# A Modular Multi-Location Anonymized Traffic Monitoring Tool for a WiFi Network

Justin Hummel, Andrew McDonald, Vatsal Shah, Riju Singh, Bradford D. Boyle,
Tingshan Huang, Nagarajan Kandasamy, Harish Sethu, and Steven Weber
Department of Electrical Engineering and Computer Engineering
Drexel University, Philadelphia, PA, 19104
{jch59,awm32,vbs27,rs557,bdb24,th423,nk78,hs42,spw26}@drexel.edu

## ABSTRACT

Network traffic anomaly detection is now considered a surer approach to early detection of malware than signature-based approaches and is best accomplished with traffic data collected from multiple locations. Existing open-source tools are primarily signature-based, or do not facilitate integration of traffic data from multiple locations for real-time analysis, or are insufficiently modular for incorporation of newly proposed approaches to anomaly detection. In this paper, we describe *DataMap*, a new modular open-source tool for the collection and real-time analysis of sampled, anonymized, and filtered traffic data from multiple WiFi locations in a network and an example of its use in anomaly detection.

## 1. INTRODUCTION

A typical piece of new malware today uses a variety of obfuscation techniques to avoid detection, especially signature-based detection typical of antivirus products currently on the market [13]. Such increasing sophistication of viruses, worms and other malware has made their early detection immediately after launch significantly harder. Some reported success rates with signature-based detection have been as low as 5% [9]. Many security professionals argue that real-time monitoring for anomalous behavior in the traffic data over a network, as opposed to signature-based detection, offers a surer approach to protecting networks [6, 7].

Real-time monitoring for anomalous behavior, however, offers its own challenges. One challenge is the matter of minimizing false positives and false negatives in the detection of patterns that deviate from the normal after the data is gathered and stored [5]. The other challenge is gathering and storing the data in real-time to enable the immediate use of detection algorithms on it. This challenge stems primarily from the fact that traffic data collected at a single location is usually insufficient to infer an anomaly; one requires traffic data from multiple locations to be able to conclude that an anomalous behavior is ongoing.


*CODASPY'14,* March 3–5, 2014, San Antonio, Texas, USA.
ACM 978-1-4503-2278-2/14/03.
http://dx.doi.org/10.1145/2557547.2557580.

While data from multiple locations is helpful to anomaly detection, the volume of data can be prohibitively large. One approach is to allow the transfer of only a statistical sample of the data, such as histograms constructed over slices of time [4], as opposed to individualized packet data. In this paper, we describe an open-source tool, called *DataMap*, developed as part of a project to facilitate real-time analysis of traffic data from multiple locations in a large-scale WiFi network. The DataMap tool is designed for adaptation to a variety of statistical sampling and aggregation techniques. It is also intended to provide a means for empirical estimation of fundamental trade-offs such as between the sampling rate of traffic data and the accuracy of inferences on anomalies possible from it.

## 2. RELATED WORK

There are a number of open-source tools available for data collection to monitor network traffic and detect intrusions. Most of these are primarily targeted for detecting known types of intrusions (as opposed to detecting an anomalous pattern of behavior) using signature-based approaches, the most widely used of these being *Snort* [11]. Snort is a lightweight sniffer, packet logger and an intrusion detection system which can generate alerts when it observes specific types of probes or attacks that indicate a potential intrusion attempt. The detection can be based on a rule set included in the snort download and updated daily or on custom rules written by the user of snort. An installation of snort runs on a single machine and it takes a complementary set of tools to gather traffic data from multiple locations and detect in real-time a gradually spreading worm such as that described in Section 1. Also, being signature-based, Snort is not able to detect anomalies whose signature is not yet known.

Open-source software tools that come closest to some of the functionality of the DataMap tool include *Security Onion* [10] and *OpenWIPS-ng* [8]. While these tools are also based on signature detection—sometimes based on tools such as Snort—they do allow data collection from multiple locations to facilitate an integrated analysis of the full network-level context. While these very useful tools are close in some functionality to the DataMap tool, they are all largely built on a foundation of signature-based detection and are not ideal for an academic study of approaches to anomaly detection when signatures are not known.

DataMap can be employed in conjunction with approaches that use any combination of signature-based algorithms to detect known patterns (with rule sets) and anomaly-based

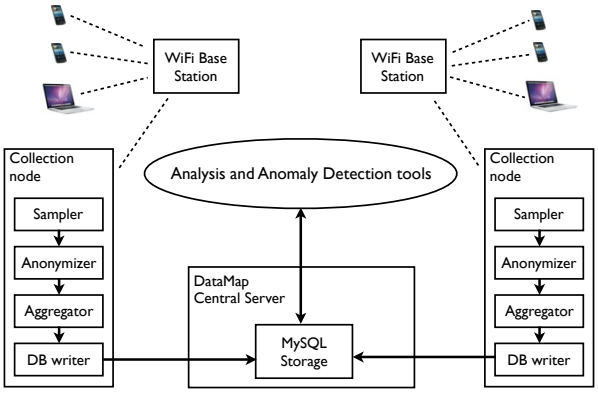



**Figure 1: A DataMap traffic monitoring system showing two collection nodes (also called traffic sensors) and the central server.**

algorithms to detect patterns not seen before (without predetermined rule sets). DataMap's modular design facilitates the sampling, aggregation, and transfer of data from multiple locations (sensors) to a central server for real-time analysis. Built on replaceable modules, the DataMap tool is intended to facilitate the study of newer techniques for anomaly detection, such as those based on compressive sampling, histogram construction, or information-theoretic metrics. While we have only reviewed the dominant open-source tools in this section, the tools that require paid licenses are even less adaptable to modification for academic research on new approaches to anomaly detection.

# 3. OVERVIEW OF THE DATAMAP COMPONENTS

The DataMap tool, a pre-release version of which is available on Github [2], utilizes a series of collection nodes and a central server to collect traffic data from a set of WiFi access points at multiple locations. The collected data is aggregated into a single database hosted on the central server. DataMap builds upon existing open-source software and, in contrast to other currently available tools, is open-source, offers concurrent multi-location traffic monitoring, enables integrated analysis on the multi-location aggregated data, and is modular to allow easy modification for academic research in anomaly detection. DataMap is designed to facilitate experiments on sampling rates and both time-domain and space-domain data aggregation strategies. It can be used with unencrypted WiFi networks or on encrypted networks with administrative authority.

Figure 1 shows a high-level block diagram of the DataMap infrastructure. The collection node consists of four primary components: the sampler module, the aggregator module, the anonymizer module and the DbWriter module. These components on the collection nodes work as part of a node daemon which is managed by the central server. The central server hosts the database which serves as the data repository and manages the collection nodes by way of a server daemon. A web interface, provided with the DataMap tool, can be used to keep track of the state of all the collection nodes and to start or stop them all at once.

The *sampler module* uses Vermont (Versatile Monitoring Toolkit [12]) to capture traffic with the desired sampling algorithm and the desired filtering strategy. It is the modular nature of Vermont that enables the DataMap project to create replaceable and independent components in the DataMap tool. The wireless network interface on a collection node is placed into monitor mode in order to collect all packets from access points on a given channel. A node can be configured to identify and channel hop to capture traffic data from multiple wireless networks in a location.

The *anonymizer module* uses *Crypto-PAn* (*Crypto*graphy-based *P*refix-preserving *An*onymization [3]) to anonymize individual IP addresses in a way that protects individual network users' privacy while preserving topological information about the network. Other identifying information such as MAC address and application-layer data is discarded in this module. Sophisticated data mining even on anonymized data can identify individuals and patterns of individual actions and privacy is preserved in the use of the DataMap tool only if it is used exclusively for the intended purpose.

The *aggregator module* summarizes the raw data gathered by the node for each pre-programmed slice of time. The aggregation can be based on any combination of packet header fields specified in an XML file. Data is passed from the sampler through the anonymizer to the aggregation module via shared memory such that the anonymization occurs before the collected data is transmitted across the network.

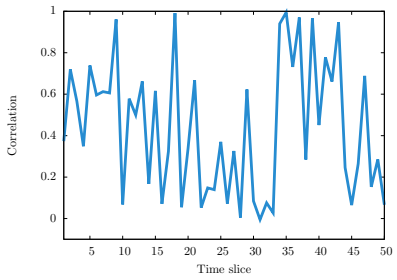
Finally, the *DbWriter module* sends the aggregated, filtered, and anonymized data to the central server. Data is labelled with a unique identifier corresponding to the location of the collection node and entered into the database.

These components on the collection nodes work as part of a node daemon which, upon start-up, sends a *Hello* message to the central server with its id and location. The daemon then waits for instructions from a server deamon on the central server, which keeps track of all the nodes and periodically pings them with a *Heartbeat* message to retrieve their latest state. A web interface, provided with the DataMap tool, can be used to keep track of the state of all the collection nodes and to start or stop them all at once.

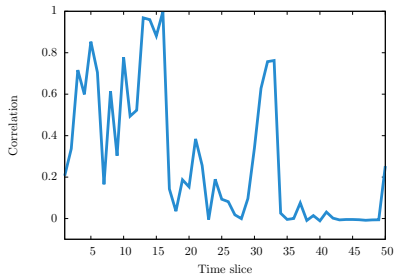
Our experience and other research suggests that patterns of data across multiple locations are best analyzed based on the statistical distribution (e.g., a histogram) of features of interest [4]. The pattern of changes in correlations over time between these distributions at different locations is not sensitive to time skews between the collection nodes (even if the correlations themselves fluctuate from one time slice to the next). This allows DataMap to dispense with clock synchronization between collection nodes and the associated overhead.

# 4. AN EXAMPLE OF DATAMAP USAGE

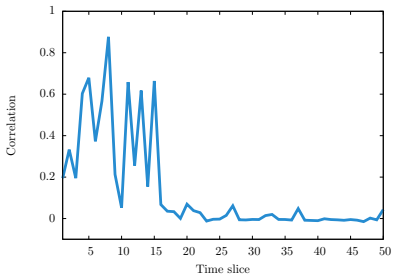
We present a simple example of data collection across multiple nodes and how it can be helpful in traffic monitoring and, potentially, the detection of an Internet worm in its early stages. A typical worm begins its life by first scanning for vulnerabilities on open ports at as many different IP addresses as possible in as short a time as possible using any of a number of sophisticated approaches. The distribution of the destination IP addresses of IP packets emerging from a worm-infected node is likely very different from that of an uninfected node. Detection of a wider range of anomalies in this or other features is made possible by constructing histograms of these features as described in [4].




(a) Correlation between histograms of source addresses at locations 1 and 2.




(b) Correlation between histograms of source addresses at locations 1 and 3.




(c) Correlation between histograms of source addresses at locations 2 and 3.

**Figure 2: An example of data collected at three locations on the Drexel University campus. The plots show correlations between the histograms computed on the traffic at these locations for source addresses.**

However, for the same reasons mentioned in Section 1, it is ideal to not rely entirely on a signature-based assessment of normal vs. anomalous histograms. A more effective approach is one that also uses a real-time assessment of deviations between the histograms computed at different nodes. Past research has shown that an analysis of correlations between traffic features through techniques such as Principal Component Analysis can achieve effective anomaly detection [1]. The DataMap tool allows precisely this kind of analysis. These correlations will likely fluctuate from one time slice to the next, but the pattern of fluctuations itself can serve as an indication of what is "normal". A sustained shift in the pattern of changes/fluctuations in the correlation between the histograms computed at two different but equally busy nodes indicates one of these nodes as a potential candidate for further examination by a system administrator. Such a sustained shift can be observed by dividing time into slices and computing the correlation between histograms at each slice.

Figure 2 presents data collected over a period of 30 minutes (or 50 time slices of 36 seconds each) from three nodes at different locations within the Drexel University campus network using the DataMap tool. Plotted is the correlation between histograms of the source IP addresses at two different locations, with each data point in a graph representing the correlation computed over a 36-second period.

The figure shows a sustained shift in the pattern of correlations between traffic at location 3 and the traffic at the other two locations beginning sometime between time slice 15 and time slice 20. This can be an indication of something anomalous or it could be a normal event in which location 3 is exhibiting some deviation for legitimate reasons. A deeper analysis using correlations between histograms of other features can complete a determination of whether or not this event deserves to be flagged with an alert.

## 5. CONCLUDING REMARKS

In this paper, we have briefly described a pre-release version of the DataMap tool for multi-location traffic monitoring and analysis. While DataMap helps detect malware threats, it does not take action to neutralize threats. It is a modular tool intended to serve as a framework for research and development of new approaches to traffic sampling, data aggregation, and analysis for effective network traffic anomaly detection.

## Acknowledgment

This work was partially funded by the National Science Foundation Award #1228847.

## 6. REFERENCES

[1] D. Brauckhoff, K. Salamatian, and M. May. Applying PCA for traffic anomaly detection: Problems and solutions. In *Proc. IEEE INFOCOM*, 2009.

[2] DataMap. `https://github.com/DataMap13/DataMap/`. Accessed: August 8, 2013.

[3] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks*, 46(2):253–272, 2004.

[4] A. Kind, M. P. Stoecklin, and X. Dimitropoulos. Histogram-based traffic anomaly detection. *IEEE Trans. Netw. Service Manag.*, 6:110–121, June 2009.

[5] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *Proc. ACM SIGCOMM*, 2005.

[6] P. Li, M. Salour, and X. Su. A survey of Internet worm detection and containment. *IEEE Communications Surveys and Tutorials*, 10:20–35, 2008.

[7] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proc. IEEE INFOCOM*, pages 1901–1910, 2003.

[8] OpenWIPS-ng. `http://www.openwips-ng.org/`. Accessed: August 8, 2013.

[9] N. Perlroth. Outmaneuvered at their own game, antivirus makers struggle to adapt. *The New York Times*, December 31, 2012.

[10] Security Onion. `https://code.google.com/p/security-onion/`. Accessed: August 8, 2013.

[11] Snort. `http://www.snort.org/`. Accessed: August 8, 2013.

[12] Vermont (VERsatile MONitoring Toolkit). `https://github.com/constcast/vermont/wiki`. Accessed: August 8, 2013.

[13] C.-H. Wu and J. D. Irwin. *Introduction to Computer Networks and Cybersecurity*. CRC Press, 2013.