Spatio-Temporal Network Anomaly Detection By Assessing Deviations Of Empirical Measures

Nihar Ranjan Sethy[#1], T. Nalini[*2], M. Padmavathy[#3]

[#] *Dept of Computer Science and Engineering, Bharath University*
Chennai India

[1,2]Student, Bharath University,
s.niharranjan8899@gmail.com
[3]krishnavathy@gmail.com

Abstract— **We introduce Internet traffic anomaly detection mechanism based on large deviations results for empirical measures. Using past traffic traces we characterize network traffic during various time-of-day intervals, assuming that it is anomaly-free. Here the data can transfer from one system to another system without any problem. Also by applying our methods, anomalies are identified within a small number of observations. We compare the two approaches presenting their advantages and disadvantages to identify and classify temporal network anomalies. We also demonstrate how our framework can be used to monitor traffic from multiple network elements in order to identify both spatial and temporal anomalies. Our techniques are validated by analyzing real traffic traces with time-stamped anomalies. If any problem occurred in any of the system like any type of virus it protects to the system. In this we detect the anomaly problems. This is the very essential part of the transformation of data from one system to another system. In this paper we can detect that how many types of anomaly create on the system through the network.**

Keywords— **Large deviations, method of types, network security, statistical anomaly detection.**

I. INTRODUCTION

Many significant progresses has been made in network monitoring instrumentation, automated on-line traffic anomaly detection is still a missing component of modern network security and traffic engineering mechanisms. Network anomaly detection approaches can be broadly grouped into two classes: signature-based anomaly detection where known patterns of past anomalies are used to identify ongoing anomalies for intrusion detection), and anomaly detection which identifies patterns that substantially deviate from normal patterns of operation. Earlier work has showed that systems based on pattern matching had detection rates below 70%. Furthermore, such systems need constant (and expensive) updating to keep up with new attack signatures. As a result, more attention has to be drawn to methods for traffic anomaly detection since they can identify even novel (unseen) types of anomalies.

In this work we focus on anomaly detection and in particular on statistical anomaly detection, where statistical methods are used to assess deviations from normal operation. Our main contribution is the introduction of a new statistical traffic anomaly detection framework that relies on identifying deviations of the empirical measure of some underlying stochastic process characterizing system behavior. In contrast with other approaches [1], [2], [6], we are not trying to characterize the abnormal operation, mainly because it is too complex to identify all the possible anomalous instances. Instead we observe past system behavior and, assuming that it is anomaly-free, we obtain a statistical characterization of "normal behavior." Then, using this knowledge we continuously monitor the system to identify time instances where system behavior does not appear to be normal. The novelty of our approach is in the way we characterize normal behavior and in how we assess deviations from it. We propose two methods to characterize normal behavior: (i) a model-free approach employing the method of types [7] to characterize the type (i.e., empirical measure) of an independent and identically-distributed (i.i.d.) sequence of appropriately averaged system activity, and (ii) a model-based approach where system activity is modeled using a Markov Modulated Process (MMP). Given these characterizations, we employ the theory of Large Deviations (LD) [7] and decision theory results to assess whether current system behavior deviates from normal.

The methods we present are statistical; as a result, our approach has the potential of detecting novel anomalies, such as previously unseen attacks. This is crucial for network security as new types of attacks are constantly being engineered. As it is common in other statistical anomaly detection approaches, we rely upon observing the system during an anomaly-free period to learn what constitutes normal behavior.

The rest of this paper is organized as follows. In Section II we present the theoretical background of our work. Section III describes the two different approaches to characterize traffic and the anomaly detection mechanism. Section IV describes the experiments and results. We conclude in Section V.

II. RELATED WORK

Spatial networks that show time-dependence serve as the underlying networks for many applications such as routing in transportation networks. Traditionally graphs have been extensively used to model spatial networks (e.g. road networks); weights assigned to nodes and edges are used to encode additional information. In a real world scenario, it is

not uncommon for these network parameters to be time-dependent. It is important to be able to formulate computationally efficient and correct algorithms for the shortest path computation that take into account the dynamic nature of the networks. Models of these networks need to capture the possible changes in topology and values of network parameters with time and provide the basis for the formulation of computationally efficient and correct algorithms for the frequent computations like shortest paths.

Intrusion detection (ID) is an important component of the defense-in-depth or layered network security mechanisms. An intrusion detection system (IDS) collects system and network activity data and analyzes the information to determine whether there is an attack occurring. Two main techniques for intrusion detection are misuse detection and anomaly detection. Misuse detection (sub)systems, use the "signatures" of known attacks, i.e., the patterns of attack behavior or effects, to identify a matched activity as an attack instance. Misuse detection are not effective against new attacks, i.e., those that don't have known signatures. Anomaly detection (sub)systems, for example, the anomaly detector of IDES, use established normal profiles, i.e., the expected behaviour, to identify any unacceptable deviation as possibly the result of an attack. Anomaly detection can be effective against new attacks. However, new legitimate behavior can also be falsely identified as an attack, resulting a false alarm. In practice, reports of attacks are often sent to security staff for investigation and appropriate actions.

In most computing environments, the behavior of a subject (e.g., a user, a program, or a network element, etc.) is observed via the available audit data logs. The basic premise for anomaly detection is that there is intrinsic characteristic or regularity in audit data that is consistent with the normal behavior and thus distinct from the abnormal behavior. The process of building an anomaly detection model should therefore involve first studying the characteristic of the data and then selecting a model that best utilizes the characteristic.

III. PROPOSED SYSTEM

We present two different approaches to characterize traffic: (A) a model-free approach based on the method of types and Sanov's theorem, and (B) a model-based approach modeling traffic using a Markov modulated process. Using these characterizations as a reference we continuously monitor traffic and employ large deviations and decision theory results to "compare" the empirical measure of the monitored traffic with the corresponding reference characterization, thus, identifying traffic anomalies in real-time. Our experimental results show that applying our methodology (even short-lived) anomalies are identified within a small number of observations.

A. A Model-Free Approach

In this section, we discuss our model-free approach and provide the structure of an algorithm to detect temporal network anomalies. As noted in the Introduction we focus on traffic at points of interest in the network, even though our approach is general enough to be applied to any trace of system activity. We assume that the traffic trace we monitor (in bits/bytes/packets/flows per time unit), corresponding to a specific time-of day interval, can be characterized by a stationary model over a certain period (e.g., a month) if no technological changes (e.g., link bandwidth upgrades) have taken place. Consider a time series of traffic activity (say, in bits/bytes/packets/flows per sample). Let the partial sum (or aggregate traffic) over the time bucket starting at and containing samples. The crucial assumption we make is that is an i.i.d. sequence for some appropriate bucket size . This is a reasonable assumption in many settings as temporal correlations tend to become weaker over longer time intervals. We quantize the values of the partial sums mapping them to the finite set of cardinality. For the rest of the paper, we will be referring to as the underlying alphabet. The quantization is done as follows: we let be the range of values takes, divide it into subintervals of equal length, and map to for . To select the appropriate size of the alphabet we follow where is the likelihood of the model with respect to a process realization. The key that tends to favor models with a larger number of free parameters. The AIC removes this bias by introducing a penalty for the number of free parameters; thus, the resulting is considered the most appropriate for the given trace (minimizing modeling and estimation error). Once we have, elements of the alphabet that are not observed in the trace are merged with neighboring ones to obtain which is the final size of the alphabet.

B. A Model-Based Approach

The approach of Section 3.1 aggregated traffic over a time bucket to yield an i.i.d. sequence. One potential disadvantage of this aggregation is that it increases the response time to an anomaly since data is being processed on the slower time-scale of time buckets. In this section, the question we are seeking to answer is whether it is possible to process data on the timescale we collect them. To that end, and because the i.i.d. assumption will no longer hold, we will impose some more structure on the stochastic nature of the traffic time-series. In particular, we will assume a Markovian structure as it is tractable and has been shown to represent traffic well [9], [10], at least for the purpose of estimating distribution-dependent metrics like loss probabilities.

C. Anomaly Detection Mechanism

Anomaly detection and in particular on statistical anomaly detection, where statistical methods are used to assess deviations from normal operation. Our main contribution is the introduction of a new statistical traffic anomaly detection framework that relies on identifying deviations of the empirical measure of some underlying stochastic process characterizing system behaviour.

IV. EXPERIMENTS AND RESULTS

The language used is java sdk2.0. Java is related to C++, which is a direct descendant of C. The trouble with C and C++

is that they are designed to be compiled for a specific target. But Java is a portable, platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments. Java can be used to create two types of programs: applications and applets. An application is a program that runs on our computer, under the Operating system of that computer. The alerting and logging subsystem is selected at run-time with command line switches. There are currently three logging and five alerting options. The logging options can be set to log packets in their decoded, human readable format to an IP-based directory structure, or in tcp dump binary format to a single log file. The decoded format logging allows fast analysis of data collected by the system. The tcp dump format is much faster to record to the disk and should be used in instances where high performance is required. Logging can also be turned off completely, leaving alerts enabled for even greater performance improvements.

1) Client Model: A client is an application or system that accesses a remote service on another computer system, known as a server, by way of a network. The term was first applied to devices that were not capable of running their own stand-alone programs, but could interact with remote computers via a network. These dumb terminals were clients of the time-sharing mainframe computer.

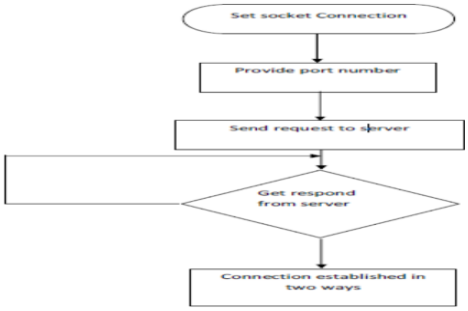

Fig. 1 Client Model

2) Server Model: In computing, a server is any combination of hardware or software designed to provide services to clients. When used alone, the term typically refers to a computer which may be running a server operating system, but is commonly used to refer to any software or dedicated hardware capable of providing services.

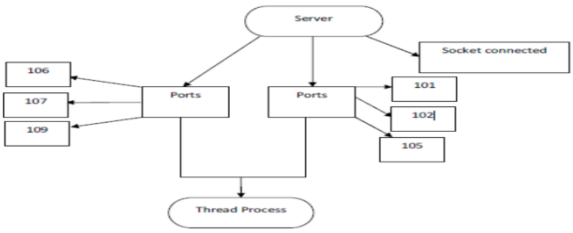

Fig. 1 Server Model

3) Network Model: Generally the channel quality is time varying For the ser-AP association decision, a user performs multiple samplings of the channel quality, and only the signal attenuation that results from long-term channel condition changes are utilized our load model can accommodate various additive load definitions such as the number of users associated with an AP. It can also deal with the multiplicative user load contributions.

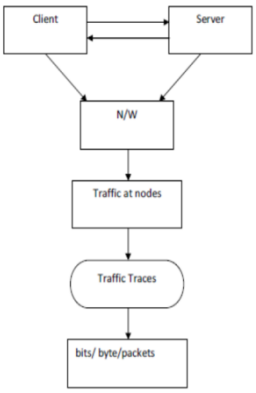

Fig-3 Network Model

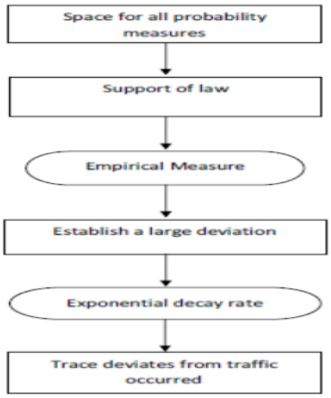

Fig-4 Large Deviations of Empirical Measure

A. Empirical Measures for Anomaly Detection

As mentioned above, the size of the alphabet and the number of states of the MMP for the Abilene data set is small when only temporal information is considered. Thus, it is easyto monitor subnets of PoPs (of low dimensionality) by specifying the group of PoPs of interest and the role of each PoP (origin or destination). We apply our framework to: (a) flows that originate (end) from (at) PoPs that are 1-hop neighbors and (b) flows that originate (end) from (at) PoPs that are many hops away from each other. In the first case study, the flows originate (end) at the Sunny Valley (SNVA) PoP with destination (originating from) the PoPs in its vicinity.

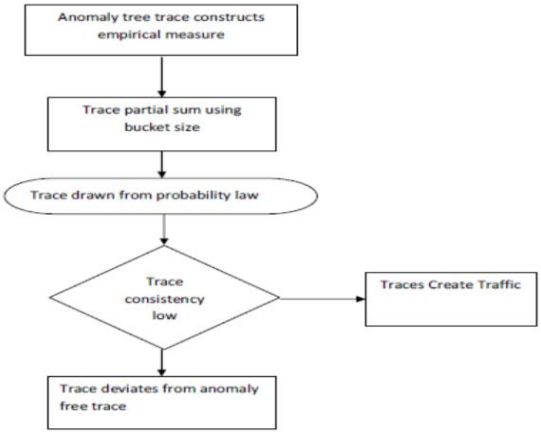

Fig. 5 Anomaly Detection

We illustrate instances of the identification of anomalies applying the model-free and the model based methods, respectively. The values of the parameters for the two methods are obtained from the temporal anomaly detection examples. It is worth noticing that the detection rate reached 100% and the false alarms rate was very low (lower than the values when only temporal anomalies were studied). This is due to two main reasons: (a) instantaneous high values in the time-series of observations that do not necessarily indicate

attacks are smoothed due to time averaging, and (b) attacks may have temporal and/or spatial correlation.

B. Congestion Traffic Minimization

In this section, we discuss the performance of our framework and we compare the two proposed methods, a model-free and a model-based one. The model-free method works on a longer time-scale processing traces of traffic aggregates over a small time interval. Using an anomaly-free trace it derives an associated probability law. Then it processes current traffic and quantifies whether it conforms to this probability law. The model-based method constructs a Markov modulated model of anomaly-free traffic measurements and relies on large deviations asymptotics and decision theory results to compare this model to ongoing traffic activity. We presented a rigorous framework to identify traffic anomalies providing asymptotic thresholds for anomaly detection. In our experimental results the model-free approach showed a somewhat better performance than the model-based one. This may be due to the fact that the former gains from the aggregation over a time-bucket in addition to the fact that the latter one requires the estimation of more parameters, hence, it may introduce a larger modeling error. For future work, it would be interesting to analyze the robustness of the anomaly detection mechanism to various model parameters.

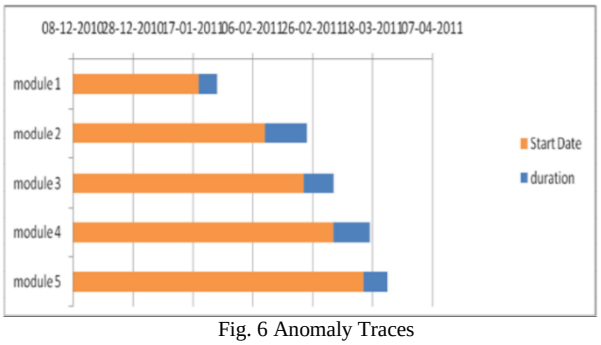

Fig. 6 Anomaly Traces

Since we monitor the detailed distributional characteristics of traffic and do not rely on the mean or the first few moments we are confident that our approach can be successful against new types of (emerging) temporal and spatial anomalies.

Our method is of low implementation complexity (only an additional counter is required), and is based on first principles, so it would be interesting to investigate how it can be embedded on routers or other network devices.

V. CONCLUSION

We introduced a general distributional fault detection scheme able to identify a large spectrum of temporal anomalies from attacks and intrusions to various volume anomalies and problems in network resource availability. We then showed how this framework can be extended to incorporate spatial information, resulting in robust spatio-temporal anomaly detection in large scale operational networks. Although most of the proposed anomaly detection frameworks are able to identify temporal or spatial anomalies, we are able to identify both as we preserve both the temporal and spatial correlation of network feature samples.

REFERENCES

[1] M. Roesch, "Snort—Lightweight intrusion detection for networks," in *LISA '99: Proc. 13th USENIX Conf. System Administration*, Seattle, WA, Nov. 1999, pp. 229–238.

[2] V. Paxson, "Bro: A system for detecting network intruders in realtime," *Computer Networks*, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[3] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Marseille, France, Nov. 2002, pp. 71–82.

[4] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation," in *Proc. DARPA Information Survivability Conf. and Expo.*, Los Alamitos, CA, Jan. 2000, pp. 12–26.

[5] 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.

[6] V. Yegneswaran, J. T. Giffin, P. Barford, and S. Jha, "An architecture for generating semantics-aware signatures," in *USENIX Security Symp.*, Baltimore, MD, Jul. 2005, pp. 97–112.

[7] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. New York: Springer-Verlag, 1998.

[8] W. Hoeffding, "Asymptotically optimal tests for multinomial distributions," *Ann. Math. Statist.*, vol. 36, pp. 369–401, 1965.

[9] I. Paschalidis and S. Vassilaras, "On the estimation of buffer overflow probabilities from measurements," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 178–191, 2001.

[10] I. Paschalidis and S. Vassilaras, "Model-based estimation of buffer overflow probabilities from measurements," in *Proc. ACM SIGMETRICS 2001/Performance 2001 Conf.*, Cambridge, MA, Jun. 16–20, 2001, pp. 154–163.

[11] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, Portland, OR, Aug.60 2004, pp. 219–230.

[12] A. Lakhina, M. Crovella, and C. Diot, " Characterization of network wide anomalies in traffic flows," in *Proc. ACM SIGCOMM Internet Measurement Conf., Taormina*, Italy, Oct. 2004, pp. 201-206.

[13] [12]A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. ACM SIGCOMM*, Philadelphia, PA, Aug. 2005, pp. 217–228.

[14] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. 2nd Int. Symp. Information Theory*, Budapest, Hungary, 1973, pp. 267–281.

[15] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.