Network Traffic Generator for Cyber Security Testbeds

Hristo Djidjev, CCS-3;
Lyudmil Aleksandrov,
Bulgarian Academy of Sciences

We have developed an algorithm for generating secure shell (SSH) network traffic that can be used as a test bed for evaluating anomaly detection and intrusion detection tools in a cybersecurity context. Given an initial dataset describing real network traffic, the generator produces synthetic traffic with characteristics close to the original. The objective is to capture complex relationships between hosts (who communicates with whom) and between sessions—such as which sessions are interrelated in time and weekly and diurnal usage patterns.

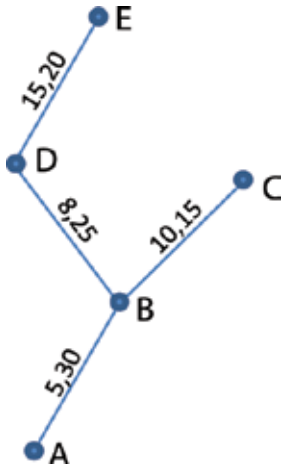

Fig. 1. A TSG. Edges have labels (s,e); S is the start time and t is the end time of the corresponding sessions. Two edges with labels (s1, e1) and (s2, e2) can appear consecutively in a TSG only if they satisfy s1 ≤ s2 and e1 ≥ e2 (the telescoping property).

Detecting malicious software and intrusions in computer networks has turned into a task of utmost importance for cyber professionals and is a major challenge for researchers [1]. One popular approach for detecting intrusions is to analyze the network traffic and find patterns of anomalous behavior [2]. Cybersecurity tools based on anomaly detection typically involve two phases. In the first, the training phase, historical data is statistically analyzed and patterns of normal behavior are extracted from the data. Specifically, such patterns may include the statistical distributions of some parameters of the traffic, for example, its volume or the session durations, and thresholds of the maximum deviation from the averages beyond which the traffic will be considered abnormal. In the second, detection phase, current traffic is analyzed and compared against the model of normal traffic and any session that deviates from that model is labeled as anomalous and is further investigated by domain experts.

In order to test existing and new anomaly and intrusion detection systems it is often preferable to use a generator that can produce network systems, traffic that resembles real observed traffic [3] and that uses adjustable parameters such as the number of hosts, number of sessions, and average session durations. Using real traffic for testing purposes has several drawbacks—network data can be difficult or impossible to get for many researchers, and sharing data between researchers can pose security or privacy issues or be forbidden by cyber policies. For anomaly detection purposes, it will be hard or impossible to distinguish between normal and malicious traffic and therefore hard to evaluate the performance of the tested anomaly detection tool. Using a traffic generator allows one to mix normal and malicious traffic in various scenarios and also vary the other traffic parameters so that the anomaly detector can be tested under different conditions.

In this report we describe our work on generating normal (as opposed to malicious) network traffic based on the SSH protocol. The traffic data we generate is in the form of a set of SSH sessions, where for each session we provide its source and target nodes (corresponding to the IP addresses of hosts in the real traffic), its start time, and its duration. Our goal is to capture patterns of real traffic associated with relationships between pairs of hosts and pairs of sessions. Specifically, we are interested in interrelated sessions that may have been started by the same user during the same activity. For instance, a legitimate user may start at host A, login to host B, go from host B to host C (as host C may not be directly reachable from A), and then close all sessions. In such a case we consider sessions (A,B) and (B,C) related. We merge such interrelated sessions in structures we call "telescoping subgraphs" (TSG) [4], see Fig. 1. In the example, the TSG is a graph consisting of three nodes, A, B, and C, and two edges (A,B) and (B,C). Intruders, unlike legitimate users, may have gained access to node A and from there can try to explore the network by hopping from host to host, forming TSGs with entirely different characteristics [5].

The difference between the patterns of legitimate users and intruders has proved useful and has been exploited in previous anomaly detectors [4,5]. Hence, one of our objectives is to generate traffic whose hosts and sessions are interrelated in a way that resembles the one observed in the original traffic. Other objectives are to preserve the temporal aspects of the traffic–namely, the different levels of activity during different days of the week and different times during the day and the session durations, the total volume of the traffic, and the distributions of the node outdegrees and indegrees, defined as the numbers of sessions with origin or that target a given host, respectively.

Our algorithm consists of two phases. During the analysis phase, the real network traffic is analyzed and a profile is computed that summarizes traffic properties that are essential for the generation process. The specific parameters of the original traffic that we compute and store for G are stored in the following date sets: (D1) the distribution of the sizes

For more information contact Hristo Djidjev at djidjev@lanl.gov.

Funding Acknowledgments
LANL, Laboratory Directed Research and Development Program

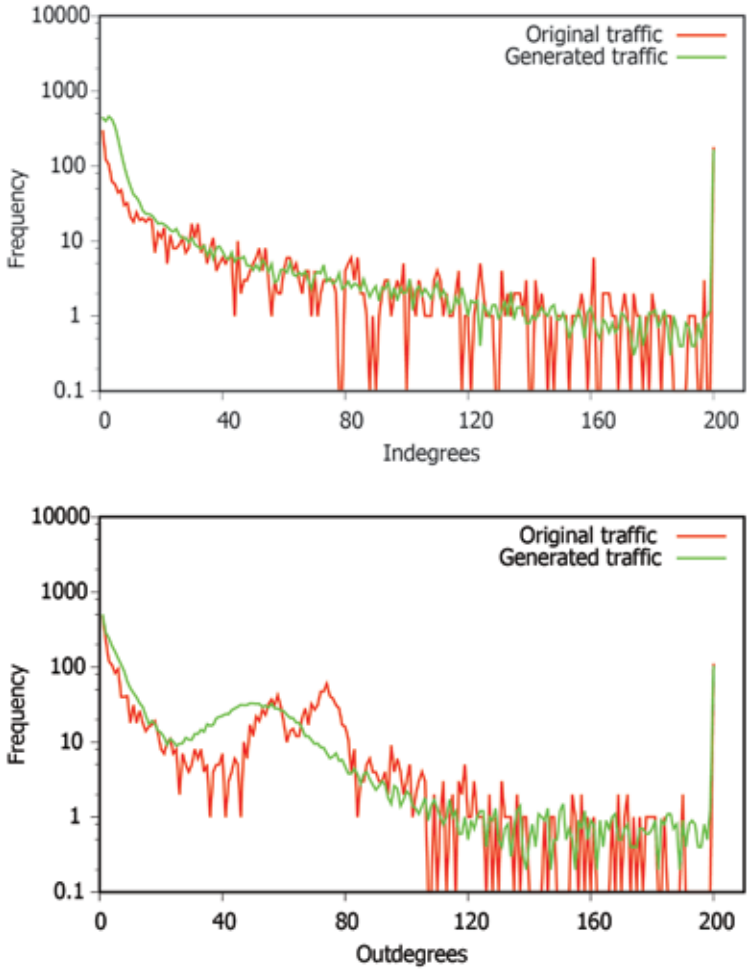

Fig. 2. Plots showing the indegree and outdegree frequencies for the original and the generated traffic. In both cases, the traffic is over a period of 28 days, but the generated-traffic data are averaged over 10 randomly generated instances.

and the total number of TSGs, (D2) the frequency of the edges (how many times each edge has been observed), (D3) the frequency of length-2 subpaths of all TSGs, and (D4) the average duration for each edge. During the generation phase, given a time window W, the profile generated in the first phase is used to generate new traffic in W. For each weekday w and each hour h, the data in the profile associated with the pair (w,h) is used. This allows us to more accurately capture the diurnal patterns of the traffic.

In order to test the method, we computed several characteristics of the original and the generated traffics and compared them. The original data were extractions of LANL network traffic collected over a period of several years. We compared TSG sizes, indegrees and outdegrees of the nodes (the hosts), and the session durations. Since our generation algorithm uses the TSG sizes information during the generation process, it was natural to expect that the TSG sizes in the original and the generated traffic would be very close. This was confirmed by the experiments. But information about the indegrees and the outdegrees was not used explicitly during the generation, so it was interesting to see if there is any correlation between these characteristics. Figure 2 shows the results of that comparison. There is a very good correlation as well as some small differences (perfect correlation is not desirable as it would indicate over-fitting). The data for the plots for the new traffic were averaged over 10 runs, which is one of the reasons they are smoother and have fractional values (between 0 and 1). The analysis of the session duration data shows a similar good correlation.

[1] Dunlavy, D.M. et al., SNL report SAND 2009-0805 (2009).

[2] Garcia-Teodoro, P. et al., *Comput Secur* **28**, 18 (2009).

[3] Sommers, J. et al, Computer Science Departmemt Technical Report 1525, University of Wisconsin (2006).

[4] Djidjev, H. et al., "Graph Based Satistical Analysis of Network Traffic," Ninth Workshop on Mining and Learning with Graphs, San Diego, CA, August 20-21, (2011).

[5] Neil, J.C., "Scan statistics for the online detection of locally anomalous subgraphs," Ph.D. Thesis, University of New Mexico (2011).