Anomaly Detection by Finding Feature Distribution Outliers

[Extended Abstract]

Marc Stoecklin*
IBM Zurich Research Laboratory
mtc@zurich.ibm.com

ABSTRACT

In our project we are developing a technique to detect traffic anomalies based on network flow behavior. We estimate baseline distributions for meaningful traffic features and derive measures of legitimate deviations thereof. Observed network behavior is then compared to the baseline behavior by means of a symmetrized version of the Kullback-Leibler divergence. The achieved dimension reduction enables effective outlier detection to flag deviations from the legitimate behavior with high precision. Our technique supports online training and provides enough information to efficiently classify observed anomalies and allows in-depth analysis on demand. First measurements confirm its resilience to seasonal effects while detecting abnormal behavior reliably.

1. INTRODUCTION

Availability and reliability of resources in computer networks deployed in companies, universities, and ISPs have become a crucial driving factor for productivity and competitiveness. As a consequence, events detrimental to a network's performance need to be detected accurately. Such undesirable events are commonly termed network anomalies and include attacks and abuse of resources, failure of mission-critical servers and devices, significant changes of user behavior, or yet unknown events.

Intrusion detection systems are deployed today in many networks to detect attacks, spreading worms, and policy violations. However, due to their rule-based nature using expert knowledge of known threats and events, these systems commonly fail to detect new traffic anomalies. Such anomalies may, for example, emerge from the presence of zero-day exploits, outbreaks of new or modified worms, or reduced performance or outages of equipment. Anomaly detection systems overcome this shortcoming by establishing a model of the normal behavior of a network. Every observed perturbation from this behavior is flagged as abnormal and reported. Modeling network behavior appropriately is, however, a complicated task because of the dynamic nature of traffic. Also, the characteristics of events can vary significantly: whereas a high traffic load may indicate a worm outbreak or a DoS attack in some cases, a similar load may be generated by legitimate applications. In general, every event leaves some traces in the distributions of traffic features (header fields, flow characteristics, etc.). Furthermore, these distributions are subject to variation with the time of day and day of the week; thus, static deviation detection is likely to generate many false positives.

*Marc Stoecklin is a graduate student at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, currently doing his Master's thesis project at IBM Research.

2. BACKGROUND AND MOTIVATION

Anomaly detection has been extensively studied throughout recent years. Most proposed techniques model network behavior in terms of traffic volume, e.g., by applying adaptive thresholds, signal analysis, edge detection, etc. Many anomalies do not exhibit significant change in traffic volumes though. Lakhina et al. [1] propose an information theoretic analysis of feature distributions of IP addresses and ports using entropy for dimension reduction. However, entropy lacks the ability to discern differing distributions that possess the same amount of uncertainty. Consequently, observed network behavior may significantly deviate from usual behavior without being reflected by entropy. Gu et al. [2] circumvent this shortcoming to some extent by comparing an observed distribution to a baseline distribution by means of relative entropy. Their threshold-based approach divides observed packets into classes according to layer 4 protocols and splits these further in terms of ports and TCP flags.

Our technique aims at taking the dynamic nature of the traffic mix into account and assumes that a network may have multiple normal behavior modes. A threshold-based approach is, thus, not sufficient for disambiguation. We select appropriate network flow features that are likely to be affected by most anomalies and observe their distributions over time. To compare feature distributions, we use a symmetrized Kullback-Leibler divergence that provides more accuracy than [2]. Moreover, we provide an online learning mechanism to further reduce the false positive rate.

3. METHODOLOGY

Our detection technique consists of two parts: a learning and a detection phase. The learning phase makes use of reference data to establish two traffic models on which the detection phase is based: (1) *baseline feature distributions* for meaningful traffic features and (2) *Clouds of Natural Be-*

havior (CNB) representing legitimate deviations from the baseline. The set of considered features $\mathcal{F}$ includes flow duration, octets per flow, packets per flow, IP addresses, TCP and UDP ports, TCP flags, and layer 2 protocol ID.

3.1 Comparing feature distributions

Our approach uses an adapted version of the Kullback-Leibler (KL) divergence as a measure of how two discrete probability mass functions $p(x)$ and $q(x)$ differ. The standard KL divergence between p and q is defined as

$$D_{\mathrm{KL}}(p||q) := \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)}.$$

In other words, the KL divergence measures discrepancies between $p(x)$ and $q(x)$ in each sample $x \in X$ and weighs them with $p(x)$.

Similar to [2], we compare an observed distribution o_f of a feature $f \in \mathcal{F}$ to the baseline distribution b_f. However, instead of only weighing the deviations in terms of o_f by computing $D_{\mathrm{KL}}(o_f||b_f)$, we use a symmetrized weighting strategy that employs both o_f and b_f. Using only a single weight limits the ability to spot reliably samples that occur significantly less frequently in one observation (i.e., $o_f(x) \to 0$ for some $x \in X$), although $b_f(x)$ is large, since $\lim_{o_f(x)\to 0} o_f(x) \log o_f(x) = 0$. Our approach weighs deviations in terms of the differences between o_f and b_f, i.e.,

$$\begin{aligned} m_f &= D_{\mathrm{KL}}(o_f||b_f) + D_{\mathrm{KL}}(b_f||o_f) \\ &= \sum_{i=1}^{n} (o_f(x_i) - b_f(x_i)) \log \frac{o_f(x_i)}{b_f(x_i)}. \end{aligned} \quad (1)$$

3.2 Learning phase

We use annotated training data, i.e., network flow files with labeled anomalies, and extract assumed anomaly-free reference data subdivided into time periods of fixed length. The learning phase is composed of three mechanisms: (i) the baseline distribution learning, (ii) the construction of the CNB, and (iii) the ability to learn good behavior online.

Currently, our method establishes the baseline distributions b_f by computing the probability mass functions for each feature $f \in \mathcal{F}$ from the reference data over all time periods, e.g., distributions on ports, flow durations, etc.

The second phase of learning consists of reusing the reference data and comparing them to the baseline distributions by means of Eq. 1. For each time period i, we construct an $|\mathcal{F}|$-dimensional vector $\vec{v}_i$ of the deviations m_f^i for all $f \in \mathcal{F}$, representing a *behavior-characteristic point* in the feature-deviation space $(\mathbb{R}^+)^{|\mathcal{F}|}$. Ideally, $b_f = o_f$ for all $f \in \mathcal{F}$ and thus $\vec{v}_i = 0$. For any deviation in some feature f, we have $v_i^f = m_f > 0$. Since normal network traffic exhibits variations in time, the ideal case is unlikely. Thus, we keep all behavior-characteristic points acquired from the reference data and form *Clouds of Natural Behavior* (CNB) of the traffic mix, representing instances of legitimate deviations from the baseline distributions.

Our approach is able to incorporate administrator feedback to improve the knowledge of good behavior patterns. Whenever an administrator identifies a false alarm (e.g., a new server/service is installed or an old one is shut down), the newly engendered behavior point may be added to the CNB to prevent this false alarm in the future and hence reduce the false positive rate.

3.3 Detection phase

The detection phase includes two steps: (i) observed feature distributions are compared to their respective baseline distributions and (ii) the corresponding representation in the feature-deviation space is analyzed to see whether it acts as an outlier with respect to normal behavior.

From observed network flow information within a time period the feature distributions o_f for each $f \in \mathcal{F}$ are computed. Then, the deviation m_f between o_f and the baseline distribution b_f is measured using Eq. 1.

Analogous to the learning phase, we represent the observed traffic as a behavior-characteristic point $\vec{v}_o$ in the feature-deviation space. Then, we determine its degree of being an outlier (i.e., an anomaly) with respect to the CNB. Our approach allows us to track back the feature(s) that caused the largest deviations in $\vec{v}_o$. Moreover, for a given abnormal feature, our technique provides, by backtracking the KL divergence computation, on-demand insight as to which sample(s) contributed most to the deviation.

4. PRELIMINARY RESULTS

First experiments of our feature distribution comparison were done on real data collected in a large production environment using NetFlow v9. A subset of 84 time periods of 5 minutes each were selected as training data to construct the baseline distributions and the CNB. Then, the remaining data was analyzed as described above (3.3-(i)). Figure 1 shows an excerpt of the plot of the values of m_f and the corresponding behavior-characteristic points. We observe a large spike between 13:55 and 15:30 caused by a port scan from outside the network; these points are reflected as outliers from the legitimate behavior in the CNB.

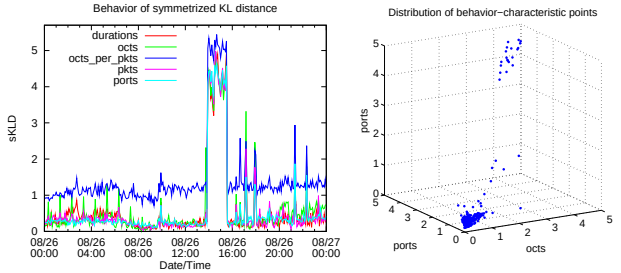

Figure 1: Detection of a port scan.

5. OUTLOOK

At the current state of the project we are evaluating various algorithms to perform outlier detection in the feature-deviation space. We are also analyzing methods to summarize or age points in the CNB in order to reduce computational overhead. Furthermore, we will evaluate the various kinds of anomalies that our technique is able to detect and determine its limits. A proof-of-concept implementation of our learning and detection phase based on feature distributions shows promising results and we will shortly implement the outlier detection part.

6. REFERENCES

[1] A. Lakhina, M. Crovella, and Ch. Diot. Mining anomalies using traffic feature distributions. In *SIGCOMM'05*, pages 217–228, 2005.

[2] Y. Gu, A. McCallum, and D. Towsley. Detecting anomalies in network traffic using maximum entropy. In *IMC'05*, 2005.