Poster Abstract: COSMO - Emulation of Internet Traffic

Daisuke Miyamoto
Information Technology Center
The University of Tokyo
2-11-16 Yayoi, Bunkyo-ku, Tokyo, JAPAN
daisu-mi@nc.u-tokyo.ac.jp

Toshiyuki Miyachi
StarBED Technology Center
National Institute of Information and
Communications Technology
2-12 Asahi-dai, Nomi, Ishikawa, JAPAN
miyachi@nict.go.jp

ABSTRACT

This paper discusses for generating realistic network traffic for the emulated Internet [5]. For emulating entire of elements of the real Internet on our testbed, we have been trying on constructing emulated inter-AS network. In this paper, we explore the suitable traffic generator which is able to emulate the real Internet traffic in the testbed. However, the existing generators do not equip functions of generating the realistic payloads. This paper therefore designs COSMO, a new traffic generator, to meet with our requirements. The key idea is to replay the real Internet traffic, rather than making it. As an initial study, we focus on improving the realism of the generated traffic. The paper captures packets at our monitoring point, divides the traffic trace file into several chunks, and replays the traffic in the emulated Internet. Based on the experiment, the paper provides our preliminary evaluation and indicates the feasibility aspect from the realism in the emulated technology.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications - Tools

General Terms

Experimentation, Security

Keywords

Traffic Generator, Denial of Service, Testbed

1. INTRODUCTION

The paper introduces COSMO, an approach for generating Internet traffic in network emulation testbeds. Recently, testbeds become an important technology for researchers to perform repeatable, realistic, and scalable experiments. Our research group have developed Internet emulation techniques [5], in which a testbed has a similar topology of the Internet, to obtain realistic experimental results. On our emulated inter-AS network, the AS is simply emulated by a single Quagga [8] router and network links on an experimental node.

Based on the emulated Internet, we considered to generate realistic network traffic. Our first motivation was to evaluate our developed Denial of Service (DoS) defense systems, named InterTrack [4]. DoS attacks consume the resources of a remote host or network. On these attacks, source IP address spoofing technique is often employed by attackers, since it makes the traditional countermeasures difficult to identify the true source of attack packets. InterTrack is designed to locate the sender of the packet based on the hash-based IP traceback technology [9].

According to Hussain et al. [6], the experimenter of the DoS defense method needs to carefully design for a testbed in consideration to network topology and background traffic. For the network topology, we decided to use the emulated Internet. Of course, it might be difficult to construct an experimental network which has same size, same facilities, and / or same characteristics of the real Internet. Therefore, we had proposed to outfit subgraphs of the Internet [5] and also proposed using Virtual Machines to increase the number of routers in the emulated Internet [7].

This paper attempts to replay packet trace as background traffic, rather than making it. Our prototype implementation, named COSMO, divides a packet trace file for AS-level chunks and controls to replay them. Our experiment constructs the Japanese Internet topology, runs COSMO to replay our traffic dataset, and compares the replayed traffic and the original one.

2. COSMO

This paper discusses to generate traffic for the emulated Internet. The emulated Internet is an experimental environment in which each node represents an Autonomous System (AS) to run BGP routing software; in short, all nodes will be AS Border Routers (ASBRs) to emulate the realistic Internet topology. In order to support generating Internet traffic in this environment, the following requirements can be considered.

1. Reality of Traffic

The generated traffic must have reality, which is similar to the traffic in the real Internet. Some of the existing traffic generators employ "0-padding" to adjust the traffic bandwidth, but it might diminish the realism. Such traffic would not be suitable for vali-

EMUTools Workshop 2013, March 05-07

DOI 10.4108/icst.simutools.2013.251708

dating payload-based anomaly detection [1]. Further, the characteristics in the real Internet, e.g., periodicity, should also be reproduced.

2. Multi-point Traffic

The traffic generator must be designed for multiple nodes, whereas the almost of traffic generators provide single-point-to-single-point traffic. Assuming if an experimenter employs the emulated Internet and attempts to observe 10 Gbps of Distributed DoS traffic. When the traffic was transmit from network leaves (attackers) to the particular network leaf (victim), the experimenter should manually adjust traffic bandwidth for the each attacker node. For expediting his experiment, the generators in the distributed environment must be orchestrated to send traffic.

3. Manipulation of Time Scale

Imagine if an experimenter wants to generate 24 hours of traffic. Naturally, the generator will finish when 24 hours past, however, it might be inconvenient while the experiments always wait for 24 hours. The generator should support fast-and-slow forward by adjusting the interval time for packet transmission. If the traffic generator runs at double the speed, his/her experiments will be finished in 12 hours. We call this function as the manipulation of time scale, and traffic generators should support it without losing the realism.

Existing traffic generators such as Harpoon [11] and D-ITG [12] are able to create network traffic by learning traffic patterns, but their generated payloads are lack of realism. In comparison to the traffic generators, capturing the packet trace in the wild and replaying it in the emulated Internet make our testbed might be realistic. Unfortunately, the current packet trace tools are designed to provide a single-point-to-single-point traffic.

For the emulated Internet topology, our testbed orchestrates multiple ASes to replay packet trace. Assuming if all Internet packets are saved in the PCAP format [10]. Given the condition, we considered the following steps for replaying the packets. At first, the traffic file is divided into several chunks by checking the source ASes. According to the routeview dataset provided by CAIDA [2], we can obtain the source AS from the recorded source IP address. We then locate the chunked file for the nodes in the emulated Internet. Finally, the generator, which runs on the each node, sends the chunked file along with the timestamp data.

For example, the source address of the packets is 130.69.x.y, the AS number is 2501 (The University of Tokyo) by checking routerview dataset and the nodes which represents AS 2501 will send the packet. If the chunked data contains the pair of the packets (P_i, P_j) that were respectively recorded at Time (t_i, t_j) where $t_j > t_i$, the node sends packet the P_i, waits Time $(t_j - t_i)$, and then sends the packet P_j.

3. EXPERIMENTS

Based on these considerations, our study will implement a traffic generator, which we called COSMO, for the emulated Internet. As a first step, we developed the prototype which attempts to meet with the requirement 1, the reality of the generated traffic.

In order to experiment COSMO, we constructed the emulated Internet and replayed the background traffic, and finally compared the replayed traffic with the original one.

3.1 Experimental Setup

Our testbed employed the emulated Internet for network topology. AnyBed [13] is a useful tool for constructing the emulated Internet. It requires two configuration files. One is the physical network configuration file which describes testbed specific information such as hardware address of nodes and wiring among network switches. Other one is the logical network configuration file which describes network topology. Given CAIDA's AS Relationships Dataset (ASRD) [3] and a Routeviews Prefix to AS mappings Dataset (PFX2AS) [2], AnyBed can generate the network configuration file in which every BGP router advertises the realistic IP addresses. Due to the facility of the testbed, we filtered ASes to extract a subgraph. This paper employed the region based filtering [5] and constructed the network topology which represented Japan Internet including 469 of ASes.

For the background-traffic dataset, we obtained network traffic dataset which was consisted of 24 hours of DNS traffic. The amount size was roughly 9.2 GB. Of the dataset, 3.4 GB was came from outside of Japan, 1.4 GB was our inner-AS traffic which was sent from our AS to our AS, and the rests were came from other ASes in Japan.

3.2 Replaying Background Traffic

This section demonstrates how COSMO works and compares the network traffic with replayed one. In our testbed, each AS node, the node represented AS in the emulated Internet, replayed the dataset which has been captured in the wild. The dataset were divided for 469 ASes which represented Japan Internet topology. Note that the testbed was completely quarantined from the real Internet. To avoid the damage suffered from connecting these nodes to the Internet, we designed our test environment to have no Internet access.

In order to facilitate our comparison, we created two KVM instances. The one had the role for sending packets that were sent from the outside of Japan. The other also had the role for sending the inner-AS traffic. The other ASes also replayed the traffic. In addition, our experiment employed tcpreplay [14] for sending packet trace.

Since our experiment did not modify the traffic dataset, the almost of all packets could not reach to the destination IP address. In this case, AS routers would send an ICMP unreachable message to the sender. Assuming if the traffic dataset also contained ICMP unreachable messages, it was hard to identify that the ICMP messages were replayed or were created by the routers. Therefore, we configured the AS node's firewall to allow tcpreplay to send ICMP unreachable messages, and also configured to OSes to prohibit sending these messages.

Given these conditions, our experiment made 468 nodes and two additional nodes send traffic, and monitored them at the AS node which represented our AS in the emulated Internet. We analyzed the frequency of the number of the received packet and the network bandwidth. Figure 1 showed the our observation of counting the number of the packets, where X axis denotes the course of time in seconds, Y axis denotes the average of incoming packets, the red broken line denotes the traffic in the wild, and the blue line denotes the

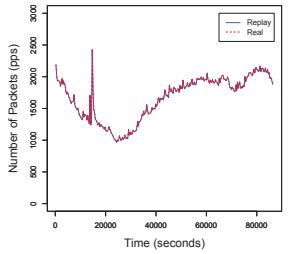

Figure 1: Number of received packets

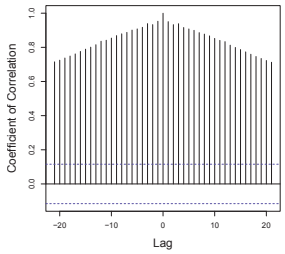

Figure 2: TSR for number of packets

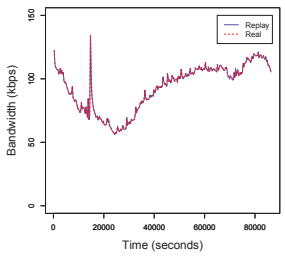

Figure 3: Traffic volume

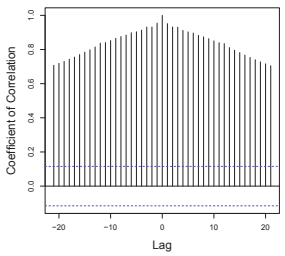

Figure 4: TSR for traffic volume

traffic in the emulated Internet. We also performed time-series regression between real traffic and replayed traffic, and the result was shown in Figure 2. We found that the maximum coefficient of correlation value was 0.999 when $lag = 0$.

Aside from the packet count, the traffic volume were also similar as shown in 3 where Y axis denotes the average bandwidth of traffic. The results of the time series regression was shown in Figure 4. The maximum coefficient of correlation value was also 0.999 when $lag = 0$. Accordingly, we assumed that the replayed traffic was quite similar to the real traffic.

4. CONCLUSION

This paper proposed COSMO, to generate background traffic in the emulated Internet. Our idea was to replay traffic from a captured packet trace, rather than making it. This paper also demonstrated that the packet trace file was divided into AS-level chunks, and an AS node in the emulated Internet replayed each chunk which had been sent from the corresponding AS in the wild. The experiment showed the replayed traffic was quite similar to the real traffic. Accordingly, we believe that our emulation technique was feasible from the aspect of the realism. In our future work, we will equip a function to meet with the rest of requirements. We might also consider creating the ideal network traffic dataset. It will require capturing traffic at multiple monitoring points and removing the duplicated packets in the multiple packet trace files, but it is beyond the scope of this paper.

5. REFERENCES

[1] Akritidis, P., Anagnostakis, K. G., and Markatos, E. P. Efficient content-based detection of zero-day worms. In *IEEE International Conference on Communications* (May 2005), vol. 2, pp. 837–843.

[2] CAIDA. Routeviews Prefix to AS mappings Dataset(pfx2as). Available at: `http://www.caida.org/data/routing/routeviews-prefix2as.xml`.

[3] CAIDA: cooperative association for internet data analysis. The CAIDA AS Relationships Dataset. Available at: `http://www.caida.org/data/active/as-relationships/`.

[4] Hazeyama, H., Kadobayashi, Y., Oe, M., and Kaizaki, R. InterTrack: A federation of IP traceback systems across borders of network operation domains. In *Proceedings of Annual Computer Security Applications Conference, Technology Blitz Session* (Dec. 2005).

[5] Hazeyama, H., Suzuki, M., Miwa, S., Miyamoto, D., and Kadobayashi, Y. Outfitting an Inter-AS Topology to a Network Emulation TestBed for Realistic Performance Tests of DDoS Countermeasures. In *Proceedings of Workshop on Cyber Security Experimentation and Test* (Aug. 2008).

[6] Hussain, A., Schwab, S., Fahmy, S., and Mirkovic, J. DDoS Experiment Methodology. In *DETER Community Workshop* (June 2006), pp. 8–14.

[7] Miwa, S., Miyachi, T., Eto, M., Yoshizumi, M., and Shinoda, Y. Design and Implementation of an Isolated Sandbox with Mimetic Internet Used to Analyze Malwares. In *Proceedings of DETER Community Workshop on Cyber Security Experimentation and Test* (Aug 2007).

[8] Quagga. Quagga Routing Suite. Available at: `http://www.nongnu.org/quagga/`.

[9] Snoeren, A. C., Partridge, C., Sanches, L. A., Jones, C. E., Tchakountio, F., Kent, S. T., and Stayer, W. T. Hash-based IP traceback. In *Proceedings of the ACM SIGCOMM 2001 Conference on Applications, Technologies, Architectures, and Protocols for computer communications* (Aug. 2001), pp. 3–14.

[10] Society, T. I. PCAP Next Generation Dump File Format. Available at: `http://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html`.

[11] Sommers, J., and Barford, P. Self-Configuring Network Traffic Generation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement* (Oct. 2004), pp. 68–81.

[12] Stefano Avallone, A. P., and Ventre, G. Distributed Internet Traffic Generator (D-ITG): analysis and experimentation over heterogeneous networks. In *Proceedings of the International Conference on Network Protocols* (Nov. 2003).

[13] Suzuki, M., Hazeyama, H., Miyamoto, D., Miwa, S., and Kadobayashi, Y. Expediting Experiments across Testbeds with AnyBed: A Testbed-Independent Topology Configuration System and Its Tool Set. *IEICE Transactions on Information and System E92-D*, 10 (Oct. 2009), 1877–1887.

[14] Tcpreplay. Availale at: `http://tcpreplay.synfin.net/trac/`.