Measuring the Effectiveness of Infrastructure-Level Detection of Large-Scale Botnets

Yuanyuan Zeng[†] Guanhua Yan[‡] Stephan Eidenbenz[‡] Kang G. Shin[†]

[†]Dept of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan 48109
{gracez, kgshin}@eecs.umich.edu

[‡]Information Sciences (CCS-3)
Los Alamos National Laboratory
Los Alamos, NM 87545
{ghyan, eidenben}@lanl.gov

***Abstract*—Botnets are one of the most serious security threats to the Internet and its end users. In recent years, utilizing P2P as a Command and Control (C&C) protocol has gained popularity due to its decentralized nature that can help hide the botmaster's identity. Most bot detection approaches targeting P2P botnets either rely on behavior monitoring or traffic flow and packet analysis, requiring fine-grained information collected locally. This requirement limits the scale of detection. In this paper, we consider detection of P2P botnets at a high-level—the infrastructure level—by exploiting their structural properties from a graph analysis perspective. Using three different P2P overlay structures, we measure the effectiveness of detecting each structure at various locations (the Autonomous System (AS), the Point of Presence (PoP), and the router rendezvous) in the Internet infrastructure.**

I. Introduction

A botnet consists of a group of coordinated bots that can mount attacks such as Distributed Denial of Service (DDoS), spamming, phishing and identity theft. Botnets are posing a serious security threat to the Internet users; they can bring down the entire system and disrupt Internet services. In a botnet, a Command and Control (C&C) channel, in which a botmaster disseminates commands to, and get response from bots, is a key element. Traditional botnets utilize the IRC or HTTP protocol to implement centralized C&C. Under this design, bots have to connect to central servers and even listen on certain channels to retrieve commands. Evidently, centralized C&C is vulnerable to a single-point-of-failure, meaning that, whenever the central servers are identified and removed, the entire botnet will be deactivated. To overcome this weakness, attackers have recently devised a decentralized C&C infrastructure exploiting the P2P protocol. A few noteworthy P2P botnets in recent years include Storm [1], Waledac [2] and Conficker [3]. Their P2P implementations are either based on an existing protocol (Storm utilized Kademila [4]) or completely customized.

The decentralized nature of P2P botnets inevitably challenges detection attempts. Approaches targeting centralized C&C structures [5], [6], [7], [8] become ineffective under the new structure in which a botmaster can join, issue commands and leave at any time at any place. Generic detection approaches [9], [10] relying on behavior monitoring and traffic correlation analysis are mostly applicable at a small scale such as in edge networks and do not scale well because they require analyzing vast amounts of fine-grained information. In addition, if there is only a small number of bots in an edge network, detection based on bots' coordination may fail due to the limited number of instances in view. Given the fact that current botnets' sizes are in the order of hundreds of thousands, an effective and efficient large-scale detection needs to function at a high level without requiring fine-grained information that can only be obtained locally. As a P2P botnet has a structured overlay and connectivity patterns different from other applications from a graph analysis perspective, naturally, we consider detection at the Internet infrastructure level by assessing the impact imposed by a P2P botnet at various network components and measuring the effectiveness of detection at such places.

In this paper, we evaluate different network components' capabilities of detecting P2P botnets at the infrastructure level. We construct three types of P2P-botnet overlays, map them to the corresponding AS (Autonomous System)-level underlays by inferring each overlay connection's AS-path, and accordingly determine the PoP (Point of Presence) path and geographical router rendezvous each connection goes through. We then take a close look at each individual AS, PoP and router rendezvous based on graph analysis. In particular, we calculate a few P2P traffic classification metrics to see whether the portion of botnet connections observed by a single network component can be identified as P2P traffic. We would like to answer the following three questions through our analysis: (1) Which network component is the best monitoring point for detection? (2) Which P2P overlay structure can help hide the botnet traffic well? (3) What are the limitations of detection at the infrastructure level? Our main contribution lies in the thorough analysis of detection potential at the three infrastructure-level network components for three different P2P overlay topologies. To the best of our knowledge, there have been only a very few previous approaches to the detection of botnets at the

infrastructure level. In [11], a method is proposed to detect and track botnets on a large Tier-1 ISP network; it can only handle traditional IRC-based botnets. The authors of [12] focused on P2P botnets, investigating only one type of P2P topology at the AS level.

Our analysis has led to three key conclusions. First, a small number of ASes can observe a large fraction of overlay traffic, but the AS-level detection is less practical. PoPs cannot capture all traffic but can still identify a reasonable number of nodes in botnets. Router rendezvous strikes a balance between detection capability and feasibility. Second, a botnet has to make a tradeoff between resilience or efficiency and the ability to evade detection. Third, the infrastructure-level detection is not a panacea for all large-scale botnets: it needs to be integrated with detection schemes in edge networks to complete a detection picture.

The remainder of the paper is structured as follows. Section II describes the related work. Section III details our methodology. Section IV presents analysis results. Section V discusses a few challenges associated with our approach. The paper concludes with Section VI.

II. Related Work

As botnets have been a major security threat for a while, numerous approaches have been proposed to detect and mitigate them. Early detection methods [5], [6], [7], [8] aim at centralized botnets, i.e., IRC-based and HTTP-based. With the popularity of P2P botnets, however, defense mechanisms [13], [14] against this new generation of botnets have been developed. All of the above-mentioned approaches only apply to specific types of botnets requiring in-depth understanding of the C&C profiles prior to their detection. A few generic approaches can detect different types of botnets regardless of the C&C structure based on network packet and flow analysis [9] or combined host and flow analysis [10]. These approaches are effective for small-scale networks, such as in a campus or an enterprise network, but do not scale to large networks, because they need to obtain fine-grained information, such as packet content, flow patterns and host behavior.

Considering the fact that P2P botnets have structured overlay topologies, our approach takes a global view, exploiting structural properties derived from graph analysis and is thus not limited by the availability of fine-grained information. In this regard, our work is closely related to graph-based traffic classification and analysis. Iliofotou *et al.* [15] proposed the use of Traffic Dispersion Graphs (TDGs) to monitor, analyze, and visualize network traffic. TDGs focus on network-wide interactions among hosts and show that graph features, such as the average degree and directionality, can be utilized to distinguish different applications. Using TDGs, they further classified P2P traffic at the Internet backbone [16]. Their scheme filters out known traffic, forms traffic clusters roughly based on applications, and finally, uses some graph metrics to identify whether a cluster belongs to a P2P application. In our analysis, we adopt some of their metrics to determine whether the portion of traffic observed by a network component is P2P. BotGrep [17] analyzes structured graph to locate bots by extracting P2P subgraphs from a communication graph containing background traffic. This approach assumes the visibility of the entire botnet communication graph, whereas our detection is at a single network component where only a fraction of botnet communication can be seen.

We are aware of two published results on AS-level underlays mapped from P2P overlays. Rasti *et al.* [18] examined the global impact of the load imposed by a P2P overlay on the AS-level underlay. They use Gnutella network snapshots to analyze diversity and load on individual AS-paths, churn among the top transit ASes and propagation of traffic within the AS-level hierarchy. Their focus was on the effect of overlay on the underlay, while our work is concerned with whether the effect can be utilized for detection. Jelasity *et al.* [12] constructed a modified Chord [19] topology and showed that the visibility of P2P botnet traffic at any single AS is limited and not sufficient for detection. Our method differs from theirs in the following aspects. First, we consider bots' geographical distribution in the overlay topology while they assume that the number of overlay nodes in each AS is proportional to the size of the AS. Second, our AS-level paths are not derived from the shortest-path algorithm they used, but a more realistic scheme. Third, we simulate three P2P overlay topologies and observe the traffic not only at the AS-level but also at PoPs and router rendezvous, providing a more thorough analysis. We will show later that our observation is not the same as theirs.

III. Methodology

A. Overview

We would like to achieve two goals as follows. First, from a defender's perspective, we would like to see how much the botnet traffic can be observed at a single network component and whether the respective traffic graph has P2P properties. Second, from an attacker's perspective, we want to study which P2P overlay topology is stealthy enough so that at a single network component the graph-level information is not sufficient for detection. Our methodology consists of four main steps as shown in Figure 1. In the first step, we construct a P2P overlay topology based on simulation and learn which end-device talks to which, i.e., the overlay connections. In the second step, to map the overlay to the AS-level underlay, we associate a connection's two end-devices' IP addresses with the corresponding ASes and calculate the AS-level path between the two ASes. Given the AS paths, we then determine PoP-level paths and geographical router rendezvous paths. With knowledge of paths of all connections, in the third step, we break

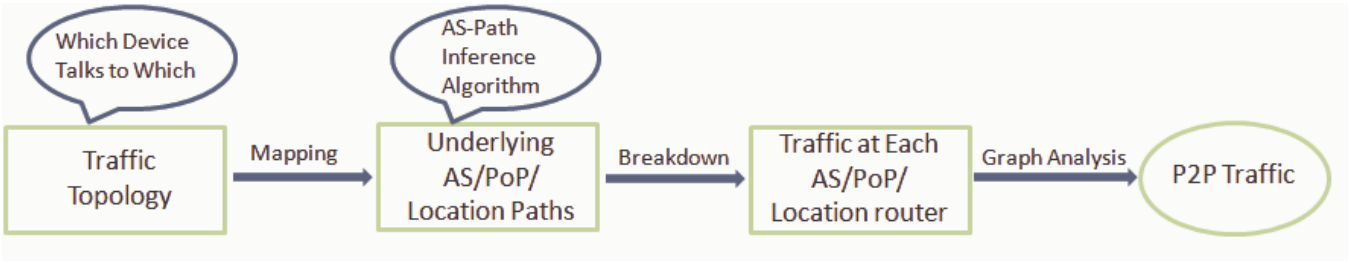

Figure 1. Overview

Table I
DATA SOURCES USED IN OUR INTERNET INFRASTRUCTURE AND END-DEVICE MODEL

Model component	Data sources
Backbone topology	Skitter dataset: http://www.caida.org/tools/measurement/skitter/ Alias clustering data from the iPlane project: http://iplane.cs.washington.edu/data/alias_lists.txt IP geolocation dataset: http://www.ip2location.com/
Internet Point of Presence	Telegeography co-location database: http://www.telegeography.com/
Internet end-devices	US census data: census-block population in each $250\times250m^2$ grid in US for a 24-hour duration [20]
Internet access routers	Dial-up service aggregators per each zip code: http://www.findanisp.com Broadband ISP market share: http://www.leichtmanresearch.com/press/081108release.html DSL central office locations: the LERG (Local Exchange Routing Guide) dataset from Telcordia Cable company service locations: Dun & Bradstreet (D&B) dataset
Internet routing	BGP routing information from the University of Oregon Route Views Project: http://www.routeviews.org/ AS prefix sets: http://www.fixedorbit.com/ AS-level path inference: Qiu and Gao's algorithm [21]

down the connections on per-AS, per-PoP, and per-router-rendezvous bases. We are especially interested in the top ASes, PoPs and router rendezvous ranked by the number of connections going through. In the last step, we inspect those top network components individually. As in [12], [18], we do not consider background traffic but focus only on the traffic coming from the P2P overlay, which is the best scenario, implying that if the P2P traffic cannot be identified under this situation, it will definitely not be captured when background traffic is present. We analyze several graph properties of the communication patterns at each top network component and determine whether it has the characteristics of P2P traffic.

B. Internet Infrastructure and End-Device Modeling

Before detailing the four main steps, we would like to briefly describe the Internet infrastructure and end-device modeling, which lays a basis for our methodology. We use multiple real-world datasets to construct a realistic model of the US Internet infrastructure. Table I lists all data sources in the model construction. In total, 73,884,296 residential computers are generated in the entire US (except Hawaii and Alaska). The distribution of Internet access routers including dial-up, DSL and Cable is based on the market share of top US broadband companies and dial-up service aggregators, and how these access routers connect to the backbone topology at Internet PoP locations is derived from AS peering relationships. We refer interested readers to [22] for details of this modeling.

C. Overlay Topology Construction

In recent years, P2P overlays have become popular in botnet construction due to their decentralized nature. Many existing P2P overlays can be utilized to facilitate botnets' C&C. We construct three types of P2P overlays: a widely-used Kademlia [4], a modified Chord [19] and a simple ring structure. We will first compare the structural properties of these three overlay topologies at each network component, the results of which will be presented in Section IV. Next, we will briefly introduce each P2P overlay followed by the way we construct the topology.

1) Kademlia: Kademlia is a Distributed-Hash-Table (DHT)-based P2P overlay protocol. Under this protocol, there is no central server and resource locations are stored throughout the network. Nodes are identified by node IDs and data items are identified by keys generated from a hash function; node IDs and keys are of the same length. Data items are stored in nodes whose IDs are close to data items' keys. The distance between two IDs, X and Y, is calculated by bitwise exclusive or (XOR) operation: $X \oplus Y$. To search a data item, a node queries its neighbors for nodes whose IDs are closest to this data item's key. After getting responses from its neighbors, the node continues to query those nodes that are closer to the key. This iterative process repeats until no closer nodes can be found. The benefit of Kademlia is its resilience to disruptions. Even if a few nodes are shut down or removed, the network will still be able to function. Kad network is an implementation of Kademlia. A few major P2P file sharing networks adopt the Kad implementation, such as Overnet and eMule. The Storm botnet was built upon Overnet.

An ideal way to construct the botnet overlay topology is to collect traffic traces from a real network, such as the Storm botnet. Since the Storm botnet is decentralized (i.e., there are no central venues that all communications can be

observed), traces captured from the Storm botnet fall into two categories each of which has its drawbacks. In the first category, the traffic data were collected from a single or a few vantage points. They can hardly provide a view of the entire botnet. In the second category, snapshots of the network were taken by network crawlers. The snapshots contain information, such as which IPs are alive or dead but cannot tell which IP connects to which IP. To characterize the effectiveness of detection at the underlay, a full picture capturing the entire network's connections is indispensable, so we have to construct a Kad network by using simulation.

We use a high-fidelity botnet simulator BotSim [23] which integrates a popular P2P client named *aMule* [24], an implementation of Kad. Considering the fact that simulating a large-scale botnet (100,000 bots) on a single or a few machines will take a prohibitively long time, our simulator was run on a distributed platform consisting of 400 machines, each with 2 Pentium III CPUs and 4Gb RAM. The simulator is a component of MIITS [25] which is built upon PRIME SSF [26], a distributed simulation engine utilizing conservative synchronization techniques. To make aMule work on our simulator, several modifications were made to the original aMule code including intercepting time-related system calls and substituting them for simulated time function calls, and replacing socket API calls with network functions developed in MIITS. The rest of the code remains intact.

In a botnet, a majority of bots are compromised residential computers and not necessarily geographically close, and hence we have to take locations into account. Constrained by data availability, all bots in our simulation are in the US and their locations follow the geographical distribution of 73 million residential computers by state. The simulation of 100,000 bots executes for 3 days in simulation time. The output files log timestamps and connections in the network. We discard the first day in which bots bootstrap and the entire botnet stabilizes, and keep the second and the third day for analysis. Now that we have log files keeping track of which node talks to which other node and each node's state information, we need to obtain the IP address of each end-device to completely construct the overlay topology. For this, we randomly choose an end-device address from the state a bot resides in. This way, we create two 1-day Kad overlay topologies of 100,000 nodes each.

2) Modified Chord: Chord is a DHT-based P2P protocol under which nodes form a ring structure. Each node has a predecessor and a successor and a few long range links. For example, there are a total of N nodes in the ring. Node i connects to nodes $(i-1) \mod N$ and $(i+1) \mod N$. It also connects to nodes $(i+2^k) \mod N$ for $k = 1, 2, \ldots, log_2 N - 1$ to form long-range links. In [12], modifications to Chord are proposed so that it is difficult to detect through graph analysis at any single AS. The main modification is to create clusters in the ring each of which has $\log_2 N$ consecutive nodes. This way, nodes in the same cluster can share the same set of long-range links for routing. This topology is of interest to us because we want to see whether using a more realistic AS-path calculation algorithm (as we will describe later) can make a difference in detection and whether this topology can successfully hide itself at PoPs and router rendezvous as well. Since this modified Chord's topology is relatively simple, we construct its overlay with 100,000 nodes directly based on its protocol without simulation. Following the same practice as in Kademlia, each end-device address is a random draw from the state a bot belongs to.

3) Simple Ring: We also consider the simplest case: each node has only two neighbors—a predecessor and a successor—to construct a ring structure. Presumably, this structure is stealthier and harder to detect than the modified Chord due to lack of connections among bots at the overlay. We will verify this assumption in later analysis. Similar to the modified chord, this overlay has 100,000 nodes constructed directly and the bots' locations follow the same geographical distribution.

D. Overlay to Underlay Mapping

1) AS-Path: Given all overlay connections, the next step is to map each connection to an AS-level path. Note that each end-device IP address is associated with an AS number and determining an AS-path of a connection is actually to determine the AS-path between two ASes. We use the AS-path inference algorithm in [21] for inter-domain routing. The key idea is to infer AS paths from existing BGP routing tables. For intra-domain routing, we use the shortest path algorithm.

2) PoP-Path: A PoP is an access point to the Internet. It is a physical location owned by an ISP or located at Internet exchange points and co-location centers. The computation of a PoP-level path is based on the respective AS-level path. Given a pair of source and destination end device IPs, the algorithm first determines the AS-level path $AS_1 AS_2 \ldots AS_n$ and then iteratively finds the shortest IP-level path between PoPs connecting every neighboring pair of ASes. We refer interested readers to [22] for details of this algorithm.

3) Router Rendezvous Path: Given an IP-level path, the geographical router rendezvous along this particular path can be determined. Thus, we can know all physical router rendezvous a connection goes through.

E. Traffic Breakdown

Since our work focuses on structural properties of traffic graph at a single network component (AS, PoP or router rendezvous), not the entire botnet overlay per se, with all the path information, we need to break the traffic down on a per AS, per PoP and per router rendezvous basis. This breakdown process is straightforward. We then rank the three

types of network components by the number of connections going through, and take a closer look at the graph properties observed at each top 10 ASes, PoPs, and router rendezvous in our analysis.

F. Graph Analysis

After breaking down the traffic, we know all connections that traverse a particular AS, PoP and router rendezvous. We can then generate directed graphs in which bots are represented by vertices and connections among them are represented by edges. For simplicity, all edges carry the same weight. Graph metrics to determine whether the traffic is P2P are proposed in [16] and adopted in analyzing the modified chord in [12]. In our analysis, we inspect the same set of features as in [12] for consistency. The features used to characterize P2P traffic include the number of weakly-connected components, size of the largest weakly-connected component, average node degree and InO (In Out) ratio. We introduce each of them as follows.

Number of Weakly-Connected Components: A weakly-connected component is a maximal subgraph of a directed graph such that in the subgraph replacing all of its directed edges with undirected edges produces a connected undirected graph. For effective detection, we expect a small number of weakly-connected components. As one can imagine, a large number of connected components usually means small-size components that are less likely to exhibit typical P2P patterns.

Size of the Largest Weakly-Connected Component: This metric is meaningful to us because as pointed in [15] the graph formed by a P2P network tends to be densely connected and have a large connected component including the majority of participating nodes.

Average Node Degree: This metric counts both the incoming and outgoing edges of a node, i.e., ignoring the directionality. A graph with a high average degree tends to be tightly-connected and P2P networks normally have high average node degrees.

InO Ratio: The metric calculates the percentage of nodes in the graph that have both incoming and outgoing edges. This metric is of interest because under client-server protocols such as HTTP and SMTP, clients usually initiate connections (outgoing edges) whereas servers normally accept connections (incoming edges). But nodes in P2P networks usually serve as both clients and servers so that P2P's InO is distinctively higher than others.

IV. Analysis Results

This section presents our analysis results. Recall that we construct three different P2P overlay topologies, namely, Kad, the modified Chord and the simple ring, and examine their traffic graphs, respectively, at three types of network components. As introduced in Section III-F, the graph features characterizing P2P patterns are the number of weakly-connected components, size of the largest weakly-connected component, average node degree and InO ratio. We conduct graph analysis first at the AS-level, then the PoP-level and finally, the router-rendezvous-level, and show the graph features at the top 10 places of each level.

A. AS-Level Analysis

We first take a look at the AS-level graphs of three different topologies. Table II shows the Kad graph properties for day1 and day2, respectively, at top 10 ASes, ranked by the number of unique connections going through. We map the AS numbers to ISPs using the AS-name lookup list [27]. It turns out that from day1 to day2 the top 10 order changes slightly but the 10 AS numbers remain the same. As expected, these top ASes belong to large ISPs such as AT&T and Verizon. Note that the traffic percentage at a single AS is calculated by the number of unique connections observed at that particular AS divided by the total number of unique connections in the entire overlay topology. In both days, top 10 ASes aggregated together can observe 98.95%—almost all of the Kad overlay's unique connections. We count unique connections not all connections across top 10 ASes, as one connection can be seen at multiple ASes. In particular, the top 1 AS (3356/Level3) alone can see two thirds of the overlay traffic with all nodes (100000) in the picture in both days. Even for ASes carrying less traffic, they have at least 99912 nodes' traffic traverse through. Most importantly, at each top AS, all nodes are weakly-connected with each other, forming one giant weakly-connected component. This property can facilitate detection because a huge portion of the overlay traffic captured in one single graph is more likely to demonstrate P2P characteristics and easier to get caught than a disconnected graph with many connected components with small sizes. As suggested in [16], two metrics can characterize P2P traffic. One is a high average degree (larger than 2.8), and the other is a high InO ratio (large than 1%). In both days, at all top ASes, the average degree and InO values are high enough for P2P classification: the lowest value of average degree is 56.8 and that of InO is 87.76%. Thus, as we can see, all top AS venues have high visibility of Kad traffic with P2P graph features, sufficient for detection.

Table III presents modified Chord graph features at top 10 ASes. Compared to Kad, top 10 AS numbers remain the same but their ranks change a bit. They in total observe 99.61%, an enormous fraction of overlay connections and the top 1 AS is still 3356 witnessing 64.25% of total traffic. Note that the AS observing the most can see 80620 while the one observing the least can only see 13900 nodes. As for the number of connected components, to the contrary of Kad, each AS's graph is not well connected and has thousands of connected components. Figure 2 shows in log scale the sizes of 10 largest weakly-connected components at top 5 ASes. Top 1 AS 3356's largest component has 36532 nodes but all other components are very small containing 15

Table II
KAD AS-LEVEL

Kad Day1 ISP	AS	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
Level3	3356	65.25%	100000	38192566	763.9	1	99.02%
AT&T	7018	35.33%	100000	20679083	413.6	1	99.02%
XO	2828	23.39%	100000	13691127	273.8	1	99.02%
Sprint	1239	8.32%	99983	4872140	97.5	1	99.01%
Verizon	19262	8.30%	100000	4859686	97.2	1	100.00%
Qwest	209	8.28%	100000	4848724	97.0	1	99.02%
NTT	2914	7.78%	99993	4556302	91.1	1	99.02%
BellSouth	6389	7.78%	100000	4554972	91.1	1	99.01%
AT&T	7132	6.78%	99995	3965587	79.3	1	100.00%
UUNET	701	5.38%	99937	3148400	63.0	1	88.13%
Kad Day2							
Level3	3356	66.69%	100000	39628509	792.6	1	99.02%
AT&T	7018	34.96%	100000	20772860	415.5	1	99.02%
XO	2828	24.18%	100000	14367036	287.3	1	99.02%
Qwest	209	8.35%	100000	4959076	99.2	1	99.02%
Sprint	1239	7.76%	99969	4611389	92.3	1	99.01%
BellSouth	6389	7.59%	100000	4509341	90.2	1	99.01%
Verizon	19262	7.23%	100000	4294952	85.9	1	100.00%
NTT	2914	7.06%	99988	4196433	83.9	1	99.02%
AT&T	7132	6.33%	99990	3761651	75.2	1	100.00%
UUNET	701	4.78%	99912	2839591	56.8	1	87.75%

Table III
MODIFIED CHORD AS-LEVEL

ISP	AS	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
Level3	3356	64.25%	80620	112431	2.8	9639	66.22%
AT&T	7018	38.09%	54272	66650	2.5	10534	51.62%
XO	2828	22.73%	36234	39784	2.2	7470	47.03%
Verizon	19262	9.43%	17365	16494	1.9	3726	37.01%
NTT	2914	8.09%	15339	14151	1.8	3384	34.45%
Sprint	1239	7.64%	14908	13366	1.8	3602	31.16%
Qwest	209	7.20%	14642	12594	1.7	3757	27.99%
AT&T	7132	7.13%	13849	12482	1.8	2956	33.29%
BellSouth	6389	6.82%	13486	11934	1.8	3080	30.47%
UUNET	701	6.27%	13900	10978	1.6	4305	16.41%

Table IV
SIMPLE RING AS-LEVEL

ISP	AS	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
Level3	3356	64.76%	79327	64755	1.6	14522	63.31%
AT&T	7018	37.51%	51316	37511	1.5	13805	46.20%
XO	2828	22.81%	32148	22805	1.4	9343	41.88%
Verizon	19262	9.30%	13632	9297	1.3	4335	36.40%
NTT	2914	8.05%	11867	8046	1.3	3821	35.60%
Sprint	1239	7.53%	11604	7532	1.3	4072	29.82%
Qwest	209	7.36%	11494	7362	1.3	4132	28.10%
AT&T	7132	7.07%	10430	7066	1.3	3364	35.49%
BellSouth	6389	6.73%	10193	6728	1.3	3465	32.01%
UUNET	701	6.17%	10831	6166	1.1	4665	13.86%

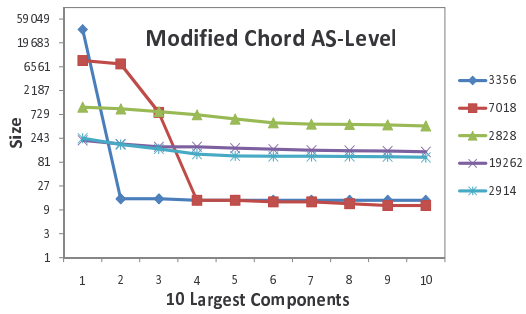

Figure 2. Modified Chord: 10 largest components at top 5 ASes

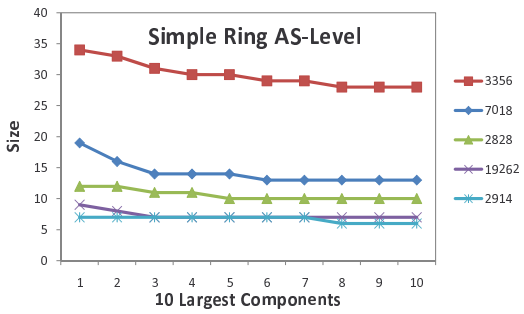

Figure 3. Simple ring: 10 largest components at top 5 ASes

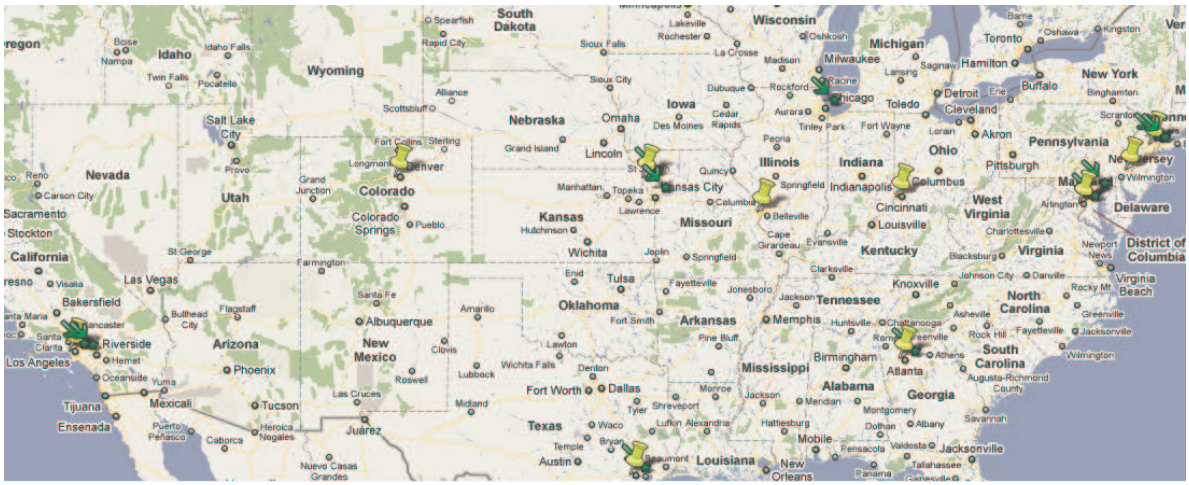

Figure 4. Top 10 PoPs (pins) and router rendezvous (arrows)

nodes or so. Top 2 AS 7018 has two large components with 8729 and 7506 nodes respectively and other components' sizes drop significantly. The component sizes remain stable at other ASes, all in the order of hundreds. Due to the topology's intent of hiding its traffic, unsurprisingly, the average degree at each AS is low—from 2.8 to 1.6, though the InO values are high—from 66.22% to 16.41%. Taking all metrics into account, AS 3356 is able to detect the P2P overlay since it can see a large portion of the overlay with typical P2P patterns, if not the entire one. If we relax the average degree threshold a bit, AS 7018 may also be a good venue to make detection efforts considering the two large connected components. We think it is hard for the rest of the ASes to do so due to their relatively fragmented views. Note that our observations on modified Chord are slightly different from those in [12] which concludes that even at the most central (top) ASes the average degrees are less than 2 and connected components are mostly of size 2 and 3 with the maximal containing 29 nodes. This difference may be attributed to the way of mapping the overlay to the underlay: they make the number of overlay nodes in each AS proportional to the size of the AS whereas we consider the geographical distribution of nodes. In addition, our AS-path inference algorithm is also different from theirs: they assume shortest paths while our inter-domain AS-pathes are derived from real-world BGP routing tables.

When it comes to the simple ring structure (Table IV), the top AS numbers do not change, and their ranks are the same as those for modified Chord. 99.62% of overlay connections traverse through top 10 ASes. Though the top1 AS 3356 can see 64.76% of the total traffic, the number of nodes visible (79327) are more than the number of edges (64755), resulting in a great number of connected components (14522) and small component sizes. As seen in Figure 3, 3356's largest component only has 34 nodes. We also verify that a majority of 3356 connected components have less than 10 nodes. The average degrees are all below 2, which is expected because each node only has a predecessor and a successor so that the average degree of the entire graph is only 2. Even though the InO values are high enough, detection based on scattered information at a single AS is difficult.

B. PoP-Level Analysis

At the PoP level, we also present graph features at each top PoP of three P2P structures. PoPs are represented by ID numbers and ranked by the number of unique connections going through as well. In Table V, as we can see, both the top 10 PoP numbers and their ranks change slightly from day1 to day2. Top 10 PoPs account for 80.88% of total traffic

Table V
KAD POP-LEVEL

Kad Day1 PoP	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
74	53.78%	100000	31479094	629.6	1	100.00%
7	10.29%	100000	6024939	120.5	1	99.94%
435	8.27%	100000	4837622	96.8	1	98.50%
11	8.14%	99998	4763870	95.3	1	99.86%
128	7.77%	99981	4550316	91.0	1	99.52%
282	7.37%	99995	4315967	86.3	1	100.00%
4	7.27%	99977	4257513	85.2	1	99.73%
267	6.72%	99992	3934199	78.7	1	100.00%
291	6.26%	99975	3661420	73.2	1	100.00%
295	6.25%	99997	3658911	73.2	1	99.97%
Kad Day2						
74	54.84%	100000	32588327	651.8	1	100.00%
7	10.06%	100000	5976120	119.5	1	99.97%
128	8.22%	99991	4883282	97.7	1	99.66%
11	8.06%	100000	4790115	95.8	1	99.87%
291	7.49%	99997	4450255	89.0	1	100.00%
435	7.41%	100000	4404198	88.1	1	98.60%
267	7.20%	99996	4279914	85.6	1	100.00%
4	7.19%	99967	4271196	85.5	1	99.67%
282	7.07%	99992	4199285	84.0	1	99.99%
239	5.88%	99879	3491615	69.9	1	99.65%

Table VI
MODIFIED CHORD POP-LEVEL

PoP	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
74	54.07%	77488	94629	2.4	16735	48.00%
7	9.27%	19927	16222	1.6	6095	21.91%
267	7.99%	14764	13981	1.9	3092	34.80%
11	7.98%	17225	13957	1.6	5334	18.75%
128	7.46%	17169	13058	1.5	5673	17.39%
4	7.25%	15962	12686	1.6	4834	20.36%
435	6.94%	13649	12151	1.8	3067	32.38%
282	6.81%	13677	11913	1.7	3184	31.41%
291	6.36%	12433	11137	1.8	2683	32.68%
295	5.84%	11877	10228	1.7	2803	29.32%

Table VII
SIMPLE RING POP-LEVEL

PoP	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
74	54.51%	75999	54506	1.4	21493	43.44%
7	9.40%	16165	9400	1.2	6765	16.30%
11	7.78%	13648	7779	1.1	5869	13.99%
128	7.63%	13765	7631	1.1	6134	10.88%
267	7.52%	11079	7521	1.4	3558	35.77%
4	7.31%	12505	7305	1.2	5200	16.83%
435	7.13%	10568	7127	1.3	3441	34.88%
282	7.08%	10587	7078	1.3	3509	33.71%
291	6.37%	9392	6373	1.4	3019	35.71%
295	5.77%	8829	5774	1.3	3055	30.80%

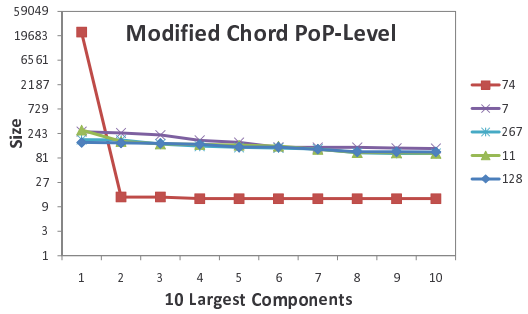

Figure 5. Modified Chord: 10 largest components at top 5 PoPs

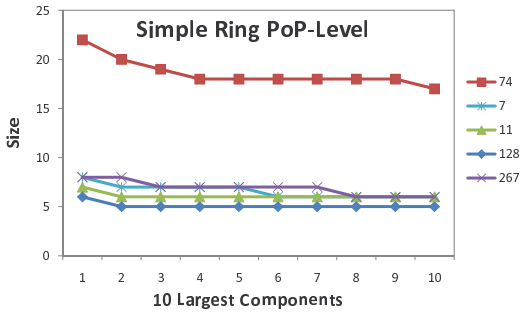

Figure 6. Simple ring: 10 largest components at top 5 PoPs

in day1 and 81.58% in day2, a slightly drop compared to traffic observed at top 10 ASes that can see more than 98%. This makes sense because PoPs, normally as traffic exchange points, are not able to see intra-domain traffic taking place within ASes. The top PoP 74 alone is able to observe 53.78% and 54.84% of traffic respectively in each day. Similar to the AS-level, not only almost all nodes (more than 99967) can be seen at each top PoP, but also they are weakly connected forming one single component. The average degrees and InO ratios are well above the P2P classification thresholds.

In the modified Chord's case as shown in Table VI, with a bit reordering, top PoPs are almost the same as those of Kad, taking up 80.29% of overlay connections in total. 74 is still the top 1 PoP observing 54.07% of total connections containing 77488 nodes, but all other PoPs observe less than 20000 nodes. As for sizes of weakly connected components, shown in Figure 5 in log scale, PoP 74's largest component is of size 23153 and others are quite small. Other PoPs' component sizes are less than 300. Considering average degree, InO ratio and the largest component size, if we tune the average degree threshold a bit, PoP 74 can be a good place for detection.

In simple ring's case (Table VII), the PoP numbers are exactly the same as those of modified Chord. Figure 4 shows the geographical locations of the top 10 PoPs denoted by pin icons. Note that they hardly change across the three structures and their locations are distributed throughout the US. 89.25% of total traffic reaches top 10 PoPs with 54.51% traversing PoP 74. Despite the fact that half of overlay connections can be observed at PoP 74, similar to AS-Level, the number of edges is smaller than the number of nodes. The largest component of PoP 74 is very small containing 22 nodes (Figure 6). It is the same case for all other top PoPs. Though InO values are moderate, low average degrees and a good many small connected components can prevent the P2P structure from being captured at any PoP.

C. Router-Rendezvous-Level Analysis

At the router-rendezvous level, we present results the same way as before. For the Kad structure, as shown in Table VIII, router rendezvous are denoted by ID numbers and ordered by the number of unique overlay connections observed. The top 10 router rendezvous are the same throughout the two days, altogether, see 89.75% of traffic in day1 and 89.27% in day2. The top 1 router rendezvous number 2 is reached by 68.77% of total connections in day1 and 68.91% in day2. A majority of nodes (more than 98579) appear in the graph as one giant component at each top router rendezvous. In addition, high average degrees and InO values make detection feasible.

Let us take a look at the modified Chord at the router-rendezvous level (Table IX). There is one new router rendezvous in the top 10 list that does not appear in that of Kad's and the ranks of the two lists are quite similar. Top 10 router rendezvous carry 89.96% of total connections and the top 1 router rendezvous is still 2 accounting for 68.76% of the traffic including 88913 nodes. As for the sizes of weakly connected components, the trend does not differ much from that at the AS- and PoP-level. The top 1 router rendezvous's largest connected component is of a big size—35943 nodes (Figure 7 in log scale) and other components have small sizes (less than 15). With a distinctive average degree and high InO value, this router rendezvous is a reasonable venue for capturing the modified Chord.

Finally, for the simple ring structure (Table X), the set of top router rendezvous is the same as that of Kad. Figure 4 illustrates all top router rendezvous for the three structures, each represented by an arrow with a star. Note that some of them are co-located with the top PoPs: in fact, PoPs are a subset of router rendezvous. Top 10 router rendezvous observe 80.54% of total traffic and router 2 sees 68.89% of traffic. With more nodes than edges at each top router rendezvous, it is difficult to get a full picture of the P2P overlay. Similar to AS- and PoP-level, the top 1 router rendezvous's largest component contains 33 nodes. The average degrees are unsurprisingly low, insufficient for detection.

D. Insights from Analysis

From the above analysis, we have several key observations worth noting. First, the visibility of Kad's overlay traffic and structure at all levels's top places is good enough for detection; the modified Chord's P2P characteristics can be captured by a few top locations but not all; and the information of the hypothetical simple ring's topology at all

Table VIII
KAD ROUTER-RENDEZVOUS-LEVEL

Kad Day1 Router	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
2	68.77%	100000	40251799	805.0	1	100.00%
2164	14.91%	99959	8728267	174.6	1	98.96%
12	11.90%	99997	6967203	139.3	1	84.22%
98	11.75%	100000	6874621	137.5	1	100.00%
222	9.26%	100000	5419174	108.4	1	99.99%
8919	8.30%	100000	4855632	97.1	1	98.50%
745	7.82%	99997	4579803	91.6	1	99.85%
82	7.33%	99978	4288889	85.8	1	99.74%
47	6.99%	98858	4090556	82.8	1	92.32%
88	6.67%	99997	3904395	78.1	1	99.71%
Kad Day2						
2	68.91%	100000	40945772	818.9	1	100.00%
2164	14.52%	99959	8626011	172.6	1	99.32%
12	11.57%	99989	6876147	137.5	1	83.77%
98	11.28%	100000	6702210	134.0	1	100.00%
222	9.05%	100000	5379049	107.6	1	99.98%
8919	7.41%	100000	4404198	88.1	1	98.60%
745	7.67%	100000	4559186	91.2	1	99.86%
82	7.24%	99973	4304038	86.1	1	99.68%
88	6.53%	99996	3881327	77.6	1	99.61%
47	6.18%	98579	3671730	74.5	1	90.15%

Table IX
MODIFIED CHORD ROUTER-RENDEZVOUS-LEVEL

Router	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
2	68.76%	88913	120337	2.7	13816	59.33%
2164	15.00%	29299	26245	1.8	8964	25.60%
12	11.57%	23682	20247	1.7	7629	20.07%
98	11.33%	21641	19821	1.8	5586	31.07%
222	8.73%	17779	15280	1.7	4771	27.68%
745	7.59%	16673	13275	1.6	5286	17.14%
82	7.29%	16133	12758	1.6	4926	19.98%
8919	6.94%	13649	12151	1.8	3067	32.38%
88	6.26%	12913	10962	1.7	3364	25.96%
57	6.16%	13606	10784	1.6	4029	19.42%

Table X
SIMPLE RING ROUTER-RENDEZVOUS-LEVEL

Router	Percentage	# of Nodes	# of Edges	Avg Degree	# of Conn Comp	InO
2	68.89%	88161	68885	1.6	19276	56.27%
2164	15.12%	25513	15122	1.2	10391	18.54%
12	11.35%	20126	11351	1.1	8775	12.80%
98	11.28%	17720	11275	1.3	6445	27.26%
222	8.93%	14243	8933	1.3	5310	25.44%
745	7.42%	13218	7419	1.1	5799	12.26%
82	7.36%	12653	7356	1.2	5297	16.27%
8919	7.13%	10568	7127	1.3	3441	34.88%
88	6.10%	9762	6102	1.3	3660	25.02%
47	6.06%	9669	6061	1.3	3608	25.37%

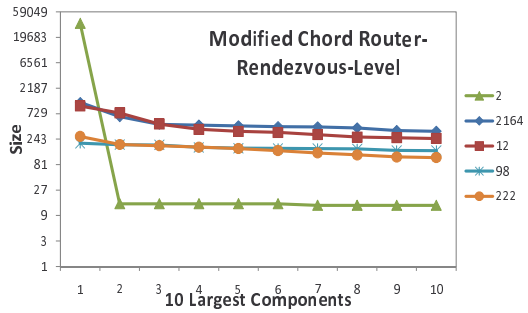

Figure 7. Modified Chord: 10 largest components at top 5 locations

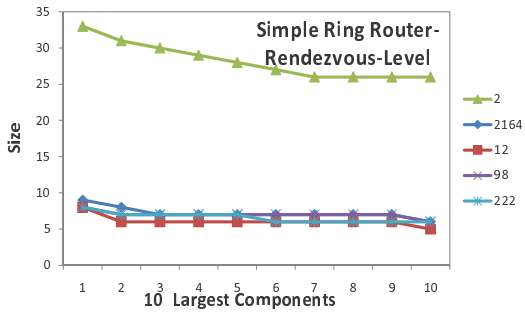

Figure 8. Simple ring: 10 largest components at top 5 locations

levels is quite fragmented and hardly useful for detection. From the attacker's viewpoint, in terms of efficiency, Kad has the most efficient routing: contacting $O(\log N)$ nodes during a search (where N is the size of the network); the modified Chord can achieve $O(\log^2 N)$ hops; and the simple ring is the worst, requiring $O(N)$ steps. From resilience's perspective, the Kad network is shown to be robust to a few types of mitigation strategies such as cutting off random nodes and removing peers learnt from bots' peerlists [28]; the simple ring structure is evidently fragile—removing a couple of nodes can disconnect the overlay; and the modified Chord structure hits the middle ground: not as resilient as Kad but better than the simple ring. We believe that, while constructing a P2P botnet, the attacker needs to strike a balance between resilience or efficiency and the ability to evade detection. Although the simple ring can hide its traffic well at various network components, to build upon this structure the botnet has to compromise resilience and C&C efficiency. The modified Chord makes a tradeoff though its structural properties cannot be concealed at some locations. Kad has been successfully utilized by the Storm botnet, but given our detection strategy, to use it for a future botnet, the attacker has to come up with techniques to mask its P2P patterns.

Second, from detection's perspective, AS-level provides better overlay traffic views overal than PoP- and router-rendezvous-level, but is less practical than the other two for actual detection deployment. Since AS is a logical concept and to capture all connections within one single AS requires collaboration and synchronization among multiple physical devices at different geographical locations, it may take some effort. PoPs normally function as traffic exchange points. From our analysis, we can see that at the PoP-level, detecting Kad and the modified Chord is very likely though the latter is only visible to the top 1 PoP. Compared to ASes and router rendezvous, PoPs observe less traffic due to the invisibility of traffic within ASes (intra-domain traffic). Moreover, the number of PoPs is small so that the points of monitoring are limited. Among the three, router rendezvous make a tradeoff. Their detection capabilities are comparable to PoPs' and they can observe intra-domain traffic with more monitoring points available, making detection more feasible.

V. Discussions

Thus far, we have measured the effectiveness of identifying P2P overlay traffic at various network components. For actual implementation of the detection at the Internet infrastructure, several challenges remain to be addressed.

First, since P2P networks implementing the same protocol may not be distinguishable at the structure-level via graph analysis, our techniques will also identify regular P2P file-sharing topologies. To differentiate between the two, our approach needs assistance from detection mechanisms at the edge networks for validation. If one node, in the traffic graph we observe, is identified as a bot by a local approach due to its malicious behavior demonstrated at the host or the network or both, we can infer that nodes in the same graph are part of the same botnet. Admittedly, although none of the state-of-the-art botnets shares the same network as a regular P2P file sharing does, if they do, more advanced detection techniques will be required to distinguish between the good and the bad.

Second, after identifying nodes of a botnet, to further mitigate or contain bots, we need to come up with efficient and effective techniques that can accommodate a large volume of traffic at the infrastructure level with minimal impact on the legitimate traffic. In the edge or local networks, fine-grained information of a particular node is available to detection mechanisms, and all incoming and outgoing traffic of the node can be controlled. Thus, after detection, taking the suspected node offline is not a difficult task. However, at the infrastructure level, a single network component may not have the ability and the confidence to remove a node completely so that advanced response mechanisms other than simply filtering or blacklisting are needed.

Finally, our models regarding the Internet infrastructure are abstracted from real-world datasets, so the accuracy depends on how well the datasets characterize the behavior and the state of the Internet, which could be error-prone. Moreover, some datasets may be outdated and may not reflect the current state of the Internet due to its fast-evolving nature. Therefore, these factors have to be taken into account when the infrastructure-level detection is put in practice.

VI. Conclusion

As P2P structures become a popular choice for recent botnets, especially large-scale ones, detection mechanisms have to keep up with this change and identify bots in an efficient and effective manner. In this paper, we propose detection of P2P botnets at a high-level—the infrastructure-level by analyzing their structural properties from a graph perspective. We construct three different P2P overlay topologies: Kad, the modified Chord and the simple ring. These overlays are mapped to the AS-level underlays and their respective AS-, PoP- and router-rendezvous-paths are inferred. Finally, we inspect these network components individually to measure their capability in identifying the P2P botnets. We find that detection at any of the three network components has its advantages and drawbacks. Overall, router-rendezvous-level detection is able to strike a balance between detection capability and feasibility. Also, a botnet needs to make a tradeoff between resilience and stealthiness.

References

[1] "Storm worm," http://en.wikipedia.org/wiki/Storm-Worm.

[2] "Waledac botnet," http://en.wikipedia.org/wiki/Waledac_botnet.

[3] "Conficker," http://en.wikipedia.org/wiki/Conficker.

[4] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of IPTPS*, 2001.

[5] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *In Proc. of NDSS*, 2008.

[6] J. R. Binkley and S. Singh, "An algorithm for anomaly-based botnet detection," in *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet*, 2006.

[7] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by irc nickname evaluation," in *Proceedings of HotBots*, 2007.

[8] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *HotBots'07*, 2007.

[9] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proceedings of the 17th USENIX Security Symposium*, 2008.

[10] Y. Zeng, X. Hu, and K. G. Shin, "Detection of botnets using combined host- and network-level information," in *Proceedings of 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2010.

[11] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in *Proceedings of HotBots*, 2007.

[12] M. Jelasity and V. Bilicki, "Towards automated detection of peer-to-peer botnets: On the limits of local approaches," in *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2009.

[13] 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 *Proc. First USENIX Workshop on Large-scale Exploits and Emergent Threats*, 2008.

[14] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the storm(peacomm)worm," SRI, Tech. Rep., 2007.

[15] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese, "Network monitoring using traffic dispersion graphs (tdgs)," in *Proceedings of IMC 2007*.

[16] M. Iliofotou, H. chul Kim, P. Pappu, M. Faloutsos, M. Mitzenmacher, and G. Varghese, "Graph-based p2p traffic classification at the internet backbone," in *Proceedings of 12th IEEE Global Internet Symposium*, 2009.

[17] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "Botgrep: Finding p2p bots with structured graph analysis," in *Proceedings of 19th USENIX Security Symposium*, 2010.

[18] A. H. Rasti, R. Rejaie, and W. Willinger, "Characterizing the global impact of p2p overlays on the as-level underlay," in *Proceedings of Passive and Active Measurement Conference (PAM)*, 2010.

[19] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM SIGCOMM*, 2001.

[20] T. N. McPherson and M. J. Brown, "Estimating daytime and night-time population distributions in u.s. cities for emergency response activities," The American Meteorological Society.

[21] J. Qiu and L. Gao, "As path inference by exploiting known as paths," in *Proceedings of IEEE GLOBECOM*, 2005.

[22] G. Yan, S. Eidenbenz, S. Thulasidasan, P. Datta, and V. Ramaswamy, "Criticality analysis of internet infrastructure," *Computer Networks*, vol. 54, no. 7, 2010.

[23] D. Ha, G. Yan, S. Eidenbenz, and H. Ngo, "On the effectiveness of structural detection and defense against p2p-based botnets," in *Proceedings of DSN 2009*.

[24] "amule," http://www.amule.org/.

[25] R. Waupotitsch, S. Eidenbenz, J. Smith, and L. Kroc, "Multi-scale integrated information and telecommunications system (miits): First results from a large-scale end-to-end network simulator," in *Proceedings of the Winter Simulation Conference*, 2006.

[26] "Prime ssf," https://www.primessf.net/bin/view/Public.

[27] "As names," http://bgp.potaroo.net/cidr/autnums.html.

[28] C. R. Davis, S. Neville, J. M. Fernandez, J.-M. Robert, and J. McHugh, "Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures?" in *Proceedings of ESORICS'08*.