Graph-Based Traffic Analysis for Network Intrusion Detection

Hristo Djidjev Gary Sandine
Los Alamos National Laboratory, Los Alamos, NM 87545

1 Motivation and objectives

There are two main approaches to detecting malware and attacks in computer systems: signature based, where a large database of attack signatures is used to identify an on-going attack, and anomaly based, in which an unusual activity pattern is looked for. The anomaly detection approach has the advantage that new types of attacks can be identified even before their signatures are discovered and catalogued. For this end, such systems analyze regular users' activity data and build a model of "normal" activity, which is then compared with real time activity data. Any anomaly, i.e., significant deviation from the normal patterns in historical data, will be considered as likely triggered by a cyber attack and further investigated.

In this work we focus on modeling the ssh traffic in a computer network as a graph and statistically analyzing the subgraphs corresponding to individual sessions for patterns that correspond to normal activities and anomalies. The goal of our analysis is to discover patterns in the network traffic data that might indicate intrusion activity or other malicious behavior. For instance, intruders often compromise a single host and then try, by scanning hosts' neighborhoods and jumping from host to host through ssh connections (e.g., from host A to host B, then from host B to host C, etc.), to explore the network topology and discover high-value information. The resulting search pattern usually contains either a long path or nodes of high degree. In contrast, legitimate users typically visit (log-in) to a single host or reach a target host after one or two hops. We plan to study and characterize the connection patterns of legitimate users by analyzing large volumes of collected traffic data and use it to discover malicious traffic, which will be identified as anomalous.

Our detector operates in two modes. In the off-line (training) mode, it analyzes a database of recorded traffic data and produces a properly organized set of "normal" traffic patterns. In the on-line (detection) mode, it analyzes the between-host traffic in real time, extracting traffic patterns, and matching them against a pattern database in order to discover anomalies signaling a possible attack. With each pattern in the database a number is associated indicating the likelihood that it corresponds to normal (legitimate) activity. Clearly, our approach requires very fast algorithms for extracting and matching patterns in order to produce results in real time, so we need to address various algorithmic challenges of our approach, which we briefly discuss below.

2 Algorithmic approach

We first convert the ssh traffic data to a graph format, which is convenient for solving combinatorial and optimization type of problems. We define a node in the graph for each host (IP address) and a directed edge for each ssh session between the corresponding host nodes. Each edge is labeled with attributes of the session including the start time and the end time of the session. As there may be multiple edges between the same pair of nodes (corresponding to different times and/or different users), the resulting graph, which we denote by G, is a multigraph. Our objective is to partition G into subgraphs that we call *telescoping graphs (TSG)* that correspond with high probability to a set of interrelated ssh sessions initiated by a single user or attacker. See the example in Figure 1. Our goal is to represent G as a union of TSGs and to design a very efficient algorithm for computing such a decomposition. We have formally defined the notions of TSG and multigraph decomposition and have shown that such decomposition can be constructed in $O(m \log n)$ time, where n is the number of nodes and m is the number of edges of G. We have implemented our decomposition algorithm and illustrated its efficiency on analyzing traffic logs collected in Los Alamos National Lab's internal network in which an average ssh multigraph for a regular work day has roughly 35000 nodes and 100 million edges. For instance, processing a multigraph of 3.3 million edges takes about 6 seconds on a desktop PC.

3 Statistical analysis

Our first approach focuses on the joint distribution of TSG sizes and diameters. We use samples of NetFlow data collected from the LANL computer network and employ a kernel smoothing method for

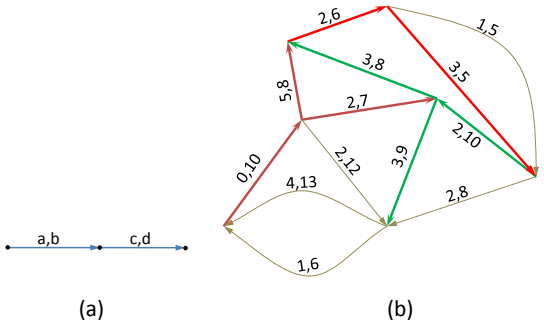

Figure 1: TSG decomposition. Pairs of numbers on each edge denote the start and the end times of the session. (a) Two consecutive edges with time labels a, b and c, d will belong to the same TSG if $a \leq b$ and $c \geq d$. (b) Decomposition of a multigraph into TSGs. Edges of the same color belong to the same TSG. Gray thin lines denote single-edge TSGs.

discrete distributions proposed by (Vander Wiel et al, 2010) to estimate the underlying distribution, against which we compare the empirical distribution obtained from a sample ssh protocol graph in order to classify the sample graph as regular or anomalous.

To these distributions we apply the asymptotic equipartition property from information theory, a consequence of the weak law of large numbers which says if $\{X_i\}$ is an iid sequence of discrete random variables with probability density function p and entropy $H(X) = -E(\log p(X)$, then the empirical entropy $-\frac{1}{n}\sum_1^n \log p(X_i)$ converges in probability to $H(X)$. For a given a sample size n and a number $\epsilon > 0$, we consider the so-called typical set $A_\epsilon^{(n)}$ which is the set of all observations $\{x_1, \ldots, x_n\}$ with empirical entropy $H(\widehat{X})$ satisfying $|H(\widehat{X}) - H(X)| < \epsilon$. For a large enough sample n, convergence in probability implies $p\{A_\epsilon^{(n)}\} > 1 - \epsilon$. Samples not in the typical set are potentially anomalous and warrant further investigation. For example, Figure 2 shows the typical set for the TSG size distribution from the ssh graph for regular work days during November 2009.

4 Comparison with work of others

Several authors, e.g., (Collins, 2008), (Ellis et al, 2006) have used graphs describing network traffic in order to discover anomalies. In all previous works there has been a single graph, viz. the graph describing all the traffic, to be analyzed–either locally, e.g., studying the *link predicates* of a node (Ellis et al), or globally, e.g., analyzing the connected component information (Collins). (Karagiannis et al, 2005) do consider patterns associated with the nodes that

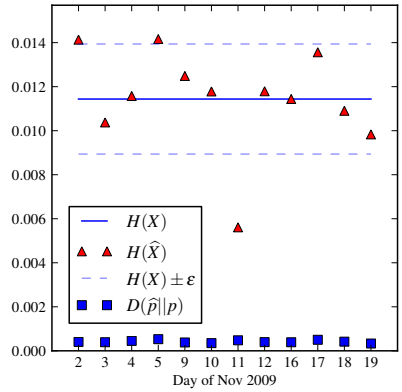

Figure 2: Typical set and an anomaly

they call *graphlets*, but they describe the application level activity, rather than intra host communications. In contrast, we decompose the original network traffic graph into subgraphs using time information and analyze the frequency and traffic attributes of these subgraphs.

5 Ongoing and future work

There still remain a number of algorithmic challenges that we plan to address in this project. One problem we will be looking into is adapting our TSG decomposition algorithm for the purposes of the on-line mode, where one needs to compute decompositions in real time. Another challenge is related to the complexity of the graph matching problem. Graph matching is needed when, for each TSG H constructed in the on-line mode, a TSG in the database to be found that is the same as (or is the closest to) H. The difficulty here is related to the fact that this problem is related to the graph isomorphism problem, for which no polynomial algorithm is known for the general case. In our special case, however, the TSGs have a special (tree) structure and polynomial algorithms are known to exist.

Furthermore, we will study model enhancements which have the potential to increase the accuracy of the predictions. For instance, one can take into account the volumes of data exchanged in a session, and encode it into weights on the edges. Another type of useful information will be the values of the nodes for a potential intruder, which might be related to system information (e.g., passwords) or to sensitive data (e.g., national infrastructure data). That information can be encoded as node weights. Finally, we will use use other data sources to label the nodes (e.g., department, or user type such as scientist, office administrator, technician) and then analyze TSGs restricted to nodes with specific labels because the statistics can vary by user type. Additionally, edges between nodes with different labels can, in some cases, be evidence of an intruder.