Challenges with collecting, anonymizing, sharing and using high-speed network-traffic data

Alastair Nottingham, Jeff W. Collyer, Brian E. Root, Molly Buchanan, Yizhe Zhang, Kolia Sadeghi (CCRi),
Yixin Sun, Don E. Brown, Jack W. Davidson, Malathi Veeraraghavan
University of Virginia (UVA)
mv5g@virginia.edu

1 Experiences with network-traffic data in a DARPA cybersecurity research project

Data collection: Fig. 1 illustrates how we collect network traffic data and host logs from the UVA campus network and Computer Science (CS) departmental network to develop machine learning methods for detecting zero-day and large-scale cyberattacks. The WAN-access ports on the campus Information and Technology Services (ITS) border routers are mirrored to links (shown in red) to a Gigamon appliance, which is operated by the UVA Information Security (InfoSec) organization. The Corelight appliance creates Zeek logs for all UVA border network traffic (which currently is around 12-14 Gbps). Our research group runs the Anonymization Host, located within the secure InfoSec network, to remove all Personally Identifiable Information (PII) from the Zeek logs before saving the datasets in our Research HPC for use by our machine-learning team. Aggregate weekday Zeek-log size is roughly 1TB. We also collect network and host logs from the CS departmental network. Supplemental data includes (i) NAT, DHCP and Asset logs to enable per UVA-host traffic baselinining, and (ii) ground-truth data from the InfoSec FireEye appliance, and the Stingar Honeypot feed from Duke University. NAT logs are particularly huge (about 250 GB per day).

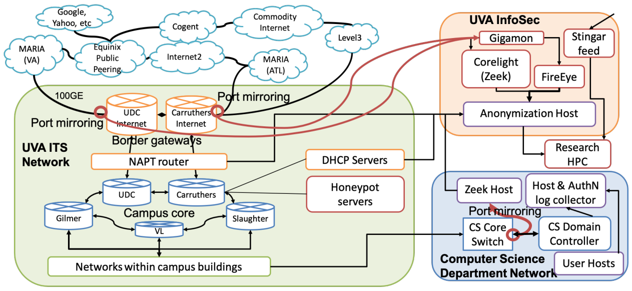

Figure 1: Data collection setup

Data anonymization:

We have developed an extensive anonymization approach, which aims to obfuscate identifying data while preserving many of the features and patterns inherent in the raw logs necessary for attack detection. Field types are handled on a case-by-case basis using a variety of methods. For example, we use the prefix-preserving IP address anonymization technique, USC cryptopANT, which is based on the GTech crypto-PAn algorithm. More challenging is the anonymization of fields that appear in HTTP Zeek logs since URIs could include Personally Identifying Information (PII), such as usernames, passwords, credit card numbers, and social security numbers. We have developed a method for anonymizing components of URIs while preserving server and parameter names, essential for attack-detection. Our C++ code, named Log Processing Platform (LPP) uses parallelism, zero-copy messaging, and extensible plugin architecture. Currently, LPP has plugins for anonymizing a subset of Zeek logs, specifically Conn, HTTP, SSL, SSH, DNS, x509, IRC, Kerberos and files.log. LPP takes 50 to 90 minutes (depending on daily traffic volume) to anonymize one-day logs on a single host with 2 GB RAM and 48 cores.

Data access for cybersecurity and other researchers: Our CISO and General Counsel approved our request to share these datasets with external researchers as long as data downloads were disallowed as part of the data-use agreement. Therefore, we created UVA accounts for these researchers, which let them “bring their code to the data” on our HPC Sentinel. We currently have about 46 users from 18 organizations. Sentinel offers OpenStack (virtual machines), SLURM (batch processing) and Jupyter Notebooks (interactive processing), and Apache Hadoop with Druid. Sentinel features and dataset information are posted on a dedicated UVA Wiki accessible to researchers and collaborators. The Wiki also provides sections where data users post suspicious/malicious findings for evaluation by the UVA Security Operations Center (SOC).

Data usage: A one-day dataset is about 1 TB, and has about 1.5M Zeek logs. Processing data at scale is crucial with such massive amounts of data. We developed a programming framework called *Features* to scale up Extract-Transform-Load (ETL) computation and machine-learning analytics. The framework is built upon PySpark, pandas and Parquet file format; it breaks out expensive operations into multiple intermediate steps, and intermediate memoized files are shared among users to avoid duplicated computation. We currently have four pipelines for processing HTTP, SSH, SSL/X509, Conn logs. We create aggregates, e.g., external FQDNs, and generate features for these aggregates. Semi-supervised and active learning methods are deployed using the limited ground-truth labels available from FireEye, Stingar and some additional threat-intelligence feeds. Attack recreation methods are used to generate network logs for communications by malware such as Wannacry and Emotet, and these logs are merged into the UVA Zeek logs to create training sets for machine-learning model generation. The models are then run on new days' data to find attacks. We have found and reported several potentially compromised UVA hosts to our SOC, some of which were true positives but others were false. Much work remains to be done to reduce our false positive rate.

2 Challenges

We have identified several challenges that arose as part of the experiences described above. Some of these challenges are generalizable to other data domains, as described below.

1. Data retention: For huge data, it is important to develop policies for what to retain and for how long. One option may be to monitor data access and base retention policies on access.
2. Sensor placement and adaptation: In-spite of the impressive variety and volume of our data-collection effort, our researchers request new sensor placement for additional data, e.g., internal network/host logs are required to detect lateral movement, which is often part of a cyber-attack kill-chain.
3. Privacy: The challenge is to find the right degree of anonymization to preserve privacy while retaining information pertinent to the application.
4. Organizational legal and ethical challenges: Institutional Review Board (IRB) approval is required for any data-oriented research that is deemed "Human Subjects Research (HSR)." A UVA policy, in combination with our anonymization approach, resulted in our research being classified as non-HSR, and hence IRB approval was waived. But as policies vary by institution, and anonymization may not be an option for some data-oriented research, legal and ethical issues need to be addressed.
5. Data labeling: Supervised methods need labeled data. We found that open-source blacklists and whitelists have quality issues, and subscription-based threat-intelligence services are expensive. For example, the use of Cloud computing for malicious activities impacts accuracy of white lists. The alternative, which is to use unsupervised methods, presents the problem of having no easy way to validate results, except to use experts, e.g., SOC analysts for our application, to rank order anomalies as malicious, for active or semi-supervised learning.
6. Sustaining data collection/use projects: Our current effort is supported by a 4-year DARPA project. While we are actively increasing our user base, we will need to develop methods for sustaining this effort. One approach would be to provide data users free initial access to generate preliminary research results, and then charge data service fees in their funded projects. Questions of whether university data can be shared with commercial organizations have ethical implications.
7. In-situ computation: Supporting external researchers on our HPC incurs HR costs for system administration and user support. Just maintaining the variety of big-data analytics software packages such as Hadoop, Spark, Druid, etc., and environments such as OpenStack, SLURM, etc., is challenging.
8. Sharing computed results: Big-data analytics have multi-step pipelines. Sharing intermediate results, such as ETL output, will save significant time and speed up scientific/engineering findings.
9. Keeping documentation up to sync with the shared data is highly important, as significant time can be lost in users deciphering the data dictionary/metadata.