Incident Forensics in Distributed High-Speed Networks

Minoo Rouhi*, Quirin Scheitle*, Oliver Gasser*, Christian Wahl*, Marcin Nawrocki†, Matthias Wählisch†, Raphael Hiesgen‡, Thomas C. Schmidt‡
*Technical University of Munich, †Freie Univesität Berlin, ‡HAW Hamburg
{minoo.rouhi, quirin.scheitle, oliver.gasser, christian.wahl}@tum.de,
{marcin.nawrocki, m.waehlisch}@fu-berlin.de, {raphael.hiesgen, t.schmidt}@haw-hamburg.de

ABSTRACT

The retention of network data for incident forensics is an unsolved challenge: High-speed networks produce gigabytes of data per second, which is infeasible to store in raw format. On the other hand, common techniques to reduce data, such as sampling or aggregation, do not provide insight into the full payload, which may be crucial for attack forensics.

In this work, we propose and evaluate a proof-of-concept framework which monitors raw traffic and selectively stores full raw packet captures of suspicious flows. This tackles several long-standing challenges with novel contributions: First, we identify suspicious flows through a distributed sensing network, which correlates anomalies from various input networks. Second, we leverage a framework capable of holding and searching minutes of traffic on high-speed networks in memory, enabling to not only capture a flow from an anomaly onward, but also minutes before. Third, we tailor the permanent storage of traffic to suspicious flows, rather than storing traffic containing mostly irrelevant packets.

1. INTRODUCTION

Many attacks cannot be predicted in advance. Network forensics is an important feedback loop when operating threatened networks, as it provides post-mortem analysis of anomalies to improve security of systems retrospectively. The success of forensic approaches depends on data quality.

Complete data which was captured during the incident is key to forensics, but challenging to achieve. The most trivial approach of storing *all* traffic independently of the occurrence of an incident is infeasible in high-speed networks, which often leaves forensics with sampled data, exported flows, or no data at all.

In this poster, we propose three components to achieve a trade-off between data completeness and storage capacities. *(i)* A system of distributed threat intelligence, which shares anomaly data among several networks. This allows for watching suspicious nodes before they may have contacted an individual network. *(ii)* A digital network oscilloscope that is capable of holding raw traffic in main memory at high-speed line rates and filter traffic from the recent past on demand. *(iii)* A middle layer for continuous queries and data sharing, without conflicting with privacy concerns.

We describe our detection system in more detail in Section 2, present a preliminary evaluation in Section 3, and conclude in Section 4.

2. DISTRIBUTED INCIDENT DETECTION SYSTEM

Proposed System: Our proposed system is comprised of three basic components: *sensors*, an *aggregation layer*, and *actors*. Sensors may be any device (or service) capable of sensing network anomalies, such as firewalls, IDS, or honeypots. As many network attacks are not local but distributed across several networks, sensors are placed in different networks. The aggregation layer collects and processes data from these sensors, and selects relevant events to export to an actor. Such an aggregation layer should operate on very low latency as we need to capture distributed traffic flows in time to comply with the requirement of complete data. As aggregation layer, we use VAST [5]. VAST enables continuous queries on event streams in near real-time, and provides a remote interface to which sensors and actors can subscribe. FlowScope captures traffic on high-speed line rates and stores this traffic in a ring buffer to allow for data access of the recent past, typically in the order of minutes—depending on memory configuration and networks speeds. Note that data storage is implemented in main memory to achieve near real-time processing. Then, by applying filter rules issued by the aggregation layer, FlowScope exports raw traffic dumps for both the past and all packets observed in the future.

Architecture: Figure 1 shows a detailed overview of the different components. For a proof of concept, we use common honeypot and IDS software as sensors. These sensors dump log files in Bro [3] format every 60 seconds, which are distributed to actors.

To extract traffic features, our aggregation layer analyses honeypot and IDS logs and makes the results available to a threat detection component. Having the distributed data available via a unified interface, we can apply multiple detection schemes to craft concise filters for the actors. We can specifically leverage correlation across multiple sensors to filter high-confidence anomalies. Based on this outcome, the aggregation layer generates filter lists to capture traffic that is used for the network forensics.

As soon as an incident is identified, using a REST-API, filter lists are pushed to corresponding FlowScope instances, which take care of continuously monitoring the upstream traffic. Finally, selected traffic from the high-speed network interfaces are dumped to PCAP files.

It is worth noting that all components can run on a single node, if necessary. This would not allow for correlating data, but still implement on-demand traffic capturing, to improve completeness for network forensics without exhausting memory resources.

Proof of Concept Setup: We deploy our distributed detection system in three campus networks. At FU Berlin

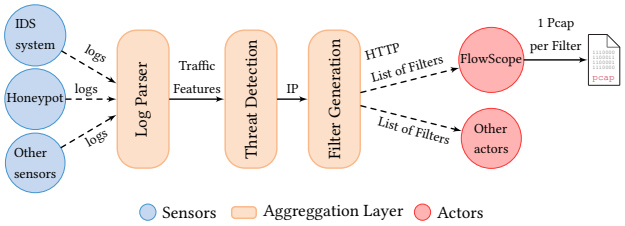

Figure 1: Processing chain of the framework

and HAW Hamburg, we operate multiple honeypot and IDS sensors. The incident detection system and the actor are deployed in the Munich scientific network. FlowScope scaled well on the 10 Gbps link, in the sensor network, even with hundreds of rules installed.

3. EVALUATION

Comparison to Other Frameworks: *(i)* Existing forensic frameworks offer the possibility to store every single packet sent down the wire [4]. The benefit of this approach is, that it provides a complete view into the full raw traffic, which then can be used for detailed anomaly analysis. Nevertheless, the approach suffers a bottleneck in high-speed links; a full capture on a monitored 10 Gbps Ethernet link equals storing 14.8 Mpps for minimum sized frames. *(ii)* Other detection frameworks solely rely on network flow data, which contains aggregated packet header values, timing and traffic volume information rather than payload data [1]. The advantages include ease of extraction of protocol level information, and low overhead for persistent retention of data across large-scale networks. However, it lacks further context for comprehensive incident root cause analysis, due to absence of payload data.

To reach a middle ground, our proposed system, focuses on capturing full record of exclusively suspicious network activity. This comes with several added virtues. First, we focus on the most relevant portion of data, which constitutes an anomaly, eliminating the limitations of storage and processig power exhaustion of technique *(i)*. Second, having access to the full record of suspicious activity, overcomes the downsides of data retention mentioned in *(ii)*. Additionally, storing payload data of only potential incidents compared to capturing everything including non-intrusive data is more well grounded from a legal point of view, as payload data typically expose personal identifiable information. Lastly, the distributed nature of the sensors and using FlowScope instances as actors allow for detection of distributed attacks and a view into past threat activities respectively.

Assessment of System Latency and Retrospective Inspection: A timely detection and access to past anomalous activities of a marked incident is pivotal to network forensics. For assessment of latency, we define the maximum detection latency as the elapsed time between creation of a log at the sensors and the application of rules to the actor. In our current setup, the sensors have a log rotation of one minute. Taking the added log upload delay from the sensors into account, we reach a maximum detection latency with granularity of about one minute after an attack has been spotted at the distributed sensing devices.

With respect to retrieving past records of activity of incidents from the moment of detection, FlowScope as the actor is able to store network packets in a fast ring buffer data structure in memory for retrospective traffic inspection. A link connectivity of 10 Gbps and available RAM storage of 256 GB, allows for peaking into past incident activities up to three minutes after-the-fact ($\frac{256\,\text{GB}}{1.25\,\text{GB/s}} = 204.8\,\text{s}$).

We conclude, that our forensics framework achieves reasonable detection latencies and can successfully handle traffic beyond 10 Gbps if enough RAM is present.

Observations from Test Deployment: We conducted a 10-hour test-run of our Proof-of-Concept setup in October 2017. In evaluating the captured raw traffic, we observe various patterns correlating to port scans. Interestingly, we find 38% of anomalies sensed at the remote sensing network to also occur at the local network. For these 38%, our system was able to store the raw traffic based on the remote anomaly sensing, highlighting that anomalies correlate across different networks, and the use of a distributed sensing/acting platform for network forensics.

4. CONCLUSION & FUTURE OUTLOOK

In this work, we presented an incident forensics framework for distributed networks. We discussed the design of our near real-time system, which addresses both the challenges of implementing incident analysis at scale of Gigabytes of data employing VAST, and capturing data on demand on links with speeds surpassing 10 Gbps using FlowScope. We provided an analogy to other exisitng incident detection systens, an assessment of the system latency and reported on deployment experiences of a first prototype. We demonstrated that our incident detection system is capable of storing and inspecting network traffic both at real-time and retrospectively when adequate hardware is provisioned.

Future Work: We are currently working on the integration of our prototype in a high-speed forensic processing platform [2] based on the distributed network visibility system VAST. Adding various data sources into the flexible VAST indexing allows to perform effective correlation analyses shortly after the data becomes exposed to continuous queries. Such queries shall detect potential anomalies and may trigger further investigations on selected data subsets.

5. REFERENCES

[1] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys & Tutorials*, 16(4):2037–2064, 2014.

[2] P. Meyer, R. Hiesgen, T. C. Schmidt, M. Nawrocki, and M. Wählisch. Towards Distributed Threat Intelligence in Real-Time. In *Proceedings of SIGCOMM Posters and Demos '17, Demo Session*, New York, NY, USA, August 2017. ACM.

[3] V. Paxson. Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463, 1999.

[4] M. Roesch et al. Snort–Lightweight Intrusion Detection for Networks. 1999.

[5] M. Vallentin, V. Paxson, and R. Sommer. VAST: A Unified Platform for Interactive Network Forensics. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, March 2016.