WebClass: Adding Rigor To Manual Labeling of Traffic Anomalies

Haakon Ringberg
Computer Science Dept
Princeton University

Augustin Soule
Thomson

Jennifer Rexford
Computer Science Dept
Princeton University

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
Authors take full responsibility for this article's technical content.
Comments can be posted through CCR Online.

ABSTRACT

Despite the flurry of anomaly-detection papers in recent years, effective ways to validate and compare proposed solutions have remained elusive. We argue that evaluating anomaly detectors on manually labeled traces is both important and unavoidable. In particular, it is important to evaluate detectors on traces from operational networks because it is in this setting that the detectors must ultimately succeed. In addition, manual labeling of such traces is unavoidable because new anomalies will be identified and characterized from manual inspection long before there are realistic models for them. It is well known, however, that manual labeling is slow and error-prone. In order to mitigate these challenges, we present WebClass, a web-based infrastructure that adds rigor to the manual labeling process. WebClass allows researchers to share, inspect, and label traffic time-series through a common graphical user interface. We are releasing WebClass to the research community in the hope that it will foster greater collaboration in creating labeled traces and that the traces will be of higher quality because the *entire* community has access to *all* the information that led to a given label.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; H.2.8 [**Database Management**]: Database Applications; H.3.5 [**Online Information Services**]: Data Sharing; C.4 [**Performance of Systems**]:

General Terms

Experimentation, Performance, Measurement

Keywords

network traffic analysis, database, data sharing, data labeling

1. INTRODUCTION

Traffic anomalies are an important concern to network operators. In response to this problem, a large number of anomaly detection techniques have been proposed [1, 3, 8, 20, 16, 19]. It is therefore important that we have sound means by which to evaluate and compare presented anomaly detectors. In order for an evaluation framework to be useful, it should provide a realistic estimate of how well the technique would help operators detect and diagnose anomalous events of interest. The realism requirement here means that both the background traffic and the types of anomalies (e.g., DDoS attacks, port scans, worms) contained in the evaluation must be representative of what operators see in the wild.

1.1 A Case For Manual Labeling

Past papers on traffic anomaly detectors have used a wide range of evaluation techniques. Determining the accuracy of detectors involves first identifying some set of events that *ought* to be found, i.e. the true-positive anomalies against which the detectors are evaluated. One employed strategy [8, 20, 19, 7] is to define this set of true positives as those found by some number of other anomaly detectors. The problem with this evaluation technique is that it does not independently verify the anomalies that are defined as true positives. That is, evaluating a new detector $\hat{D}$ based on an old detector D_i is somewhat dubious unless either (1) D_i was itself evaluated using independently verified anomalies, or (2) D_i, in turn, was evaluated based on an older detector D_j, which was evaluated using independently verified anomalies. In essence, this evaluation technique cannot achieve independent evaluation (of an anomaly detector) on its own.

Alternatively, the true positive anomalies can be defined as those that have certain specified properties [6, 15]. [6], for example, states that "any source which generated SYNs but no FINs for more than [n destinations is defined] as being a true port scan." While this is a reasonable strategy for some types of anomalies, it cannot capture the full diversity of anomalies seen in the wild. We simply do not have an adequate understanding of most anomalies to characterize them so precisely. Furthermore, the very precise definitions of anomalies that are inherent to this model will also invariably lead to false classifications. In the above definition, for example, it is difficult to argue for the appropriateness of a specific value of n.

A third possible evaluation approach leverages synthetic models of the anomalies one wishes to detect. This can either be done within the confines of a simulator/emulator for complete control, or as a form of anomaly injection into a preexisting trace [8, 16]. While simulation has promise to be a very useful component of a rigorous evaluation strat-

egy [13], it is (1) not sufficient because of the importance of *also* evaluating detectors on traces from operational networks, and (2) at present we simply do not have adequately realistic models of many anomalies to simulate them. Anomaly injection also suffers from the latter weakness in addition to also being unable to accurately capture feedback effects between the anomaly and background traffic because the background traffic is provided by the trace whereas the anomaly is simulated.

Manual labeling is the lone remaining evaluation strategy, has been the most widely utilized one [1, 9, 16, 10, 5], and is important not only by a process of elimination. We define manual labeling as a domain expert inspecting a set of potential anomalies and verifying whether they are indeed true positives. It is imperative that some of these labeled traces are from operational networks since it is in this setting that they must ultimately succeed. Also, in a new or evolving field such as anomaly detection, manual inspection builds knowledge of the underlying data. In particular, new anomalies or adaptations of old anomalies will necessarily be observed first "in the wild". Clearly, the domain experts that perform this initial diagnosis will be able to identify these anomalies long before reasonable models of their operation can be constructed for other evaluation techniques.

Moreover, manual labeling of traces is not a sign of poor methodology whatsoever. In fact, the entire field of supervised learning is devoted to designing effective algorithms that can learn from labeled traces. Manual labeling of datasets is invariably the first step towards training and testing algorithms in such disparate fields as parts-of-speech classification [12], image classification [18], and intrusion detection [11]. It should therefore not be surprising that manual labeling of traces has an important role to play in training and evaluating traffic anomaly detectors.

1.2 The Challenges of Manual Labeling

Despite its importance, manual labeling is not a panacea. In particular, its reliance on human intervention can introduce both errors and disagreements. There are multiple potential explanations for differences in the labels associated with a trace by various domain experts. First, modern networks carry vast amounts of high-dimensional data, which humans are generally ill-equipped to efficiency and reliably analyze. Additionally, the data we analyze is often incomplete. That is, due to the tremendous magnitude of measurement data, networks are forced to employ data-reduction techniques such as sampling, temporal aggregation, and spatial aggregation in order for the data-collection infrastructure to keep up with the large volume of data being transmitted. These techniques degrade the data and thus inherently make it more difficult to correctly diagnose an anomaly.

While we cannot easily improve the quality of the underlying data, we can remedy the other two important challenges to reliable manual labeling of traces: (1) that researchers utilize different tools to visualize, inspect, and label the data, or that (2) different methodologies are employed in order to distinguish true-positive anomalies from false-positive anomalies. We have designed and implemented a system we call WebClass in order to address these challenges. WebClass allows its users to share, label, and inspect labels associated with traffic timeseries. It has several important features: (1) a shared and visually expressive user interface that allows all labelers to see the same information, and (2) the ability to inspect, and thus critique, others' labels, which will improve the overall accuracy of labels. We will release WebClass to the community in the hope that it will lead to more sound evaluation of traffic anomaly detectors.

2. WebClass

WebClass is a Web-based software system that allows a user to inspect and label potential anomalies on timeseries of traffic measurements. It has a database back-end and an AJAX-based [4] front-end. WebClass stores the labels affixed by all users along with information about the human labeler, which allows users either to exclude labels for which there is too much disagreement and/or provide confidence figures for individual anomalies. Finally, egregious labeling mistakes can be detected because each user's labels can be inspected by all others.

2.1 Shared Back-End Database

WebClass is a system for *evaluating* anomaly detectors, not an anomaly detector. Running an anomaly detector is performed separately. Once this has been completed, a user must upload two sources of data into WebClass: (1) the traffic trace that the detector analyzed, and (2) the locations (spatial and temporal) of the alarms that the detector raised. WebClass presently supports traces that are timeseries of traffic measurements, viz., the IP 4-tuple [9] and flow/packet/byte counts. WebClass must also have access to the underlying flow traces (currently only NetFlow [2] is supported). Traffic timeseries stored in the WebClass database are always associated with the network from which the data was collected. The timeseries may also contain information regarding data-reduction techniques that have been employed, e.g., sampling, temporal aggregation, and spatial aggregation. The set of potential anomalies is parameterized by the detection algorithm used and how the algorithm was tuned, e.g., the detection threshold.

The vast majority of data used by WebClass is stored in a database. This includes the static information described above—i.e., the timeseries and potential anomalies—in addition to the labels and descriptions that are inserted when users classify anomalies. Retrieving data for further analysis is thus as simple as a SQL table dump. It is therefore possible to access the WebClass database from other applications, which can be designed to perform custom analysis depending on the needs of a given research group.

2.2 Graphical User Interface

The WebClass user interface is presented in figure 1. Users who wish to label anomalies using WebClass must first log in, as seen by box **A**. Using the fields in boxes **B** and **C**, the user may specify that only anomalies that match certain criteria (e.g., a specific network, an ingress router, a specific moment in time, etc.) should be classified. WebClass then iteratively displays the potential anomalies that match these requirements so that the user may classify them.

WebClass has multiple tools to help the user arrive at a decision about the type of any given potential anomaly. The most prominent aspect of the UI is the plot of the traffic timeseries: source IP address entropy, source port number entropy, destination IP address entropy, destination port number entropy, #packets, and #bytes/packet (i.e. average packet size for the given time window). Potential anoma-

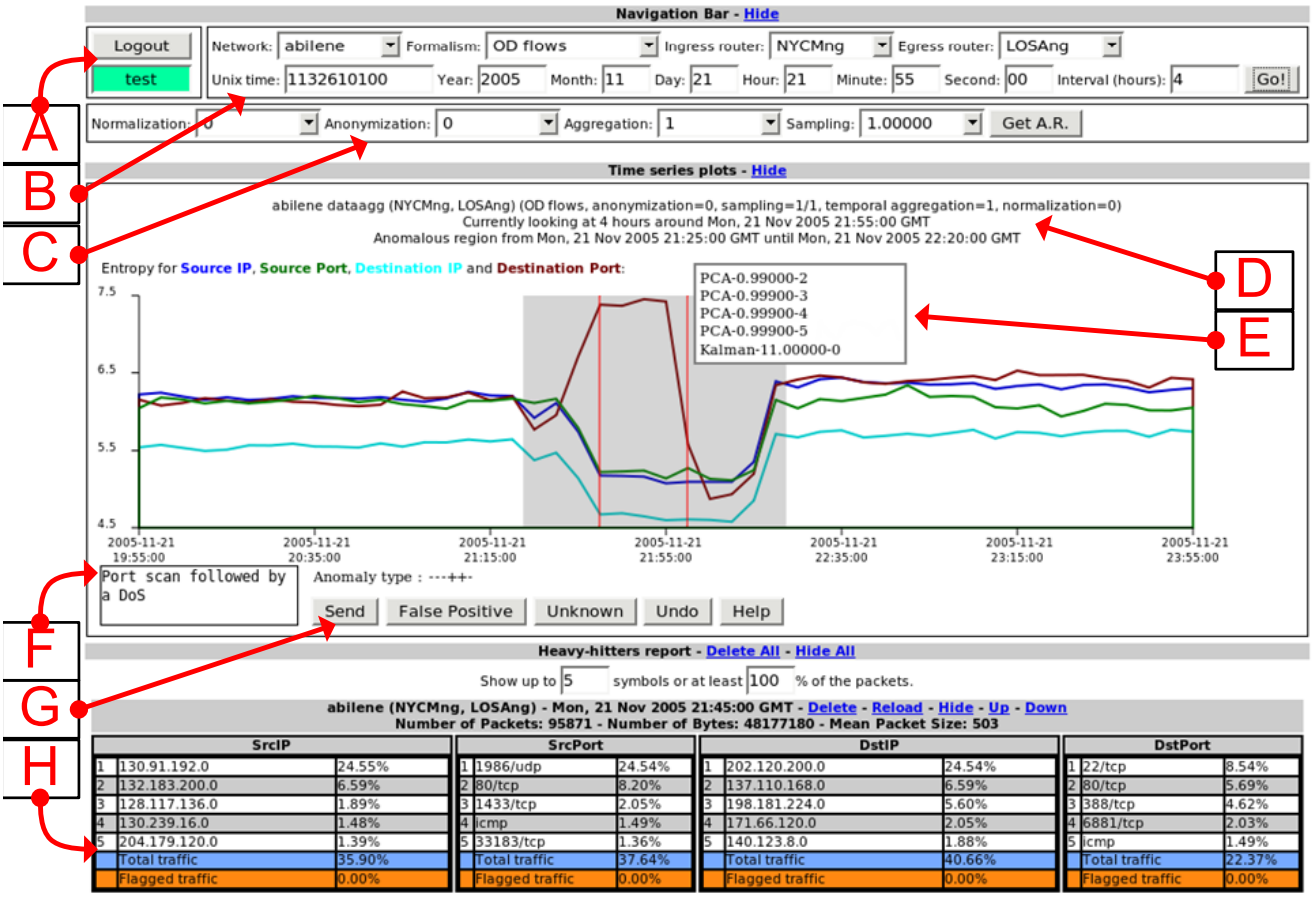

Figure 1: WebClass: A Web-based Anomaly Classification Tool

lies are indicated by the vertical bars on the timeseries plot. Box **D** gives the user information about the portion of the traffic trace that the user is currently viewing. If a user places the mouse over an anomaly then WebClass displays a box containing information how which algorithm and tuning detected it. A sample such box can be seen with **E**.

A user also has the ability to scroll forwards or backwards in the timeseries, or zoom out/in, which is equivalent to inspecting a longer/shorter time interval around any shown anomaly. Both of these features are designed to aid the user in distinguishing anomalies from regular patterns that might not be of interest. A user may also inspect "heavy hitter" [20] reports like the one shown in box **H**. These indicate the top-n most prevalent keys in a given time window for the IP 4-tuple. Lastly, the user may compare any two time windows either to detect changes or to evaluate automated change detectors [7].

Once the user is adequately confident about what label to associate with the anomaly, she can press the appropriate button in box **G**. The same area also has an "Unknown" button, in the event that the user is unable to make a determination regarding the given anomaly. In either case, the user may associate an arbitrary description with the anomaly using box **F**, which can potentially elaborate on peculiar features of this anomaly or reasons for certainty/uncertainty. Once the user has submitted the label by clicking the appropriate button, WebClass presents the next anomaly in the sequence. Finally, in the event that the user made a mistake, she can press "Undo" to unroll the previous commit.

3. CONCLUSIONS

We have argued that manual labeling of traces is an important and unavoidable evaluation methodology for traffic anomaly detectors. This is so partly because anomaly detectors must ultimately succeed on traces from operational networks, and manual labeling is the only method-independent technique to identify the true-positive anomalies that ought to be identified in these traces. In addition, domain experts will be able to identify new or altered anomalies in these traces long before realistic models of their behavior can be constructed for other evaluation techniques.

Despite the importance of manual labeling, there are accuracy concerns due to its reliance no human intervention. We have presented WebClass, a web-based software system that allows users to share, label, and inspect traffic time-series, in order to mitigate challenges associated with manual labeling. In particular, the shared and visually expressive user interface is aimed to improve the ease and reliability of labeling whereas the ability inspect others' labels should allow the community as a whole to converge on some set of true-positive anomalies. We have successfully used WebClass in our research [14, 17], and will release it to the community at large in the hope that it will improve the quality of anomaly detection evaluation.

4. ACKNOWLEDGMENTS

The authors would like to thank Melissa Carroll for useful discussion on supervised learning and manual labeling in other areas of computer science.

5. REFERENCES

[1] Barford, P., Kline, J., Plonka, D., and Ron, A. A signal analysis of network traffic anomalies. In *ACM Internet Measurement Workshop* (Marseille, France, 2002), pp. 71–82.

[2] Cisco NetFlow. `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`.

[3] Estan, C., Savage, S., and Varghese, G. Automatically inferring patterns of resource consumption in network traffic. In *ACM SIGCOMM* (Karlsruhe, Germany, 2003), pp. 137–148.

[4] Garrett, J. J. Ajax: A new approach to web applications. `http://www.adaptivepath.com/publications/essays/archives/000385.php`.

[5] Huang, Y., Feamster, N., Lakhina, A., and Xu, J. J. Diagnosing network disruptions with network-wide analysis. In *ACM SIGMETRICS* (San Diego, CA, USA, 2007).

[6] Kompella, R. R., Singh, S., and Varghese, G. On scalable attack detection in the network. In *ACM Internet Measurement Conference* (New York, NY, USA, 2004), pp. 187–200.

[7] Krishnamurthy, B., Sen, S., Zhang, Y., and Chen, Y. Sketch-based change detection: Methods, evaluation, and applications. In *ACM Internet Measurement Conference* (Miami Beach, FL, USA, 2003), pp. 234–247.

[8] Lakhina, A., Crovella, M., and Diot, C. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM* (Portland, Oregon, USA, 2004), pp. 219–230.

[9] Lakhina, A., Crovella, M., and Diot, C. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM* (Philadelphia, Pennsylvania, USA, 2005), pp. 217–228.

[10] Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., and Lakhina, A. Detection and identification of network anomalies using sketch subspaces. In *ACM Internet Measurement Conference* (Rio de Janeiro, Brazil, October 2006).

[11] Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., and Zissman, M. A. Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. *discex 02* (2000), 1012.

[12] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist. 19*, 2 (1993), 313–330.

[13] Ringberg, H., Roughan, M., and Rexford, J. The need for simulation in evaluating anomaly detectors. *SIGCOMM Comput. Commun. Rev.* (2008).

[14] Ringberg, H., Soule, A., Rexford, J., and Diot, C. Sensitivity of PCA for traffic anomaly detection. In *ACM SIGMETRICS* (San Diego, CA, USA, 2007).

[15] Sekar, V., Duffield, N. G., Spatscheck, O., van der Merwe, J. E., and Zhang, H. LADS: Large-scale automated DDoS detection system. In *USENIX Technical Conference* (2006), pp. 171–184.

[16] Soule, A., Salamatian, K., and Taft, N. Combining filtering and statistical methods for anomaly detection. In *ACM Internet Measurement Conference* (Berkeley, California, USA, October 2005).

[17] Soule, A., Silveira, F., Ringberg, H., and Diot, C. Challenging the supremacy of traffic matrices in anomaly detection. In *ACM Internet Measurement Conference* (2007), pp. 105–110.

[18] von Ahn, L., and Dabbish, L. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2004), pp. 319–326.

[19] Zhang, Y., Ge, Z., Greenberg, A., and Roughan, M. Network anomography. In *ACM Internet Measurement Conference* (Berkeley, California, USA, October 2005).

[20] Zhang, Y., Singh, S., Sen, S., Duffield, N., and Lund, C. Online identification of hierarchical heavy hitters: Algorithms, evaluation, and applications. In *ACM Internet Measurement Conference* (Taormina, Sicily, Italy, 2004), pp. 101–114.