Annotating Network Trace Data for Anomaly Detection Research

Andreas Löf
University of Waikato
Hamilton
New Zealand
Email: andreas.lof@cs.waikato.ac.nz

Richard Nelson
University of Waikato
Hamilton
New Zealand
Email: richardn@waikato.ac.nz

***Abstract*—Anomaly detection holds significant promise for automating network operations and security monitoring. Many detection techniques have been proposed. To evaluate and compare such techniques requires up to date datasets, useful truth data and the ability to record the outputs of the techniques in a common format. Existing datasets for network anomaly detection are either limited / aged or lacking in truth data.**

This paper presents a new annotation format allowing network datasets to be annotated with arbitrary event data. Use of the new format is demonstrated in a method to create new datasets that retain more information than a simple network capture. The supporting tools for the annotation format allow for incorporating events from multiple different sources. The ability to record and share network data and detected anomalies is a key component in moving anomaly detection research forward.

***Index Terms*—anomaly detection, passive data collection, annotations**

I. Introduction

Anomaly detection is the process of automatically finding unusual occurrences in a sequence of network measurement data. Anomalies in network data can lead to the identification of events that have occurred. These anomalies can be hidden by the large amounts of the very data that modern network management systems generate trying to discover them. Automating the process of detection can make the process faster and more reliable. The hope is that this will, in turn, lead to faster diagnosis and appropriate resolution of the causes of the events.

Events may be operational events such as faults, configuration errors or unusual network behaviour or they may be security events such as probes or malicious payloads. All such events can have major impact on the network service and ultimately users. Fast resolution can mitigate against significant service degradation, malicious damage and cost.

Because of this potential, many detection techniques have been proposed (e.g. [3]). To move from proposed techniques to valid network too anomaly detection techniques need to be carefully evaluated. Selecting and improving techniques involves comparison which requires common input data. Evaluating the metrics requires some idea of the actual events that are contained within the input data or the ground truth.

This paper is concerned with establishing a system for providing useful input data for common evaluation of anomaly detection techniques across a range of researchers. In particular we are interested in network trace data collected for behavioural anomaly detection. This has the potential to detect events originating both within and outside the network, as well as detect previously unknown event types. Packet content data is used for signature matching as deployed in Intrusion Detection Systems (IDS) and active measurement data is easily shared as it causes no privacy concerns. Therefore these types of data are not considered in the paper.

Section II defines the concepts required of data to be used for anomaly detection research, discusses the sources of such data and develops the requirements for datasets. Section III discussed existing datasets and data formats used in anomaly detection. A new data format that meets the identified requirements is introduced in section IV. An example dataset collection using the new data format that uses multiple sources for improved truth is described in section V and we conclude in section VI.

II. Concepts

A. Network Data

Collecting network trace data is passive measurement and is well understood although not always well performed. There are many formats for network traces but they have common characteristics. They only store packet data, normally metadata is limited to per-packet information such as a timestamp and possibly the port on which it was captured. The packet contents may be complete or truncated. There are no facilities for appending annotations on a per packet or any other basis.

For anomaly detection research, datasets need to be relevant and up to date. Network traffic types are changing on an ongoing basis as new applications and network devices are introduced an network speeds increase. These changes in turn result in changes of the normal use patterns within which anomalies need to be detected. Security anomalies are changing quickly too as new vulnerabilities are discovered and exploited.

Datasets also need to be shareable. To compare techniques, different researchers need to run their algorithms against the same data. In practice the best way to do this is to make the dataset publicly available. However, making real world

978-1-4799-3784-4/14/$31.00 ©2014 IEEE

datasets publicly available conflicts with the need for privacy of network users.

There are two main ways to create datasets: synthetic (simulated) traffic, collecting real network traffic and semi-synthetic, a mix of the previous two. A synthetic dataset is created by using a network traffic generator to create both anomalies and background traffic. A dataset created from real network traffic has had network data captured from one or more network links. Semi-synthetic traffic mixes real world network traffic with known synthetic anomalies. Each method suffers from different drawbacks.

1) Synthetic Datasets: Since all of the traffic is introduced by the researcher when creating a synthetic dataset, only the intended anomalies will be present. This allows the creation of complete truth data for the dataset; an advantage that can be achieved no other way. The second advantage of synthetic datasets is that since there are no actual network users there are no privacy concerns and so the complete dataset can be publicly released and easily shared.

However, generating a synthetic dataset that is representative of real world network is impractical because of the difficulties with creating background traffic that appears authentic yet does not contain any undetected anomalies. The anomalies must also be realistic. Any bias in, or artefacts from, the process of creating background traffic or anomalies will either bias anomaly detection techniques developed with the dataset or provide incorrect results when techniques are evaluated. Efforts to improving the accuracy of the representation of the dataset increase the cost and effort required so new datasets become infrequent.

2) Real World Data: There are many examples of real world traces that are publicly available [14] [20] . Real world datasets are relatively quick and easy to collect. Permission of the network operator is normally the main impediment. Although traffic mix and characteristics vary from network to network, each trace is at least representative some set of users with all the complexity that implies. These traces do not suffer from any artefacts from the capture process, assuming that sufficient resources are available at the capture point and the capture software is bug free.

The disadvantages are the converse of synthetic dataset advantages, they lack truth data and privacy concerns make them difficult to share without significant anonymisation. There are many examples of such data being collected and used for anomaly detection research but being held privately so preventing comparison of results by other researchers e.g. [7] [8] [16].

3) Semi-Synthetic Datasets: To combine the advantages of realistic traffic with known truth data some researchers have attempted to combine known synthetic anomalies with real world background traffic. Making the synthetic data integrate well with the real network data that it is mixed with is difficult, particularly in choosing the types and volumes of anomalies. Also the amalous traffic, being synthetic, is unlikely to perfectly represent real world anomalies and so will bias any evaluation. Finally the truth data based on the synthetic traffic can only be partial because of the possibility of anomalies occuring the in the background traffic.

B. Truth

There are many different ways to evaluate anomaly detection methods. They all have one elements in common; they want to establish how well a method behaves given a specific metric. All of these methods are dependant on having various datasets to evaluate them against, consisting of network traces and truth data. Truth data is also used for training of machine learning anomaly detection techniques and the calibration of parameters in other techniques.

Frequently the term ”Ground truth” is used to denote the underlying reality. This can refer to events that occur on the network or the anomalies they cause within the measurement data. When anomalies are identified they are recorded by placing an identifying *label* into an *annotation*. There are many possible sources for labels including human experts, intrusion detection systems or anomaly detection algorithms.

Having annotations leads to the concept of *correctness* which is how closely a set of annotations approach the truth of the anomalies present in the data. Errors may be present due to incorrect transcription due to software bugs and human error, loss of precision in timestamps or network addresses. For many sources errors can also be due to mistakes in estimation of the anomalies (so called false positives or false negatives) or the inability of a source to detect particular types of anomaly.

Ground truth of the anomalies present in network datasets is difficult to achieve because of the size of complexity of the data. It is only possible to achieve complete truth in synthetic datasets, however due to the other problems with such datasets, other ways of finding truth data are important. For real world data, truth can come from human labelling or algorithmic processing, but both of these sources are fallible.

There are no perfect automatic anomaly detection systems. The pursuit of such a thing is the point of the research these datasets support. Signature detection systems can reliably detect all occurrences of a given signature but the patterns they look for require constant updating and a pattern may occur naturally in data that is compressed or random.

Expert Labelling is suitable for small datasets but rapidly becomes infeasible when the size of the dataset grows. It can be very difficult to manually distinguish an anomaly from normal data.

It does not appear that there have been any formal studies on expert performance in the field of network anomaly detection, but the findings from research performed on expert performance in other fields suggests that they are not infallible. In fact studies such as [2] and [17] have shown very low reliability for human experts.

In network anomaly detection, a dataset will commonly consist in the order of millions of flows and hundreds of millions of packets. For example in the trace set Waikato 8 [20], 2011-06-02 contains 349,717,701 packets and 13,837,822 flows. In order to establish an exhaustive set of labels, each and every instance would need to be reviewed by (preferably multiple) labellers, ideally a combination of expert and non-experts would be used to achieve a high level of agreement between the labellers. Without a high level of agreement, the produced labels would not be reliable.

Our conclusion is, expert, or non-expert, exhaustive labelling is impossible to achieve for a modern network trace due to the volume of the data and the number of labellers needed to establish reliable labels.

C. Annotations

1) Requirements on Annotations: The primary purpose of trace annotations is to facilitate communication and sharing of information between both researchers and end users. To be able to accommodate this, they need to be expressible and flexible. To be useful for information sharing, the annotations need to provide both accuracy for identifying specific events and allow for a wide range of descriptions of what the annotation signifies. To be useful to researchers, the annotations also need to be able to show how much confidence the researchers have in a particular annotation. Also, the annotation format should be extensible in such a way that different versions of the format can co-exist without any difficulties. These simple use cases can be used to derive the following requirements:

1) An annotation should be able to represent events of different scopes (per packet, per flow, flows and/or packets over a period).
2) The annotation should uniquely identify the packets and/or flows in the original trace that comprise the anomaly. This can be achieved through timestamps and network addresses.
3) Annotations from multiple sources should be able to be stored and compared. The format should be extensible to allow new sources in future.

III. Previous Work

A. Datasets

DARPA sponsored a sequence of three intrusion detection evaluations performed by Lincoln Laboratories. Three corresponding synthetic datasets were released in 1998, 1999 and 2000. The evaluations were focusing on both host and network intrusion, but only the network aspect of them is relevant here.

These datasets exemplify perfectly the advantages and disadvantages of synthetic datasets. They have been and are still widely used, the DARPA 1999 dataset [6] receiving over 60 citations per year on average between 2007 and 2012 according to Google Scholar. Examination of a subset of these papers shows that the majority of these papers are using the datasets for evaluations. This is despite new web services such as YouTube, protocols such as bittorrent and security threats such as Slammer and Stuxnet all appearing after these datasets. Further, the datasets are often regarded as flawed. McHugh [10] wrote a critique that is mainly focused on the DARPA 1998 evaluation. The critique focuses on both issues with the dataset and how the evaluation itself was performed.

The specific flaws are not relevant here, the important thing is that the most popular datasets for anomaly detection evaluation, created with the backing of DARPA, have a range of flaws and are now over a decade old. McHugh [10] documents some of these flaws, the most well known one being the different TTL between attackers and any other traffic. As a result, new datasets are badly needed.

Real World datasets are much more common, however very few (e.g. [9] [19] [18]) have any truth data. In all cases the truth is limited to a small number of well defined anomaly types. Sperotto et. al.'s work [18] is interesting because it takes a similar approach to what is outlined in this paper, using honeypots as the source for flow annotations. The dataset itself is however not publicly accessible and can only be obtained by contacting the authors directly. These dataset are inadequate since they only focus on specific types of events or are not open.

B. Annotation Formats

ADMD [5] has been developed with the purpose of sharing scores for anomaly detection methods. It is an XML based format. ADMD supports annotations of packets or of slices. Packets are identified by a hash of the packet. A slice, as used by ADMD, is a complete or partial network flow between two hosts. The slice is identified by source and destination IP addresses, protocol, source and destination ports, and a duration. Packet and slice (or flow) are the only scopes supported - it is not possible to annotate events that occur across multiple flows. Further each annotation file only supports a single scope limiting the ability to compare annotations from multiple sources that might appear in different scopes.

ADMD allows for the addition of a description of the annotation file as well as information about the algorithm that created the annotations, including parameters. This is potentially useful additional meta data but limits the extensibility of the format.

Microsoft Network Monitor [11] has support for annotations when it is using its own custom annotation format. However, it is not possible to annotate a captured trace without saving and reopening it beforehand. The format only supports annotating packets and it has to be a manual procedure. This means that it is impossible to add annotations that are not tied to a specific packet such as data volume anomalies. It is also impossible to automatically label the trace, or read back the labels. These problems make it unsuitable for anomaly detection research.

WebClass [13] consists of a web application and a supporting SQL database to manually label and classify datasets. The system focuses on labelling existing datasets at a timeseries or aggregated level. Individual packets or flows are not supported. As such, it is of limited usefulness for comparing automated anomaly detection methods, but provide a valuable tool to verify automatically labelled data.

Formats such as sFlow, NetFlow, IPFIX are not used to annotate network traffic, rather to provide information about the traffic or specific flows. As a result they lack provision to express a measure of confidence for any particular event, even if they support expressing events at all.

IV. New Annotation Format

We have developed a new annotation format with the goal of fulfilling the requirements identified in section II-C1. To improve processing speed and uncompressed storage size, it is a binary format, with all data stored in network byte order. This format supports a time resolution as fine as the trace format, the ability to identify trace wide events, hosts, services, flows, and packets and is arbitrarily extensible. An implementation of the necessary functions to read and write the data has been released as open source software and is available at http://research.wand.net.nz/andreasl.php.

Each different record is identified by an IANA enterprise number. This number can be used to identify the implementation of the format that created that specific record, which also means that the records should conform to the format as defined by that organisation. This would allow different institutions to add additional information in their own implementations of the records without explicitly breaking any existing design. The University of Waikato IANA enterprise number should be used if no modifications are made to the format.

A. Format Outline

Figure 1a shows an annotation as it will be stored in the file. Each actual annotation is encapsulated in a container that identifies the content type, the enterprise that created the annotation and the total length of the record. An annotation, as encapsulated by the container, actually consists of three parts; the annotation, the identifier and the description. Each container record can be of variable of variable length, depending on the contents, and is written sequentially to disk in the binary file format.

The annotation part defines the event scope, the timestamp, a confidence number and a divisor. In addition to this, it also tells what type of identification is used for the annotation. The confidence number defines the belief held in the annotation's accuracy and the divisor is used when serialising annotations to disk in order to not lose precision.

There are currently four supported identification types. The most descriptive is Ethernet, the second most is Service, then Host and finally None. Figure 1b shows how these identification types are related and what each contains. The Ethernet type is used to identify packets and flows captured over an Ethernet link. It is possible that an additional identification type for ATM links, or not using the link layer addresses at all might be added in the future.

Normally, the descriptions for an annotation is made in the style of an semantic URL. The URL should directing the user of the annotation format to a web page with more information about the specific annotation. A semantic URL is structured in such a way that information is conveyed by the URL itself. The main benefit of having a semantic URL is that it still conveys information about the annotation even though the resource it points to might be missing.

Figure 2 shows an example semantic URL used to label the annotations. This URL consists of several parts. **`www.wand.net.nz`** tells us which organisation created the annotations. The name of the dataset the annotation is associated with is **`waikato8`**. The source of the annotation if **`snort`** and the Snort rule with the **`sid 128`** was triggered. The final part of the URL shows us that revision 1 of the rule was used.

The format introduced here fulfils all of the requirements established in Section II-C1. The format supports multiple scopes of events, both on a per packet/flow basis, as well as on more aggregated events. The different way of expressing identification information combined with high timestamp accuracy allows the format to unambiguously identify a specific packet/flow, thus allowing for high accuracy. Combining multiple sources of annotations is trivial as the format itself does not distinguish between any originating method for the annotations recorded, it is rather up to the format to identify itself using the semantic URLs. Finally, the format is open and designed in such a way that it can easily be extended in the future without breaking any backwards compatibility.

V. Collecting a new Dataset

To demonstrate the advantages of the annotation format supporting multiple sources a new dataset was captured using a labelling during capture approach.

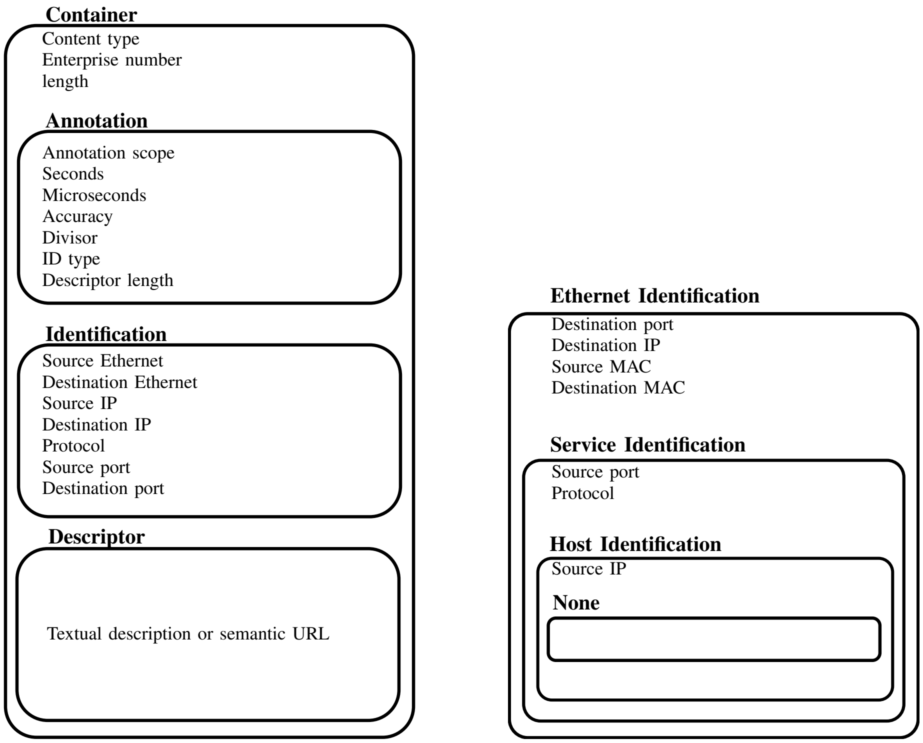
The dataset itself was captured using best practice passive measurement techniques. A measurement point using an Endace DAG 4.3GE card was attached to a Span port of the University of Waikato edge router, outside the firewall. The capture software was libtrace [1] version 3.0.12 and WDCap[21] version 3.1.12. The measurement timestamps were GPS synchronised and the capture ran from from 2011-06-02 to 2011-06-20. The data rate averaged 2.3 MB/s with a peak throughput of 31.7 MB/s and inspection of the logs from the DAG card showed that no packets were dropped.

As the data was obtained from a production network the trace was anonymised during capture. The capture software used dynamic snapping to identify the end of the transport headers (including options) then each packet was truncated four bytes after that point. All IP addresses were anonymised using Crypto-PAn [4]. The World IPv6 Day occurred during the capture so some IPv6 traffic was observed.

This dynamic snapping allows for the complete capture of link layer headers, network headers and TCP, UDP and ICMP headers.

A. Labelling During Capture

The labels applied during capture were supplied by the output of two intrusion detection systems running simultaneously with the capture. The system was arranged so that these systems received complete packets with anonymised addresses so that the annotations match the addresses stored in the trace file. The two systems ran Snort [15] and Bro [12]; chosen because they are mainstream deep packet inspection



(a) Overview of the annotation format

(b) The identification tuple

Fig. 1: The annotation dataformat

`http://www.wand.net.nz/annotations/waikato8/snort/sid/128/revision/1` .

Fig. 2: An example semantic URL

based network intrusion detection systems that see large scale deployment in many different organisations.

Both systems were monitored by watchdogs that restarted them when the systems terminated prematurely. This proved to be necessary for both Bro and Snort, but for different reasons. Snort was unable to restart gracefully when the binary rules were updated, instead Snort attempted a restart and then aborted. Bro suffered from stability issues and would sometimes segfault.

Unfortunately, this problem led to some of the captured data not being analysed by the systems, but by using buffering of the data and the watchdogs we were able to mitigate this as much as possible.

Snort was running both the Sourcefire VRT and the Emerging Threats open rule sets. At midnight each night the signature files were updated to the latest versions and snort was restarted. The Snort version used during the data capture is 2.9.0.5 and the VRT rule sets starting with the 2011-04-28 release and ending with the 2011-05-17 release. The EmergingThreats rules were also updated on a daily basis at midnight.

The Bro installation was configured using the scripts that was distributed with the Bro source. There were no attempts made to write custom detectors for Bro, but rather to see what it could detect using the policies and detectors distributed with the system.

Custom matching software was written to take the output of each system and convert into the new annotation format. This software had to identify the original packet or flow that triggered the output. The Snort unified2 output format gives complete network addressing information and, by reading the Snort output, a timestamp matching that of the original packet and so the identification is straightforward. Bro does not provide Ethernet MAC addresses in its output, nor accurate timestamps. Further it has a detection delay that can vary from five to thirty minutes. Consequently the matching process can only identify trigger flows.

Although IDS systems are different from anomaly detection there is often some commonality in the underlying events. The signatures are lost from the dataset when the packets are truncated. The dataset described here shows that it is possible to retain useful extra information in a common format that can easily be compared with the outputs of different anomaly detection systems. The dataset is available as Waikato VIII from [20].

VI. Conclusion

To make progress in the field of network anomaly detection a basis for evaluating detection techniques is required. To date this has most commonly been performed using synthetic datasets with complete truth data, but this is fundamentally flawed as it produces techniques biased by the abstractions that are used in producing the synthetic traffic.

Using real world datasets for evaluation is limited by the lack of complete truth data. Consequently the only way forward is to record the imperfect outputs from a range of different sources and use a comparison process to select the best anomaly detection techniques. The comparison process has not been addressed in this paper, but the first requirement is for a common, extensible, format for anomaly detection outputs so that it might become possible.

We present here a new annotation format to allow labels from an arbitrary range of sources to be recorded against network trace data, as well as a freely available new dataset with automatically generated labels from two popular Network Intrusion Detection Systems. This is accompanied by tools to allow the writing and reading of the annotations. The tools and formats can scale well to large datasets and multiple sources of inputs. Collecting traces with labels from external sources during capture allows extra information to be recorded that would normally be lost during anonymisation.

Using these tools provides the basis for collecting new real world network datasets with labels from multiple sources. This enables the traces to retain more information than the trace files alone. Putting annotations from different anomaly detection techniques in a common format provides a basis for comparison. This is the first step in moving towards consistently evaluating such techniques against up to date real world datasets.

A. Further Work

With this data format the next step is to find a comparison process that allows us to approach the ground truth using the inputs of multiple detector types. This will require practical implementations of a range of anomaly detection algorithms. Then we plan to apply data fusion methods to the outputs to create an approximation to the truth.

References

[1] S. Alcock, P. Lorier, and R. Nelson, "Libtrace: a packet capture and analysis library," *SIGCOMM Computer Communications Review*, vol. 42, no. 2, pp. 42–48, Mar. 2012.

[2] C. F. Camerer and E. J. Johnson, *Research on Judgment and Decision Making*, W. M. Goldstein and R. M. Hogarth, Eds. Cambridge University Press, 1997.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[4] 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*, vol. 46, no. 2, pp. 253 – 272, 2004.

[5] K. Fukuda, B. Abry, P. Borgnat, F. Claffy, K. Claffy, and E. Aben, "ADMD," 2008, last accessed on 2014-06-22. [Online]. Available: http://admd.sf.net

[6] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579 – 595, 2000, recent Advances in Intrusion Detection Systems.

[7] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 1, 2009.

[8] M. Mahoney and P. Chan, "Learning rules for anomaly detection of hostile network traffic," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, Nov. 2003, pp. 601–604.

[9] MAWI, "MAWI working group traffic archive," n. d., last accessed on 2013-02-06. [Online]. Available: http://mawi.wide.ad.jp/mawi/

[10] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[11] Microsoft, "Microsoft Network Monitor," 2010, last accessed 2013-09-09. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=4865

[12] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, ser. SSYM'98. Berkeley, CA, USA: USENIX Association, 1998, pp. 3–3.

[13] H. Ringberg, A. Soule, and J. Rexford, "Webclass: adding rigor to manual labeling of traffic anomalies," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 1, pp. 35–38, 2008.

[14] RIPE, "RIPE data repository," n. d., last accessed on 2013-03-13. [Online]. Available: https://labs.ripe.net/datarepository

[15] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX conference on System administration*, ser. LISA '99. Berkeley, CA, USA: USENIX Association, 1999, pp. 229–238. [Online]. Available: www.snort.org

[16] T. Seppälä, T. Alapaholuoma, O. Knuuti, J. Ylinen, P. Loula, and K. Hätönen, "Implicit malpractice and suspicious traffic detection in large scale ip networks," in *Fifth International Conference on Internet Monitoring and Protection (ICIMP)*, May 2010, pp. 135–140.

[17] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 254–263.

[18] A. Sperotto, R. Sadre, F. Vliet, and A. Pras, "A labeled data set for flow-based intrusion detection," in *IP Operations and Management*, ser. Lecture Notes in Computer Science, G. Nunzi, C. Scoglio, and X. Li, Eds. Springer Berlin Heidelberg, 2009, vol. 5843, pp. 39–50. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04968-2_4

[19] University of Massachusetts, "UMass trace repository," n. d., last accessed on 2014-06-22. [Online]. Available: http://traces.cs.umass.edu/

[20] Waikato Internet Traffic Storage, "WITS Website," n. d., last accessed on 2014-06-22. [Online]. Available: http://www.wand.net.nz/wits

[21] WAND Network Research Group, "WDCap," n. d., last visited on 2014-06-22. [Online]. Available: http://research.wand.net.nz/software/wdcap.php