A Framework of Evaluation Methodologies for Network Anomaly Detectors

Xin Zhao, Yekui Qian, Changsheng Wang
Air Defense Forces Academy, Zhengzhou, 450052, China
zhaoxin.remerci@gmail.com

***Abstract*—Anomaly detection has been a field of intensive research over the last years. Along with that several works to evaluate anomaly detectors have been proposed. In this paper we argue four properties regarding ideal evaluation methodologies that cannot be answered by single current evaluation technique employed today. We therefore present an framework of an evaluation methodology that leverages traces from operational networks, simulation and emulation to satisfy the four properties.**

Keywords-emulation; framework; anomaly detector; netflow; flow matrix

I. INTRODUCTION

Network security has always been a hot topic to network users and managers. Nowadays new threats or mutations of existing ones appear at a very fast rate. It is therefore not surprising that there has been an explosion in research on network anomaly detection in recent years [1], [2] [3]. With the flurry of anomaly detection papers, several works to validate and compare the proposed solutions have been proposed [1][4]-[7]. The data traces used in those works are captured from operational networks, simulations and emulations.

Generally, an ideal evaluation method for network anomaly detection requires credibility and fidelity of the data traces and the ability of entire control and reproducibility the experiments. However, it's difficult to find a single evaluation technique employed today satisfying all those ideal conditions.

A general way to evaluation in the anomaly detection domain is using data traces taken from operational networks. But it's difficult to get such data traces since network traffic data is very privacy-sensitive, especially when it contains payloads or IP addresses. Strict laws have been issued in many countries that prohibit the public sharing of network data. Moreover, the available datasets are usually not labeled, i.e., the instances of malicious or disruptive activity in the trace are not known beforehand and need to be added by a human expert.

Simulation has played a vital role in networking research over years. It has unmitigated access and control over networks to evaluate some aspects of complex ideas. However, simulation often simplifies some parts of a real environment. The full interaction between all parts is difficult to model or simulate as it requires such a detail that it is not feasible. Therefore the synthetic data generated cannot substitute for their counterparts in real operational networks.

Emulation goes half way between simulation and real world testing, by modeling some parts and running live other parts. It provides more fidelity than simulation. But emulators have in common that they try to address the problems of scaling, management and reproducibility. Moreover emulation cannot always be a substitute for real world experiments.

We therefore present a framework of evaluation methodologies that leverages traces from operational networks, simulation and emulation to satisfy the four properties of ideal evaluation methods.

The rest of this paper is organized as follows. Section II we briefly present the most important approaches to evaluate anomaly detectors that have been published so far; In Section III we enumerate four requirements of a thorough evaluation of anomaly detectors that are not always met by currently employed evaluation methodologies. In Section IV we propose the framework of an ideal evaluation method. We conclude in Section V.

II. RELATED WORKS

Recent work has proposed using ways such as Principal Component Analysis (PCA) and subspace-based analysis [1], etc. over global traffic matrix statistics to effectively isolate network-wide anomalies including worms, DDoS attacks, and IP scans.

Captured datasets are typically very privacy-sensitive. Although there is a large body of literature on anomaly detection techniques only few authors share their data within the community.

One notable data source is the packet traces from the WIDE backbone network [8] that are maintained by the MAWI working group. The repository includes a variety of anonymized traces from 1999 to 2009 from six different measurement points. Abilene [9] and Géant [10] are the other two backbone networks that provide researchers with anonymized NetFlow traces upon request. Traces from these networks have been used for evaluation in [1][11][12].

Labels that identify when an anomaly has happened are, however, absolutely required for evaluating whether an anomaly detection system is accurate or not. Ringberg et al. [13] have recognized the problem and proposed a tool called webclass that provides functionality to store and compare labels that have been assigned by different domain experts to a trace.

A more recent advance in dataset simulation and emulation is due to Sommers et al [14]. The authors present Harpoon, a tool for generating representative benign packet

Published by Atlantis Press, Paris, France.
© the authors

traffic at the IP flow level. Harpoon generates application-independent UDP and TCP traffic.

Mirkovic et al. [15] have developed a tool called AProf that extracts the traffic caused by Denial of Service attacks from packet traces based on connection heuristics. The extracted attack traces are used as benchmarks for experimentation in the DETER testbed [16].

III. REQUIREMENTS

In this section we list four properties of an ideal evaluation methodology for network wide anomaly detectors.

A. Credibility

"Ground truth" in the context of network anomaly detection requires a complete list of all anomalies existing in a given data set. While the requirement itself might seems obvious, it is much less clear how to obtain this ground truth. Identifying the true-positive anomalies requires combing through vast amounts of data that are sometimes of poor quality due to data-reduction techniques such as sampling. The challenges of obtaining high-quality data have led to many compromises in the evaluation of anomaly detectors, which in turn leads to "partial" ground truth.

In order to quantify the accuracy of a detector it is necessary to first identify a set of "true" anomalies that ought to be found by the detector. This set must obviously be identified by a procedure that is independent of the detector being evaluated.

By far the most common way to accomplish this identification is to rely on manual labeling of traces by domain experts or automated injection of anomalies into traces. In the manual labeling procedure, the human domain expert inspects a trace and certifies some events as being true-positive anomalies. The detector is then evaluated based on its ability to identify this set of events. Anomaly injection leverages models of anomalies in order to introduce them into traces taken from operational networks. It cannot guarantee accurate FPP and FNP measurements due to its reliance on existing traces. That is, these existing traces presumably came with an unknown number of anomalies. Neither automated algorithms nor human domain experts can identify all these anomalies with complete confidence.

B. Fidelity

Fidelity to "real" networks is important. There are several dimensions to fidelity: (1) the number of nodes, (2) realism, i.e., reproducing real router and end-system behavior, and (3) realistic heterogeneity of hardware and software, and (4) a realistic mix of link bandwidths and delays.

Fidelity has costs for the purchase, maintenance, and operation of hardware and software. The hardware-related costs of a testbed increase linearly with the number of nodes, and faster than linearly when the cost of switches is considered since some switch ports have to be used for inter-switch bandwidth. A central aspect of the experimental science on testbeds is to construct idealized abstractions of the real Internet with enough fidelity for specific experiments.

Some experimenters will want to run experiments that require more nodes than are available. It is possible to run multiple virtual nodes on each physical node to enable such experiments, but virtualization introduces artifacts which must be considered when evaluating experimental results.

C. Experimental Control

There are important questions about the effectiveness of network anomaly detectors that cannot be answered without having complete control of the entire evaluation experiment. Having complete control requires that one has the power to change the location, magnitude, and type of individual anomalies as well as for the background traffic.

For data traces from operational networks, manual labeling would be unable to provide such control due to the fixed nature of the underlying trace. While automated anomaly injection into existing traces only provides partial control over anomaly itself and not the background traffic.

Complete control over the evaluation experiment is also necessary in order to train and test the detector on clean data. That is, an evaluation methodology that leverages existing traces should guarantee that all anomalies in those traces have been identified. Only through simulation can one ensure that all anomalies in an evaluation trace are known.

D. Reproducibility

The ability to reproduce an experiment is a central characteristic of the scientific method. A researcher might wish to (1) verify published results by evaluating the same algorithm on the same data, (2) investigate the robustness of a published algorithm by applying it to different data, or (3) compare a novel algorithm against the published one by using the same data.

The problem of reproducibility in network anomaly detection is particularly dire due to a general lack of public data sets. The problem is exacerbated because most detectors are evaluated using traces from operational networks, and there are numerous valid reasons why such traces cannot be shared with the community. A significant fraction of these traces come from commercial networks, which means that both the data and software is likely proprietary. Even for traces from educational networks, there will be privacy concerns. Furthermore, there are stringent laws that restrict the distribution of certain types of telecommunications data. Finally, traces can often be on the order of many terabytes and it may be practically infeasible to share them. Traces can grow to this size because modern networks carry vast amounts of traffic and therefore any algorithm that is claimed to be able to operate in an online setting should be evaluated on representative traces.

IV. FRAMEWORK

The evaluation framework presented is for the anomaly detection and classification method ODC [17] that we have previously proposed and future works. Due to the virtues and shortcomings of each single evaluation method, it's obvious that an anomaly detector will be thought more valuable by testing and verifying it through more methods, as shown in figure 1, to real networks. The common models of several methods can be reused with minor modification.

Published by Atlantis Press, Paris, France.
© the authors

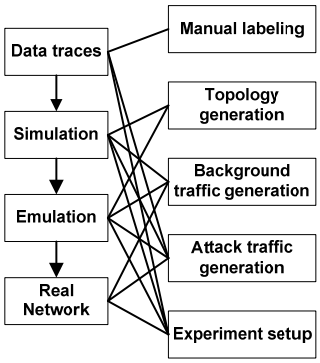

Figure 1. Framework of evaluation methods for anomaly detectors

A. *Data traces*

Data traces with manual labeling or attack traffic injection can be directly used for evaluation of anomaly detectors. It's simple and the first method in our framework in which we use four weeks of sampled NetFlow data for the period March 1, 2010 to March 28, 2010, collected from all access links of two backbone networks: Abilene and Géant.

Abilene is the Internet2 backbone network, connecting over 200 US universities and peering with research networks in Europe and Asia. It consists of 9 Points of Presence (PoPs), spanning the continental US. We collected sampled IP-level traffic flow data from every PoP in Abilene. Sampling is periodic, at a rate of 1 out of 100 packets. Abilene anonymizes destination and source IP addresses by masking out their last 11 bits.

Géant is the European Research network, and is twice as large as Abilene, with 22 PoPs, located in the major European capitals. Data from Géant is sampled periodically, at a rate of 1 every 1000 packets. The Géant flow records are not anonymized.

Both networks report flow statistics every 5 minutes; this allows us to construct traffic timeseries with bins of size 5 minutes. The prevalence of experimental and academic traffic on both networks make them attractive testbeds for developing and validating methods for anomaly diagnosis.

The download flow data were sampled in both backbone networks and anonymized in Abilene networks. This could lead to potential bias when evaluating anomaly detectors, which should be considered and is the reason of using successive evaluation methods.

The data traces should be manual labeled or anomaly injected before using for evaluation. Therefore, anomalies in the data were identified using available manual labeling methods: visual inspection of timeseries and top-n queries directly on the flow data. Anomaly injection needs attack traffic generation. We use MACE [18] as the malicious traffic generator. MACE is a modular attack composition framework that consists of three primary components: (i) exploit, (ii) obfuscation, and (iii) propagation, as well as a number of functions to support interpretation, execution, and exception handling of attack profiles.

B. *Simulation*

Due to the controllability, reproducibility and scalability of simulation, researchers have found it valuable in evaluating anomaly detection systems over the last years [19][20]. Currently the available simulators include NS [21], OPNET [22], GNS [7], and so on. We use GNS3 as the simulation method in the framework.

GNS3 is a multi-platform, open-source Graphical Network Simulator. GNS3 allows the emulation of complex network topologies by emulating many Cisco IOS router platforms, IPS, PIX and ASA firewalls, and JunOS with the help of Dynamips and Dynagen. Dynamips is the core program behind the emulation process and the Dynagen tool runs on top of it to create a user-friendly, text-based environment. GNS3 provides the graphical front-end for Dynagen, so that users can create the topologies in a graphical and user-friendly environment. GNS3 also allows the emulation of ATM and Frame Relay switches, enables packet capture using Wireshark.

The topology of the simulation network is created in NS2 format. Each backbone node was paired with an external interface. The external interfaces represent traffic to network resources external to the simulation network. NS2 is a TCL based scripting language which can use iterative control structures.

MACE can use traffic generation tools as complements used to generate legitimate (benign) background traffic. Harpoon reproduces network traffic in an application-oblivious manner [23]. Numerous application-aware traffic generators like SURGE produce workloads to stress-test web servers [24]. The LTProf tool produces legitimate traffic models that describe communication between a set of active clients and a network that is the target of a DDoS attack.

The experiment setup generator receives as input (1) AS-level and edge-network topologies from the topology library, (2) legitimate and malicious traffic models generated by the MACE. It glues these elements together into an ns file containing topology specification and a collection of Perl scripts, one for each attack from the list and one script for legitimate-traffic- only testing.

C. *Emulation*

The emulation network in this framework is like DETER [5]. The DETER testbed allows security researchers to replicate threats of interest in a secure environment and to develop, deploy and evaluate potential solutions. The emulation network topology is shown in figure 2, including three cluster of experimental PC nodes, with a common control plane. There are roughly 50 nodes in total, currently. These nodes are interconnected by a "programmable backplane" of high-speed Ethernet switches, trunked to form a single logical switch. Each experimental PC has four experimental interfaces and one control interface to this switch.

© the authors

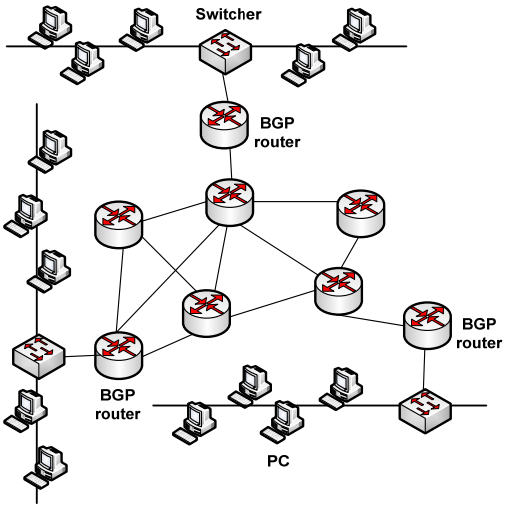

Figure 2. Topology of the simulation network

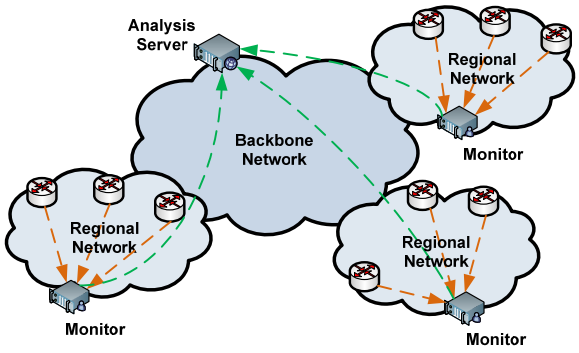

Figure 3. Distributed monitoring system of the Testbed

As shown is figure 3, in each cluster there is a monitor capable of generating NetFlow about passing traffic (both directions of the link are monitored without sampling or packet loss) in the form of compressed nfdump files. The traffic matrix is generated at analysis server which receives compressed nfdump files at 1:00 a.m. each day. Further, precomputed time series are provided from the whole measurement period. This includes volumes of flows, packets and bytes per 5 minutes interval differentiated by a protocol (TCP, UDP, ICMP).

V. CONCLUSION

This paper addresses several issues related to experimenting with current methods and enumerates four properties of an ideal evaluation method and proposed a general framework for experimenting with network anomaly detection methods. To this end, the framework consists of various functions and methods used for anomaly detection.

ACKNOWLEDGMENT

The authors would like to thank Brian Cashman for valuable help in the process of downloading Abilene data traces. This work was part of the National Defense Foundation of China.

REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in ACM SIGCOMM ’04, Aug. 2004.

[2] 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.

[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] 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).

[5] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. Experiences With DETER: A Testbed for Security Research. In 2nd IEEE TridentCom, March 2006.

[6] EMIST project. Evaluation methods for internet security technology. http://www.isi.edu/deter/emist.temp.html.

[7] Graphical network emulator - GNS3, http://www.gns3.net/, 2009.

[8] Wide project. http://www.wide.ad.jp/.

[9] Abilene Network operations center. http://www.abilene.iu.edu/.

[10] GEANT. The panEuropean research network. http://www.geant2.net/.

[11] Xin Li, Fang Bian, Mark Crovella, Christophe Diot, Ramesh Govindan, Gianluca Iannaccone, and Anukool Lakhina. Detection and identification of network anomalies using sketch subspaces. In IMC ’06: Proceedings of the 6th ACM.

[12] Steve Uhlig, Bruno Quoitin, Jean Lepropre, and Simon Balon. Providing public intradomain traffic matrices to the research community. SIGCOMM Comput. Commun. Rev., 36(1):83–86, 2006.

[13] Haakon Ringberg, Augustin Soule, and Jennifer Rexford. Webclass: adding rigor to manual labeling of traffic anomalies. SIGCOMM Comput. Commun. Rev., 38(1):35–38, 2008.

[14] Joel Sommers and Paul Barford. Self-configuring network traffic generation. In Internet Measurement Conference, 2004.

[15] Jelena Mirkovic, Songjie Wei, Alefiya Hussain, Brett Wilson, Roshan Thomas, Stephen Schwab, Sonia Fahmy, Roman Chertov, and Peter Reiher. DDoS benchmarks and experimenter’s workbench for the DETER testbed. In Tridentcom 2007.

[16] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. Experiences With DETER: A Testbed for Security Research. In 2nd IEEE TridentCom, March 2006.

[17] Yekui Qian, Min Chen, et al. ODC-a method for online detecting and classifying network-wide traffic anomalies. Joumal on Comlnunications, 32(1):111-120, 2011.

[18] J. Sommers, V. Yegneswaran, and P. Barford. A Framework for Malicious Workload Generation. In Proceedings of ACM SIGCOMM/USENIX Internet Measurement Conference, Taormina, Italy, October 2004.

[19] Sally Floyd and Eddie Kohler. Internet research needs better models. SIGCOMM Comput. Commun. Rev., 33(1):29–34, 2003.

[20] Haakon Ringberg, Matthew Roughan, and Jennifer Rexford. The need for simulation in evaluating anomaly detectors. SIGCOMM Comput.Commun. Rev., 38(1):55–59, 2008.

[21] The Network Simulator, http://www.nsnam.org/.

[22] OPNET, http://www.opnet.com/.

[23] J. Sommers and P. Barford. Self-Configuring Network Traffic Generation. In Proceedings of ACM SIGCOMM Internet Measurement Conference, 2004.

[24] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Perfromance Evaluation. In Proceedings of ACM SIGMETRICS, 1998.

© the authors