Classifying Internet One-way Traffic

Eduard Glatz
ETH Zurich
Zurich, Switzerland
eglatz@tik.ee.ethz.ch

Xenofontas Dimitropoulos
ETH Zurich
Zurich, Switzerland
fontas@tik.ee.ethz.ch

ABSTRACT

Internet background radiation (IBR) is a very interesting piece of Internet traffic as it is the result of attacks and misconfigurations. Previous work primarily analyzed IBR traffic to large unused IP address blocks called network telescopes. In this work, we build new techniques for monitoring one-way traffic in live networks with the main goals of 1) expanding our understanding of this interesting type of traffic towards live networks as well as of 2) making it useful for detecting and analyzing the impact of outages. Our first contribution is a classification scheme for dissecting one-way traffic into useful classes, including one-way traffic due to unreachable services, scanning, peer-to-peer applications, and backscatter. Our classification scheme is helpful for monitoring IBR traffic in live networks solely based on flow-level data. After thoroughly validating our classifier, we use it to analyze a massive data-set that covers 7.41 petabytes of traffic from a large backbone network to shed light into the composition of one-way traffic. We find that the main sources of one-way traffic are malicious scanning, peer-to-peer applications, and outages. In addition, we report a number of interesting observations including that one-way traffic makes a very large fraction, i.e., between 34% and 67%, of the total number of flows to the monitored network, although it only accounts for 3.4% of the number of packets on average, which suggests a new conceptual model for Internet traffic in which IBR traffic is dominant in terms of flows. Finally, we demonstrate the utility of one-way traffic of the particularly interesting class of unreachable services for monitoring network and service outages by analyzing the impact of interesting events we detected in the network of our university.

Categories and Subject Descriptors

C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations—*Network monitoring*

General Terms

Measurement, Security

Keywords

Measurement methods, Traffic Analysis (anomaly detection, classification)

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

1. INTRODUCTION

Studying the background radiation of the Internet has been instrumental for understanding Internet threats, like the prevalence of Denial of Service (DoS) attacks [29] and the propagation of Slammer [27] and Blaster [4] worms. IBR traffic has been primarily studied with the help of network telescopes [28], which are large unpopulated IP address blocks (e.g. /8) that seamlessly filter out all benign traffic. In live networks, i.e., popuated networks IBR traffic can be extracted by dissecting two-way from one-way traffic, i.e., traffic connections that do not receive a network reply. In this work, we build techniques for monitoring one-way traffic in live networks.

We make three main contributions. First, we *design and validate a comprehensive classification scheme for dissecting one-way traffic* that relies solely on flow-level data, e.g., NetFlow. Our scheme uses 13 carefully-designed rules, which encode novel heuristics as well as leverage and combine existing techniques that have been proven effective for certain types of traffic. We associate each one-way flow with up to 17 different *signs*, which help dig into the causes of one-way flows. The key features of our scheme are that: 1) it relies on flow data; 2) it is very easy to configure; 3) it provides transparency into the classification process; and 4) it is easily extensible.

Our second contribution is that *we use our classification scheme to shed light into IBR traffic towards a large live network by analyzing a massive data set of unsampled flow records summarizing 7.41 petabytes of traffic* recorded over eight years (from 2004 to 2011). We find that in the studied years one-way traffic makes between 34% and 67% of the total number of flows, but it only accounts on average for 3.4% of the packets. This suggests a new conceptual model of Internet traffic in which IBR traffic is a needle in a haystack in terms of packets (and bytes), but it is very dominant in terms of flows. Besides, we find that scanning is the largest component of one-way traffic corresponding to 83.5% and 62.6% of the one-way flows and packets, respectively. The second major contributor is traffic caused by peer-to-peer (P2P) applications. Finally, we find that in 2011 the vol-

ume of IBR traffic in terms of flows is almost equal to 2004 with small fluctuations in the years between. The fraction of IBR traffic in the total number of flows has been consistently decreasing from 67% in 2004 to 34% in 2011.

Third, *we show how one-way flows assigned to the class of unreachable services provide new possibilities for service availability monitoring enabling in particular to passively assess the impact of outages*, which is not possible with traditional approaches based on active probing. To demonstrate the utility of our scheme, we describe and measure the impact of interesting outages and misconfiguration we discovered in the network of our university.

In the next section we provide preliminary insights into the one-way traffic classification problem. Next, in Section 3 we describe our data sets and their sanitization steps. Our classification scheme is introduced in Section 4 and validated in Section 5. In Section 6 we use our scheme to characterize the composition of IBR traffic in a live network, while in Section 7 we outline how our work is useful for service availability monitoring. Finally, we present related work and conclude our paper in Section 8 and 9.

2. PRELIMINARIES

Network communication inherently reflects dialogs taking place between applications running on distributed systems including any accompanied control and error messages. Regular communication involves two-way traffic consisting of packets in opposite direction. In contrast, one-way traffic results from communication errors or other unusual situations that result in packets in one direction. In this work, we look at one-way communication at the network level. We define a *two-way flow* (or equivalently a *bi-flow*) as the aggregate of two standard 5-tuple flows that have reverse values in the source and destination IP address and port number fields. A 5-tuple traffic flow is a flow of packets that have common values in the IP address, port number, and layer-4 protocol fields. We define an *one-way flow* as a flow that does not have a matching reverse flow. A two-way flow may still be the result of a failed communication at the transport or application level, e.g., a TCP reset packet produces a two-way flow, therefore *one-way flows provide a conservative view of failed communication attempts* in the Internet.

One-way traffic is important as it is associated with interesting events, such as unreachable services, scanning, and misconfigurations. To draw useful information from one-way traffic we need to infer what is causing it. The goal of one-way traffic classification is to classify one-way traffic into classes of malicious and benign causes. We identify three main causes of one-way traffic:

- **Failures & Policies:** this group comprises *attempts to access a service that do not succeed* due to failures or policies. Possible reasons are firewall blocking, temporary or permanent outages of network elements, stopped service processes on end-hosts, and routing misconfigurations. For example, the use of laptops outside of enterprise networks they are configured for produces one-way traffic. Similarly, one-way traffic is the result of attempts to use services that are blocked by security policies, like when testing connectivity with the ping tool to non-responding destinations. Furthermore, one-way traffic can in special cases result from network congestion.

- **Attacks:** *vulnerability scanning* is a very common attack vector. However, unsolicited probes typically do no receive a reply resulting in many one-way flows. Another attack case is *backscatter traffic* from DoS attacks using spoofed source IP addresses, which also results in one-way flows towards the spoofed IP addresses. Third, *prefix hijacking* results in one-way traffic towards a "black hole".

- **Special application behavior:** *P2P applications* try to reconnect to systems they have been in contact before as such systems are often preferred peers by application design. However, the dynamic nature of P2P networks and in particular nodes unexpectedly leaving a network result in one-way reconnection attempts. A second, but rare scenario is *applications using sessions that run the two communication directions over different connections*. For example, an interesting case we have identified is the application layer protocol of LANsource [15] that runs over UDP and uses different destination ports for each communication direction.

Properly identifying Internet one-way traffic from measurement data imposes specific requirements on monitoring. In particular, traffic sampling and poor spatial network coverage, i.e., missing one direction of a communication that may take a different path due to asymmetric routing, may result in artificial one-way flows. In principle, dealing with asymmetric routing requires to either know how flows are routed or to monitor every link along a network cut between a monitored network and the Internet. For these reasons, it is practically easier to measure one-way traffic near the edge, e.g., in enterprise, university, or small ISP networks, where it is possible to monitor the entire border traffic activity without sampling. Edge networks comprise the vast majority of Internet domains.

3. DATASETS AND SANITIZATION

We use data from a regional academic backbone network that serves 46 single-homed universities and research institutes. The monitored address range contains 2.2 million IP addresses, which correspond to a continuous block slightly larger than a /11. We have been archiving unsampled flow records since 2003 from hardware-based NetFlow meters on the border routers of the monitored network. The meters capture all the traffic that crosses the border destined to or coming from the Internet. In a single peering link, we observe in 2011 on average 108.1 million flows per hour, which corresponds to 3,064 million packets. The stored flow records are not anonymized. From each record we extract for our classification scheme the following fields: IP addresses, port numbers, protocol number, byte/packet counts, and timestamps. We do not use TCP flags because they are not supported by the fast hardware-based NetFlow, although they can be easily integrated in our classification. Over time we have accumulated a massive archive with more than 100 TBytes of compressed NetFlow records.

A challenge in analyzing our data is computational overhead especially for tasks that cannot be parallelized. For example, using a optimized C++ program to parse one day of data and compute the average number of one-way flows per host takes 10 hours and 10-20 Gbytes of main memory. The computational overhead includes time for disk I/O, de-

compression (*bzip2*), sorting flows by their start time, preprocessing (as discussed in the following paragraphs), and updating a hash table, which keeps per host state. Memory consumption is dominated by the hash table. The computation time can be drastically reduced by using flow meters that support on-the-fly bidirectional flow monitoring, like YAF [16]. To make the computational overhead more tractable, we use two 400-hour samples per year, which correspond to approximately one month per year. In particular, we analyze the first 400 hours of each February and August between 2004 and 2011. The studied traffic data correspond to 457 gigaflows and 7.41 petabytes and cover approximately 9% of the total number of flows observed during the eight year period. This is one of the largest traffic traces that has been analyzed in the literature.

For each 400-hour sample, in the top rows of Table 1 we report the time the monitoring and collection infrastructure was functioning properly and the traffic volume it corresponds to. Overall the availability of the infrastructure was higher than 99.5% with only short interruptions due to router restarts and corrupted data. To eliminate the impact of interruptions in our analysis, we use daily averages and discard the days in which the short interruptions occurred.

3.1 Data Sanitization

In this section we describe how we addressed a number of data sanitization challenges for extracting one-way flows. In particular, we outline our methodology for eliminating double-counting, defragmentating flows, and pairing bi-flows. To speed up searching for matching flows, we split time into intervals and limit our search within an interval. In Section 3.1.4 we analyze the impact of the interval size. In the future, the standardization of bidirectional flow monitoring [34] will likely make one-way flow extraction much easier.

3.1.1 Double-Counting Elimination

Although we collect flows from border routers only, if a flow is routed through multiple border routers, it will be reported multiple times in our NetFlow data. To address this problem we use a map provided to us by network operators of SWITCH (ISP serving Swiss Universities) that specifies which ports of the border routers are connected to the internal backbone and which to the Internet. Our NetFlow data carry information about the input and output port of a flow. We use this information to filter out flows. We only keep flows that are routed between the Internet and the intranet. Double-counting elimination has a significant impact on our data reducing the total traffic volume by 32.3%.

3.1.2 Defragmentation

We call the standard 5-tuple NetFlow records *raw flows*. NetFlow may fragment a traffic flow when the active timeout expires or when the flow cache is filled up and flushed. In addition, the optional NetFlow *fast aging* feature, with which our meters are configured, leads to additional fragmentation. Fast flow aging reduces the utilization of the flow table by exporting a flow if no more than a few packets are observed within the first few seconds of the creation of a flow table entry. It realizes an efficient process to age out entries for short flows. To address flow fragmentation due to fast aging, we split time into intervals and for each raw flow we search within its time interval for other raw flows with the same 5-tuple. We merge matching flows updating their byte/packet counts and start/end timestamps. We call the merged flows *defragmented flows*. Note that if a flow spans multiple intervals it will yield one defragmented flow for each interval. In the bi-flow pairing process we account for these cases. In the third and fourth row of Table 1 we show the numbers of raw and defragmented flows. We observe that defragmentation has a significant impact reducing the number of flows by a fraction ranging between 20.6% and 39.6% for different years.

3.1.3 Bi-flow Pairing

For TCP and UDP a two-way flow is the aggregate of two defragmented flows that have the same 5-tuple with reverse values in the source and destination IP address and port fields. For each defragmented flow in an observation interval, we search for flows in the same or in the adjacent intervals and group matched flows into a bi-flow. A one-way flow is a flow that does not have a matching reverse flow in the same or in an adjacent interval.

A special situation exists for other protocols, like ICMP, Encapsulated IPv6, and Encapsulated Secure Payloads (ESP). The port fields in our NetFlow data do not carry interesting semantics for protocols other than TCP and UDP. For these protocols we define a flow and match bi-flows based on a 3-tuple consisting of the IP addresses and protocol fields. This definition may underestimate the number of one-way flows involving protocols other than TCP and UDP.

Interestingly, it is not officially documented how NetFlow defines a flow for these protocols. According to certain sources NetFlow aggregates packets with common values in the source/destination IP address, protocol, and ICMP type/code fields into an ICMP flow [37]. For protocols other than TCP, UDP, and ICMP, NetFlow sets port fields to zero [37]. We find that this behavior is not met by our NetFlow data. Only 5% of all ICMP flows carry the ICMP type and code information, while all remaining ICMP flows have this information set to zero.

3.1.4 Impact of the Interval Size

We next analyze the impact of the interval size by computing flow metrics for a full day of data using different interval sizes. We select a 10-minute reference interval within which we can search efficiently for flow fragments and pairs. In Figure 1 we show the impact of the interval size with respect to the 10-minute (600 secs) interval. We observe, as expected, that longer intervals lead to slightly lower flow counts as fewer flows cross interval boundaries. This decrease is less for two-way flows due to their longer duration. We observe that doubling the interval size slightly decreases absolute count metrics by 3-5%. On the other hand, the decrease of the relative volume of one-way flows is only 1.2% and does not decrease further with an increasing interval size. Therefore, we conclude that with a 10-minute interval absolute count metrics slightly overestimate the number of flows, while relative count metrics, which is the type of metrics we primarily report, are very accurate.

4. ONE-WAY TRAFFIC CLASSIFICATION

Our classification scheme has the following key features:

- **Flow-based Classification:** We leverage solely flow monitoring data, i.e., NetFlow. Compared to packet-

	2004	2005	2006	2007	2008	2009	2010	2011
Hours	799.0	800.0	792.0	780.0	799.0	799.0	800.0	800.0
Total terabytes	270.8	304.5	424.4	641.5	842.6	1035.2	1763.2	2132.7
Raw flows (1e9)	30.96	29.35	30.32	49.03	63.16	77.05	90.91	86.49
Defragmented flows (1e9)	24.31	22.47	22.53	29.62	44.03	58.55	72.21	66.91
Two-way flows (1e9)	6.117	6.712	8.046	10.84	17.61	22.97	28.71	26.55
Two-way packets (1e9)	381.1	418.6	558.0	834.0	1080.8	1286.8	2092.3	2372.7
One-way flows (1e9)	12.19	9.102	6.502	8.040	8.967	12.78	15.01	14.06
One-way packets (1e9)	34.09	20.89	24.04	31.09	27.59	49.65	39.56	79.03

Table 1: Size of data sets per year in total hours, bytes, flows and packets. The different flow counts represent the processing steps from raw data to defragmented flows and finally to flows separated into two- and one-way flows.

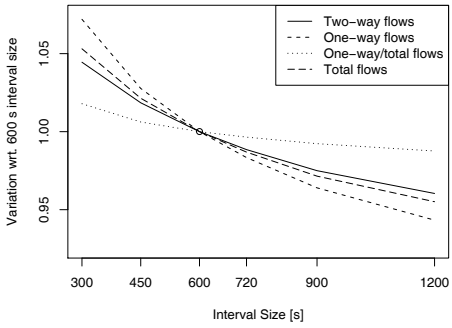

Figure 1: Impact of time interval size on flow metrics aggregated for a full day. The decrease of the fraction of one-way to total flows is not more than 1.2% for a doubled interval size.

level data and DPI that have been used in the past to monitor IBR traffic, flow monitoring provides fewer details, but is less expensive, scales better to monitoring large IP address blocks, and is more generally available.

- **Easy Configuration:** Our classifier does not require training, which is a drawback of many traffic classification techniques. It can be applied with minimal configuration, without extensive overhead.
- **Comprehensibility:** We use legible rules instead of complex classification structures, like self-organizing maps, to make our classifier comprehensible to non-machine-learning experts. Our scheme is based on the rules summarized in Table 3 that can be easily reviewed and verified both by researchers and network operators. Traffic classification schemes should be easily understandable to be engaged and deployed by network operators.
- **Extensibility:** Our classifier can be easily extended with additional traffic features and classification rules. For example, if a flow monitor is configured to collect additional data, like TCP flags, then these can be easily integrated into new or existing classification rules.

We classify one-way flows using a two step process that resembles the way medical diagnosis identifies the cause of a health problem. We first define a set of *signs*, which are informative attributes for digging into the cause of one-way flows. We then check each flow against all signs and find the ones it matches. Finally, we classify a flow based on the collection of signs that are present or absent. Our classification scheme is based on a set of rules that take as input the signs of a flow and determine the appropriate class, including the unknown class.

4.1 Signs

Flow signs are derived from *sign tests*. A sign test determines if a flow exhibits a specific sign and may be as simple as checking a flow feature value or as complex as performing a behavioral assessment of an involved end-host. We introduce in total 17 signs exploiting in 4 cases techniques from the literature that have been shown effective for detecting scan and peer-to-peer traffic. We classify our signs in four categories based on the entity they characterize: 1) a pair of communicating hosts, 2) a remote host as a source of a one-way flow, 3) a local host as a target of a one-way flow, or 4) a flow. In Table 2 we summarize our signs and show in which category they belong.

4.1.1 Host-Pair Signs

Host-pair signs characterize a pair of end-hosts with a one-way flow. They reflect the mix of one- and two-way flows of a host pair. In Figure 2 the graphs a to d illustrate communication patterns that correspond to the four possible combinations of a biflow and an outflow between a host pair with an inflow. Hosts are represented by nodes, where the local host is the node in the left and directed links represent the presence of one or more flows. We use the sign *end-hosts-communicating* to mark one-way flows between hosts that are involved in a productive communication, i.e., they have one or more biflows between them, with occasional failed connections, i.e., one-way flows. We assign this sign to one-way flows with the communication pattern shown in the graphs b and d in Figure 2.

Furthermore, we introduce a sign called "InOut" to capture the pattern of graph c in Figure 2. In particular, the "InOut" sign identifies host pair communication situations in which a one-way ICMP flow is a reply to another outgoing one-way flow exchanged between the same host pair. We assign ICMP one-way flows with the sign "InOut" to a special class that includes suspected benign one-way flows.

4.1.2 Remote Host Signs

We use four signs to characterize remote hosts that generate one-way flows towards the monitored network. If we observe one-way flows from a well-known source port of a remote host to a local ephemeral port and at the same time we do not observe any two-way flows nor opposite direction

Sign Type	Sign Name	Detection Criterion/Algorithm	Sign Short Name for Rules
Host pair behavior	End-hosts-communicating	One-way flow between productive host pair	PotOK
	Limited dialog	One-way flows between unproductive host pair	InOut
Remote host behavior	Service sole reply	no biflow on srcIP $\wedge$ dstPort$\geq$1024 $\wedge$ srcPort $<$ 1024	Backsc
	Remote scanner 1	TRW algorithm (suspected scanner)	TRWscan
	Remote scanner 2	Host classification (suspected scanner)	HCscan
	Remote non-scanner	TRW algorithm (suspected regular host)	TRWnom
Local host behavior	Unused local address	Unpopulated local IP address	GreyIP
	Service unreachable	Unanswered request to local service	Unreach
	Peer-to-peer	Flow towards local P2P host	P2P
Flow feature	Artifact	UDP/TCP flow with both port numbers=0	Artef
	Single packet	Flow contains one packet only	Onepkt
	Large flow	Flow carries $\geq$ 10 packets or $\geq$ 10240 bytes	Large
	Bogon	Source IP belongs to bogon space	Bogon
	Protocol	IP protocol type of flow	TCP, UDP, ICMP, OTHER

Table 2: Overview of defined signs based on the behavior of one or both involved hosts or based on the numeric values of one or more flow attributes.

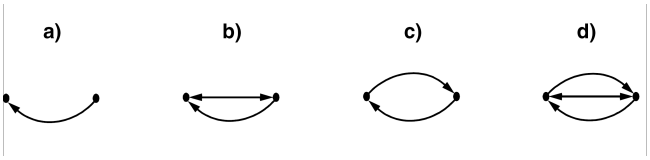

Figure 2: Mixture of incoming one- and two-way flows exchanged between a host pair shown as a graph. Hosts are represented by nodes and the presence of inflow/outflow/biflows by arrows.

one-way flows between the two hosts, then we assign the *service sole reply* sign. We use this sign to detect backscatter from attack traffic towards a service of the remote host. We check for the absence of biflows within a 30-minute time interval to limit computational overhead.

Besides, we leverage two state-of-the-art algorithms to detect remote scanners. We introduce three signs that describe either a positive or a negative test result. The first two signs are based on the well-known Threshold Random Walk (TRW) [19] algorithm, which uses sequential hypothesis testing to qualify a remote host as scanner or benign (or undecidable). We denote these signs with "TRWscan" and "TRWnom". The second algorithm [3] is a variant of TRW that inspects the behavior of a remote host in a relaxed way and classifies it as a scanner if a test condition is met. A negative test result is not supported, i.e., it does not qualify a host as benign. The test assesses the *service fanout* of a remote host, i.e., the count of 2-tuples {dstIP,dstPort} the host attempted to access, and its overall success in connection establishment. Each connection attempt is qualified either as "good" if it results in a two-way flow or as "bad" if it results in a one-way flow. This way, a remote host is classified as a scanner if it has a bad service fanout larger than two times the good service fanout (and at least 4). The host classification is done in a first pass. During a second pass any flow sourced by a classified scanner is labeled with the sign "HCscan". Both algorithms restrict their tests to 30 minute time windows to keep memory demands in bounds.

4.1.3 Local Host Signs

We use three signs to characterize local hosts within the monitored network. We mark with the sign *unused local address* ("GreyIP") local IP addresses that never sourced any flow over a full observation period of 400 hours and, therefore, are unpopulated.

We introduce the sign *service unreachable* to trace outages of well-known services. We define a service as an endpoint described by the 3-tuple {localIP, protocol, localPort} with the constraint that the port number is well-known, i.e., below 1024. We consider only TCP and UDP flows. We consider a local service valid if it serves at least 20 clients during any 30-minute interval over a full observation period. We label one-way flows that target valid services with the sign *service unreachable* ("Unreach").

The sign *peer-to-peer* marks one-way flows caused by stale host caches of P2P applications. Behavioral P2P application identification has been studied extensively [20, 31, 18]. We make use of three rules from the literature (H1,2,3 from [18]) that identify a local host as a member of a P2P network based on the observation of typical P2P behavioral patterns. We require that at least 2 out of 3 P2P rules are matched or a match occurs for at least 5 time windows of 30 minute duration. First, we prepare a list with detected local P2P hosts by analyzing two-way flows of a full observation period. Then, we mark any TCP/UDP one-way flow targeted at a high port towards such hosts with the *peer-to-peer* sign.

4.1.4 Flow Signs

The remaining signs are either direct flow attributes or are derived from them by the rules shown in Table 2. The sign *artifact* is used for TCP and UDP flows with both port numbers set to zero. Such flows are the result of packet fragments for which the correct port numbers are not available due to the missing layer-4 header. If a flow contains a source address that belongs to bogon space using the data of [9] then we mark it with the sign *bogon*.

4.2 Classifier

A class is associated with one or more rules. We define the following classes:

- **Service Unreachable:** access attempt to temporary unavailable services.
- **Malicious Scanning:** probing for the exploitation of vulnerabilities in end systems.
- **Benign P2P:** P2P applications trying to access peers listed in their local host cache that are not anymore available.

Class Name	Rule #	Flow Membership Rules
Malicious Scanning	1	$\{TRWscan, \overline{PotOk}\} \Rightarrow Scanner$
	2	$\{HCscan, \overline{PotOk}\} \Rightarrow Scanner$
	3	$\{GreyIP, \overline{Backsc}, \overline{ICMP}, \overline{bogon}\} \Rightarrow Scanner$
	4	$\{Onepkt, \overline{GreyIP}, \overline{ICMP}, \overline{TRWnom}, \overline{bogon}, \overline{P2P}, \overline{Unreach}, \overline{PotOk}, \overline{Backsc}, \overline{Large}\} \Rightarrow Scanner$
	5	$\{ICMP, \overline{TRWnom}, \overline{InOut}, \overline{bogon}, \overline{PotOk}\} \Rightarrow Scanner$
Backscatter	6	$\{Backsc, \overline{TRWscan}, \overline{HCscan}, \overline{P2P}, \overline{InOut}, \overline{PotOk}\} \Rightarrow Backscatter$
Service Unreachable	7	$\{Unreach, \overline{TRWscan}, \overline{HCscan}, \overline{bogon}, \overline{P2P}\} \Rightarrow Unreachable$
Benign P2P Scanning	8	$\{P2P, \overline{TRWscan}, \overline{HCscan}, \overline{bogon}\} \Rightarrow P2P$
Suspected Benign	9	$\{PotOk, \overline{Unreach}, \overline{P2P}, \overline{bogon}\} \Rightarrow Benign$
	10	$\{Large, \overline{GreyIP}, \overline{TRWscan}, \overline{HCscan}, \overline{P2P}, \overline{Unreach}, \overline{ICMP}, \overline{Backsc}, \overline{bogon}\} \Rightarrow Benign$
	11	$\{TRWnom, \overline{GreyIP}, \overline{HCscan}, \overline{P2P}, \overline{Unreach}, \overline{bogon}, \overline{Backsc}\} \Rightarrow Benign$
	12	$\{ICMP, InOut, \overline{TRWscan}, \overline{HCscan}, \overline{bogon}\} \Rightarrow Benign$
Bogon	13	$\{bogon, \overline{TRWscan}, \overline{HCscan}, \overline{Backsc}\} \Rightarrow Bogon$

Table 3: Rules used to classify one-way flows. Each rule specifies which signs have to be present or absent (sign names with overbars). An overview of defined signs can be found in Table 2.

- **Backscatter:** replies to DoS attack traffic that uses randomly chosen source IP addresses to hide the real identity of an attacker.
- **Suspected Benign:** one-way flows may exist as part of benign applications using data and control connections in parallel and employing one of them for acknowledgment only. Another cause may be temporary failures within an otherwise productive communication.
- **Bogon:** one-way flows originating from bogon IP space.
- **Other:** one-way flows that do not match any of the above classes.

We derive our classification rules with a systematic process shown in Figure 3. We start with an initial set of rules and in each iteration we classify flows, compute a conflict report of the resulting classification and update the rules. The conflict report describes flows classified in multiple classes. Specifically, it reports: 1) the classes that intersect; 2) the size of the intersections; 3) an ordered (by their popularity) list of sign combinations in the intersections; and 4) the number of remaining unclassified flows. Based on the conflict report and in particular the sign combinations, we then manually update the rules to resolve conflicts and to reduce the number of unclassified flows. Each flow is checked against 17 different signs, which in theory yields a maximum of 131,072 possible sign combinations. However, most combinations do not occur in practice. In our data, we observe a total of 1,035 different sign combinations. Moreover, one could ignore sign combinations that are only observed in a small number of flows resulting in a sharp reduction on the number of interesting combinations. We iteratively repeat this procedure until we resolve all conflicts and cannot further significantly reduce the number of unclassified flows. Based on the described procedure we derived a first version of our classification rules after eight iterations. Then, we further refine our rules based on our validation. The final classifier includes 13 classification rules shown in Table 3.

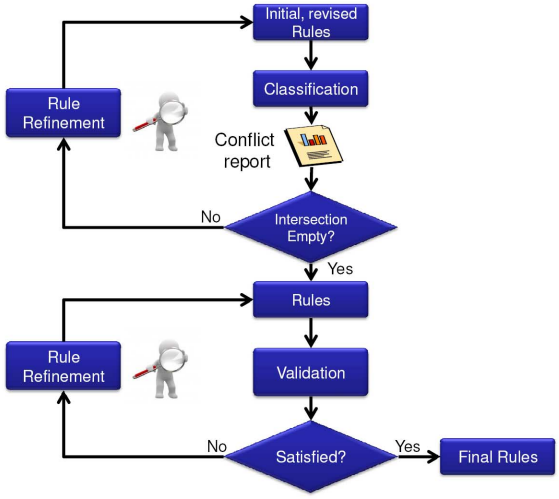

Figure 3: Rule refinement stages

For the initial rule set we defined rules that contained only signs that must be present based on expert knowledge. This resulted in many flows that were classified in multiple classes, summing up on average to 21.9% of the flows with a peak of 40.6%. Through carefully studying conflicting sign sets provided by the conflict report we revised the rules by adding extra predicates describing signs that must be absent to resolve all class conflicts. Finally, we investigated all sign sets of significant size that were not yet catched by any rule to minimize the count of unclassified flows (assigned to class "Other"). This way we could classify on average 98.0% of all one-way flows with a low of 95.6% in August 2008.

Our classifier processes input flows in three passes. In a first pass it initializes local services, local P2P hosts and in-

active local IP addresses. In a second pass matching signs are assigned to flows and in a final pass rules are applied to account each flow to a unique class. We have implemented our classifier in C++ and run it on a dedicated Linux cluster with a fast RAID disk subsystem. We collect logs of the overhead of the three passes combined for all 16 runs. In general, the overhead scales with the size of the dataset, but also depends on dataset characteristics. On average a run processed 28.6 gigaflows in 23.1 hours (with a maximum of 37.4 hours) and required 3.4 GB of memory (with a peak of 6.1 GB). Considering the monitoring period length of 400 hours these figures are well within real-time bounds set by the speed of the flow data collection process. Note that we exclude the overhead for preprocessing (time binning, flow defragmentation, biflow-pairing) as it largely depends on the available input data (e.g. NetFlow export or YAF-created flow records). On our NetFlow records the preprocessing required the most resources while still safely remaining within real-time bounds.

Our classification scheme has one parameter that needs configuration: the time window size used for P2P and services identification and for scanner detection. We set it by default to 30 minutes, but on a smaller network the window size should be increased (see also Section 5).

5. VALIDATION

In this section we evaluate the accuracy of our classifier and optimize its rule set. Validating a one-way traffic classification scheme is very challenging. A major challenge is the scarcity of information available from one-way traffic consisting of very short flows carrying few packets and frequently no payload. We address this challenge, first, by building more accurate host profiles based on flow data over long time windows, second, by carefully examining one-way packets for violation of relevant protocol state machines, and third, by exploiting DPI and application identification techniques.

We first use a traffic summarization technique [5] based on frequent item-set mining (FIM) to summarize flows of different classes into frequent item-sets, which can be then easily inspected by an analyst. We apply this approach in each class and measurement period between 2004 and 2011. In summary, we find that for different classes and periods up to 75% of the flows were correctly classified. However, with this approach we cannot assess the remaining flows and the number of false negatives. In addition, a validation of one-way traffic classification solely based on flow data often provides insufficient evidence, as we miss information like TCP flag and ICMP types/codes. For this reason, we built a dedicated validation setup in a smaller network to obtain more detailed data.

5.1 Validation Setup

We built a monitoring setup and collected packet traces at the Internet gateway of the Swiss university Hochschule für Technik Rapperswil (HSR) occupying a /16 IP address range. The campus network is sparsely populated with a total of 3,949 active IP addresses seen during our experiment. We use a dedicated host that is tapping traffic between the border router and the firewall. On the border router no filtering is activated, enabling us to see all traffic routed to this IP range. To reliably collect packet data, we use an Endace DAG card that provides exact packet timestamps. Mandated by the IT security policy, we anonymize all IP addresses using a prefix-preserving scheme. Note that anonymization does not allow us to use active probing for our validation as proved to be useful in [30].

We collected packet traces for 19 consecutive days and extracted flows using the YAF flow meter [16] configured for bidirectional flow export. We then split flows into 10-minute intervals and defragmented flows within each interval resulting in a total of 322.7 million (mio) flows, which include 219.6 mio (68.1%) incoming one-way flows, 8.06 mio outgoing one-way flows (2.5%) and 95.1 mio two-way flows (29.5%). The small fraction of two-way flows can be explained by the scarce population of the monitored network. We retain packet data for a sub-period of three randomly chosen days.

To obtain application labels we configured YAF with its optional DPI application identification feature that assigns application labels to flows it recognizes. However, the coverage achieved by YAF is limited. Thus, we used an additional DPI application identifier to add a second set of application labels for almost all flows. This additional DPI application identifier originally was developed to evaluate the BLINC classifier [21] and in an improved version to evaluate several competing approaches to application identification [24].

To learn more on the occurrence of one-way flows using bogus source IP addresses, we periodically downloaded the full bogon list provided by [9] throughout the data collection process. This list not only contains the official IANA-reserved address ranges, but includes all IP ranges assigned to regional registrars that have not yet been handed out to costumers.

5.2 Validation Criterias

To determine class memberships we built a DPI classifier that uses an extensive set of 33 rules that make heavy use of details available only from packet-level data. Due to space limitations, we describe our DPI-based classification in the companion technical report [12]. In the following paragraphs, we summarize the key additional information we extracted from the packet-level data and how we exploited it.

Extended Host Profiles: For each host we maintain a host profile over the full observation period of 457 hours (19 days) that tracks its connection success and subsequently is used to feed the TRW algorithm. This is feasible in our /16 network, i.e., it required 12 GBytes of memory for 10.2 mio extended host profiles, but does not reasonably scale for larger networks carrying more traffic. Besides, extended host profiles in combination with application identification are useful for identifying P2P hosts and local services.

ICMP types and codes: For ICMP flows we analyze if type and code information and the communication situation of the involved host pair fit a class. For example, we deem a flow as backscatter if it is an echo reply and the receiver never sent a request to the sender. In addition, we check if the source of an ICMP flow is not already identified as a scanner, which interestingly is true for more than 92.1% of all incoming one-way ICMP flows.

Protocol State Machine: As an additional way to identify malicious scan traffic, we analyze how well individual flows follow the transport layer protocol state machine. Malformed packets are frequently used to exploit weaknesses of protocol stacks or to penetrate non-stateful firewalls. For

TCP flows we test if the flags of consecutive packets fit into acceptable state changes in the TCP protocol machine and run sanity checks on the sequence numbers of segments.

Application Identification: We apply application identification techniques [16, 21, 24] on two-way flows to discover local services and to detect hosts running P2P applications.

Precise Timestamps: We rely on packets timestamps to identify the initiator of a connection, which is not possible using NetFlow timestamps. Initiator detection is useful in combination with DPI and extended host profiles for mapping local services.

5.3 Validation Results

We run both the flow-based classifier and our DPI classifier over the evaluation dataset. Then, we compare the results of both classification runs and fill-in a *confusion matrix* showing predicted (flow-based) versus actual (DPI-based) class memberships. Based on the confusion matrix, we calculate the *recall* (also called sensitivity), *specificity* and *precision* metrics for all defined classes. Recall measures the portion of actual positives which are correctly identified as such; specificity measures the proportion of actual negatives which are correctly identified; and precision measures the portion of the classified positives which are correctly identified.

For the flow-based classification we use the same settings as we apply to the full dataset with the exception of the time window parameter for remote host profiling, which we increase by a factor of 32 to 16 hours instead of 30 mins. This compensates for the smaller probability to observe randomly targeted scan flows in the evaluation dataset caused by the smaller IP address range. Similarly to the full-dataset analysis, we classify incoming one-way flows. Our validation uses the full evaluation dataset to build host profiles and a subset of 34.8 mio incoming one-way flows for DPI. For this subset we could validate 99.4% of all one-way flows.

The test results are summarized in Table 4. We obtain very good results for the classes "Malicious Scanning", "Service Unreachable" and "Benign P2P Scanning" with recalls above 95.3%, specificitiy values greater than 91.3% and precision exceeding 95.4%. The class "Suspected Benign" has a recall of 85.1% and a precision of 75% caused by flows in the borderline between the classes "Suspected Benign" and "Malicious Scanning" that are hard to separate. On the other hand, its specificity is excellent at 99.9%. For the class "Backscatter" we observe a recall of 62.4%. We miss primarily ICMP replies and error messages as ICMP type and code information is not available in our flow data. However, the recall can be easily improved by using ICMP type/code fields when available to find ICMP replies. Besides, a specificity of 100% and a precision of 88.4% shows this class is still very useful. Finally, we observe a low recall for the class "Bogon" of 40.4% and at the same time a perfect specificity and precision of 100.0%. This is because the bogon list used by the flow-based classifier is a subset of the bogon list used by our DPI classifier. We intentionally exclude the full-bogon information from flow-based classification because historical full-bogon data is not available. Again, it is trivial to extend the flow-based classifier to use full-bogon lists for analyzing present or future data.

Class Name	Recall [%]	Specificity [%]	Precision [%]
Malicious Scanning	99.9 (99.6)	91.2 (88.8)	99.8 (99.6)
Backscatter	62.4 (62.4)	100.0 (100.0)	88.4 (87.2)
Service Unreachable	99.6 (99.5)	100.0 (100.0)	96.1 (92.5)
Benign P2P Scanning	95.3 (91.8)	99.9 (99.8)	95.5 (85.0)
Suspected Benign	85.1 (70.5)	99.9 (99.9)	75.0 (70.5)
Bogon	40.4 (40.2)	100.0 (100.0)	100.0 (100.0)

Table 4: Results of validation. The values in parentheses provide a lower bound by counting non-validated flows as classification mismatches.

5.4 Impact on Flow Classifier

For flow-based scan detection we use two sets of rules. The first set is based on scan detection algorithms providing strong evidence. The second set defines rules that point towards scanning, but provide weaker evidence. In particular, it includes flows 1) towards unpopulated IP addresses or 2) that consist only of a single packet and do not match any other rule. The validation shows that the second rule set is in fact useful for scan detection, which led us to add this rule set to the class "Malicious Scanning" eliminating an initially defined class "Suspicious Other". In particular, 65.1% of the flows matching these rules are actually detected as scanners when observing host behavior for an extended time period.

Another change introduced during validation concerns the assignment of ICMP flows. We observe that 92.1% of all ICMP flows seen are sourced by identified scanners. Without ICMP type/code information, it is difficult to satisfactorily distinguish backscatter ICMP flows from scanning. Consequently, we moved a rule initially assigned to the class "Backscatter", assuming ICMP replies from DoS attacks, to the class "Malicious Scanning". Furthermore, we added the sign "InOut" that helps to identify host pair communication situations in which an ICMP flow is a reply to a one-way flow exchanged between the same host pair. We use this sign in the class "Suspected Benign".

6. ONE-WAY TRAFFIC COMPOSITION

In this section we apply our classification scheme on a massive dataset of flows records collected between 2004 and 2011 to shed light into the composition and characteristics of one-way traffic.

Aggregate Statistics: We first find that in terms of flows one-way traffic is a very large component of Internet traffic. During the studied period one-way flows correspond to between 34% and 67% of the total number of flows towards the monitored network. This is important for systems that need to keep per flow state, like stateful firewalls and flow meters. On the other hand, due to their short-lived nature, one-way traffic corresponds on average only to 3.4% and 0.79% of the total number of packets and bytes, respectively. This indicates that IBR traffic is not a significant in terms of additional bandwidth consumption and packet processing overhead.

Table 5 provides for each measurement period and class the fraction of one-way traffic in terms of flows and packets. On average the class "Malicious Scanning" accounts for 83.5% of all one-way flows. It is followed by the classes "Benign P2P" (6.7%), "Service Unreachable" (4.8%), "Suspected Benign" (2.6%), "Other" (2.2%), "Backscatter" (0.3%) and "Bogon" (0.1%), where in parentheses we show the average fraction of one-way flows per class over the eight year period. The packet perspective provides a different view. "Mali-

Period	Malicious Scanning	Backscatter	Unreachable	Benign P2P	Suspected Benign	Bogon	Other
2004-02	87.2%/57.9%	0.1%/0.2%	4.8%/9.2%	5.5%/10.3%	1.1%/3.7%	0.1%/0.1%	1.1%/18.7%
2004-08	93.9%/82.0%	0.0%/0.1%	0.6%/1.3%	3.4%/9.8%	1.0%/4.1%	0.0%/0.0%	1.0%/2.7%
2005-02	87.2%/65.6%	0.0%/0.1%	5.8%/12.1%	4.1%/9.1%	1.8%/10.5%	0.0%/0.0%	1.0%/2.7%
2005-08	95.4%/74.7%	0.0%/0.1%	0.4%/0.7%	1.8%/4.3%	1.5%/5.3%	0.3%/0.1%	0.7%/14.9%
2006-02	89.3%/71.8%	0.1%/0.2%	0.6%/4.4%	4.2%/9.9%	3.3%/10.2%	0.2%/0.1%	2.3%/3.5%
2006-08	87.8%/47.3%	0.1%/0.0%	0.4%/0.3%	5.3%/4.1%	3.2%/46.9%	0.0%/0.0%	3.1%/1.4%
2007-02	86.9%/83.6%	0.1%/0.1%	1.7%/1.2%	6.5%/7.1%	2.8%/6.2%	0.0%/0.0%	2.0%/1.8%
2007-08	80.7%/63.5%	0.2%/0.2%	2.2%/2.9%	9.5%/17.6%	3.5%/10.3%	0.2%/0.1%	3.7%/5.5%
2008-02	68.3%/34.4%	0.2%/0.2%	1.1%/1.9%	23.9%/49.4%	3.0%/10.4%	0.0%/0.0%	3.5%/3.7%
2008-08	70.5%/47.8%	0.3%/0.4%	15.2%/26.3%	6.5%/11.4%	3.2%/8.5%	0.0%/0.0%	4.4%/5.5%
2009-02	74.2%/55.7%	0.1%/0.0%	15.3%/25.3%	5.9%/11.1%	2.7%/5.6%	0.0%/0.0%	1.8%/2.3%
2009-08	86.2%/75.2%	0.1%/0.1%	3.7%/6.0%	5.7%/10.2%	2.3%/5.2%	0.0%/0.0%	2.0%/3.3%
2010-02	88.2%/77.5%	0.3%/0.2%	2.6%/5.1%	4.0%/7.7%	2.9%/5.9%	0.1%/0.0%	2.0%/3.5%
2010-08	85.8%/74.3%	0.2%/0.2%	3.1%/4.7%	6.3%/12.5%	2.6%/5.5%	0.0%/0.0%	1.9%/2.7%
2011-02	81.4%/61.5%	0.6%/1.2%	6.2%/12.9%	7.3%/18.7%	2.6%/3.5%	0.0%/0.0%	1.9%/2.3%
2011-08	72.8%/28.3%	2.2%/4.3%	12.3%/46.7%	6.9%/15.1%	3.7%/4.3%	0.0%/0.0%	2.0%/1.4%

Table 5: Fraction of flows/packets falling into the defined one-way flow classes. The class *Other* represents the remainder of flows not captured by the other classes. Note, that we do not list rules that achieve a low coverage eliminating the flow artifacts class that results from fragmented packets without layer-4 header.

cious Scanning" accounts for 62.6% of all one-way packets, "Benign P2P" for 13.0%, "Service Unreachable" for 10.1%, "Suspected Benign" for 9.1%, "Other" for 4.7%, "Backscatter" for 0.5%, and "Bogon" for 0.03%.

We observe that one-way traffic is clearly dominated by scanning. In terms of packets, scanning accounts for a smaller fraction of 62.6% of the total number of one-way packets. This is because one-way flows classified as scanning consist on average of 1.6 packets per flow. In contrast, one-way flows of the classes "Service Unreachable" and "Benign P2P" consist of 4.1 and 12.1 packets per flow, respectively. This difference is because TCP one-way flows to unreachable services are much more persistent in SYN packet retransmission attempts than TCP scanning. In addition, we find that one-way flows in the class "Benign P2P" are often multi-packet UDP flows that sharply increase the average number of packets per flow.

Changes Over Time: Besides, we observe a number of notable changes over time. In Figure 4 we compare how the mean daily number of one- and two-way flows evolved between 2004 and 2011. The volume of one-way flows has exhibited small fluctuations between 2004 and 2011 and, interestingly, in 2011 it is almost equal to 2004. On the other hand, the number of two-way flows has grown significantly by 343%. A study by Akamai estimated the global Internet penetration increase in 2010 as 17% [2]. If we extrapolate this figure over eight years, it yields a growth by 351%, which is very close to our observations.

Besides, we observe that the fraction of one-way flows has dropped significantly over time as shown in Figure 5. In 2004 one-way flows accounted for 67% of the total number of flows. Their share gradually dropped and since Aug. 2007 one-way flows account consistently for one out of three incoming flows. Between 2004 and 2007, a very large fraction of one-way flows targeted a small number of destination port numbers, which were used by loud worms, like Sasser. For this reason, we connect the decline of share of one-way flows to that presently aggresive scanning is not as common malware propagation vector as in the period between 2004 and 2007.

Figure 6 shows how one-way traffic breaks down into different classes in terms of flows. The fraction of "Malicious Scanning" flows varies, started at a high of 87.2% in 2004

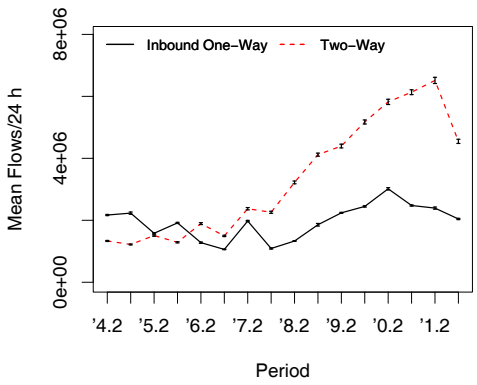

Figure 4: Evolution of one and two-way mean daily flow counts over time. The vertical bars mark 95% confidence intervals.

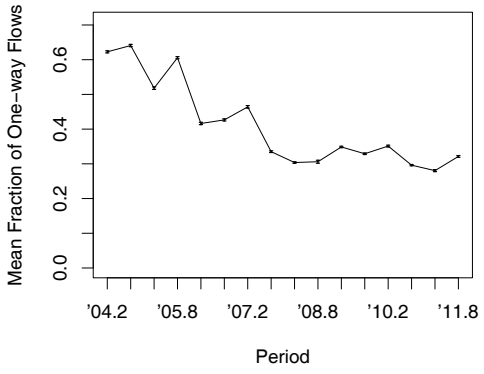

Figure 5: One-way flows as a (mean) fraction of the total number of flows, i.e., both one- and two-way flows, between 2004 and 2011. The share of one-way flows has declined between 2004 and 2007 and remains almost constant since then.

and stayed at this or a higher level until 2006. Then, in 2007 we note a decline that leads to a low of 68.3% in Aug. 2008.

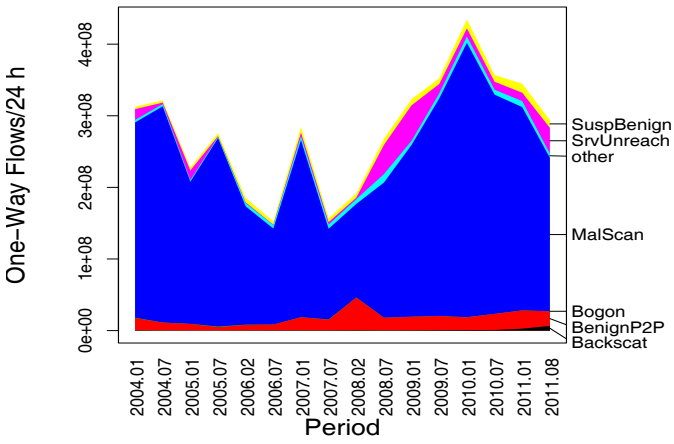

Figure 6: Composition of one-way traffic in flow counts per class.

This decline correlates with an increase of the class P2P one-way traffic that falls early into the same period and also an increase of class "Unreachable" later on. The high counts in class "Service Unreachable" between Aug. 2008 and Feb. 2009 are caused by unanswered NTP requests towards an NTP server. Investigating further this incident, we found that the increased number of requests pushed the operator to introduce access restrictions to this server. As we analyze more in the next section, this class of one-way flows is very useful to administrators to monitor the reachability of services. The peak in benign P2P one-way traffic between Aug. 2007 and Feb. 2008 can be attributed to two IP addresses that are reported to be fake servers by the eMule Bulletin Board. Fake eMule servers are commonly used to collect eMule user data for research or legal prosecution. Minor contributors to this class are XBSlink traffic sourced by Xbox 360 and PS2/3 game equipment, and the MMORPG (massively multiplayer online role-playing game) Heroes of Might and Magic.

7. SERVICE AVAILABILITY MONITORING

In this section, we show how one-way flows of the class "Service Unreachable" are very helpful for monitoring the availability of local networked services. Most existing schemes for monitoring the availability of services within an enterprise, a data center, or a university network are based on active probing and/or server logs. In contrast, flow data provide a new passive approach to monitor local services. The key intuition is that outages result in both 1) a drop (or halt) in the number of two-way flows and 2) a correlated rise in the number of one-way flows towards a destination. These changes can be easily detected by monitoring the number of one- and two-way flows towards local services. Compared to previous approaches, flow-based outage detection enables to leverage the regular traffic of a network as probe traffic. This approach offers the following key advantages: 1) *It exploits passive measurements* in contrast to injecting network overhead for active probing; 2) *It provides a tangible assessment of the impact of outages* by enabling to count the actual number of remote client IP addresses that fail to reach a service; 3) *It automatically discovers running local services* circumventing the need to manually configure (new) services; and 4) *It enables to concurrently monitor many services* by taking advantage of data from the strategical network location of gateways.

In the next paragraphs, we first outline our analysis methodology and secondly we show how our approach enables to assess the impact of outages using as a case study illustrating interesting misconfigurations and outages we detected in the campus network of ETH Zurich by applying our scheme on one week of data.

7.1 Methodology

Using one-way flows classified with our scheme in the class "Service Unreachable" as well as bidirectional flows as input, we compute the following three metrics to evaluate the reachability of local services within the network of ETH Zurich. We focus on services running on well-known port numbers.

We define the *availability* of a service as the number of time intervals a service is alive, i.e., it has bidirectional flows, over the number of intervals it is either alive or unresponsive, i.e., it only has incoming one-way flows. We ignore intervals in which a service does not have any incoming flows. The availability metric helps group services into permanently and temporarily offered ones.

Available services are not necessarily reachable to *all* clients due to access policies, misconfigurations, or other reachability problems. We define the *reachability* of a service as the fraction of the number of distinct client IP addresses with bidirectional flows to the service over the total number of distinct client IP addresses to the service during a time interval. The reachability metric is useful for identifying misconfigurations and outages.

Finally, the *outage impact* of a service is the number of unique client IP addresses that are involved in one-way flows of the class "Service Unreachable" during a time interval. As several clients can be behind a remote NAT, this metric provides a lower bound on the actual number of clients that are affected by an outage and is very useful for assessing the impact of failures.

7.2 Outages and Misconfigurations

In this section we demonstrate the utility of one-way flows detected with our classification scheme for monitoring the availability of networked services.

We applied our scheme on a randomly-selected week of NetFlow data from our archive and initially focused on services discovered within the address range of the Department of Information Technology and Electrical Engineering of ETH Zurich. We discovered services as 2-tuples $\{localIP, localPort\}$ with a well-known port number that had more than 20 unique clients during any time interval. As with bi-flow pairing, we used a 10-minute time interval. We discovered in total 43 remotely accessible departmental services for which we computed our three metrics. To validate our observations, we met and discussed with the network administrators of our department providing them details about the availability, reachability, and outages of the services. The administrators confirmed the validity of the discovered services and provided feedback on a number of interesting outages we summarize in the next paragraphs.

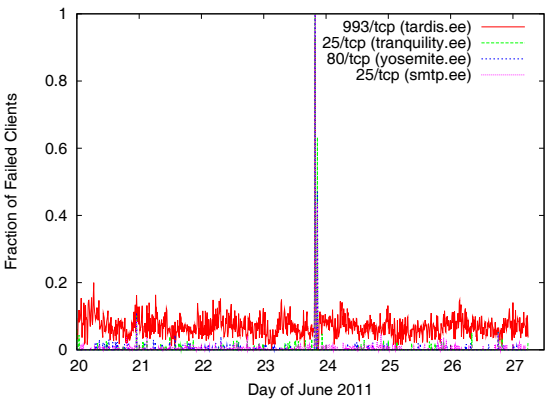

Figure 7: Coinciding outage observed on the 23rd of June 2011 19:40 UTC on most university services resulting in failed accesses from 287,583 unique clients.

We identified a group of 32 services available at least 99% of the time and 11 services that either fail very often or are not operated all the time.

For all frequently used services we found a coinciding global outage on the 23rd of June 2011 starting at approximately 19:40 UTC and lasting approximately 15 minutes as shown in Figure 7. This was a general problem in the reachability of the department network from the outside. To learn more about this outage, we analyzed three key services of the entire university: the main web, webmail, and software distribution services of ETH Zurich. Surprisingly, we found that at the same point in time the services were not accessible indicating a campus-wide reachability problem. During the identified 15-minute interval the outage impact metric revealed that 287,583 unique clients failed to access target services! Our investigation showed that the outage occurred during a planned router software upgrade. Although, temporal disruptions due to the upgrade were expected, the scale of the disruption revealed was suprisingly larger than expected. Present server availability monitoring techniques based on active probing and server logs miss this important information about the impact of an outage. Understanding the impact of outages in terms of the number of affected clients is very useful since this information helps to better provision network repairs and upgrades so that the number of affected clients is minimized. In addition, the number of failed client connections relates to the quality of an offered service and to lost revenues. In the examined case, although the detected outage resulted from a planned router software upgrade scheduled to take place during late evening hours, the number of affected clients, which our analysis revealed, turned out to be much higher than expected, which is important for better planning of such events. In addition, we visually inspected the time series of the volume of one-way flows for the top departmental services and found that the outage resulted in a sharp increase in requests targeting mail services due to the automated retry behavior of mail clients. On the other hand, for web services the volume of one-way flows during the outage did not substantially differ from the volume of two-way flows during regular operation, indicating a more network-friendly retry behavior.

Our inspection of the reachability metric for the discovered services revealed two actionable cases. First, we found an NTP server that was consistently not reachable on average by 12.9% of its clients. The outage impact metric showed that in this case too, a suprisingly large number of clients failed to receive a reply: during the studied week in total 2.2 million unique clients were involved in one-way flows to the NTP server. Investigating this further unveiled that standard NTP "hello world" configuration examples use the `swisstime.ee.ethz.ch` server in our department, making it likely one of the most popular NTP servers in the Internet. In addition, `swisstime.ee.ethz.ch` is often preconfigured in embedded devices. The failure rate, i.e., fraction of one-way flows, varies for different countries of origin. In the top client countries we observe Belgium with failure rate 65.7%, India with 20.2%, and Germany with 16.8%. Our investigation found two main causes of low reachability: 1) an IP filter was configured to prevent access from a /8 address block, which was frequently the source of attacks; and 2) certain devices appear to compute invalid CRC checksums resulting in dropped requests. From these findings we learned both that the traffic flow volume to the studied network was heavily affected from the popular NTP server and that a misconfigured application computes invalid CRC checksums.

Second, we noticed that the SMB service (Microsoft remote file server access) offered on 445/tcp was targeted by far more one-way flows (112% more) than two-way flows. An in-depth analysis revealed that almost all one-way flows were sourced by a likely misconfigured client that persistently tried to connect to this service. All remaining services achieved a reachability of 97.1% or more with most of them ranking at the 99.9% figure.

8. RELATED WORK

The related work falls into the following three categories.

8.1 IBR Traffic in Network Telescopes

Studying IBR traffic has been very useful for understanding Internet threats. In 2001, Moore *et al.* [29] were the first to use network telescopes to monitor background radiation and answer questions such as "how prevalent are denial-of-service attacks in the Internet today?". Yegneswaran *et al.* [39] employed scan logs collected by Dshield [11] during four months in 2001 and 2002 to analyze quantitative characteristics of intrusion activity in the global Internet. They made a number of interesting observations including that a very small collection of correlated sources are responsible for a significant fraction of intrusion attempts. In 2004, Pang *et al.* [30] used traces, lasting 80-hours to one week, from three sites to conduct a detailed characterization of traffic towards four large unused address blocks. Among other observations, they highlighted the dynamism of background radiation, which unlike normal traffic, can exhibit significant changes on a nearly daily basis. In 2010, Wustrow *et al.* [38] examined background radiation between 2006 and 2010 to four /8 unused blocks. They observed among other findings that background radiation continues to be very dynamic and identified interesting artifacts that lead to address space pollution. In addition to characterizing traces from network telescopes, in this work we consider the need to dissect one-way traffic to live networks, introduce and validate a one-way traffic classifier to facilitate this goal and characterize a massive data set of IBR traffic towards a live network.

8.2 IBR Traffic in Live Networks

A number of more recent studies have focused on one-way traffic. In 2012, Brownlee [6] described a tool called *iatmon* for classifying one-way traffic based on packet interarrival times and remote scanner patterns of the schema of Treurinet [35]. In addition, the author described observations about the different types of scanning in six months of data from the UCSD Network Telescope [7]. Our work is complementary to the *iatmon* tool, as we focus on flow instead of packet-based classification. Besides, the work by Lee and Brownlee [26] used three one-day packet traces captured in 2000, 2003, and 2006 in the University of Auckland to analyze the lifetime and size distribution of one and two-way flows. In our work, we provide new insights into one-way flows based on a substantially larger trace and introduce novel methods to classify their causes. Allman *et al.* [3] used TCP connection logs from the Lawrence Berkeley National Laboratory collected between 1994 and 2006 to characterize how scanning changed over time. In addition, they correlated observed patterns to well-known malware activity. Our study follows on until 2011 and focuses primarily on how to classify one-way traffic. Jin *et al.* [17] used flow data towards grey IP space of a campus network and developed a heuristic to identify external scanners. We follow along this direction and highlight the need to classify one-way traffic and identify a number of different causes in addition to scanning. Guha *et al.* [13] characterized the fraction of useful flows generated by mobile hosts in an enterprise network and found that 34% of the flows fail mainly due to hosts mobility. Compared to this work, we focus on understanding the nature of incoming (instead of out-going) one-way traffic.

8.3 Network Outages

Many previous studies have developed techniques for detecting network outages caused by events like prefix hijacking using control plane data (e.g., BGP updates) [36, 8, 25], active probing with tools like traceroute and ping [40, 41], or both [14, 22, 23, 33].

Recently, two studies have provided first insights on using passive network traffic measurements, like NetFlow or packet traces, to detect and characterize network outages. In particular, Schatzmann *et al.* [32] described a flow-based approach to identify reachability problems in remote networks caused by events like prefix hijacking. Dainotti *et al.* [10] used IBR traffic to a /8 network telescope to characterize large-scale remote outages caused by earthquakes and country-wide censorship. Compared to these studies, we use IBR traffic towards a live network to detect and characterize fine-grained outages affecting local networked services.

9. CONCLUSIONS

One-way traffic towards live networks is an exotic piece of Internet traffic. In the past it has been primarily studied with the help of a small number of large network telescopes. Although network telescopes have been very useful for understanding Internet attacks and misconfigurations, only a very small number of telescopes have been available to researchers, while the exhaustion of the IPv4 address space is likely to further drain their availability.

To shed light into IBR traffic towards live networks, in this work we introduce a classification scheme for one-way traffic. Classifying one-way traffic based solely on flow-level data is very challenging as one-way flows contain very few packets often without any payload. To address these challenges, we design, carefully optimize, and validate 13 classification rules that leverage communication patterns of involved end hosts to classify one-way traffic. Our techniques are useful for monitoring IBR traffic in any edge network and therefore significantly broaden the availability of network telescopes. They are also useful to network operators for passively monitoring the reachability of local networked services and in particular for analyzing the impact of outages, which is not possible with existing service monitoring techniques based on active probing or server logs.

We use our classifier to analyze the composition of a massive dataset of traffic flow records from a regional backbone network and make the following key observations:

- One-way traffic makes between 34% and 67% of the total number of flows, although it only accounts for only 3.4% and 0.79% of the total number of packets and bytes, respectively.
- The main sources of one-way traffic are: 1) scanning, which accounts for 83.5% of the one-way flows and 62.6% of the one-way packets; 2) peer-to-peer applications, which account for 6.7% of the one-way flows and 13.0% of the one-way packets; and 3) unreachable services, which account for 4.8% of the one-way flows and 10.1% of the one-way packets.
- The fraction of one-way flows to the total number of flows has declined from 67% in 2004 to 34% in 2011.
- The number of two-way flows has grown significantly by 343% between 2004 and 2011. In contrast, the absolute volume of one-way flows in 2011 is almost equal to 2004.

Finally, we focus on the particularly interesting class of one-way traffic caused by unreachable services and show, using the network of ETH Zurich as a case study, how this class is very useful for detecting important network outages and for assessing their impact.

With this paper we make the code of our classification scheme publicly available [1]. In addition, to shed further light into the composition of one-way traffic and to facilitate correlating data from multiple IBR traffic monitors, we release non-sensitive details about the evolution of one-way and two-way traffic between 2004 and 2011 based on the studied datasets. In particular, for each 10-minute time interval, we make available statistics about the synthesis of one-way traffic, the popularity of different signs, the targeted services, and the geographical distribution of sources.

Acknowledgements

We are grateful to SWITCH for providing us precious data for studying one-way traffic. In addition, we are thankful to Dominik Schatzmann for helping with double-counting elimination and to Simon Leinen for providing us valuable feedback on service reachability monitoring. Lastly, we wish to acknowledge anonymous reviewers for their helpful comments and suggestions.

10. REFERENCES

[1] One-way Traffic Classification Website. http://www.ow-class.ethz.ch/.

[2] Akamai Technologies. The state of the internet report (3rd quarter, 2009). Technical report, 2009.

[3] M. Allman, V. Paxson, and J. Terrell. A brief history of scanning. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, page 82. ACM, 2007.

[4] M. Bailey, E. Cooke, F. Jahanian, D. Watson, and J. Nazario. The blaster worm: Then and now. *IEEE Security and Privacy*, 3:26–31, July 2005.

[5] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 28–34, New York, NY, USA, 2009. ACM.

[6] N. Brownlee. One-way traffic monitoring with iatmon. In *Passive and Active Measurement Conference*, 2012.

[7] CAIDA. UCSD Network Telescope. http://www.caida.org/data/realtime/telescope/.

[8] Y.-J. Chi, R. Oliveira, and L. Zhang. Cyclops: the as-level connectivity observatory. *SIGCOMM Comput. Commun. Rev.*, 38(5), Sept. 2008.

[9] T. Cymru. The Bogon Reference. http://www.team-cymru.org/Services/Bogons/, 2012.

[10] A. Dainotti, R. Amman, E. Aben, and K. C. Claffy. Extracting benefit from harm: using malware pollution to analyze the impact of political and geophysical events on the internet. *SIGCOMM Comput. Commun. Rev.*, 42(1):31–39, 2012.

[11] Cooperative Network Security Community - Internet Security. www.dshield.org.

[12] E. Glatz and X. Dimitropoulos. Classifying internet one-way traffic. TIK-Report 336, ETH Zurich, May 2012.

[13] S. Guha, J. Chandrashekar, N. Taft, and K. Papagiannaki. How healthy are today's enterprise networks? In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 145–150. ACM, 2008.

[14] X. Hu and Z. M. Mao. Accurate real-time identification of ip prefix hijacking. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, 2007.

[15] IANA - Internet Assigned Numbers Authority. PORT NUMBERS. http://www.iana.org/assignments/port-numbers, 2011.

[16] C. Inacio and B. Trammell. Yaf: yet another flowmeter. In *Proceedings of the 24th international conference on Large installation system administration*, pages 1–16. USENIX Association, 2010.

[17] Y. Jin, Z. Zhang, K. Xu, F. Cao, and S. Sahu. Identifying and tracking suspicious activities through IP gray space analysis. In *Proceedings of the 3rd annual ACM workshop on Mining network data*, page 12. ACM, 2007.

[18] W. John and S. Tafvelin. Heuristics to classify internet backbone traffic based on connection patterns. *International Conference on Information Networking (ICOIN)*, pages 1–5, 2008.

[19] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 211–225, 2004.

[20] T. Karagiannis, A. Broido, and M. Faloutsos. Transport layer identification of p2p traffic. *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134, 2004.

[21] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. Blinc: multilevel traffic classification in the dark. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, 2005.

[22] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying black holes in the internet with hubble. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, 2008.

[23] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. Lifeguard: practical repair of persistent route failures. In *Proceedings of the ACM SIGCOMM 2012*, 2012.

[24] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT conference*, page 11. ACM, 2008.

[25] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: A prefix hijack alert system. In *In Proc. USENIX Security Symposium*, 2006.

[26] D. Lee and N. Brownlee. Passive measurement of one-way and two-way flow lifetimes. *SIGCOMM Comput. Commun. Rev.*, 37(3):17–28, 2007.

[27] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security and Privacy*, 1:33–39, July 2003.

[28] D. Moore, C. Shannon, G. Voelker, and S. Savage. Network telescopes: Technical report. Technical report, CAIDA, 2004.

[29] D. Moore, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. In *SSYM'01: Proceedings of the 10th conference on USENIX Security Symposium*, Berkeley, USA, 2001.

[30] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet background radiation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM New York, NY, USA, 2004.

[31] M. Perényi, T. D. Dang, A. Gefferth, and S. Molnr. Identification and analysis of peer-to-peer traffic. *JOURNAL OF COMMUNICATIONS*, 1(7), 2006.

[32] D. Schatzmann, S. Leinen, J. Kogel, and W. Muhlbauer. Fact: Flow-based approach for connectivity tracking. In *Passive and Active Measurement conference*, Mar. 2011.

[33] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu.

Detecting prefix hijackings in the internet with argus. In *Proceedings of the 12th ACM SIGCOMM conference on Internet measurement*. ACM, 2012.

[34] B. Trammell and E. Boschi. Bidirectional flow export using IPFIX. RFC 5103, January 2008.

[35] J. Treurniet. A network activity classification schema and its application to scan detection. *IEEE/ACM Trans. Netw.*, 19(5):1396–1404, Oct. 2011.

[36] US Homeland Security. BGPmon. `http://bgpmon.net/`.

[37] Wikipedia. Netflow. `http://en.wikipedia.org/wiki/Netflow`.

[38] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston. Internet background radiation revisited. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 62–74, New York, NY, USA, 2010. ACM.

[39] V. Yegneswaran, P. Barford, and J. Ullrich. Internet intrusions: global characteristics and prevalence. *SIGMETRICS Perform. Eval. Rev.*, 31(1):138–147, 2003.

[40] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush. ispy: detecting ip prefix hijacking on my own. *IEEE/ACM Trans. Netw.*, 18(6), Dec. 2010.

[41] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '07, 2007.