will delete itself from the system.

8.2 Signaling the IP Address

The measurement server used to sinkhole the malicious traffic is located at 34.67.67.23. Signaling this IP address to the blockchain requires two transactions of respectively 17,186 and 5,955 satoshi, which in total is 2.73 USD at time of writing, which is transacted to the address owned by the adversary. These transactions were made on Friday, August 14, 2020, at 8:12 and 8:19 am, 2.5 hours after the last transaction by the adversaries. The adversaries took back the network 2 days later at the August 16 at 11:51 am.

The first request to the C&C server came in merely 2 minutes after the takeover transaction went through, at 8:21 am. Figure 13 shows the infections per 10 minutes during the takeover period, where the bulk of infections happen close to the time of the takeover. This high number of infections can be attributed to the distribution method of this malware strain being phishing, where many infections occur right after the emails are sent. In total, we have observed 275 unique infections during a two-day period, mainly in the first few hours of the takeover. According to [15], the median click-through rate in phishing emails is 16.7%, in which case the adversaries would have sent 1647 phishing emails.

Figure 14 shows a CDF of the inter-arrival time of two consecutive infections, which has a high rate in which 60% of the infections are less than 100 seconds apart. This infection speed only occurs in the initial burst of infections, as the infections become more sparse as time progresses, with 1% of the infections occurring more than 3 hours apart.

8.3 Geographical Bias

By directing the incoming requests to another server, we obtain unique insights in the spread of this particular malware campaign. Figure 15 shows the spread of infections in terms of IP addresses and originating ASes, where the largest part of infections are located in Russia, followed by the US and Belarus. When looking at the individual machines that are infected however, a different picture shows. While the infections in Russia originate from residential IP addresses, the malware infections in the US originate solely from malware sandboxes, internet proxies or security firms. This also holds for west-European countries such as the Netherlands, Germany and Italy. In fact, infections can be traced back to countries where the Russian language is known: Russia, Belarus, Romania and Ukraine. In the other countries the infections stem either from proxy servers, or security vendors running the malware in a sandbox environment. As the malware is distributed through phishing emails, the adversaries are able to target their attacks towards this specific group of victims by sending emails in Russian.

8.4 Adversarial Behavior after Takeover

During the takeover, the adversaries did not receive the requests from infected machines anymore, and did not steal any credentials. This loss of revenue would be noticed by the adversaries, on which they would have to respond by taking back the network and perhaps modify the malware to prevent another takeover attempt. The IP address signaled by the 1BkeG address was generally updated by the adversaries once a day, but after the takeover the IP address was not for two days. One would expect the adversaries to take back the infrastructure as quickly as possible to not lose out on revenue, but given that the adversaries first need to send out phishing emails for infections to come in, revenue loss is minimal after the initial boost in infections, eliminating the need to immediately take the system back.

To further understand the behavior of the adversaries, three additional takeovers were conducted to identify whether the adversaries respond differently or have fixed the flaw in the malware allowing for the takeover. In the subsequent takeovers, the number of infections contacting the measurement server decreased significantly, with only the occasional security vendor contacting the server in total. While this could be the result of a change in the malware, where it did not use IP addresses that were not sent by the correct Bitcoin address, malware samples submitted to VirusTotal during this time period had queried the measurement server and were thus not instrumented differently. As the adversaries did take back the malware, the distribution of new phishing emails could be synchronized to these distributions, where the bulk of the infections would go to the intended C&C server. In this case, we would only observe the long tail of the infections.

After the initial takeover the adversaries went back to their normal operation, but after the second takeover the adversaries completely abandoned their original pattern in which they updated the C&C every day at the times shown in Figure 7. Instead the update pattern shifted until later in the afternoon, several hours after the takeover. However, the last takeover had been active for multiple days, without the adversaries signaling a new IP address, and without any new samples being reported to VirusTotal. After the third takeover of the infrastructure, the adversaries modified their setup to address the critical design flaw, in that the C&C information was no longer signaled through the *incoming* transaction into the wallet, but rather the *outgoing* payment, thus mirroring the design previously used in Cerber.

9 DISCUSSION

The advent and proliferation of malware coordination via the blockchain marks an important turning point in the development of C&C paradigms. As discussed in Section 5, its use essentially renders typically used detection and mitigation techniques ineffective, and thus means that master drawing on this principle will be difficult to suppress in practice. Although one might be tempted to simply block any outgoing request to a blockchain website as an ad-hoc measure, such a list will on the one hand be futile to keep up to date, and could on the other hand result in collateral damage. Aside from signaling C&C information, this mechanism could also be utilized for other malicious purposes, such as covert channels or data exfiltration. It is therefore paramount to develop new strategies to reliably detect and eliminate this new type of channels that piggyback upon existing infrastructures.

Also for to this very reason, the takeover we report on in this paper was only a temporary disruption of the perpetrators' malware campaign. While we were able to suspend their activities for a total of 17 days and saved hundreds of users from infection, their activities ultimately resumed. Contrary to for example domain name-based control channels where a takedown is possible and has been accomplished before [32], as we cannot by design exert control over the blockchain, any actions taken in this context will only be temporary and we see that the adversaries in the end recovered and continued their activities. Is the squeeze therefore worth the juice, and should the research and security vendor community take the effort to disrupt – even if only temporary – such infrastructures? We believe this to be a worthwhile endeavor, even if the effects were only temporary, as in the process we were able to obtain valuable insights into the operation and scale of the actors that would not be possible through other angles and datasets commonly in use, such as for example passive DNS or sandboxes such as Virustotal. These insights help us on the one hand to characterize the threat landscape, and might in the end provide new insights towards better techniques for detection and disruption of these threat vectors.

Takedowns or takeovers are therefore one piece in the puzzle to prevent infections and safeguard users, but there is also another component in this ethical debate that should be considered. By means of takedowns, defenders essentially exert pressure on the ecosystem and thus drive threat actors towards innovation. This side effect was clearly visible in our study in that we could conclusively show that the threat actors understood our disruptions, identified their weak points and remedied it. Should defenders therefore block infrastructures, given that this would lead to innovation that might in the end lead to a potentially undetectable mechanism in future campaigns? Also here we believe the answer should be yes - if we as defenders would not continuously try to disrupt adversarial systems, we would essentially give in to cyber crime and at the same time sacrifice users who could have been saved from being a victim. We can see the validity of this argument, if we scale it to a macroscopic level, for example: *people should not use a virus scanner, because otherwise malware would evolve that could no longer be reliably detected by an antivirus, so let's rather get infected by simple malware.* Single cyber security measures are unlikely to change the threat landscape once and for all, but what we can hope for is they become enough of a hindering to actors to pursue their activities at the level they currently do and with ease. Philosophically speaking, what security measures can therefore accomplish is that they might price lower end adversaries out of the market as we make the juice too sour and no longer worth the effort. A well-funded, well-equipped and well determined attacker will always find a way around any measure devised.

10 CONCLUSIONS

DGAs are widely used in malware families, allowing malware authors to dynamically change the location of the infrastructure and thereby protecting against takedowns or blacklists. DGAs are however efficiently detected by defenders due to the traces in the network of the victim, and consequently preemptively blacklisted. For this reason, malware authors find and exploit new paradigms to dynamically update the location of their infrastructure. In this paper, we report on malware coordination using the Bitcoin blockchain, signaling the location of the infrastructure through specific Bitcoin transactions. This allows the malware authors to instantaneously update the IP address contacted by the malware, and is harder to block using network-based anomaly detection.

We follow the evolution of the Pony malware leveraging this concept for 12 months. During this time, we are able to identify how the malware authors updated their malware and cycled their infrastructure. We also show the flaw in the signaling strategy as used by the malware authors, that we eventually use to take over command of the malware and obtain unique insights into malware infection behavior.

REFERENCES

[1] 2006. McAfee, "W32/sality.m". Malware description by McAfee.

[2] Jasper Abbink and Christian Doerr. 2017. Popularity-Based Detection of Domain Generation Algorithms. In *2nd International Workshop on Malware Analysis*.

[3] Syed Taha Ali, Patrick McCorry, Peter Hyun-Jeen Lee, and Feng Hao. 2015. Zombiecoin: Powering next-generation botnets with bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 34–48.

[4] Syed Taha Ali, Patrick McCorry, Peter Hyun-Jeen Lee, and Feng Hao. 2018. ZombieCoin 2.0: managing next-generation botnets using Bitcoin. *International Journal of Information Security* 17, 4 (2018), 411–422.

[5] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 34–51.

[6] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From throw-away traffic to bots: detecting the rise of DGA-based malware. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. 491–506.

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

[8] Mathis Baden, Christof Ferreira Torres, Beltran Borja Fiz Pontiveros, and Radu State. 2019. Whispering Botnet Command and Control Instructions. In *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 77–81.

[9] Federica Bisio, Salvatore Saeli, Pierangelo Lombardo, Davide Bernardi, Alan Perotti, and Danilo Massa. 2017. Real-time behavioral DGA detection through machine learning. In *2017 International Carnahan Conference on Security Technology (ICCST)*.

[10] Blueliv. 2018. The Credential Theft Ecosystem. https://www.blueliv.com/the-credential-theft-ecosystem/

[11] Christian J Dietrich and Christian Rossow. 2009. Empirical research of ip blacklists. In *ISSE 2008 Securing Electronic Business Processes*. Springer, 163–171.

[12] Kobi Eisenkraft and Arie Olshtein. 2019. *Pony's C&C servers hidden inside the Bitcoin blockchain.* Technical Report. Check Point. https://research.checkpoint.com/2019/ponys-cc-servers-hidden-inside-the-bitcoin-blockchain/

[13] Europol. 2018. Five Arrested for Spreading Ransomware Throughout Europe and US. https://www.europol.europa.eu/newsroom/news/five-arrested-for-spreading-ransomware-throughout-europe-and-us

[14] Davor Frkat, Robert Annessi, and Tanja Zseby. 2018. Chainchannels: Private botnet communication over public blockchains. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE.

[15] William J. Gordon, Adam Wright, Ranjit Aiyagari, Leslie Corbo, Robert J. Glynn, Jigar Kadakia, Jack Kufahl, Christina Mazzone, James Noga, Mark Parkulo, Brad Sanford, Paul Scheib, and Adam B. Landman. 2019. Assessment of Employee Susceptibility to Phishing Attacks at US Health Care Institutions. *JAMA Network Open* 2, 3 (03 2019). https://doi.org/10.1001/jamanetworkopen.2019.0393

[16] Harm Griffioen, Tim M. Booij, and Christian Doerr. 2020. Quality Evaluation of Cyber Threat Intelligence Feeds. In *International Conference on Applied Cryptography and Network Security (ACNS)*.

[17] Guofei Gu, Junjie Zhang, and Wenke Lee. 2008. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSSâĂŹ08)*.

[18] Philip Koshy, Diana Koshy, and Patrick McDaniel. 2014. An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*. Springer, 469–485.

[19] Chaz Lever, Platon Kotzias, Davide Balzarotti, Juan Caballero, and Manos Antonakakis. 2017. A lustrum of malware network communication: Evolution and insights. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 788–804.

[20] Kevin Liao, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. 2016. Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In *2016 APWG Symposium on Electronic Crime Research (eCrime)*. IEEE, 1–13.

[21] Hieu Mac, Duc Tran, Van Tong, Linh Giang Nguyen, and Hai Anh Tran. 2017. DGA botnet detection using supervised learning methods. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*.

[22] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*. 127–140.

[23] Satoshi Nakamoto. 2008. *Bitcoin: A peer-to-peer electronic cash system*. Technical Report. Manubot.

[24] Kris Oosthoek and Christian Doerr. 2020. From Hodl to Heist: Analysis of Cyber Security Threats to Bitcoin Exchanges. (2020).

[25] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoit Dupont. 2019. Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity* 5, 1 (2019), tyz003.

[26] Stijn Pletinckx, Cyril Trap, and Christian Doerr. 2018. Malware Coordination using the Blockchain: An Analysis of the Cerber Ransomware. In *IEEE Conference on Communications and Network Security*.

[27] Daniel Plohmann, Khaled Yakdan, Michael Klatt, Johannes Bader, and Elmar Gerhards-Padilla. 2016. A comprehensive measurement study of domain generating malware. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*.

[28] Fergal Reid and Martin Harrigan. 2013. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*. Springer, 197–223.

[29] Dorit Ron and Adi Shamir. 2013. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*. Springer, 6–24.

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

[31] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. 2014. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*. Springer, 457–468.

[32] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. 2009. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (Chicago, Illinois, USA) *(CCS '09)*. Association for Computing Machinery, New York, NY, USA, 635–647. https://doi.org/10.1145/1653662.1653738

[33] Mingkai Tong, Xiaoqing Sun, Jiahai Yang, Hui Zhang, Shuang Zhu, Xinran Liu, and Heng Liu. 2019. D3n: Dga detection with deep-learning through nxdomain. In *International Conference on Knowledge Science, Engineering and Management*. Springer.

[34] Duc Tran, Hieu Mac, Van Tong, Hai Anh Tran, and Linh Giang Nguyen. 2018. A LSTM based framework for handling multiclass imbalance in DGA botnet detection. *Neurocomputing* 275 (2018).

[35] R Vinayakumar, KP Soman, Prabaharan Poornachandran, Mamoun Alazab, and Alireza Jolfaei. 2019. DBD: deep learning DGA-based botnet detection. In *Deep Learning Applications for Cyber Security*. Springer, 127–149.

[36] Andrey Yakovlev. 2019. The Dark Side of Russia: How New Internet Laws and Nationalism Fuel Russian Cybercrime. https://intsights.com/resources/how-new-internet-laws-and-nationalism-fuel-russian-cybercrime

[37] Luhui Yang, Guangjie Liu, Jiangtao Zhai, Yuewei Dai, Zhaozhi Yan, Yuguang Zou, and Wenchao Huang. 2018. A novel detection method for word-based DGA. In *International Conference on Cloud Computing and Security*. Springer, 472–483.

[38] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. 2019. Tracing transactions across cryptocurrency ledgers. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 837–850.

[39] Bin Yu, Daniel L Gray, Jie Pan, Martine De Cock, and Anderson CA Nascimento. 2017. Inline DGA detection with deep networks. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 683–692.

[40] Yonglin Zhou, Qing-shan Li, Qidi Miao, and Kangbin Yim. 2013. DGA-Based Botnet Detection Using DNS Traffic. *J. Internet Serv. Inf. Secur.* 3, 3/4 (2013).

A BITCOIN ADDRESSES

Table 1: Abbreviation of Bitcoin addresses

Bitcoin address	Abbreviation
1BkeGqpo8M5KNVYXW3obmQt1R58zXAqLBQ	1BkeG
1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y	1CeLg
19hi8BJ7HxKK45aLVdMbzE6oTSW5mGYC82	19hi8
1PFSS4kdTxvVhrti4fM3jK9FLhUt5zZf6i	1PFSS
1N94rYBBCZSnLoK56omRkAPRFrpr5t8C1y	1N94r

Table 2: Summary of abused Bitcoin addresses as of August 13, 2020

Abbreviation	First seen	Last seen	Transactions
1BkeG	August 28, 2019	August 13, 2020	427
1CeLg	August 27, 2019	December 11, 2019	229
19hi8	December 10, 2019	August 13, 2020	582
1PFSS	September 16, 2019	December 10, 2019	37
1N94r	December 10, 2019	August 13, 2020	919

Table 3: Bitcoin addresses identified using Orbit

Address
1NvZiEcbwJb5K1korvVxwgbyiqo1xAWYJc
1P4NJHpyVUGa6W5PVr199dbHfj8WrPZrEv
1P96228rJLwcz1c7nEwVhLY6eVq5oztggz
1Pf9BDgoYYS1CsLbjDThuXpJHvtihS1Y8Y
1Prp6CayFNqVECivfGUsBGxi1pd4GqCes4
1PsyExbT4umCdX1gQLJykXdw1heZBCjhoD
1PW1nBXj1J8fauCF849WxoWX4RiNgb8umZ
1Q2UrUiAtRi7zJc3K5GeXCb4emtm5Jwt4Q
1QFRXDRJ2r5kspqnmLZWJham4w8ShYcvTe
1WXfZpsVHTFE8f7Zo67JPtJpyVaiYocLi
12WTUYP1rB8o41qJgRVLGgYQwKyqdxxHJ7
13bFSyRuADZaAA789hPtWSsEAX39k5dHAo
13gnMm2yQokbqBdYVjjyD2H2qWASNXFRRd
13P7pi2sqVHVUqusc3aCxuncnn4xBKBpTt
13Sg1cgzyk4aFtqSgMtX1TRrb5r8Fq4Hhb
16gMQTkrYhBamJt1rBFYdEa7m4z9kpjcsk
18xUPexL5vWwEn7wx9NxJmTr4NKCMpU9YV
19J6yko1EQjkMP7PggCnYKrzxzdNLXX3AJ
1Ad9dCSmkAVgrkri5Pg5HQkLn8sWqAoMNJ
1AieTH8jpSYGCfgh7yuYQkXxNanZBD1tHj
1ARzwaew4ZQHAPWwwnhhJSFwWBqY9LgFLh
1BpvQKPpxgJx5fBbEaNddwbskWqHsJWFA4
1BQoxrf8sdqfukSetH9ovhdzVJTHgp16yK
1CQKE5ydCRm8ojoe4HMRqydtNrWCVEEMc4
1DEwsFaHqWVr6ZB9ThturHedmbt6uoTPfD
1DhLJr2NFFuQuDDcziiTSSh6tyuJyuicSZ
1Dv8fsUUf4hds64WsnUvp9fUuCJgnPEFUz
1F9pidNKVh8AqAG2bkea3zzbQi5NALmYyJ
1G6EvUZcan2sBEE7omyYqFeEFcvEnZAZZP
1GCayv4GrMKW8BxRQsLCko7H7mso5dVaWR
1GtpPsfNSPmar53pAY2gNEWtGS3WtnJ7Pk
1HGJRJp1mzKdpnxQGgUaVzLaqNff8DiR82
1LFJkLe4smeBC2zufL3PH7rUBJ3x9i4jaN
1LH3j3VMrcVCCYn8jVYSoCifywkYxQzquC
1LRGS5niSw2bYRchCzQtxARUxEVR8VQ2Zv
1Men8hUQjqUTZY5hi4w6YniB13taNkcxcp
1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y
1PFSS4kdTxvVhrti4fM3jK9FLhUt5zZf6i
1N94rYBBCZSnLoK56omRkAPRFrpr5t8C1y
1BkeGqpo8M5KNVYXW3obmQt1R58zXAqLBQ

B C&C IP INFORMATION

Table 4: C&C IPs decoded from 1BkeG: August 28, 2019 (08-28) to March 26, 2020 (03-26)

ID	Date	C&C IP	ID	Date	C&C IP
1	08-28	195.123.227.99	54	12-24	195.123.220.236
2	09-03	212.73.150.254	55	12-24	195.123.245.106
3	09-06	185.234.72.50	56	01-09	185.14.30.190
4	09-11	91.200.100.136	57	01-13	195.123.233.133
5	09-11	91.200.100.136	58	01-14	5.34.177.9
6	09-13	91.200.100.66	59	01-16	195.123.220.107
7	09-16	91.200.100.136	60	01-19	195.123.218.204
8	09-16	91.200.100.134	61	01-20	195.123.222.104
9	09-17	91.200.100.134	62	01-21	195.123.218.204
10	09-17	172.105.69.5	63	01-21	195.123.222.114
11	09-19	172.104.54.151	64	01-22	85.217.171.218
12	09-21	78.108.216.39	65	01-23	185.14.28.186
13	09-23	91.200.103.136	66	01-26	5.34.178.122
14	09-24	91.200.102.39	67	01-27	195.123.234.110
15	09-25	91.200.100.174	68	01-28	195.123.233.167
16	09-29	185.203.118.16	69	01-29	5.34.177.65
17	09-30	85.217.170.51	70	01-30	195.123.240.67
18	10-02	185.203.119.18	71	02-03	45.90.57.16
19	10-05	94.156.35.216	72	02-04	195.123.246.145
20	10-08	185.177.59.149	73	02-04	195.123.226.86
21	10-09	85.217.171.48	74	02-05	195.123.234.158
22	10-13	185.203.119.169	75	02-06	195.123.233.231
23	10-14	185.203.117.49	76	02-07	195.123.234.158
24	10-16	185.203.116.47	77	02-09	5.34.178.60
25	10-17	185.203.117.49	78	02-10	195.123.246.145
26	10-21	94.156.35.35	79	02-10	45.90.57.186
27	10-22	185.203.116.121	80	02-15	195.123.228.78
28	10-23	94.156.144.25	81	02-17	195.123.246.145
29	10-24	91.92.136.33	82	02-17	82.118.21.170
30	10-26	94.156.144.27	83	02-18	195.123.246.145
31	10-29	91.200.103.110	84	02-18	195.123.234.158
32	10-31	91.200.100.4	85	02-24	195.123.225.9
33	11-04	94.156.189.124	86	02-25	185.234.72.142
34	11-12	193.37.212.24	87	02-26	195.123.225.11
35	11-13	185.205.209.13	88	03-01	193.37.213.160
36	11-14	185.205.210.51	89	03-02	195.123.226.189
37	11-17	185.205.210.96	90	03-03	195.123.226.191
38	11-18	185.203.117.29	91	03-11	195.123.227.225
39	11-19	94.156.144.42	92	03-12	195.123.226.190
40	11-20	193.37.213.133	93	03-12	195.123.226.81
41	11-24	185.203.116.157	94	03-15	195.123.225.152
42	11-25	85.217.171.43	95	03-16	195.123.226.81
43	12-03	94.156.35.31	96	03-16	195.123.225.170
44	12-10	94.156.189.251	97	03-17	195.123.226.81
45	12-11	94.156.189.251	98	03-17	109.94.110.80
46	12-11	195.123.220.179	99	03-18	195.123.226.135
47	12-11	195.123.220.191	100	03-20	195.123.226.81
48	12-16	195.123.220.222	101	03-22	142.202.188.245
49	12-17	185.205.210.132	102	03-23	142.202.188.243
50	12-18	195.123.220.236	103	03-24	142.202.188.241
51	12-22	195.123.220.77	104	03-25	195.123.225.240
52	12-23	195.123.220.236	105	03-26	195.123.224.42
53	12-23	185.14.29.65	106	03-26	195.123.224.44

Table 5: C&C IPs decoded from 1BkeG: April 5 (04-05) to August 13, 2020 (08-13)

ID	Date	C&C IP	ID	Date	C&C IP
107	04-05	185.141.61.178	153	06-08	151.80.194.90
108	04-06	185.205.210.43	154	06-09	137.74.131.213
109	04-07	193.37.212.100	155	06-11	137.74.131.213
110	04-08	94.156.35.48	156	06-11	185.81.98.49
111	04-09	94.156.35.14	157	06-11	185.234.72.106
112	04-10	193.37.212.100	158	06-12	137.74.157.159
113	04-11	94.156.189.44	159	06-16	185.234.72.106
114	04-14	185.141.61.81	160	06-16	137.74.157.159
115	04-15	94.156.189.48	161	06-17	51.254.87.67
116	04-15	185.141.61.81	162	06-17	142.202.190.44
117	04-15	142.202.190.18	163	06-23	172.86.75.54
118	04-17	94.156.35.48	164	06-24	161.35.105.177
119	04-17	142.202.190.18	165	06-25	45.61.138.160
120	04-19	94.156.189.113	166	06-30	172.105.104.213
121	04-20	91.92.136.155	167	07-01	45.61.136.140
122	04-22	142.202.190.17	168	07-03	139.180.214.192
123	04-22	185.141.61.81	169	07-03	142.202.188.204
124	04-22	142.202.190.17	170	07-06	45.61.139.50
125	04-23	142.202.190.26	171	07-07	45.61.136.126
126	04-26	142.202.188.236	172	07-08	45.61.136.168
127	04-27	142.202.188.216	173	07-08	51.38.94.172
128	04-28	91.200.101.10	174	07-09	172.105.48.152
129	04-29	142.202.188.249	175	07-09	140.82.0.67
130	04-29	142.202.190.34	176	07-14	45.79.126.239
131	05-06	185.203.118.73	177	07-14	45.79.126.239
132	05-07	82.118.21.243	178	07-14	45.77.63.37
133	05-11	91.200.102.153	179	07-15	51.38.94.172
134	05-12	142.202.188.254	180	07-17	95.179.243.62
135	05-18	94.156.144.97	181	07-17	51.38.94.172
136	05-19	195.123.240.92	182	07-19	149.248.7.219
137	05-19	142.202.188.248	183	07-20	149.248.7.219
138	05-20	142.202.188.254	184	07-20	51.38.94.172
139	05-21	142.202.190.22	185	07-21	139.180.165.173
140	05-26	142.202.190.19	186	07-21	139.180.165.173
141	05-27	142.202.190.43	187	07-21	139.180.165.173
142	05-28	142.202.190.19	188	07-21	51.38.94.172
143	05-28	142.202.190.55	189	07-22	172.105.59.15
144	05-28	185.177.59.58	190	07-22	172.105.59.15
145	06-01	195.123.228.246	191	07-24	134.122.24.253
146	06-02	185.177.59.58	192	07-24	134.122.24.253
147	06-03	185.203.119.165	193	07-27	185.92.222.127
148	06-03	212.73.150.176	194	07-28	142.202.188.249
149	06-03	195.123.228.197	195	07-29	167.172.200.71
150	06-04	94.156.189.177	196	07-31	45.61.138.109
151	06-04	142.202.190.43	197	08-03	45.61.139.16
152	06-05	151.80.194.85	198	08-13	67.205.148.45

C INFRASTRUCTURE COUNTRY LOCATION

Table 6: Summary of countries which C&C IP is located in

Countries	First seen	Last seen
Bulgaria	August 28, 2019	June 4, 2020
Germany	September 6, 2019	June 16, 2020
Singapore	September 19, 2019	July 3, 2020
Netherlands	December 11, 2019	July 27, 2020
Czechia	December 24, 2019	February 18, 2020
United States	January 26, 2020	August 13, 2020
Switzerland	February 3, 2020	February 10, 2020
Ukraine	February 17, 2020	May 7, 2020
Italy	June 5, 2020	June 8, 2020
France	June 9, 2020	July 21, 2020
United Kingdom	June 25, 2020	August 3, 2020
Canada	June 30, 2020	June 30, 2020
India	July 9, 2020	July 22, 2020
Japan	July 21, 2020	July 21, 2020

Table 7: Summary of abused providers

Provider	First seen	Last seen
ITL LLC	August 28, 2019	June 3, 2020
Belcloud LTD	September 3, 2019	June 4, 2020
combahton GmbH	September 6, 2019	June 16, 2020
Linode LLC	September 17, 2019	July 22, 2020
ITL-Bulgaria	December 24, 2019	May 7, 2020
DYNU	March 22, 2020	July 28, 2020
OVH SAS	June 5, 2020	July 21, 2020
WorldStream B.V.	June 11, 2020	June 11, 2020
BL Networks	June 23, 2020	August 3, 2020
DIGITALOCEAN-ASN	June 24, 2020	August 13, 2020
AS-CHOOPA	July 3, 2020	July 27, 2020