Sentinels: Guarding ISP Networks from Forwarding Anomalies

Tobias Bühler
ETH Zürich
buehlert@ethz.ch

Ingmar Poese
BENOCS
ipoese@benocs.com

Laurent Vanbever
ETH Zürich
lvanbever@ethz.ch

1. INTRODUCTION

Where is the Skype traffic entering an ISP network leaving and does the chosen egress point match the configuration from the control plane? A compelling answer to this seemingly easy questions is actually quite hard to find. Internet traffic enters, flows and leaves an ISP network in unpredictable ways, according to dynamic routing decisions. A lot of changes are triggered by uncontrollable events from outside the network. Even worse, the already available measurement tools such as NetFlow [1] or SNMP [2], currently often used for billing, provide only imperfect, coarse-grain data on a per-device basis. The data is heavily sampled and there is no connection between different observation points.

We want to build a system which combines the already existing measurement data from e.g. NetFlow with control plane configurations to answer specific queries from operators (such as the ones above). In the end, we want to have similar end-to-end statistics as e.g. [3] achieves for data center networks and a query language comparable to [4] to answer different requests.

Verifying forwarding properties inherently requires to keep track of how packets are forwarded, which is often done by embedding tags on the packets. Yet, flexibly embedding tags in ISPs is challenging as they do not usually possess programmable hardware. Instead, our system aims at identifying some packet headers that show the required property, e.g. that traffic with this header enters via a point X.

Figure 1 shows a high-level view of our whole system. Section 2 describes how NetFlow data is used to compute so called *Source Sentinels*. Intuitively, a src sentinel could be seen as a tag uniquely identifying an ingress router. Unlike in a data center or in a enterprise network where tags are set at the ingress, src sentinels need to be computed and updated (when traffic shifts). A first evaluation of found sentinels in real ISP network traffic is provided in Section 4. The sentinels as well as BGP and NetFlow data are used to execute sentinel-based applications (described in Section 3) to answer network operator specific queries. Because the input data is imperfect, the results are forwarded to a probabilistic analysis function (outlined in Section 5) to produce final results with a corresponding probability.

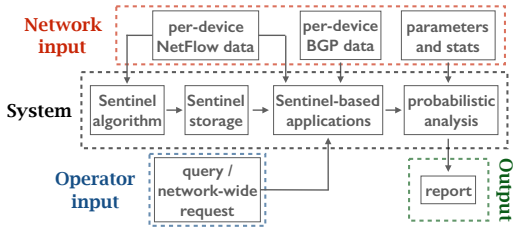

Figure 1: Overview of our system.

Novelty: Other works looked at marking and analyzing packets at the ingress points. IP traceback systems aim at finding the real origin of spoofed packets. For example, [5, 6] mark packets on the first ingress router with the IP of the used interface. In contrast, our src sentinels aim to find the ingress point inside one ISP network only. We are therefore independent of non-participating neighbor ISPs. Furthermore, once we have found the sentinels, they can be used to detect a variety of problems (e.g. forwarding anomalies or policy validations) and not only the origin of spoofed IP packets.

2. SRC SENTINELS

Definition: Given a number of ingress routers $\{R_1, R_2, ..., R_n\}$ of a network, an IP subnet S is a valid src sentinel for router R_x if and only if all flows with a src IP s in S are entering the network only over R_x. Figure 2 shows some valid (e.g. 60.0.0.0/28 and 90.0.0.0/26) and invalid (10.0.0.0/24) src sentinels. The sentinels are based on observed network data.

Algorithm: We use a Binary-Tree to find the src sentinels. We collect NetFlow data of flows entering the ISP network over any ingress point. For these flows we save a tuple `<src_IP, router>` at the root node of the Binary-Tree. For every saved tuple, we divide the src IPs based on the highest bit and save the tuple in the matching leaf. We repeat the process with the two new leaves and the next lower bits until either of the fol-

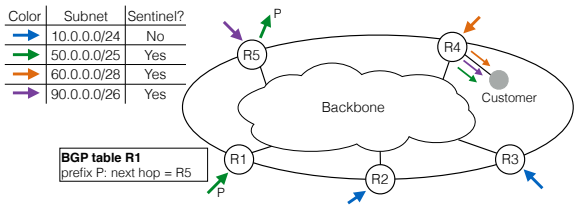

Figure 2: Src sentinels indicate src prefixes known to enter via a single ingress. Here, the network has 3 valid src sentinels and one invalid one (10.0.0.0/24, as it enters via R2 and R3).

lowing two conditions occurs: *(i)* we have a node which contains only IPs from one ingress router in which case the corresponding prefix is a src sentinel; or *(ii)* we reach a /32 node and we still have observations from multiple ingress routers, we therefore cannot find a src sentinel with the given IP. Because the measurement data provided by NetFlow is imperfect, we cannot guarantee that a found sentinel is actually valid and we need a probabilistic analysis (Section 5). The found sentinels are saved in a sentinel storage and continuously updated with new NetFlow data.

Different Sentinel Types: The src sentinels are useful to observe flows on ingress and egress points. To observe and verify other network properties it could be better to use different sentinel versions. For example AS sentinels, where we can replace the tuple `<src_IP, router>` with `<src_IP, AS_number>`. Other interesting sentinel types are interface sentinels or geographical sentinels.

3. APPLICATIONS

Egress Monitoring: We look at all the ingress points and collect the flows entering from a src sentinel subnet. We save the src sentinel to dst prefix pairs and look at matching flows at egress points. Once we observe a flow at an ingress and one (or more) egresses we compare the observations with the BGP table of the ingress router. If the observed egress point does not match the BGP table, we may have found a discrepancy between the control plane and the actual forwarding path taken by the flows. Figure 2 shows an example of this application. On router $R1$ and $R5$ we observe the same flow from the src sentinel 50.0.0.0/25 towards a dst prefix P. We can compare this observation with the BGP table of ingress router $R1$ and conclude that the flow is taking the expected egress router (next hop) $R5$. In a similar way, we can also detect forwarding anomalies. An ingress observation without a matching egress one could indicate e.g. a blackhole or a forwarding loop.

DDoS Detection: Sentinels can also be used to detect some DDoS attacks in near real-time by tracking the evolution of the number of sentinels seen. For instance, if we observe a sudden increase in the number of sentinels leaving to one customer (Figure 2, customer near $R4$), it means that the customer starts seeing traffic from more parts of the Internet. Due to the known ingresses of the corresponding sentinels, the DDoS traffic can be dropped before it enters the network.

4. EVALUATION

We evaluated our system on multiple hours of real NetFlow and BGP data (around peak time) from a big ISP in Europe. We were searching for src sentinels and could verify the following points:

- **Existence:** Every 5 minutes, we found an average of **900 000 src sentinels** with an average size of **/26**. Our system is therefore able to operate all the time. The sentinels contain multiple IP addresses which increases the probability that we observe sentinel flows in the NetFlow data;
- **Coverage:** On average, the found src sentinels cover **more than 60%** of the incoming traffic on every ingress router and contain flows towards **at least a quarter** of all other edge routers in the network. We are therefore able to detect network problems for a high traffic amount and between a lot of edge routers;
- **Computation Time:** A first implementation of the src sentinel search algorithm needs around 40 seconds to find all sentinels in 5 minutes of NetFlow data.

5. FURTHER WORK

Probabilistic Analysis: As NetFlow data is heavily sampled, a precise estimation of the total traffic amount and number of flows is difficult. Also, the sampling rates at different network points can vary greatly. An observation of a sentinel flow on an ingress only is therefore not automatically a sign for a loop or blackhole in the network. We may just not get any matching NetFlow entries on the egress due to the high sampling rates.

To minimize the impact of the different sampling rates we are currently looking into multiple solutions. We can increase the observation times or we can only track flows with a traffic amount bigger than a certain threshold. Both of these approaches increase the probability that we see the flows at an ingress *and* an egress. Furthermore, we are working on a likelihood function which computes a probability that e.g. an ingress observation only or a src sentinel is valid.

Query Language: We are also working on a query language similar to the query language used in [4]. Given a query, our system should first check if we have matching src sentinels available and then pick the best sentinel type to answer the query. Finally, we observe the corresponding flows from the NetFlow data and use the probabilistic analysis to answer the query (Figure 1).

6. REFERENCES

[1] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Internet Engineering Task Force, Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3954.txt

[2] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157 (Historic), May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1157.txt

[3] C. Guo, L. Yuan, D. Xiang, Y. Dang, R. Huang, D. Maltz, Z. Liu, V. Wang, B. Pang, H. Chen, Z.-W. Lin, and V. Kurien, "Pingmesh: A Large-Scale System for Data Center Network Latency Measurement and Analysis," in *SIGCOMM*, 2015.

[4] S. Narayana, M. Tahmasbi, J. Rexford, and D. Walker, "Compiling Path Queries," in *NSDI*, 2016.

[5] A. Belenky and N. Ansari, "IP Traceback With Deterministic Packet Marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, 2003.

[6] G. Jin and J. Yang, "Deterministic Packet Marking based on Redundant Decomposition for IP Traceback," *IEEE Communications Letters*, vol. 10, no. 3, pp. 204–206, 2006.