Analysis of a "/0" Stealth Scan from a Botnet

Alberto Dainotti[1], Alistair King[1], Kimberly Claffy[1], Ferdinando Papale[2], and Antonio Pescapé[2]

[1]CAIDA, University of California San Diego, La Jolla, CA USA
[2]Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Universitá degli Studi di Napoli Federico II, Italy

Botnets are the most common vehicle of cyber-criminal activity. They are used for spamming, phishing, denial of service attacks, brute-force cracking, stealing private information, and cyber warfare. Botnets carry out network scans for several reasons, including searching for vulnerable machines to infect and recruit into the botnet, probing networks for enumeration or penetration, etc. We present the measurement and analysis of a horizontal scan of the entire IPv4 address space conducted by the Sality botnet in February 2011. This 12-day scan originated from approximately 3 million distinct IP addresses, and used a heavily coordinated and unusually covert scanning strategy to try to discover and compromise VoIP-related (SIP server) infrastructure. We observed this event through the UCSD Network Telescope, a /8 darknet continuously receiving large amounts of unsolicited traffic, and we correlate this traffic data with other public sources of data to validate our inferences. Sality is one of the largest botnets ever identified by researchers, its behavior represents ominous advances in the evolution of modern malware: the use of more sophisticated stealth scanning strategies by millions of coordinated bots, targeting critical voice communications infrastructure. This work offers a detailed dissection of the botnet's scanning behavior, including general methods to correlate, visualize, and extrapolate botnet behavior across the global Internet.

I. Introduction

Botnets are collections of Internet hosts ("*bots*") that through malware infection have fallen under the control of a single entity ("*botmaster*"). Botnets of up to few million hosts have been observed [4], [24], [58]. Innocent users carry on with their legitimate activities, unaware that their infected PCs are executing various types of malicious activity in the background, including spamming, phishing, denial-of-service (DOS) attacks, brute-force password cracking, stealing of credentials, espionage and cyber warfare. The news media and scientific literature have documented many criminal activities carried out by botnets over the last few years [15], [21], [36], [63], including on mobile phones [46].

We would like to thank: Joe Stewart of SecureWorks for helping us to identify the sipscan binary; Ken Chiang at Sandia National Labs for helping reverse engineering the binary; Saverio Niccolini at NEC for brainstorming on the SIP header of the sipscan; and Marco Stendardo for helping with the scripts used in the analysis. We are also grateful to all the CAIDA folks for their support, and in particular to Dan Andersen for enabling the storage, transport, and processing of massive data volumes on systems available 24/7. Support for the UCSD network telescope operations and data collection, curation, analysis, and sharing is provided by NSF CRI CNS-1059439, DHS S&T NBCHC070133 and UCSD. Antonio Pescapè was partially funded by PLATINO (PON01_01007) by MIUR and by a Google Faculty Award for the UBICA project.

Botnets perform network scanning for different reasons: propagation, enumeration, penetration. One common type of scanning, called "horizontal scanning", systematically probes the same protocol port across a given range of IP addresses, sometimes selecting random IP addresses as targets. To infect new hosts in order to recruit them as bots, some botnets, e.g., Conficker [28], [50], perform a horizontal scan continuously using self-propagating worm code that exploits a known system vulnerability. In this work we focus on a different type of botnet scan – one performed under the explicit command and control of the botmaster, occurring over a well-delimited interval.

Several botnets have been analyzed in the literature, including characterizing botnet scanning techniques either based on packet captures from darknets and honeynets [42], [43], or by examining botnet source code [11]. Documented scans by botnets have been of relatively small size (e.g. around 3000 bots) [43] and lightly coordinated, e.g., many bots randomly (typically uniformly randomly [43]) probing the same target address range.

In February 2011, the UCSD /8 Network Telescope instrumentation [7] captured traffic reflecting a previously undocumented large-scale stealth scanning behavior (across the entire IPv4 space, we believe) from a botnet using about 3 million unique source IP addresses. We identified the malware responsible for this massive and sophisticated scanning activity as a binary module of the Sality botnet [24] known to target SIP (Session Initiation Protocol [53]) servers [23]. We hence refer to this interesting scanning event as *"sipscan"* throughout the rest of this paper.

Our contributions in this study include techniques to characterize a large-scale intentionally surreptitious scan of the entire IPv4 space (that is, a *"/0"* scan), including use of additional data to confirm that the scan was not using spoofed source IP addresses, but rather was being sourced by a large botnet. We correlated darknet traffic over this period with two other publicly available sources of Internet traffic data that strongly suggest the scan was not just of this /8 but over the entire IPv4 Internet address space. Finally, we created animations and visualization to help us understand the strictly ordered progression of the entire /0 scan, and correlate its address space and geographic coverage with its traffic volume. These tools also enabled us to delineate different phases of its scanning activity and its adaptation to changing network conditions. These methods and tools have already yielded

substantial insight into the first observed /0 scan by a botnet, but we anticipate a wide range of applicability to other analyses of unidirectional or even bidirectional traffic.

Section II summarizes related work. Sections III describes the anatomy of the scan, including high-level characteristics and validation that it was indeed carried out by a botnet targeting the entire IPv4 space. Section IV analyzes more detailed properties of the scan, including the impressively covert scanning strategy, bot turnover rate, coverage and overlap in target addresses, and highly orchestrated adaptivity and coordination of the bots. Section VI summarizes our findings and contributions.

II. Background and Related Work

Botnets have been an active area of research for almost a decade, starting with early generation botnets that used IRC channels to implement centralized Command & Control (C&C) infrastructures [9], [17]. In 2007 the Storm botnet signaled a new generation of botnet capabilities, including the use of peer-to-peer protocols to support distributed C&C channels [34], [55], [62]. These botnets are harder to detect and dismantle because there is no single point of failure, and they often use sophisticated techniques such as encrypted communication [62] and *Fast flux* DNS resolution [14]. Researchers have also studied methods for automated discovery of botnets [32], [44], [60], formal models of botnet communication [16], [18], and their use for orchestrated spam campaigns [39], [49].

Botnets commonly scan large segments of Internet address space, either seeking hosts to infect or compromise, or for the purpose of network mapping and service discovery. Analyzing and detecting these events can improve our understanding of evolving botnet characteristics and spreading techniques, our ability to distinguish them from benign traffic sources, and our ability to mitigate attacks. But analysis of network probing activities of botnets has received little attention in the literature.

In 2005, Yegneswaran, Barford, and Paxson analyzed six months of network traffic captured by honeynets [66]. Based on statistical properties of traffic, they characterized and classified 22 large-scale events into three categories: worm outbreaks, misconfigurations, and botnet probings. These first-generation botnets were less evolved in several ways than those we see today: in size (a maximum of 26,000 bots), scope (largest target scope was a /8 network), and communication capabilities (centralized IRC-based command and control). Li, Goyal, and Chen [42] analyzed traffic data they collected from 10 contiguous /24 networks operating as honeynets throughout 2006. Through analysis of the probing traffic they were able to infer properties of the botnet, e.g., geographical location of, and operating system running on infected machines. We use a similar approach to infer characteristics of the botnet scan we study in this paper. These three authors collaborated with Paxson on a more comprehensive analysis of data from both 2006 and 2007, which was corroborated both by data from the DShield project [35] and by the inspection of botnet source code [43]. Analyzing the traffic from 10 contiguous /24 darknets/honeynets they identified 203 botnet scans with different characteristics, all scanning at most a /8 network, and all with inferred bot populations significantly smaller (200-3700) than the February 2011 scan captured at our darknet (3 million IP addresses), They found that these first-generation botnets employed simple scanning strategies, either sequential or uniform random scanning, and elementary orchestration capabilities: many bots scanning the same address range independently, with high redundancy and large overlap in target addresses. Other studies have found similar results via examination of botnet source code to understand the scanning strategies [10], [11]. Barford and Yegneswaran [11] inspected four widely-used IRC botnet code bases, finding only primitive scanning capabilities with *"no means for efficient distribution of a target address space among a collection of bots"*. However these studies did not analyze any new-generation botnets.

The scan that we observe and analyze in this study differs from previous work in several ways: (i) it is recent (2011) and related to a new-generation, widely-deployed, peer-to-peer botnet (Sality [24]); (ii) it is observed from a larger darknet (a /8 network); (iii) the population of bots participating in the scan is several orders of magnitude larger; (iv) the target scope is the entire IPv4 address space; (v) it adopts a well-orchestrated stealth scan strategy with little redundancy and overlap.

This last point is the most surprising finding in terms of novelty and impact. The remarkably stealth scanning employed by new-generation botnets gives us reason to suspect that many large-scale scans may have occurred in recent years but gone unnoticed by any modern instrumentation for studying it. Despite the lack of any literature documenting the observation of highly coordinated large-scale network scans from botnets, the concept has been discussed, both in a worst-case theoretical analysis of attack potential [61], and for the more benign application of Internet-wide service discovery [40]. For service discovery, these authors considered a scan strategy based on reverse-byte sequential increments of target IP addresses, which they named "Reverse IP Sequential (RIS)". Although they dismissed this option for being difficult to extrapolate metrics from partial scans, we discovered that this was exactly the technique used by the Internet-wide scan ("sipscan") we study in this paper (Section IV-A). Heidemann's *et al.* reachability census was Internet-wide but ran independently from two hosts, not coordinated in the way botnets are [33].

Another relatively novel aspect of the scan we analyze is that it targets SIP infrastructure, which is not typically in published lists of services probed by botnets [43]. Only in the past 3 years have SIP servers been reported as the object of large-scale attacks [51], [56], [68]. As more of the world's voice communications move to an IP substrate, fraudulent activity targeting SIP-based VoIP services offers an attractive source of revenue to cybercrime [25]. In April 2010, Sheldon reported a series of brute-force password-guessing attacks on SIP servers worldwide, sourced from the Amazon EC2 cloud [56], [68]. Later in 2010, several sources reported on a new malware named "*sundayaddr*", which behaved like a few-hundred node botnet comprised of unix-like machines (e.g., Linux, FreeBSD) trying to brute-force accounts on SIP servers [31], [51]. The layout of the SIP headers in the attacking

packets was almost identical to that of SIPVicious, a tool suite written in Python designed to perform security auditing of SIP services [27]. It seems therefore likely that the attack code was a slightly modified version of SIPVicious [51].

In November 2010, the author of SIPVicious reported another large-scale attack against several SIP servers worldwide, using a more significantly different SIP header than used by SIPVicious [26], [57]. Both of these events were reported by several parties and were largely discussed on public SIP operational mailing lists [6], [8]. In contrast, to the best of our knowledge the scan that we document in this study was not publicly reported with respect to either observed network traffic or server activity (e.g. logs). Symantec identified and analyzed the binary responsible for what we call the "sipscan", which they discovered while monitoring Sality, a large peer-to-peer based botnet [23], [24]. A host infected by Sality downloads the scanning binary via a component of the main bot executable, which is responsible for downloading and executing additional malware whose URLs are communicated by other botnet peers [24]. During our analysis we had access to the same binary code and verified that it matches the SIP headers we observed in the sipscan. Symantec did not publish any information about the stealth scanning strategy or in particular on the reverse byte order adopted by the sipscan (Section IV-A). Our study, based instead on network traffic measurement and analysis, is complementary to what has been found by reverse-engineering the code running on the bots, showing novel insights into the botnet population and the orchestration and coordination of the scan. Since Sality is one of the largest known botnets but relatively undocumented in research literature, another contribution of our study is to shed light on the scanning behavior of this new-generation botnet.

III. Analysis Part I: Anatomy of the Sipscan

A. Overview

The sipscan probes each target IP address with two packets: (i) a UDP packet to port 5060 carrying a SIP header and (ii) a TCP SYN packet attempting to open a connection on port 80. We observe the sipscan at a darknet – i.e., there are no devices on it responding to incoming traffic – so we do not observe any further packets for the same flows except for TCP SYN retransmits.

Figure 1 depicts the SIP header of the packets sent by the sipscan. This SIP header is a request to register a random user account on a SIP server, but random account registrations are usually not accepted by SIP servers. Thus, if the targeted host is a SIP server, the registration will likely fail but will result in a "404 Not Found" response code, which is enough to reveal to the bot that the target is indeed a SIP server. We presume that the goal is to identify SIP servers for later use, e.g., to perform brute-force attempts to register user accounts.

The sipscan SIP header is similar to the header built by the SIPVicious security auditing tool suite to generate probe packets [27]. In November 2010, the author of this tool reported a large distributed attack against SIP servers with headers similar to those his tool used; this attack was observed by several parties and was likely carried out by a botnet [26], [57]. In the case of both the November 2010 scan and the February 2011 scan we observed, the botnet developers probably used the Python code of SIPVicious as a reference to write their attack code. The most notable difference between such attacks and SIPVicious headers is in the "User-Agent" header, where the attack code replaced the string "friendly-scanner" with the less suspicious "Asterisk PBX"[1].

The observed sipscan header has two distinctive characteristics compared to the attack of November 2010 (and in general compared to the miscellaneous SIP malware packets observed at the UCSD telescope): the user name, which is always composed of ten digits, and the "To:"/"From:" fields, which contains a SIP URI instead of simply the number [53]. Based on the properties of its SIP header, we defined a payload signature to identify all the sipscan packets seen by the UCSD Network Telescope. Each source host sends the TCP packet together with the UDP packets, allowing us to easily infer which TCP SYN packets on port 80, among all those received by the telescope, were associated with the sipscan).

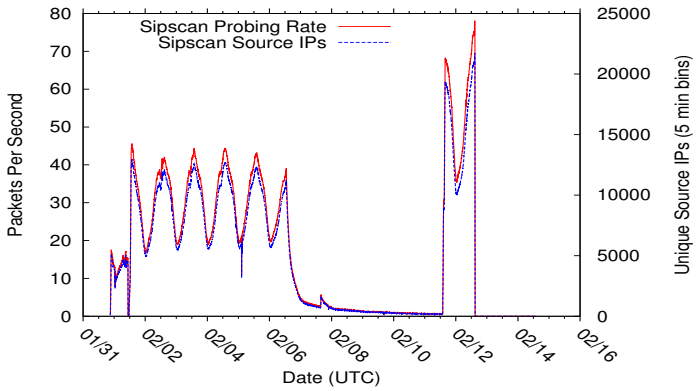


Fig. 2: Overview of the scan. The continuous line shows the packets per second, in 5 minute bins, of UDP probing packets from the sipscan observed by the UCSD Network Telescope. The dashed line represents the corresponding number of distinct source IP addresses per bin.

Figure 2 shows the packet rate of the sipscan UDP packets (left axis) and the number of unique IPs per hour (right axis) sending such packets to addresses in the UCSD Network Telescope. The scan goes through different phases over approximately 12 days: it starts with a packet received on Monday 31 January 2011 at 21:07 UTC, and ends with a sharp drop of packets on Saturday 12 February around 15:00 UTC. Approximately 100 residual packets were observed in the following two days. During the scan, peaks of 21,000 hosts with distinct IPs probed the telescope's /8 address space in a single 5-minute interval.

Table I lists the main characteristics of the scan. The portion of the scan observed by the UCSD Network Telescope involved around 3 million distinct source addresses, generating 20 million probes – we define a probe as a UDP scanning packet with the payload signature from Figure 1, plus TCP SYN packets to the same destination. These probes covered

[1]Asterisk is a widely deployed open-source PBX software supporting both PSTN and VoIP.

```
1  2011-02-02 12:15:18.913184 IP (tos 0x0, ttl 36, id 20335, offset 0, flags [none], proto UDP
       (17), length 412) XX.10.100.90.1878 > XX.164.30.56.5060: [udp sum ok] SIP, length: 384
2          REGISTER sip:3982516068@XX.164.30.56 SIP/2.0
3          Via: SIP/2.0/UDP XX.164.30.56:5060;branch=1F8b5C6T44G2CJt;rport
4          Content-Length: 0
5          From: <sip:3982516068@XX.164.30.56>; tag=14718138184028634232183426 68
6          Accept: application/sdp
7          User-Agent: Asterisk PBX
8          To: <sip:3982516068@XX.164.30.56>
9          Contact: sip:3982516068@XX.164.30.56
10         CSeq: 1 REGISTER
11         Call-ID: 4731021211
12         Max-Forwards: 70
```

Fig. 1: Example of the payload of a UDP packet generated by the sipscan (line 1 is tcpdump output [5] with timestamp and information from IP and UDP headers). The payload contains a SIP request to register a user on the contacted host. A variant of the signature (which we also matched) has the string ":5060" appended to the "Contact: " header field (line 9). In the figure we replaced the value of the most significant byte of the destination address with "XX".

# of probes (1 probe = 1 UDP + multiple TCP pkts)	20,255,721
#of source IP addresses	2,954,108
# of destination IP addresses	14,534,793
% of telescope IP space covered	86,6%
# of unique couples (source IP - destination IP)	20,241,109
max probes per second	78.3
max # of distinct source IPs in 1 hour	160,264
max # of distinct source IPs in 5 minutes	21,829
average # of probes received by a /24	309
max # of probes received by a /24	442
average # of sources targeting a destination	1.39
max # of sources targeting a destination	14
average # of destinations a source targets	6.85
max # of destination a source targets	17613

TABLE I: Summary of the scanning event characteristics. The scan originated from almost 3 million distinct IP addresses and hit about 14.5 million addresses of the address space observed by the UCSD Network Telescope.

more than 14.5 million target IP addresses, that is, 86.6% of the darknet address space.

B. Verification of unspoofed source addresses

Because darknet addresses do not respond to received packets, we cannot generally assume that packets are not using spoofed (fake) source IP addresses. Effective scanning requires the use of real source addresses to receive responses, so there is reason to assume that these IP addresses are not spoofed. Conversely, evidence that the addresses are not spoofed would increase our confidence in the hypothesis that this behavior is in fact a large-scale scan. We found the following evidence that the observed packets were not actually spoofed.

- In [20] we studied the country-wide outage that occurred in Egypt between the 27th of January and the 2nd of February 2011. During the last two days of the outage - which overlap with the period of activity of the sipscan - most of the country was completely isolated from the rest of the Internet. We verified that no sipscan packets with source IP addresses that geolocated to Egypt were observed by the telescope during the outage. Figure 3 shows the re-announcement of all the BGP prefixes geolocated to Egypt that were withdrawn during the outage (continuous line, left y axis), and the packet rate of UDP packets from the sipscan geolocated to the same country (dashed line, second y axis). The graph shows Egyptian hosts contributing to the scanning activity only after the country is reconnected to the Internet. We used the same methodology described in [20] to analyze BGP data from the RIPE RIS [3] and Routeviews [64] repositories, and geolocation data from MaxMind [45] and Afrinic [1].
- Random IP spoofing would use also source IPs from our /8 darknet set of addresses, which we never see in this set of packets. We also mapped the source addresses of the scan to originating ASes (autonomous systems, or independent networks in the global routing system) using BGP data, and verified that they matched only assigned ranges of IP addresses.
- In Section III-D we analyze source port numbers in transport-layer headers from selected scanning bots. The consistency of these parameters over time suggests that the source addresses are not spoofed: IP spoofing requires the use of raw sockets and usually involves random selection of spoofed addresses, whereas the progression of source ports followed by these bots is typical of packets sent through standard sockets that use ephemeral ports assigned by the operating system based on a single, global counter.

C. Botnet activity

This convincing evidence that the source IP addresses are authentic supports our hypothesis that a botnet is generating the packets, rather than one or a few hosts, or a worm spreading. Over the course of twelve days, we observed about 3 million source addresses, which mapped to countries and networks all over the world (Section IV-D). Figure 2 displays a clearly delimited beginning and end of the behavior, with strong diurnal periodicity and variations of intensity. Spreading worms tend to exhibit closer to exponential growth in IP addresses infected and trying to spread further [69].

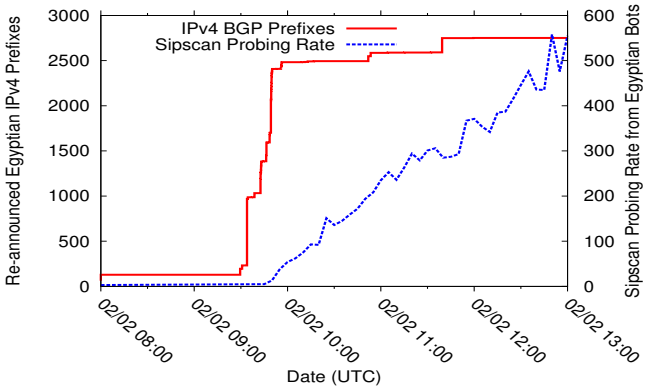


Fig. 3: The case of the Internet black-out in Egypt helps to verify that source addresses from the sipscan are not spoofed. The continuous line shows the reannouncement of routes to Egyptian IPv4 prefixes when the country reappears on the Internet on 2 February 2011. The sipscan starts approximately on the 1 February, but we start seeing probes from source IPs geolocated to Egypt only when the Egyptian networks get reannounced through BGP updates.

We discovered an even more compelling piece of evidence that this traffic was generated by a botnet, when we examined traffic data during the nation-wide censorship episode happened in 2011 in Egypt. In [20] we showed that, during the Egyptian outage, some Conficker-infected hosts were still able to randomly send infecting packets to the Internet, even if they were in networks not visible via BGP. Outbound connectivity (from Egyptian hosts "upstream" to the rest of the Internet) was still possible from some networks in Egypt through the use of default routes. But while we saw Conficker traffic originating from IPs geolocated in Egypt, we saw no sipscan traffic from Egypt, consistent with the sipscan hosts not acting independently, but rather receiving instructions from a command & control 'botmaster' host (i.e., requiring bidirectional connectivity) outside of Egypt.

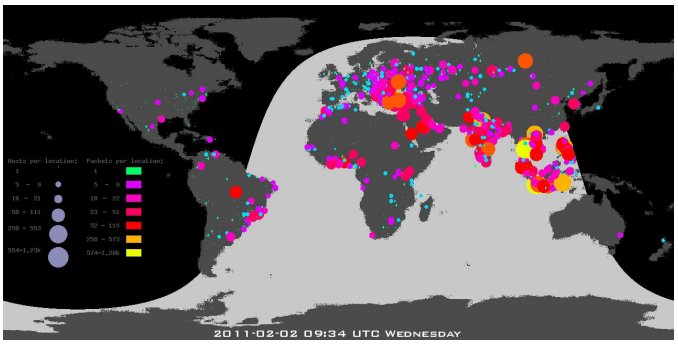


Fig. 4: Snapshot of our "World Map" animation of the sipscan available at [13] (Wed Feb 2 09:34:00 2011). The animation shows, in 5:20-minutes of data represented per frame, circles at the geographical coordinates of source hosts (bots) with size proportional to the number of hosts geolocated to those coordinates, and color to the number of packets sent. The animation depicts the spatial and temporal dynamics of the scan.

To simultaneously represent both the temporal and spatial dynamics of the event, we created a "World Map" animation available at [13]. Figure 4 is a single frame of the animation

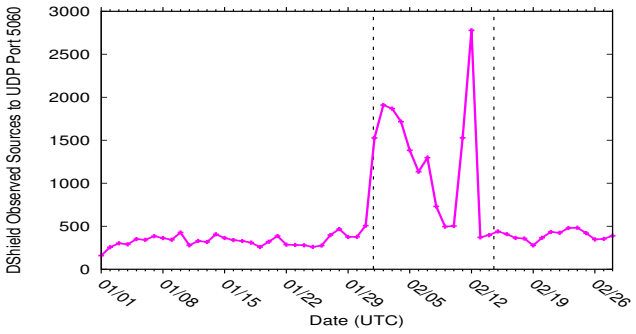


Fig. 5: Daily count of unique source IP addresses in packets to port 5060 extracted from DShield sensor data [35]. The unique source IP count, for the months of January and February 2011, shows an increase of almost one order of magnitude between the 1st and the 12th of February. Its profile matches the sipscan shown in Figure 2, suggesting that sensors (darknets and honeynets) in other /8 networks received the same kind of traffic. The start and end times of the sipscan are denoted in this graph by the two dashed vertical lines.

(capturing a window of 5 minutes and 20 seconds of data) from Wed 2 Feburary 09:34:00 2011. The circles are centered at the geographical coordinates of source IP addresses. For each time bin, the size of the circle is proportional to the number of hosts geolocated to those coordinates, whereas the color reflects the number of packets sent (these two values are not proportional because, as we show in Section IV, there are both hosts sending a single probe and hosts sending multiple probes at different rates). The animation illustrates the traffic volume and geographic scope of the scan over time. Geolocation of IP addresses was done using the MaxMind GeoLite database released on March 1st, 2011, temporally proximate to the event [45]. The software used to create the animation is an improved version of the code originally developed at CAIDA by Huffaker *et al.* and available at [2]. The animation visually represents, for the first time, an Internet-wide scan conducted by a large botnet.

D. A "/0" scan

Observation from the UCSD Network Telescope is limited to packets destined to the corresponding /8 network. However, we also discovered evidence that the scan targeted the entire IPv4 address space (a /0 scan): similar traffic patterns observable on other network segments, and a continuity in source port usage in the packets we observed.

1) Targeting the UCSD Network Telescope

Even if approximately 15% of addresses of our darknet were not hit by the scan, the sipscan uniformly targeted the entire address range of the /8 network. In Section IV-C we show that the missing (15%) targets may be due to a specific configuration parameter that would trade completeness of IPv4 address space coverage for redundancy in the utilization of the bots.

2) DShield repository

We have found circumstantial evidence of sipscan traffic in the DShield repository [35]. DShield is a constantly updated repository of scanning and attack reports. In particular it reports aggregated data of traffic observed on several "sensors"

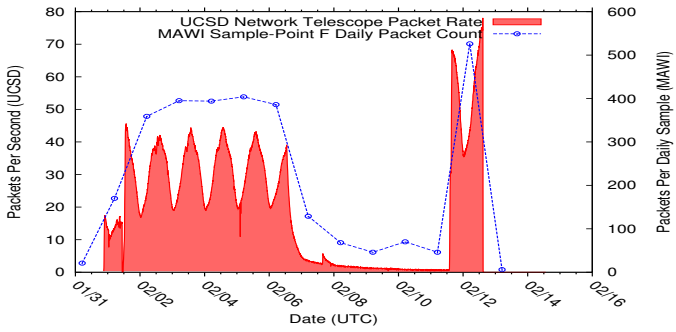


Fig. 6: Sipscan UDP packets observed by (i) the UCSD Network Telescope (y axis, packets per second) and (ii) MAWI WIDE Samplepoint-F (y2 axis, packets per daily sample of xx minutes). The samples found on the small link monitored by the MAWI working group perfectly follow the profile of the sipscan observed by the UCSD Network Telescope, strongly suggesting that the sipscan targeted also other /8 networks.

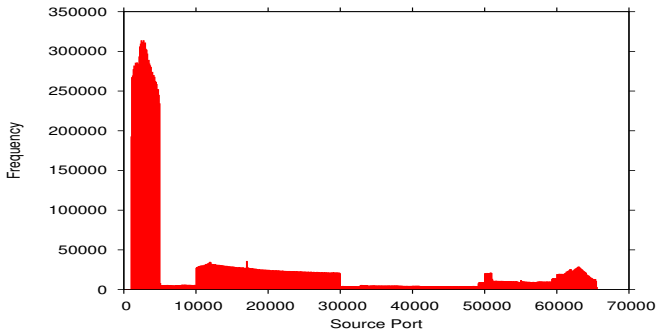


Fig. 7: Distribution of the source port numbers (bin size = 100) The most common range is 1025-5000, used by several versions of OSs from the Microsoft Windows family.

(i.e., small honeynets and darknets) operated by different participating organizations. Figure 5 shows the number of distinct source IP addresses per day observed by the DShield sensors on port 5060 from the 1 January to 28 February 2011. The large spikes in the traffic profile of the source IP addresses match the sipscan profile shown in Figure 2, indicating that the same phenomenon was probably targeting other networks besides the /8 monitored by the UCSD Network Telescope.

3) MAWI WIDE Samplepoint-F

We also examined traffic traces from a 150Mbps link on a trans-Pacific line that are made available by the MAWI WIDE project [30] (link "samplepoint-F"). The trace set is made of daily traces in pcap format, of 15 minutes each, where the IP addresses are anonymized and the transport-layer payload is removed [29]. This anonymization scheme prevented us for searching the trace specifically for the sipscan packets, since we can see neither the UDP payload signature nor the source IP addresses of the packets. Instead, from the analysis of the sipscan SIP headers (Figure 1), we built a flow-level signature with the following conditions for each UDP flow: (i) destination port 5060; (ii) made of a single packet; (iii) flow-size (in this case matching the packet size) between 382 and 451 bytes. We obtained the packet size range by examining all SIP header fields that were not fixed size, and how they varied (e.g. IP addresses in ascii format take between 7 and 15 bytes). We further sanitized the remaining flows considering some isolated cases of spikes in the MAWI traces which were using source ports outside of the most common ranges observed on the telescope (see Figure 7). The final result, depicted in Figure 6, is that there are almost no packets matching the flow-level signature in the days outside of the sipscan, whereas their profile during that period roughly follows the profile of the sipscan (The lack of tight precision between the two data sets in Figure 6 is due to the MAWI samples being coarser-grained, 15 minutes each once per day, and from a relatively small link).

This finding is important because the anonymization technique used for MAWI traces preserves matching prefixes and IP classes between IP addresses [29]. The analysis of this data therefore revealed that, on average, 8 different /8 classes were targeted every day by the packets traveling on this link.

4) Exploiting source-port continuity

The positive correlations of our data source with the DShield and MAWI data sources convinced us that the sipscan hit other /8 networks as well as our own. We also found the following evidence that the sipscan most likely targeted all the /8 networks in the IPv4 address space.

We identified a few bots scanning at a roughly constant pace over several days. Analyzing the sequence of source ports in their scanning packets revealed that some of these bots used incremental source ports within a specific range assigned by the operating system. For example, Windows XP and other Microsoft operating systems assign a new ephemeral source port in the range 1025-5000 by incrementing a global counter for each opened TCP or UDP socket [48]. We inferred how many other connections/sessions a bot opened between each probe sent to the darknet by following the sequence of source ports the bot used and "unwrapping" them, taking into account their range. In [43], Li *et al.* used the same methodology to estimate the global scope of botnet scans. We could only apply this technique to the few persistent bots (see Section IV-B) running on an operating system configured to assign source ports in this manner.

Figure 8 depicts the behavior of three of these bots (the bot number indicates its rank based on the number of probes they sent). The continuous lines represent the count of probes (a UDP packet plus at least one TCP SYN packet) observed by the UCSD Network Telescope (y axis), whereas the dashed lines represent the number of connections/sessions opened by each bot as inferred by unwrapping its source port numbers (second y axis). For each bot the two curves follow approximately the same trend, suggesting that the view from the telescope is representative of the global behavior of the bot. The UCSD Network Telescope covers 1/256th of the entire IPv4 address space, so a uniformly random scanning bot will probe this /8 darknet approximately every 256 probes, or every 512 new connections opened (every probe includes a UDP and TCP connection attempt). We find these subclass of bots actually hitting our darknet every 570 packets (on average), which would be consistent with their hosting computer opening other connections/sessions unrelated to the

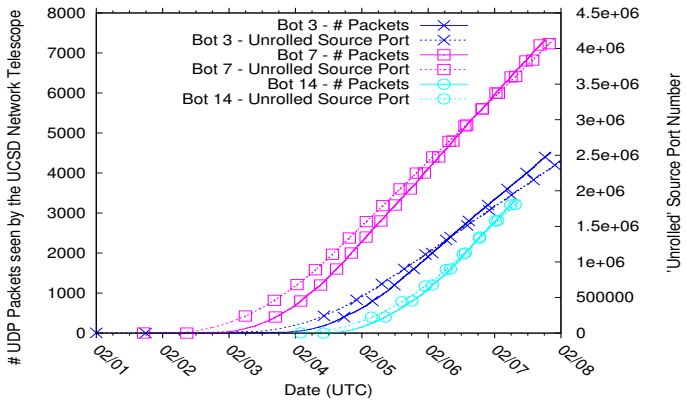


Fig. 8: Estimating the global scan scope by exploiting source port continuity in scanning bots: continuous lines represent the count of probes (a UDP packet plus at a least one TCP SYN packet) observed by the UCSD Network Telescope (y axis), whereas the dashed lines represent the number of connections/sessions opened by each bot as inferred by unwrapping its source port numbers (second y axis). Each bot probes the darknet on average (approximately) every 285 global probes, suggesting that during its absence reaches the remaining 255 /8 networks in the IPv4 address space.

scan, such as legitimate user activity or communication with the botmaster. In the next section we will show how the bots select their target IP addresses by first incrementing the most significant byte. Therefore we can assume that the external 255 probes from the bot reach all the other /8 networks in the IPv4 address space. In Section IV we will also explore another feature of the data in Figure 8: the bots proceed at different rates and are active over different time intervals. We will refer to this finding later in the paper.

IV. Analysis Part II: Properties of the Sipscan and of the Botnet

A. Reverse IP Sequential order

A first manual observation of the sipscan destination addresses revealed that the bots were coordinated, presumably by a botmaster, to choose targets in a pre-defined sequence while scanning the entire IPv4 address space. Such coordination has not yet been documented in botnet-related research literature (see Section II). Even more interesting, the target IP addresses incremented in reverse-byte order – likely to make the scan covert. Reverse-byte order scanning was considered in the context of supporting network-friendly Internet-wide service discovery [40], but was discarded for being difficult to extrapolate metrics from partial scans. A pseudo-random approach in selecting target addresses was also used as a technique for non-aggressive Internet-wide measurement surveys [33]. But to the best of our knowledge, this reverse-byte order scanning has been neither empirically observed in malicious scans nor discovered in botnet source code.

Manual examination of a sequence of 20 million addresses is practically infeasible; even its visual representation is a challenge. We used a visual map based on the space-filling Hilbert Curve [47], [54] to verify that the target IP addresses incremented in reverse-byte order for the three bytes that we could observe (the most significant byte is fixed in our data to the /8 of the darknet observation point).

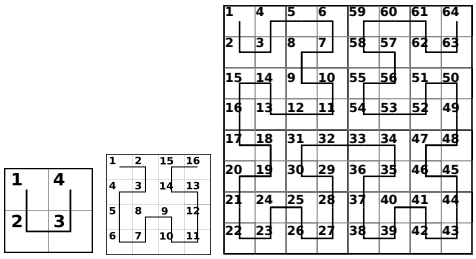


Fig. 9: Examples of Hilbert's space-filling curves: orders 1, 2 and 3.

The Hilbert curve is a continuous fractal curve that can be used to map one-dimensional data into two dimensions filling a square, such as shown in Figure 9. Other researchers have effectively used the Hilbert space layout to visualize results of Internet-wide scanning or other Internet-wide data [22], [33]. The original order of the data is preserved along the Hilbert curve in two dimensions, and conveniently displays data that is structured in powers of two. Hilbert curves of order 4, 8, and 12 have 2^8, 2^{16}, 2^{24} points, respectively, which in turn correspond to the masks for Class C (/24), Class B (/16), and Class A (/8) address blocks in the IPv4 numbering space. When mapping IP addresses to these two-dimensional Hilbert curves, adjacent address blocks appear as adjacent squares, even CIDR blocks (in between Class A, B, and C block sizes) are always represented as squares or rectangles.

We visualized the progression of the IP addresses targeted by the sipscan through an animation. Each frame represents the IPv4 address space of our darknet using a Hilbert curve of order 12, in which each cell corresponds to one IP address of the darknet, thus varying the 3 least significant bytes through all the possible combinations. The curve is displayed as a bitmap of size 4096x4096, with each pixel being assigned an IP address. For each frame, the pixels corresponding to the IP addresses that have been probed prior to that point in time are highlighted. We also added a brightness decay effect to better highlight the addresses probed in the last few frames while displaying the animation.

Drawing the Hilbert curve using IP addresses sequenced in their natural byte order does not reveal a particular pattern in the target progression, showing the square uniformly filling across the 12 days of the scan. This animation of target progression is available at [13]. In contrast, reversing the order of the three varying (i.e., least significant) bytes yields a representation that clearly illustrates the reverse sequential IP order rigorously followed by the sipscan: throughout the 12 days all the bots "march" together toward filling the entire address space. Figure 10 shows the frame for 5 February 2011 11:47 GMT from the full reverse-byte order animation available at [13]. This animation proves the strong coordination of bot activity: the progression is strictly observed by all the bots for the entire execution of the scan, independent of (i) variations in global scanning speed, (ii) the rates at which different bots proceed (see Section III-D), (iii) the large number of hosts involved at the same time and thus the possible distributed architecture of the botnet (e.g., multiple C&C channels).

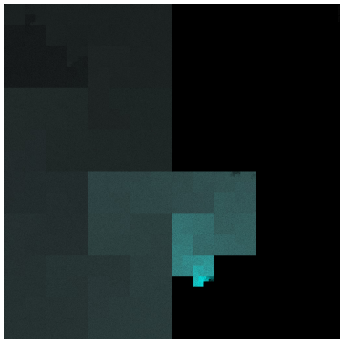


Fig. 10: Snapshot of our animation representing the progression over time of the IP addresses targeted by the sipscan [13]. The darknet address space is represented as a Hilbert curve of order 12 in which the order of the three least significant bytes of each address is reversed before mapping it into the curve. Highlighted pixels correspond to IP addresses that have been probed up to that time (5 Feb 2011 11:47 GMT, in this frame). The animation proves the reverse-byte order progression is rigorously followed by the bots during the entire 12 days, independent of the varying rate of the sipscan. [The above snapshot is a modified version of the original frame from the reverse byte order animation at [13]; we over-emphasized the fading effect to better illustrate, in a single picture, the path the scan took through the address space.]

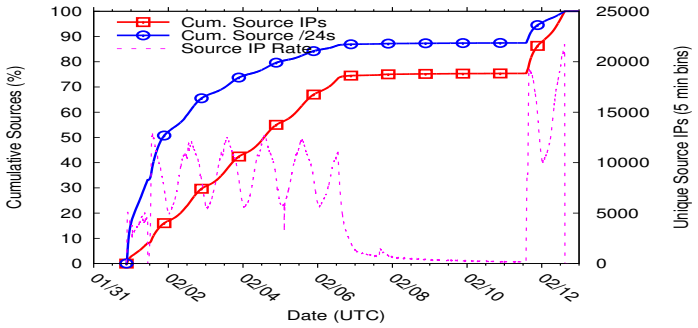


Fig. 11: Arrival of new bots. The continuous line with squares shows the cumulative percentage of bots that probed our darknet throughout the 12-day scan. The continuous line with circles is the cumulative percentage of source /24 networks. The slope of these curves indicates a constant arrival (during the botnet's active periods) of new bots participating in the scan. The dashed line represents the number of unique source IPs scanning per 5-minute interval, representing the evolution of the scan over time.

We also created a composite animation which combines both the natural and reverse byte order heatmaps with the world map animation into a single synchronized view of both the sources and the targets of the sipscan. This composite animation is available at [13].

The reverse IP sequential order used in this scan has significant implications. Observing this scan from a generic /24 network, would result in a very low number of packets per day: the average speed, during the largest phase of the scan – from the 2nd to the 6th of February – increments the least significant byte 34 times per day, unlikely to be detected by many automated systems [41]. This stealth technique is even more effective when combined with the constant turnover of bots that we illustrate in the next section.

B. Bot Turnover

The scanning statistics in Table I, in particular the number of unique source IPs (about 3 million), total number of probes (about 20 million), and the average number of destinations a source targets (6.85), suggest that there is a large turnover in the use of the bots. Figure 11 shows the constant use of new bots throughout the entire scan, except for the interval from approximately 7 February 00:00 GMT to 11 February 12:00 GMT, which exhibits significantly reduced botnet activity. The continuous line with square symbols shows the cumulative percentage of bots that probed our darknet over the 12-day scan. Its linear slope indicates a constant arrival of new bots participating in the scan. To partially take into account the effect of dynamic IP address assignment, we also plot the cumulative sum of unique /24 networks containing the source IP addresses (continuous line with circles). The slope of this curve proves that new bots take part in the scan for its entire duration.

Figure 12 shows the distribution of the number of packets sent by each bot. The diagram on the left uses a log-log scale to show all the data, whereas the diagram on the right uses a linear scale to zoom in to the left side of the distribution up to 10 packets. More than 1 million bots (more than 1/3 of the total) sent a single probe and never participated further in the scan. The number of bots that sent more than 100 packets during the scan is two orders of magnitude smaller. This difference suggests rapid turnover of bots during the scan. We hypothesize that this behavior is related to how the C&C channels managed and assigned tasks to bots. For example, a C&C channel may assign a list of target IP ranges to a queue of bots, in which case it is unlikely that a single bot could reach the head of the queue twice. In such a situation, bots that reappear in the scan would have likely been assigned to a C&C channel with a smaller pool of bots.

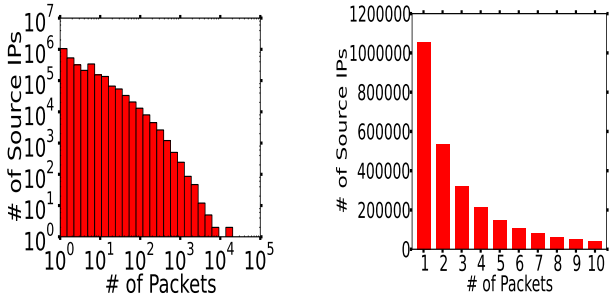


Fig. 12: (left) Full histogram of packets sent per bot (log-log scale); (right) zoomed histogram of packets sent per bot for bots that sent up to 10 packets (linear axis) Most bots sent few packets, e.g., over a third of the bots sent a single packet during the entire scan.

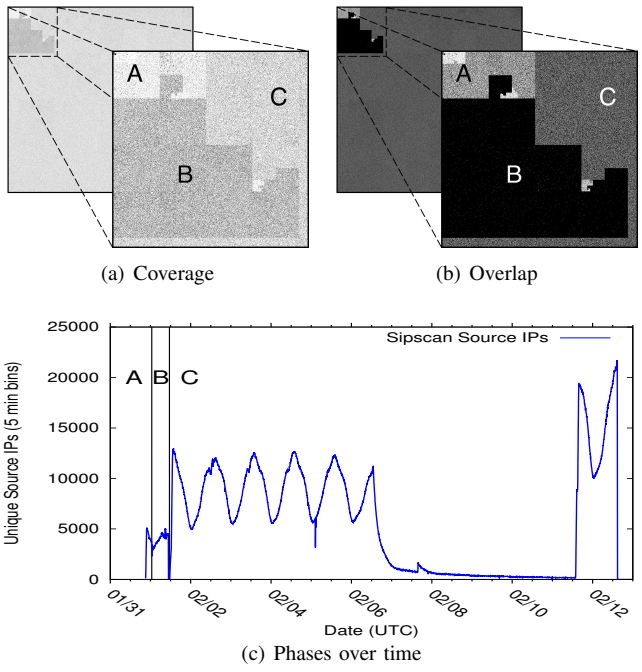
In combination with the reverse-byte order property of the scan, the high bot turnover rate makes the scan impressively covert. Not only would an automated intrusion detection system on a /24 network see only 34 packets to the same port in a single day, but they would most likely arrive from

34 distinct IP addresses, making detection highly unlikely (see Section VI.

C. Coordination and Adaptation

1) Coverage and Overlap

The scan fails to cover the entire darknet's /8 address space, probing only 86.6% of it (Table I). On the other hand, there is a non-negligible overlap in terms of bots hitting the same target: about 5.7 million IP addresses were probed by more than one bot, and on average a targeted IP is probed by 1.39 distinct bots. Whether probed zero, one, or multiple times, the probed IP addresses are scattered all over the address space without clusters or holes, in both the standard and reverse representation of the address bytes. These properties – coverage and overlap of target addresses – are independent of the number of bots active at any given time, the overall rate of the scan, or specific subnets being scanned. But we did discover a correlation between coverage and overlap in targets, which we believe is likely a function of a parameter of the scan configured by the botmaster to support trading off completeness and redundancy of scanning.



(a) Coverage (b) Overlap

(c) Phases over time

Fig. 13: Different phases (A, B, C) of the scan characterized by different but correlated rates of coverage and overlap of the target IP space, (a) Slice of the Hilbert-curve map (with reversed-byte order IP addresses) highlighting areas of different density indicating different coverage of the target space. (b) shows the same phenomenon in terms of overlap: the lit pixels in the map represent target addresses probed by more than one bot. The three regions perfectly match between the two maps. (c) Scanning source IPs throughout scan, showing the transitions from Phase A to B and from Phase B to C.

The representation with the Hilbert curve of the probed IP addresses in reverse byte order reveals three regions with different densities. These regions are labeled A, B, C, in a detail of the Hilbert-curve map in Figure 13(a) and correspond to three different phases of the scan as indicated in Figure 13(c). Brighter areas indicate a greater coverage of the corresponding address space: the scan starts with a very high percentage of targets probed ("A"), after few hours a parameter is changed and the coverage significantly drops ("B"), finally the parameter is adjusted again and an intermediate level of target coverage remains for the rest of the scan ("C"). The same regions are visible in Figure 13(b), where we use the Hilbert-curve map to highlight the overlap in targets: IP addresses (in reverse-byte order) that were probed more than once are depicted in white.

Start time	Jan 31 21:00	Feb 1 00:45	Feb 1 11:20
# of probes	179,143	486,394	19,590,184
% of IP space covered	93.81%	76.27%	86.98%
Average bots per target	1.66	1.01	1.40

TABLE II: Characteristics of the three phases of the scan, with different coverage and overlap of the target address space, show a trade-off between the two properties.

Table II shows statistics calculated separately for the three phases of the scan. The correlation between coverage and overlap of the scan is evident, and is consistent with a probabilistic mechanism in the choice of the targets that can be configured by the botmaster to trade off completeness and redundancy of scanning. The finding illustrated in Figure 14 further substantiates the hypothesis that the three phases correspond to different configurations of the scan. The figure shows, for each phase, the distribution of the number of packets sent in each "reverse /16 subnet" (we define a reverse /16 subnet as the set of all possible IP addresses obtainable when the least two significant bytes are fixed). The three curves refer to populations of different size, which explains the different smoothness of their shapes (e.g., phase C is considerably longer thus covering a larger number of reverse /16 subnets). However, all of them are highly centered around a different value (average values are 395.6 (A), 196.3 (B), 312.6 (C)) and mostly non-overlapping, reflecting a consistent and distinctive behavior in each phase.

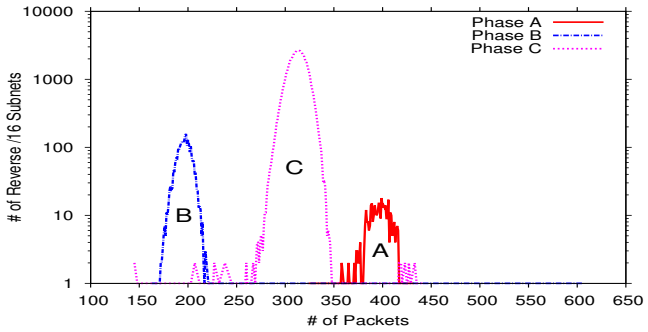


Fig. 14: Consistent and distinctive behavior of the different phases of the scan. The curves represent, for each phase A, B, C, the distributions of packets observed at the UCSD Network Telescope in each "reverse /16 subnet". The distributions are all centered around different values and mostly non overlapping.

Finally, in both Figures 13(a) and 13(b), we also observe better coverage and larger overlap in the transition from one region to the other, suggesting that the botmaster re-issued a command to scan those IP ranges to the bots after changing

the configuration parameter (possibly because the scan was stopped without collecting the results of the previous command). The higher coverage in these transition areas provides further evidence of a probabilistic approach in the choice of the target IPs (probably happening at the level of the single bots): re-issuing the commands for that range of target IPs results in a partially different set of probed targets.

Even given non-negligible redundancy, an average of 1.39 bots hitting the same target is small compared to the large number of bots involved. Such low redundancy is novel, or at least undocumented in the literature, which has mostly reported on bots that independently scan the same address range in a random uniform fashion [12], [43]. The small overlap and thus high efficiency in terms of completeness vs. redundancy achieved by this botnet is an impressive consequence of strongly orchestrated behavior.

2) Adaptivity

The strong coordination of bot activity is also visible in terms of adaptation capabilities. Starting around 7 February 00:00 GMT through around 11 February 12:00 GMT, the scan proceeds very slowly, with only a few active bots (Figure 2). A possible hypothesis is that most of the C&C channels are down during this period. However, we observe that the target IP ranges that would have normally been assigned to these C&C channels were automatically redirected to those channels that were still up.

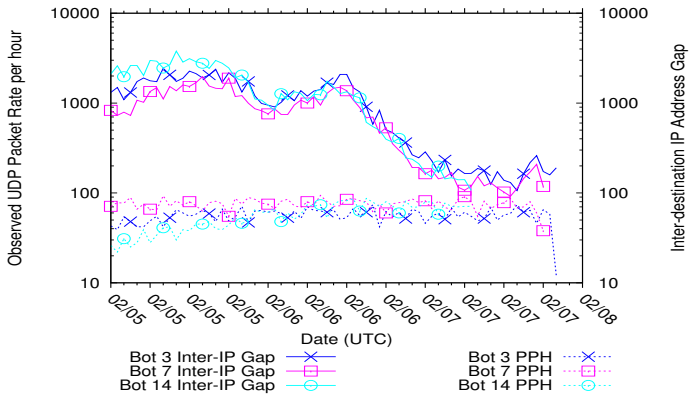


Fig. 15: Adaptive assignment of target IP ranges to different C&C channels. Dashed lines represent probes per hour (PPH) carried out by 3 different bots. Their speed did not change significantly on 7 February but the global speed of the scan decreased considerably, probably because some C&C channels went off-line. However, the target IP ranges assigned to these bots became denser during this period, to compensate for the absence of other C&C channels: continuous lines represent the distance between subsequent target IPs of each respective bot, showing an order of magnitude decrease in that time interval.

Figure 15 illustrates this behavior. Dashed lines in the graph represent the probing rate per hour of the three bots discussed in Section III-D. During this period the bots do not change their speed, suggesting that the C&C channel they refer to has not changed its characteristics in terms of numbers of bots managed, etc. (i.e., the number of bots competing for a certain C&C channel does not change, therefore the rate at which each bot gets assigned a new "reverse" /24 stays the same). However, over this same time interval we observe a significant change in the sequences of IP ranges assigned to these bots. The continuous lines in Figure 15 show, for each of the three bots, the distance between subsequent target IPs, calculated by subtracting the target IPs after reversing their byte order and converting them into 32-bit numbers. The graph shows a drop of about one order of magnitude in the distance, meaning that the corresponding C&C channel(s) receive a "denser" list of targets to compensate for the disappearance of the other C&C channels.

D. Botnet characteristics

Observing a horizontal scan of this magnitude from such a large darknet allows unique insight into the characteristics of the botnet that performed it. The size of the darknet, combined with the reverse IP sequential ordering of the targets, allowed the telescope to capture probes across the entire life of the scan, providing an unprecedented view of the population of the Sality botnet.

A white paper from Symantec [24] estimated the size of the Sality botnet at approximately a million bots, by measuring the number of hosts that a 'rogue' server under their control communicated with. We identified a total of 2,954,108 unique source IPs for bots that participated in the sipscan. As the authors of [63] demonstrate, it is difficult to accurately determine the size of the botnet population when using source IP addresses collected from traffic sent by infected hosts. This difficulty arises due to the effects of dynamic IP address assignment (DHCP), which can result in several IP addresses being used by a single bot (especially over a 12-day interval), and NAT, which can cause multiple bots to appear as a single IP. However, Figure 11 shows continuous growth in the number of unique /24 networks hosting bots over the entire duration of the scan. This diversity of /24 networks can be used as an approximation for the number of new bots that arrive over the course of the scan.

We leverage the large population of source addresses observed to further understand how hosts compromised by botnets are distributed globally. To this end, we determine the Autonomous System Number (ASN) for each bot using a Routeviews BGP routing snapshot [64] taken on Monday 14 February 2011 at 12:00 UTC, proximate to the scanning episode. Using this table, we perform longest-prefix matching to resolve each source IP to its origin AS.

The ASes enumerated in Table III are the 10 most common across the bots used by the sipscan botmaster. We also list the AS name and home country extracted from whois data. Similar to the Conficker [59] and Mega-D [4] botnets, we see a dominant AS at the top of the list (TTNet), which alone accounts for over 10% all participating bots, followed by a long tail of small ASes. However, although the scale of the leading ASes may resemble other botnets, the networks featured in the top 10 are quite different (Table IV). Only four of the ASes in the top 10 of the sipscan appear in the top 10 of either Conficker [59] or Mega-D [59]. Notably, TTNet in Turkey, which [59] lists in 10th place, represents the largest AS by more than a factor of two in the sipscan botnet.

Rank	%	ASN	AS Name	Country
1	10.81	9121	TTNet	Turkey
2	4.57	8452	TE	Egypt
3	4.40	9829	BSNL-NIB	India
4	4.22	17974	TELKOMNET	Indonesia
5	4.20	45899	VNPT	Vietnam
6	3.01	7738	TELEMAR	Brazil
7	2.65	8708	RDSNET	Romania
8	2.51	24560	AIRTELBROADBAND	India
9	2.07	9050	RTD	Romania
10	1.94	9737	TOTNET	Thailand

TABLE III: Top 10 origin ASes of bots used in the sipscan. As noted in other work [59], we see a dominant AS at the top of the list (Turkey, with 10% of the overall bot population), followed by a long tail. The country and AS name data have been extracted from whois data for each AS.

	Conficker [59]		Mega-D [59]		Sipscan	
Rank	ASN	Country	ASN	Country	ASN	Country
1	4134	China	3352	Spain	9121	Turkey
2	4837	China	3269	Italy	8452	Egypt
3	7738	Brazil	6739	Spain	9829	India
4	3462	Taiwan	9121	Turkey	17974	Indonesia
5	45899	Vietnam	6147	Peru	45899	Vietnam
6	27699	Brazil	19262	USA	7738	Brazil
7	9829	India	4134	China	8708	Romania
8	8167	Brazil	7738	Brazil	24560	India
9	3269	Italy	7418	Chile	9050	Romania
10	9121	Turkey	22927	Argentina	9737	Thailand

TABLE IV: Comparison of the top 10 ASes observed in three different botnets: the Conficker botnet as surveyed by [59], the Mega-D botnet as reported by [4], [59], and the Sality (sipscan) botnet. We observe a trend toward Eastern European countries which have not featured as prominently in previous botnets.

Both the Conficker and Mega-D AS distributions indicate a move toward larger representation of bots in Asian and South American countries, corroborating the results of [59]. However, we see a considerable rise in bots in Eastern European countries, which becomes even more apparent on a per-country level (Table V).

Simply aggregating bots by their ASN can be misleading because many large organizations/providers have multiple ASNs. To complement our AS findings, we geolocate the bot's IP address using a MaxMind GeoLite database [45] snapshot from March 1 2011 (again, proximate to the scan episode). Table V presents the top 10 countries for bots in both the sipscan and the Mega-D botnets [4]. Once we aggregate bots to a country granularity, the distribution of locations changes appreciably, with the Russian Federation making an appearance in the top 10 lists of both Mega-D and sipscan[2].

	Mega-D [4]		Sipscan	
Rank	%	Country	%	Country
1	14.82	USA	12.55	Turkey
2	11.74	Russian Federation	12.54	India
3	6.33	Turkey	8.64	Brazil
4	6.32	Poland	7.23	Egypt
5	5.32	Thailand	5.77	Indonesia
6	4.11	Brazil	5.59	Romania
7	3.89	Germany	5.58	Russian Federation
8	3.23	United Kingdom	5.36	Vietnam
9	2.53	India	5.10	Thailand
10	2.25	Spain	3.01	Ukraine

TABLE V: Top 10 Countries of bots used in the sipscan compared to the Mega-D botnet. Geolocation data for sipscan sources was obtained using the MaxMind GeoLite database [45]. Aggregating bots by country rather than AS helps identify regions that are heavily compromised by bots but have many small ASes, such as the Russian Federation, which is not in the list of top 10 ASes.

Contrary to similarly large botnets [4], [52], [59], [65], the sipscan bots do not have a dominant presence in China. China has been recorded in the top ten lists of these other botnets, but in the sipscan, China is in 27th place (0.57%) - close to U.S.'s 29th place position (0.44%). Heatmaps of overall Sality bot locations [24] also indicate a corresponding lack of Sality bot presence in China. We believe this under-representation of China, when compared to previous botnets, may be considered a limitation of the Sality botnet rather than a specific design choice by the botmaster. Although the data presented in [24] is largely in aggregated graphical form, it does appear to corroborate our findings in terms of geographical distribution. As noted earlier however, we are able to identify a much larger bot population.

In addition to analyzing the networks that host the bots, we also investigated the bots themselves. Output of the p0f passive OS fingerprinting tool [67] reported that more than 97% of bots were running operating systems of the Microsoft Windows family. The distribution of UDP source port values shown in Figure 7 also shows that the majority of packets fall into the 1025-5000 range of ports, which was used by Microsoft Windows until Vista and Server 2003. There are, however, a non-negligible number of bots that p0f identified as running the Linux operating system. We believe these machines are likely not bots but rather NAT gateways proxying packets from infected hosts.

V. Binary Analysis

We had the opportunity to analyze the binary code responsible for this scanning. The binary is a separate executable that Sality-infected computers download via a URL as directed by the peer-to-peer botnet infrastructure [23], [24]. Although our work focuses on the Internet measurement aspect of the event, we partially reverse engineered this code to validate some of our inferences. Here we summarize the most relevant findings.

We found that each bot contacts a hard-coded IP address (the C&C channel) in order to receive a probing command from the botmaster. The command followed by the bots we observed is one of three different command types that the binary supports. Through this command, the botmaster sends the target IP to the bot in the form of an ASCII string (dotted quad decimal format). By analyzing the code, we verified that this address is the actual address probed by the bot. In particular, the bot properly manages the endianness of the target IP addresses, e.g., when converting the ASCII string into binary and then when contacting the target.

Each bot reports through the C&C channel the results of a probe immediately after receiving a response from the victim. It then selects and probes a new target by incrementing the most significant byte of the target address received by the

[2] [59] only provides Conficker results at an AS level.

botmaster. This increment is repeated 15 times, for a total of 16 targets probed, each one from a different /8 network. The bot then sleeps for a fixed amount of time before contacting the botmaster again to receive a new target IP.

These findings, along with the progression of the target IP addresses observed through the UCSD Network Telescope, indicate that both the botmaster and each bot incremented the target IPs in reverse-byte order, and that the sequence followed by the scan reflected the original orders of the botmaster (who was sending addresses as quad decimal dot-separated ASCII strings). In other words, the reverse byte order probing was most likely not due to a bug or error in managing the endianness of the target IP addresses.

Inspecting the binary also revealed that several interesting properties of the scan would have not been visible by relying solely on the reverse-engineering the bot binary. For example, the code running on a single bot shows only the selection of 16 target addresses (whose increments to the most significant byte could have been attributed to a coding mistake, without the knowledge of the overall pattern). But analysis of traffic from the UCSD Network Telescope revealed a heavily coordinated behavior of many bots around the world, allowing inference of the mechanisms adopted by the botmaster in orchestrating the scan.

Finally, our analysis of the sipscan code binary confirmed the ability of the bots to perform break-in attempts – trying to register users with the SIP server – based either on brute-forcing or using specific lists of user/password pairs communicated to the bot by the botmaster. The software included code to try selected lists of common user/password pairs in case a web administrative panel was found active on the SIP server, trying to gain administrative rights. Symantec also reported the presence of both functionalities in the binary module [23]. It is credible that the purpose of registering users or gathering full control of the SIP server was to perform fraudolent VoIP activities [25].

VI. Discussion

Botnets commonly scan large segments of the Internet's address space for various purposes, such as infecting or compromising hosts, recruiting hosts into a botnet, or collating a list of future targets. Awareness of evolving botnet characteristics and spreading techniques can improve our ability to navigate and mitigate their impacts. As mentioned in Section II, although many aspects of botnet behavior have been documented, we are not aware of any published investigation of a million node botnet covertly scanning the entire IPv4 space. Most of the available literature are studies of older generation (pre-2007) botnets that are substantially smaller in size, scope, and capability than newer-generation botnets. Studies of newer generation bots have focused on aspects other than the scanning behavior, such as the command and control, peer-to-peer infrastructure, or the domain of abuse, e.g., spam campaigns inflicted by the botnet. We presented a new angle on the study of new-generation botnets: their scanning activity as observable in large darknets, most aspects of which cannot be inferred by reverse-engineering the bot malware.

This work offers contributions in two areas: documenting and visualizing behavioral aspects of a current generation botnet, and thoroughly analyzing the multiple synergistic characteristics of its extraordinarily well-coordinated scanning. The scan that we analyzed in this study was new, or at least not previously documented, in four ways. It was sourced by a current-generation (2011), widely-deployed, peer-to-peer botnet (Sality [24]). Although earlier-generation versions of Sality were first reported in June 2003, it was not until February 2011 that Sality operators deployed a new module designed to locate and compromise SIP servers in a distributed, heavily coordinated manner. The population of bots participating in the scan was several orders of magnitude larger than any previously documented botmaster-orchestrated scanning. Previous Internet-wide scanning behavior perpetrated by botnets was due to worm-spreading modules inside the bot, e.g, in Conficker, rather than botmaster-coordinated scanning. Not only was this sipscan coordinated, but it was impressively well-engineered to maximize coverage, minimize redundancy and overlap among target IP addresses by scanning bots, and evade detection by even state-of-the-art intrusion detection capabilities.

We used the detailed packet traces captured by the darknet to richly analyze many properties of the botnet, including several interacting properties of the botnet's heavily coordinated scanning. The size of the botnet, the fact that it was a /0 scan, i.e., of the entire IPv4 address space, and the reverse-byte ordering sequence of IP addresses targets were unprecedented and impressive enough characteristics. Time-series analysis of the active IP addresses operating as bots revealed an unusually rapid turnover rate and associated low re-use rate of the bot population, all tightly coordinated by the botmaster to scan in a extremely regular, stealth pattern.

In a recent work [41], Leonard et al. performed a stochastic analysis of horizontal IP scans and of detection techniques implemented in modern intrusion detection systems (IDS). The authors formalized under which conditions current IDS implementations would be able to detect a horizontal scan (based on pattern, number of source IP addresses used for probing, etc.) when monitoring a network of a given size. The numbers we found in the case of the sipscan, show that detecting a scan with similar characteristics would be impossible for state-of-the-art IDS implementations. A typical IDS raises an alert when observing a number of probes, from a single IP address, greater than a threshold a_s within a predefined time window Δ_s. The longest time window and the minimum threshold values (i.e., highest detection sensitivity) in default settings of current IDSes are respectively 3600s and 5 probes (these values try to optimize the false positives/false negatives trade off, as well as limit memory consumption) [41]. A scan from approximately 3 million distinct source IP addresses over a duration of 12 days, load-balancing its packets across the scanned space, would avoid detection even by an IDS monitoring an entire /8 network (with $\Delta_s = 3600$ and $a_s = 5$).

We developed animation and visualization techniques to facilitate our exploration of the sipscan. The Hilbert-curve map visualization clearly revealed the strictly ordered reverse-byte

incrementing behavior of the progression of the entire scan; without this visualization technique it is not clear that we would have verified this sequence (for all the three observable changing bytes). Animations over time [13] exposed the three phases of the scanning, and juxtaposing the Hilbert maps with a geographic map of bot activity as well as a traffic time-series allowed us to simultaneously visualize multiple dimensions of the scanning behavior. We expect this technique will be useful for analysis of other large-scale Internet probing behavior [38].

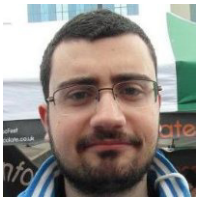
Analysis of this scan provided an eye-opening if ominous indicator of the more sophisticated capabilities of modern malware to surreptitiously survey and exploit critical infrastructure vulnerabilities on a planetary scale. Our darknet packet capture allowed us to precisely analyze this botnet's comprehensive and covert scanning behavior, and in the process we developed generalizable methods to correlate, visualize, and extrapolate botnet behavior across the global Internet. Finally, another contribution of this work is the dataset available at [13], with detailed information (e.g., timestamp and source IP geolocation) for each sipscan UDP probing packet we captured.

This work leaves open the question of how to automatically detect such macroscopic events. In [19] we suggested that the time series of distinct source IP addresses per destination port is a better indicator than packet rate, but we also reported that commonly scanned ports (such as TCP 80 and 445) receive so much probing traffic that it would be difficult to spot large-scale coordinated scans with traditional change-point detection approaches. Novel methodologies should instead correlate distinctive features (such as common source IP addresses, automatically extracted common substrings in payload, etc.) from traffic captured simultaneously on different large darknets and live networks. To support our own and other efforts to automate the analysis of darknet as well as two-way traffic, we release as open source the Corsaro [37] software suite we developed to perform high-speed analysis of packet trace data, which we used to analyze the sipscan. We are currently using Corsaro to extract in quasi realtime a large number ($>$ 1M) of time series from traffic and experiment with visualization techniques for the identification of specific scanning patterns.

Alberto Dainotti is a research scientist at the Cooperative Association for Internet Data Analysis (CAIDA), SDSC, UC San Diego, USA. In 2008 he received his Ph.D. in Computer Engineering and Systems at the Department of Computer Engineering and Systems of University of Napoli "Federico II", Italy. His main research interests are in the field of Internet measurement and network security, with a focus on the analysis of large-scale Internet events. In 2012 he was awarded the IRTF Applied Networking Research Prize.

Alistair King is a research programmer at the Cooperative Association for Internet Data Analysis (CAIDA), SDSC, UC San Diego. He received a Masters Degree in Science in 2010 from The University of Waikato, New Zealand. His current interests are centered around software and infrastructure development for efficient, realtime analysis of large-scale Internet measurement datasets.

KC Claffy is director of the Cooperative Association for Internet Data Analysis (CAIDA), which she founded at the UC San Diego Supercomputer Center in 1996. CAIDA provides Internet measurement tools, data, analyses and research to promote a robust, scalable global Internet infrastructure. As a research scientist at SDSC and Adjunct Professor of Computer Science & Engineering at UCSD, her research interests include Internet (workload, performance, topology, routing, and economics) data collection, analysis, visualization, and enabling others to make use of CAIDA data and results. She has been at SDSC since 1991 and holds a Ph.D. in Computer Science from UC San Diego.



Ferdinando Papale is currently a second year student in the Computer Science and Engineering MSc program at the Technical University of Denmark (DTU). In 2011 he received his Bachelor's Degree in Computer Engineering from the University of Napoli "Federico II", Italy.

Antonio Pescapé is an Assistant Professor at the Electrical Engineering and Information Technology Department at the University of Napoli Federico II (Italy) and Honorary Visiting Senior Research Fellow at the School of Computing, Informatics and Media of the University of Bradford (UK). He received the M.S. Laurea Degree in Computer Engineering and the Ph.D. in Computer Engineering and Systems, both at University of Napoli Federico II. His research interests are in the networking field with focus on Internet Monitoring, Measurements and Management and Network Security. He is a Senior Member of the IEEE.

REFERENCES

[1] AfriNIC: The Registry of Internet Number Resources for Africa. http://www.afrinic.net.

[2] Cuttlefish. http://www.caida.org/tools/visualization/cuttlefish/.

[3] RIPE NCC: Routing Information Service (RIS). http://www.ripe.net/data-tools/stats/ris/routing-information-service.

[4] Secureworks. ozdok/mega-d trojan analysis. http://www.secureworks.com/research/threats/ozdok/.

[5] tcpdump. http://www.tcpdump.org.

[6] The asterisk-users mailing-list archives. http://lists.digium.com/pipermail/asterisk-users/2010-November/thread.html, November 2010.

[7] UCSD Network Telescope, 2010. http://www.caida.org/data/passive/network_telescope.xml.

[8] The voipsec mailing-list archives. http://voipsa.org/pipermail/voipsec_voipsa.org/2010-November/thread.html, November 2010.

[9] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM.

[10] P. Bacher, T. Holz, M. Kotter, and G. Wicherski. Know your enemy: Tracking botnets. http://www.honeynet.org/papers/bots, 2008.

[11] P. Barford and V. Yegneswaran. An Inside Look at Botnets. In M. Christodorescu, S. Jha, D. Maughan, D. Song, and C. Wang, editors, *Malware Detection*, volume 27 of *Advanced in Information Security*. Springer, 2006.

[12] P. Barford and V. Yegneswaran. An Inside Look at Botnets. *Advances in Information Security, Malware Detection*, vol. 27, 2007, Springer.

[13] CAIDA. Supplemental data: Analysis of a "/0" Stealth Scan from a Botnet. http://www.caida.org/publications/papers/2012/analysis_slash_zero/supplemental/, 2012.

[14] C. Castelluccia, M. A. Kaafar, P. Manils, and D. Perito. Geolocalization of proxied services and its application to fast-flux hidden servers. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 184–189, New York, NY, USA, 2009. ACM.

[15] J. Cheng. Symantec: Flashback botnet could generate up to $10k per day in ad clicks. http://arstechnica.com/apple/2012/05/symantec-flashback-botnet-could-generate-up-to-10k-per-day-in-ad-clicks/, May 1 2012.

[16] C. Y. Cho, D. Babi ć, E. C. R. Shin, and D. Song. Inference and analysis of formal models of botnet command and control protocols. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 426–439, New York, NY, USA, 2010. ACM.

[17] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet*, SRUTI'05, pages 6–6, Berkeley, CA, USA, 2005. USENIX Association.

[18] D. Dagon, G. Gu, C. Lee, and W. Lee. A taxonomy of botnet structures. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*, pages 325 –339, dec. 2007.

[19] A. Dainotti, A. King, and K. Claffy. Analysis of internet-wide probing using darknets. In *Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security*, BADGERS '12, pages 13–14, New York, NY, USA, 2012. ACM.

[20] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé. Analysis of country-wide internet outages caused by censorship. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, pages 1–18, New York, NY, USA, 2011. ACM.

[21] J. Davis. Hackers take down the most wired country in europe. http://www.wired.com/politics/security/magazine/15-09/ff_estonia, July 1 2011.

[22] Duane Wessels. Mapping the IPv4 address space, 2009. http://maps.measurement-factory.com/.

[23] N. Falliere. A distributed cracker for voip. http://www.symantec.com/connect/blogs/distributed-cracker-voip, February 15 2011.

[24] N. Falliere. Sality: Story of a peer-to-peer viral network. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/sality_peer_to_peer_viral_network.pdf, July 2011.

[25] S. Gauci. 11 million Euro loss in VoIP fraud.. and my VoIP logs, December 2010. http://blog.sipvicious.org/2010/12/11-million-euro-loss-in-voip-fraud-and.html.

[26] S. Gauci. Distributed sip scanning during halloween weekend. http://blog.sipvicious.org/2010/11/distributed-sip-scanning-during.html, Nov 4 2010.

[27] S. Gauci. Sipvicious. tools for auditing sip based voip systems. http://code.google.com/p/sipvicious/, Apr 2012.

[28] C. W. Group. Conficker working group lessons learned. http://www.confickerworkinggroup.org/wiki/uploads/Conficker_Working_Group_Lessons_Learned_17_June_2010_final.pdf, June 2010.

[29] M. W. group. Guidelines for protecting user privacy in wide traffic traces. http://mawi.wide.ad.jp/mawi/guideline.txt, Oct 1999.

[30] M. W. group. Mawi working group traffic archive. http://mawi.wide.ad.jp, Apr 2012.

[31] M. Gruber, F. Fankhauser, S. Taber, C. Schanes, and T. Grechenig. Security status of voip based on the observation of real-world attacks on a honeynet. In *Privacy, security, risk and trust (passat), IEEE third international conference on social computing (socialcom)*, pages 1041 –1047, oct. 2011.

[32] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)*, February 2008.

[33] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister. Census and survey of the visible internet. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 169–182, New York, NY, USA, 2008. ACM.

[34] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, pages 9:1–9:9, Berkeley, CA, USA, 2008.

[35] S. Institute. Dshield.org: Distributed intrusion detection system. http://www.dshield.org, Apr 2012.

[36] C. Kanich, N. Weavery, D. McCoy, T. Halvorson, C. Kreibichy, K. Levchenko, V. Paxson, G. M. Voelker, and S. Savage. Show me the money: characterizing spam-advertised revenue. In *Proceedings of the 20th USENIX conference on Security*, SEC'11, pages 15–15, Berkeley, CA, USA, 2011. USENIX Association.

[37] A. King and A. Dainotti. Corsaro. www.caida.org/tools/measurement/corsaro/, 2012.

[38] A. King, A. Dainotti, B. Huffaker, and k. claffy. A Coordinated View of the Temporal Evolution of Large-scale Internet Events. *Computing*, Jan 2013.

[39] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamcraft: an inside look at spam campaign orchestration. In *Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*, LEET'09, pages 4–4, Berkeley, CA, USA, 2009. USENIX Association.

[40] D. Leonard and D. Loguinov. Demystifying service discovery: implementing an internet-wide scanner. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 109–122, New York, NY, USA, 2010. ACM.

[41] D. Leonard, Z. Yao, X. Wang, and D. Loguinov. Stochastic analysis of horizontal ip scanning. In *INFOCOM, 2012 Proceedings IEEE*, pages 2077–2085, 2012.

[42] Z. Li, A. Goyal, and Y. Chen. Honeynet-based botnet scan traffic analysis. In W. Lee, C. Wang, and D. Dagon, editors, *Botnet Detection*, volume 36 of *Advances in Information Security*, pages 25–44. Springer, 2008.

[43] Z. Li, A. Goyal, Y. Chen, and V. Paxson. Towards situational awareness of large-scale botnet probing events. *Information Forensics and Security, IEEE Transactions on*, 6(1):175 –188, march 2011.

[44] W. Lu, M. Tavallaee, and A. A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. ASIACCS '09, pages 1–10, New York, NY, USA, 2009. ACM.

[45] MaxMind. MaxMind GeoLite Country: Open Source IP Address to Country Database. http://www.maxmind.com/app/geolitecountry.

[46] C. Mullaney. Android.bmaster: A million-dollar mobile botnet. http://www.symantec.com/connect/blogs/androidbmaster-million-dollar-mobile-botnet, February 9 2012.

[47] R. Munroe. xkcd: Map of the Internet. http://xkcd.com/195/, 2006.

[48] M. D. Network. bind function. http://msdn.microsoft.com/en-us/library/ms737550%28VS.85%29.aspx, 2012.

[49] A. Pathak, F. Qian, Y. C. Hu, Z. M. Mao, and S. Ranjan. Botnet spam campaigns can be long lasting: evidence, implications, and analysis. SIGMETRICS '09, pages 13–24, New York, NY, USA, 2009. ACM.

[50] P. Porras, H. Saidi, and V. Yegneswaran. Conficker. Technical report, SRI International, Mar 2009.

[51] A. H. Project. Sip brute force attack originating from amazon ec2 hosts. http://honeynet.org.au/?q=sunday_scanner, October 25 2010.

[52] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. SIGCOMM '06, pages 291–302, New York, NY, USA, 2006. ACM.
[53] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002.
[54] H. Sagan. *Space-filling curves.* Universitext. New York: Springer-Verlag. xv, 193 p. DM 54.00; öS 421.20; sFr. 54.00 , 1994.
[55] S. Sarat and A. Terzis. Measuring the storm worm network, October 2007.
[56] S. Sheldon. Sip brute force attack originating from amazon ec2 hosts. http://www.stuartsheldon.org/blog/2010/04/sip-brute-force-attack-originating-from-amazon-ec2-hosts/, April 11 2010.
[57] S. Sheldon. Sip brute force attacks escalate over halloween weekend. http://www.stuartsheldon.org/blog/2010/11/sip-brute-force-attacks-escalate-over-halloween-weekend/, Nov 1 2010.
[58] S. Shin and G. Gu. Conficker and beyond: a large-scale empirical study. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 151–160, New York, NY, USA, 2010. ACM.
[59] S. Shin, G. Gu, N. Reddy, and C. Lee. A large-scale empirical study of conficker. *Information Forensics and Security, IEEE Transactions on*, 7(2):676 –690, april 2012.
[60] S. Shin, Z. Xu, and G. Gu. EFFORT: Efficient and Effective Bot Malware Detection. In *Proceedings of the 31th Annual IEEE Conference on Computer Communications (INFOCOM'12) Mini-Conference*, March 2012.
[61] S. Staniford, V. Paxson, and N. Weaver. How to own the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167, Berkeley, CA, USA, 2002. USENIX Association.
[62] J. Stewart. Protocols and encryption of the storm botnet. http://www.blackhat.com/presentations/bh-usa-08/Stewart/BH_US_08_Stewart_Protocols_of_the_Storm.pdf., 2008.
[63] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. 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.
[64] University of Oregon. University of Oregon Route Views project. http://www.routeviews.org.
[65] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, pages 171–182, New York, NY, USA, 2008. ACM.
[66] V. Yegneswaran, P. Barford, and V. Paxson. Using honeynets for internet situational awareness. In *Fourth Workshop on Hot Topics in Networks (HotNets IV)*, 2005.
[67] M. Zalewski. p0f v3. http://lcamtuf.coredump.cx/p0f3/, 2012.
[68] L. Zeltser. Targeting VoIP: Increase in SIP Connections on UDP port 5060. http://isc.sans.edu/diary.html?storyid=9193, July 2010.
[69] C. C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and early warning for internet worms. In *Proceedings of the 10th ACM conference on Computer and communications security*, CCS '03, pages 190–199, New York, NY, USA, 2003. ACM.