On The Power and Limitations of Detecting Network Filtering via Passive Observation

Matthew Sargent[1], Jakub Czyz[2], Mark Allman[3], and Michael Bailey[4]

[1] Case Western Reserve University, Cleveland, OH, United States
[2] University of Michigan, Ann Arbor, MI, United States
[3] Intl. Computer Science Institute, Berkeley, CA, United States
[4] University of Illinois at Urbana-Champaign, Champaign, IL, United States

Abstract. Network operators often apply policy-based traffic filtering at the egress of edge networks. These policies can be detected by performing active measurements; however, doing so involves instrumenting every network one wishes to study. We investigate a methodology for detecting policy-based service-level traffic filtering from passive observation of *traffic markers* within darknets. Such markers represent traffic we expect to arrive and, therefore, whose absence is suggestive of network filtering. We study the approach with data from five large darknets over the course of one week. While we show the approach has utility to expose filtering in some cases, there are also limits to the methodology.

1 Introduction

In this paper we develop a methodology for broadly understanding policy-based network filtering across the Internet. We begin with three observations from previous work:

Policy-based Filtering Happens: We understand from experience and anecdote that network operators apply policy-based filters to traffic leaving their networks. These filters are used for myriad reasons, including (*i*) because particular traffic types are not meant to traverse wide-area networks (e.g., internal file sharing), (*ii*) to prevent services from being leveraged by external devices (e.g., using an internal mail server as an open relay), (*iii*) to funnel all user traffic through some proxy (e.g., to implement capacity-saving caching or content-based filtering) and (*iv*) to prevent propagation of malware. The community has previously taken modest steps to empirically understand such filtering. For instance, the Netalyzr [12] tool determines whether 25 popular services are blocked or not via active probing from within the network under study.

Missing Traffic Illuminates Network Behavior: Previous research shows that we can detect broad network outages by monitoring dark address space for the *curious absence* of traffic. In other words, when a large darknet suddenly receives no background radiation from a previously active network, we can conclude there is a change in policy. This has been studied in the context of both political events [10] which cause authorities to sever ties with the Internet, as well as natural disasters [3] which have the same impact on network traffic, even if these do not share the goal of policies that thwart communication of political adversaries.

Malware is Ubiquitous: A wealth of compromised devices on edge networks try to indiscriminately propagate using a set of vulnerabilities that span services [1, 18].

We believe the above suggests we can leverage the ubiquity of background radiation to form an expectation that specific *marker traffic* should arrive from a given origin network. When the expectation fails to hold, we are left with the strong suggestion of a policy-based filter hindering the specific kind of traffic in a given origin network. As a concrete exemplar, we study this technique in the context of over 96 billion Conficker packets that arrive at our darknet to form a broad understanding of TCP port 445 filtering in origin networks across the Internet.

By studying one week of traffic arriving at five /8 darknets—roughly 2.25% of the IPv4 address space—we find evidence that both supports and refutes our hypothesis. We find that in the case of Conficker—a large malware outbreak—detecting silence from a given origin network for a given kind of traffic does in fact allow us to understand the policy filters in place across the Internet. On the other hand, while we observe much malware in our datasets, we find each specific kind of traffic rarely spans enough of the origin networks to broadly develop an expectation that the given traffic should be present and thus develop conclusions based on the absence of such traffic. Therefore, we also learn that searching for silence in darknet traffic is limited to only significant events—i.e., full outages or large malware outbreaks. However, even with the limitations, we will show that the general approach does increase our broad understanding of policy-based traffic filtering.

2 Related Work

We leverage a number of technologies and techniques that have been developed by the community, including observing background radiation (e.g., [14, 18]), and using darknets as an observatory (e.g., [2]). None of this previous work addresses the topic of inferring service-level network policy via passive observation, which we tackle in this paper.

Meanwhile, studying policy-based network filtering of various kinds has previously been conducted via active measurements from the edge network under study (e.g., [8], [12], [4], [5]). The policies the previous work addresses are myriad—from the impact of bogon filtering to the ability to spoof packets to service-level policies. The wealth of work illustrates the interest in this topic. Our goals are similar to some of this previous work; however, our approach is to leverage passive measurements to understand the Internet broadly without the need to instrument every edge network, which is at best a large logistical undertaking.

The closest work to ours is in using the lack of background radiation from a given network to detect large scale outages that stem from natural disasters [3] or political events [10]. Our work shares their general notion that a lack of background radiation destined to a darknet can illuminate events within the network. We take this notion a step further and detect service-level policies applied to network traffic.

3 Data Collection

We use two primary sources of data for this study. The first dataset is a list of known Conficker infected hosts obtained via the Conficker domain sinkhole [13]. The Conficker worm [15] has been plaguing the Internet since 2008 and, six years later, continues to be

Table 1: Darknet data characterization.

Address Block	Packets (billions)	Bytes (trillions)	Rate (Mbps)	Rate (Kpps)	Source /24s (millions)
100/8	22.1	1.7	22.5	36.7	3.1
105/8	17.1	1.1	15.0	28.2	2.1
23/8	16.9	1.8	23.4	28.0	2.6
37/8	21.7	1.5	20.3	35.9	2.4
45/8	18.2	1.3	16.6	30.1	2.3
All	96.1	7.4	97.8	159	4.1

the top globally-detected worm in the first half of 2014 [11]. It propagates via several vulnerabilities in Microsoft Windows, as well as via dictionary attacks on passwords. Propagation via the network vector involves scanning random IPs on TCP port 445 [6]. A flaw in the random number generator results in Conficker only targeting IP addresses with both second and fourth octets less than 128, which effectively excludes more than three-quarters of addresses from ever being scanned [16]. One of the main ways that Conficker has been disabled by researchers is to pre-emptively determine and register botnet-related domain names—which are generated algorithmically—that the malware uses for command and control. Subsequently, by observing communication to these domains, we are able to discover IP addresses of Conficker-infected hosts [13]. The list of infected IP addresses we use in this study was collected at the same time as our darknet data (described below) and contains 17.5M Conficker infected hosts from 1.6M /24 networks.

The second dataset is a set of packet traces of traffic arriving at five unallocated IPv4 darknets: 23.0.0.0/8, 37.0.0.0/8, 45.0.0.0/8, 100.0.0.0/8, and 105.0.0.0/8. We obtained permission from the Regional Internet Registrars (RIRs) to simultaneously announce these network blocks for one week, January 14–20, 2011. We validated that our routes for these prefixes were globally visible to the majority of Route Views' [17] 121 peers during the week of our data collection. In aggregate, our darknet observes traffic to nearly 84M IPv4 addresses or roughly 2.25% of the usable IPv4 address space. While using darknets is a well-known technique (e.g., [18]), to our knowledge, this is the largest simultaneous IPv4 darknet collection to date.

In total, our darknet data comprises roughly 96.1B packets from 4.1M /24 address blocks in the Internet. Table 1 gives a broad characterization of our darknet data. Due to the lack of two-way traffic, we are unable to directly estimate how much measurement-based packet loss impacts our dataset. However, we have previously used the monitor to capture traffic at 1 Gbps without significant loss and the average rate of the darknet data is less than 98 Mbps. Therefore, we do not believe the amount of traffic our monitor failed to collect rises to the point of impacting our high-order conclusions.

Next, we classify the darknet data into five categories: (*i*) *Conficker* traffic represents TCP SYNs to port 445 from a known Conficker-infected host; (*ii*) *Likely Conficker* traffic includes TCP SYNs to port 445 from hosts not on the Conficker-infected host list but to an IP address that Conficker is known to target; (*iii*) *Scanning* traffic represents TCP SYNs that could not be produced by Conficker processes; (*iv*) *Backscatter* traffic represents SYN+ACK packets that are likely the result of SYNs spoofed to be from our

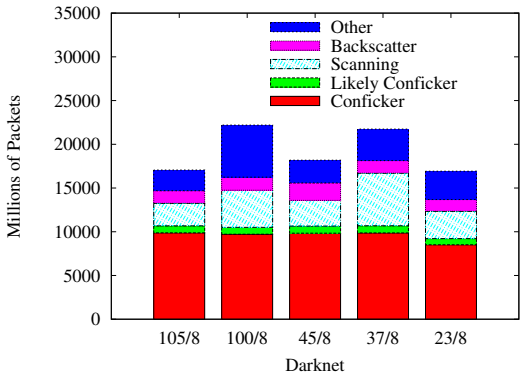

Fig. 1: Traffic volume by category for each darknet.

darknet; and (*v*) *Other* traffic, which includes all traffic not falling into one of the other categories. Figure 1 shows the breakdown of the traffic captured to each /8 we monitor. We note that the amount of Conficker traffic is relatively uniform across the /8 blocks we monitor.

A final caveat is that we cannot verify the source addresses in packets arriving at our monitor. We know spoofing is both possible and likely present—e.g., see the amount of backscatter in Figure 1 as an indication of the prevalence of spoofing. Therefore, in the remainder of the paper we take care to include this ambiguousness in our interpretation of the results.

4 Preliminaries

As we discuss in § 1, our hypothesis is that we can use the background radiation from malware to infer filtering policies across the Internet. In this section we offer several comments on the efficacy of this approach in general and also for specifically detecting policy-based TCP port 445 filtering.

General Coverage: A natural first question is whether we in fact observe traffic in our darknet from a broad spectrum of Internet endpoints. To quantify the fraction of the Internet that transmits traffic to our darknet we use routing tables from Route Views at the beginning of our darknet collection (January 13, 2011) to determine that 2.43B addresses are routed. The set of /24 networks we receive traffic from corresponds to 2.40B IP addresses when taking into account routed prefix size—or, 98.8% of the routed IP addresses. Some of this traffic is no doubt spoofed, so we compute the number of addresses belonging to /24s that send at least five scanning or backscatter packets.[5] We find 1.85B such addresses—or, 76.1% of the routed IP addresses. This analysis leads us to conclude that background radiation—and the lack thereof—arrives at our darknet

[5] Five is a somewhat arbitrary choice that weeds out /24 address blocks that send exceedingly little traffic for illustrative purposes.

from a broad spectrum of the Internet and therefore is a potential source of information about policy-based filtering in the Internet.

Conficker Coverage: While the amount and breadth of background radiation offers hope that we can broadly detect filtering policy, Conficker is an imperfect marker. As we note above, Conficker-infected endpoints are known to inhabit 1.6M of 4.1M /24 address blocks we observe sending traffic to our darknet. This partially stems from the error in Conficker that prevents it from scanning three-quarters of the network. While the footprint of the marker scopes the amount of the network we can assess, we are unaware of any other technique that achieves this level of coverage. While not ideal, we believe even an imperfect marker can provide a better understanding than we have today.

Conficker Behavior: Another preliminary question we must tackle pertains to the behavior of Conficker. Before we can infer that we are missing some marker traffic, we must have an expectation about how much such traffic we should observe. In order to remain undetected, Conficker infectees only scan after five minutes of keyboard inactivity on a given host [7]. Further, Conficker has four scanning modes—a number of them localized in scope. Finally, an infected host obviously cannot scan when the host is powered off. Given these constraints, we cannot simply compute an expectation based on a model of each host scanning continually and uniformly.

We can determine a rough idea of whether we should expect to observe traffic from each infectee, as follows. We know that, when scanning, each infected machine pauses between 100 msec and 2 sec between probes [7]. Given that we observe nearly 84M IP addresses, we would expect to observe one out of every 52 probes—or, one probe every 104 seconds if we assume the slowest scanning rate. Or, if we are to observe 10 probes from a given infected machine on each /8 we monitor, the host would have to scan for 86 minutes over the course of the week—or less than 1% of the week. Therefore, our first order assumption—which we revisit in § 5—is that we should observe Conficker activity from all infected hosts.

5 Validation

While the cursory analysis in § 4 suggests inferring policy-based filtering of TCP port 445 should be possible given both the proliferation of Conficker and our broad vantage point, this section tests our assumptions and frames the confidence we can gain from the results. We note that given the breadth with which we aim to develop understanding, we have no ground truth. Therefore, we cannot absolutely prove our inferences correct, but aim to illustrate that they are likely to be so.

An Anecdote: Comcast provides a list of ports that are subject to policy filtering for its residential customers—including TCP/445 [9]. In our darknet data we find nearly 3M packets from Comcast's 76.102.0.0/15 address block. As expected, we find no TCP/445 traffic even though our list indicates 81 Conficker-infected hosts within the given address block. While this is an obviously anecdotal case, it is illustrative of our goal to detect policy from the absence of specific traffic from given address blocks.

Conficker Sending Behavior: The preliminary analysis in § 4 suggests our darknet is big enough to observe all Conficker-infected hosts scanning with high probability based on what we know about Conficker's behavior. To check this we consider all Conficker

infectees from /24 address blocks where we observe some traffic to TCP port 445. In this case, we do not believe there is a general policy against TCP/445 traffic at the /24 level. However, we find TCP/445 traffic from only 51% of the infected hosts across these cases. Our data does not shed light on why we do not observe 49% of the Conficker hosts. The reasons could be many, including policy at finer granularity than a /24 (even to the host granularity), reactive filtering in response to scanning and removal of Conficker from the machine. We combat this situation by requiring multiple Conficker infectees per address block to overcome the seeming failure of some Conficker hosts to send scanning traffic.

Active Measurement: As part of its suite of active measurements, Netalyzr [12] attempts to establish a TCP/445 connection to a known server. We have obtained the Netalyzr test results starting one month before and ending one month after our darknet data collection. We find 1,555 hosts in the Netalyzr data that are also infected with Conficker. We therefore can evaluate our technique using the Netalyzr results as ground truth. First, we find 176 hosts (11%) where Netalyzr is run multiple times and shows inconsistent results. This shows that filtering policy and end-host behavior are not consistent across two months and therefore that the Netalyzr data is at best an approximation of ground truth with respect to the darknet data. For another 647 hosts, Netalyzr concludes a port-based filter is in place. The darknet data agrees with this assessment in 97% of the cases. In the 3% of the cases where Netalyzr concludes port filtering, we find a minimum of 17 TCP/445 packets from each host, with a median of 1,369 TCP/445 packets—and therefore we conclude that no filter is in place. We believe the likely cause for this is a policy change. Finally, Netalyzr finds 732 hosts to be unfiltered. However, we only observe 279 (38%) send traffic to our darknet, seemingly leaving our method with a large error. However, we note that the analysis in the last paragraph shows that we can only expect traffic from roughly half the infected Conficker hosts. Applying that expectation, the accuracy of the inference from the darknet data increases to 76%. As we note previously, the error can come from myriad places. Further, we show below that using multiple infected hosts can increase our confidence in our inferences.

Broad Comparison: Finally, we again compare our darknet observations with Netalyzr's results, but instead of using single IP addresses we will now aggregate results across /24 address block, routed block (determined from Route Views) and autonomous system. This allows us to bring multiple infected hosts to bear on our inference, but at the expense of possibly observing multiple policy domains.

Figure 2 shows the accuracy of our inference with respect to the Netalyzr results as a function of the number of Conficker infected hosts for the given aggregate block.[6] This plot first illustrates that regardless of level of aggregation the accuracy roughly levels off once a handful of Conficker infectees are present within the block. Second, the tighter we scope the block the better the accuracy, with /24 blocks showing the best accuracy, followed by routed blocks and then autonomous systems. We believe this is because as we increase the aggregation the instances of multiple policy domains also increases. Therefore, trying to treat the entire block the same leads to incorrect inferences.

We find that approximately half the hosts that contact the Conficker command and control structure ultimately show up in our darknet data. We see this manifests in the

[6] There are more Conficker infected hosts in some of the routed blocks and ASes, however, we truncate the plot at 255 for comparison with /24 blocks.

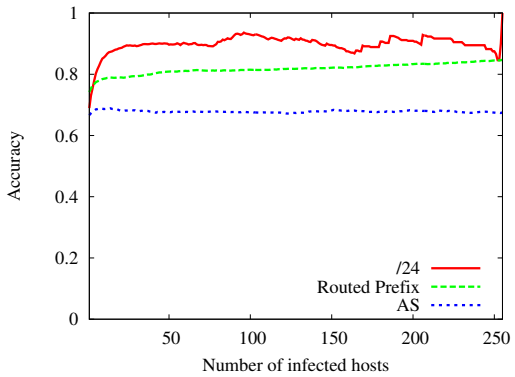

Fig. 2: Accuracy of the three methods when varying the number of infected hosts required before making comparisons with Netalyzr.

accuracy rate in Figure 2. Requiring five infected hosts per /24 should mean one of the Conficker infectees sends traffic with a 96% likelihood. When applying this threshold and comparing with the Netalyzr results we find an accuracy of 80%. In approximately 6% of the cases Netalyzr determines the network is filtering traffic while we observe Conficker from the given /24 in our darknet data. Finally, in 14% of the cases Netalyzr is able to establish a TCP/445 connection while we find no Conficker in our darknet collection and hence infer the given /24 is filtering TCP/445. While the reason for this discrepancy is not clear, we note that it will cause an over-estimate of the amount of filtering in the network.

Summary: As we show in this section, looking for the curious absence of traffic to understand fine-grain network filtering policy is not a clean process. We clearly need to understand the signal we expect to find. However, our conclusion is that, while this process is not perfect, we can use it to gain an approximate understanding of policy filtering in the network. Finally, while active measurements may be more precise, they are much more difficult to obtain on a large scale basis and therefore we are trading absolute precision for breadth of understanding.

6 Data Analysis

After establishing the promise of our methodology in § 4 and § 5, we now return to our high-level goal from § 1 to understand network filtering of TCP port 445 traffic using Conficker as a marker.

6.1 /24-Based Policy

As we develop above, we believe Conficker is a marker that will illuminate network filtering policy for the broad regions of the network where it is known to exist—even if

Table 2: Labels assigned to routed prefixes /23 or larger based on their component /24s.

Classification	Amount	Percentage
No Filtering	10,084	13%
Filtering	27,351	35%
Multiple Policies	14,536	18%
Low Signal	22,075	28%
Muddled/No Filtering	5,178	7%

the marker is less than ideal in some situations. As a starting point, we aggregate and label traffic based on the source /24 address block, our expectations of Conficker, and the traffic that arrives in our darknet.

First, as we sketch in § 4, we do not expect Conficker from roughly 60% of the /24 blocks observed at our darknet monitors. For roughly 0.2% of the /24 blocks from which we do not expect Conficker traffic we do in fact observe Likely Conficker at our darknet. This shows that the list of Conficker-infected hosts is comprehensive and not missing a significant portion of hosts infected with the malware. We do not further consider address blocks where we do not expect Conficker as we can infer nothing from its absence in these cases.

This leaves us with Conficker infectees in roughly 40% of the /24 address blocks in our darknet data. We now need a process to label each /24 address block by its filtering policy. Given our validation work in § 5, we proceed in two steps. First, when we observe Conficker traffic from a /24 block we determine there is no general TCP/445 filtering. Second, we know we cannot expect Conficker from all infectees, and so the absence of the marker does not necessarily indicate a network filter. Rather, we determine a /24 block is filtering TCP/445 when (*i*) we find no TCP/445 traffic in our darknet data and (*ii*) the /24 block has at least five infectees. As we develop in § 5 the second criteria gives us at least 96% confidence that Conficker should arrive and therefore when it does not we infer a policy-based filter.

We find 434K (27%) of the 1.6M /24 blocks with Conficker infectees are not imposing TCP/445 filtering on their traffic. Meanwhile, we infer that 448K /24 blocks (28%) filter TCP/445 traffic. That is, we are able to confidently characterize the filtering policy of 882K /24 networks—or 9.3% of all the routed address space. This is, by far, a larger portion than previous methodologies can claim—e.g., Netalyzr runs from the month surrounding our data collection cover 23K /24 networks. Our analysis leaves 747K /24 blocks (45%) from which we do not observe TCP/445 traffic but which do not contain five infectees. These are cases where we have an indication of possible filtering, but cannot develop high confidence in this determination.

6.2 Routed Prefix-Based Policy

We next turn to a larger aggregation of address blocks to better understand filtering policy at a coarser granularity. We leverage routed prefixes as found in Route Views at the time of our darknet data collection for this analysis. Our general method to infer

whether filtering happens for an entire prefix is to look for consistent behavior from the /24 blocks within the prefix. Since we tackle /24 address blocks above, in this section we only study the 140K routed prefixes that are at least a /23 (out of 254K total routed prefixes).

Of the 140K prefixes we consider, we find no Conficker infectees and no TCP/445 traffic for 61K of the prefixes. We cannot further study these prefixes as we have no expectation of TCP/445 traffic and therefore the absence of such traffic does not inform our assessment of filtering. This leaves roughly 79K prefixes on which we have some expectation of observing TCP/445 traffic. We summarize our results in Table 2.

First, when each /24 block containing at least one Conficker infectee within the routed prefix produces TCP/445 traffic we conclude the network applies no general TCP/445 filtering. Table 2 shows 13% of the prefixes do not filter TCP/445. Similarly, when we observe no TCP/445 traffic for each /24 block containing at least one infectee across a prefix with at least five total infectees we conclude filtering is in place for the entire prefix. We find prefix-wide filtering in 35% of the prefixes. We also find cases where no TCP/445 traffic arrives at our darknet, but the routed prefix contains fewer than five infectees. We cannot confidently determine that these prefixes filter TCP/445—even if the data suggests this may be the case. We denote these cases "low signal" in the table and find 28% of the prefixes fall into this category.

Finally, we are left with prefixes that have indications of both no filtering—i.e., we observe TCP/445 traffic—and filtering—i.e., the infectee list suggests we should observe more TCP/445 traffic than we do. For cases where we observe traffic from at least five infectees we conclude that the prefix has multiple policies. In other words, we are confident in our determination that filtering is occurring within the prefix and yet we still observe TCP/445 traffic from the prefix. We find this happens in 18% of the cases. As the size of the address blocks we consider increases this is a natural finding that follows our intuition—i.e., that the block would be split up into multiple policy domains. Finally, we have cases where we observe TCP/445 traffic and there are also indications we should see additional traffic, but from less than five infectees. In this case, we know filtering is not in use across the entire prefix and, even though we have some indication that filtering may be happening, we cannot conclude it is with confidence. We find 7% of the prefixes in this "muddled" state.

We next consider the fraction of each prefix we use to determine its filtering policy. For each routed prefix, we calculate the fraction of the constituent /24 blocks (*i*) with a known Conficker infectee and (*ii*) where we conclusively determine that filtering is or is not present. Figure 3 shows the distribution of prefixes according to these fractions. The "all" distribution in the plot shows the expected prefix coverage based on the Conficker infectee list, whereas the "classified" distribution shows the fraction of /24 blocks we actually use in concrete prefix classifications. Comparing the distributions shows that, when making a classification, we generally use more of the prefix (i.e., more /24s) than the expectation predicts, which adds to our confidence in the classifications.

Next, we examine the size of the routed prefixes we are able to concretely classify. The distribution of the size of all routed prefixes we consider, as well as the distributions of the routed prefix sizes for each concrete classification we make are given in Figure 4. The figure shows that the distribution of network size for networks we can concretely

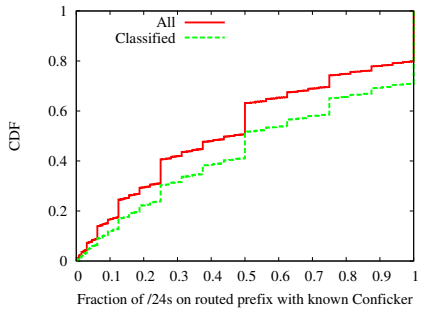

Fig. 3: CDF of the fraction of /24s on a routed prefix with known Conficker.

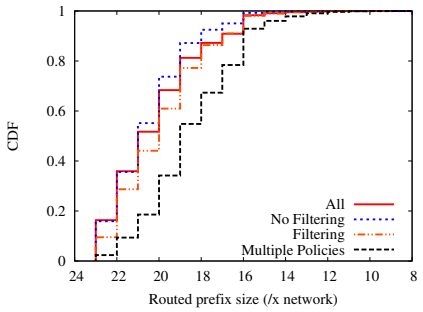

Fig. 4: CDF of the routed prefix sizes on which we make judgements.

detect filtering policy is similar to the distribution of the size of all origin networks. In other words, neither our detection nor results are biased by prefix size. Further, we find that networks that filter TCP/445 are slightly larger than networks that do not filter TCP/445. This perhaps indicates that operators of larger networks are more diligent about security policy than those of smaller networks. Finally, we find that networks with multiple policies are larger than networks with a single policy. As we note above, this is natural because as network sizes increase the tendency to have multiple administrative and policy domains to cope with a variety of situations arises.

Finally, we note that we are able to confidently determine a single filtering policy in roughly half of the /23 and larger routed prefixes. This corresponds to 699M IP addresses or 28% of the routable addresses during the week of our darknet data collection.

7 Limitations

From previous research we understand that full network outages—whether caused by policy decisions or natural disasters—can be detected by the absence of traffic arriving at darknets. Further, in the previous sections we illustrate that we can use similar strategies to infer finer-grained policy such as port blocking. As developed thus far, both the course- and fine-grained policy discovery requires big events—i.e., a broad swatch of the Internet becoming unreachable or malware that is both prevalent and energetically propagating.

A natural next question is whether the aggregate background radiation that appears at darknet monitors provides enough information to form further general understanding of policies across the Internet. To address this question we first determine the top TCP ports arriving at our darknet.[7] We then calculate the number of origin /24 networks that source each kind of traffic and compare this to the total number of origin /24s we observe. Table 3 shows the results. In the best case—port 80—we find SYNs from only 18% of origin /24s we observe. This either means 82% of the /24s either (*i*) are subject to policy blocking or (*ii*) do not source radiation to port 80. We believe the latter is far more likely

[7] We included UDP in our analysis, but elide it from this discussion due to space constraints and its similarity with the TCP results.

Table 3: Percentage of /24s observed sending TCP SYNs to other prevalent destination ports in the measured darknets.

Darknet	# /24s Receiving SYNs	% /24s w/SYN for TCP/80	TCP/139	TCP/1433	TCP/22
100/8	2.0M	14.2%	1.5%	<1%	<1%
105/8	1.5M	4.0%	1.1%	<1%	<1%
23/8	1.7M	6.2%	1.0%	<1%	<1%
37/8	1.6M	21.6%	1.0%	<1%	<1%
45/8	1.6M	5.6%	1.1%	<1%	<1%
All	3.1M	18.2%	1.3%	<1%	<1%

than the former. That is, background radiation does not in general energetically target our darknet enough to develop a solid expectation that the traffic should be there and hence draw conclusions about its absence. Further, for the other top ports the prevalence is even smaller than for port 80 and, hence, makes any conclusions about policy even more tenuous.

Therefore, our conclusion is that while the general technique of searching for the absence of traffic can be useful, it has its limits.

8 Conclusions

This paper makes several high-order contributions:

Methodology: We develop a novel methodology for detecting service-level network filtering based on passive observation of traffic markers. While this aspect of the Internet has been previously studied, our passive observation-based technique allows for developing an understanding at a breadth previously unattainable. Using Conficker as our exemplar, we are able to conclusively determine the network filtering policy of 699M IP addresses or roughly 28% of the routed IPv4 address space. Although this is a modest fraction of the Internet, it is much larger than previous attempts. For instance, the original Netalyzr study [12] reports results from 100K test runs. Even if each Netalyzr run represents a /24 network our results cover 27 times as much of the Internet.

State of TCP/445: Of the address space we can conclusively assess, we find filtering of outgoing TCP/445 traffic occurs in two-thirds of the cases. We also note that as the size of the routed prefix under study increases the chance of finding multiple service-level filtering policies within the prefix also increases. While we believe it is a natural and expected result that larger networks would encompass more than one administrative and policy domain, we believe this offers a cautionary note in that aggregating too much of the network can dilute any understanding we derive.

Methodological Limitations: Finally, we illustrate that there are limits to the methodology of using the absence of background radiation to infer policy. In particular, we can leverage large events to infer policy, but more run-of-the-mill instances of background radiation are not energetic and wide-spread enough to allow us to form the expectation of traffic and hence draw conclusions when the expectation fails.

Acknowledgments

We would like to thank Christian Kreibich for the Netalyzr data, Phillip Porras for the Conficker sinkhole data, and Vern Paxson for comments on an earlier draft. This work is sponsored by NSF grants CNS-1213157, CNS-1237265, CNS-1505790 and CNS-1111699.

References

1. M. Allman, V. Paxson, and J. Terrell. A Brief History of Scanning. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*, IMC'07, Oct. 2007.
2. M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson. The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In *Proceedings of Network and Distributed System Security Symposium*, NDSS'05, pages 167–179, 2005.
3. K. Benson, A. Dainotti, k. claffy, and E. Aben. Gaining Insight into AS-level Outages through Analysis of Internet Background Radiation. In *Traffic Monitoring and Analysis Workshop*, TMA'13, Apr 2013.
4. R. Beverly, A. Berger, Y. Hyun, and k. claffy. Understanding the efficacy of deployed internet source address validation filtering. In *Proceedings of the ACM SIGCOMM conference on Internet Measurement*, IMC'09, 2009.
5. R. Bush, J. Hiebert, O. Maennel, M. Roughan, and S. Uhlig. Testing the reachability of (new) address space. In *Proceedings of the SIGCOMM workshop on Internet Network Management*, INM'07, pages 236–241, New York, NY, USA, 2007. ACM.
6. CAIDA. Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope. http://www.caida.org/research/security/ms08-067/conficker.xml, 2013.
7. E. Chien. Downadup: Attempts at Smart Network Scanning. http://www.symantec.com/connect/blogs/downadup-attempts-smart-network-scanning, Jan. 2009.
8. D. R. Choffnes, F. E. Bustamante, and Z. Ge. Crowdsourcing service-level network event monitoring. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM'10, 2010.
9. Comcast. Blocked Ports List. https://customer.comcast.com/help-and-support/internet/list-of-blocked-ports/.
10. 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. IMC '11, 2011.
11. F-Secure. Threat Report H1 2014. http://www.f-secure.com/documents/996508/1030743/Threat_Report_H1_2014.pdf, 2014.
12. C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: illuminating the edge network. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*, IMC'10, 2010.
13. J. Kristoff. Experiences with conficker c sinkhole operation and analysis. In *Proceedings of Australian Computer Emergency Response Team Conference*, 2009.
14. R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. In *Proceedings of the ACM SIGCOMM conference on Internet Measurement*, IMC'04, 2004.
15. P. Porras, H. Saidi, and V. Yegneswaran. An analysis of conficker's logic and rendezvous points. Technical report, SRI International, Mar. 2009.
16. M. Richard and M. Ligh. Making fun of your malware. Defcon 17, 2009.
17. University of Oregon. Route Views project. http://www.routeviews.org/.
18. E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Houston. Internet Background Radiation Revisited. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*, IMC'10, 2010.