On Automatic Network Environment Cloning for Facilitating Cybersecurity Training and Testing

Cuong Pham
Télécom SudParis

Gregory Blanc
Télécom SudParis

Hervé Debar
Télécom SudParis

Abstract—**Cyber attacks are happening worldwide on a daily basis. It results in a great need to have a system that can construct a practical network environment, in terms of both network functions and background traffic flows, for the purposes of security tool assessment and cyber range creation. The concept of "cloning" traditionally is to create an identical copy of a single entity, such as a machine or a software. In this paper, we propose an extension of this concept through building a system that can clone a production environment, both in terms of network functions and legitimate traffic flows. We also present our current progress in building the first prototype of the system.**

I. Introduction

Network environment construction and traffic generation are important to networking and cybersecurity researchers. As cyber attacks are growing in terms of volume and sophistication, such tools turn out to be useful in many cases. One usage is the assessment of network security tools, including intrusion detection systems (IDSes). In this situation, an easy and convenient way is needed to construct a production-like environment and generate a background traffic flow, for testing the reliability and accuracy of an IDS. Another utilization is the realization of cybersecurity training and Capture The Flag competition environments (CTFs). Though there exist automation tools to prepare network architecture and launch attacks [1], [2], they often lack the function to generate background traffic during the competitions, to make the environment more lifelike. This function is rather performed manually when it is included in the cyber ranges, which is time consuming and inefficient.

To address this difficulty, we propose the idea of implementing a network environment clone system, as a solution of reliably creating an emulated production-like environment in a reproducible manner. Given a production environment, the system firstly studies its properties, such as the topology, softwares, available services, system configurations, and the characteristics of the background traffic flows, such as IP and port distributions, protocol volumes, intensity, etc. It then creates an emulated environment that has the similar topology, and functions in a similar manner as the reference one. After that, it produces background traffic flows in the emulated environment, following the characteristics extracted above. The traffic is generated based on a number of protocol agents, which contains various actions linked together by probability models for representing real user activity patterns.

II. Research Background

Network security deals with network related indicators, such as IP addresses, service ports, protocol usage, etc. In general, the most ideal condition for network security tool assessment or network cybersecurity training, is to have a real production environment, with a flow of legitimate traffic generated by user daily activities, running in the background. However, because of security concerns, this condition is rarely achieved. It therefore leads to an alternative, which is to construct a closed network environment that resembles or mimics the production one, and artificially generate a background traffic flow. IDSes and trainees then can be tested with the traffic, either in the *live* mode (at the generation time), or in the *replay* mode (afterwards, once the traffic has been recorded).

There have been many research works published on the topic of generating background traffic, in which they follow two main methods, which are *traffic characteristic modeling*, and *user behavior modeling*. Regarding the first approach, a traffic model is extracted and built from a real trace following a set of predefined characteristics. For example, D-ITG is a powerful generator for both legitimate and attack traffic [3]. It models background traffic with measured distributions of flow behavior on a target link, such as IP spatial distribution, inter-session start times, session duration, etc. The tool has one sending server and one receiving server, in which they use a signaling channel to exchange information on the generation process. Swing is another tool for network traffic generation [4]. It is aimed to reproduce several essential characteristics from a given trace, including packet inter-arrival rate and burstiness across a range of time scales, packet size distributions, flow arrival time and length distribution. Besides, there exist plenty other similar tools according to a survey by Behal and Kumar [5], such as Curl Loader6 that can generate HTTP, FTP, and SSL traffic, and can simulate hundreds of users with different IP addresses, or RUDE7 for generating and collecting UDP traffic. Generally, this approach is useful to obtain a trace which has a range of characteristics of an original trace. However, contents of the packets' payloads in the generated traffic are usually unrealistic (content of email messages, http urls, etc.), which poses a problem that the trace might not be representative of actual network traffic.

The second method focuses on the behavior patterns of actual users, and try to generate the background traffic following these patterns. The DARPA project, which was run in 1998

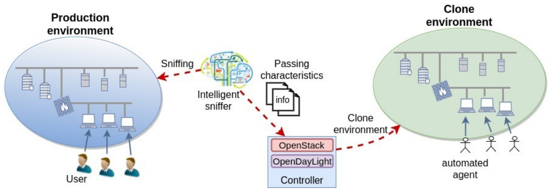

Fig. 1. The work flow of the system.

and 1999 by the MIT Lincoln Laboratory [6], follows this method. To generate background traffic, various usage patterns of common services were modeled and deployed mimicking real users using real services and protocols from a custom network environment. Another model is proposed by Shiravi et al. [7], which also generates both legitimate and malicious traffic for testing IDSes. A number of agents are implemented to generate traffic for different protocols, following predefined distributions in terms of number of requests and their time intervals. Besides, we have the work of Wang et. al., in which they propose a Web traffic generator on Many-Core processors system (TGMP) [8]. They divide the process of a user visiting a web site into three stages: selecting a link, clicking another link in a relatively short time, and staying for a longer time with an inactive state when obtaining the information. The system is designed for large scale experiments, as that it is equipped with a lot of cores, and can simulate up to 50,000 users accessing the web server simultaneously. These works, while they are efficient to obtain a legitimate traffic flow that reflects real user activities, have some downsides. The DARPA project has been criticized [9], with the unrealistic nature of the network architecture, some questionable methodologies used in the evaluation, as well as the validation of the dataset. The model of Shiravi, while it appears to be well-rounded, heavily relies on its fixed network infrastructure and topology, and its protocol agents have only simple activities for each protocol. TGMP focuses only on HTTP traffic, and only models the activity of web page browsing and no other types, such as live streaming, etc.

Being aware of the advantages and drawbacks of the previous works, our research aims at building a network environment cloning system, in both network functions and background traffic characteristics perspectives. We set to address two main requirements with our system. Firstly, the environment for the traffic generation process must be similar to a production one, and is constructed in a reproducible manner. Secondly, the traffic is generated based on a number of protocol agents, containing various user activity patterns. Furthermore, these agents must be monitored throughout the process, so that the system can manipulate their actions properly for achieving the reference characteristics in the generated traffic.

III. Research Objectives and Research Plan

Figure 1 describes our vision of how a network environment cloning system works. Given an input as a reference production environment, the system firstly studies the network architecture, along with the background traffic characteristics. It then creates a similar network environment using virtualization technologies, and from there generates the traffic following the characteristics extracted above. In more details, the system has three main features as listed below:

- **Network sniffing**. This is the ability of extracting the network functions (network topology, softwares, system configurations, firewall rules, NATs, etc.), and the characteristics of the traffic flow in a network environment. The knowledge about the network functions can be gained directly from network sniffing, or is partly given by network administrators. About traffic characteristics, they can be analyzed at a macroscopic level (IP flow and ports, arrival rate, protocol volumes distribution, etc.), and a microscopic level (packet size distribution, session start time distribution, etc.). Moreover, the traffic should be recorded over a sufficient period of time, which allows the sniffer to model the environment in the most comprehensive perspective.
- **Network construction**. This is the ability of creating a network environment with the topology extracted from the above feature. Production environments can be complex (e.g, multi-cloud topology, or multiple networks from different locations). The features therefore needs to be efficient to virtualize such topologies. Furthermore, it should be able to set up necessary services and configurations in the system, to make the cloned environment functioning in a similar manner as the reference one.
- **Background traffic generation.** This feature is for producing a flow of background network traffic, which matches the characteristics obtained from the first feature. The traffic is generated from various actions for representing normal user activities in the environment. In addition, the generation process should be able to flexibly adjust the traffic characteristics by allowing the users to change the network conditions, or by following an appropriate underlying random distribution.

The proposed research comprises the following tasks:

1) Study and find a list of characteristics that fully models a network environment. The work of accurately generating network traffic involves many difficulties. There exists no consensus in characterizing or comparing network traffic, as it varies among different conditions [10]. The rapid evolution of the Internet and the presence of new applications make the job of modeling traffic even more challenging.
2) Study the state of the art tools on the subject of network sniffing, and implement the mentioned sniffer. One of the main challenges in this task is to obtain the network topology from the captured traffic. The information acquired from Layer 2 or from some protocol such as SNMP could help collect the basic knowledge of the topology. However there are still many issues, such as how to detect in case a machine has multiple interfaces, or in case of complex topologies such as multi-cloud. We

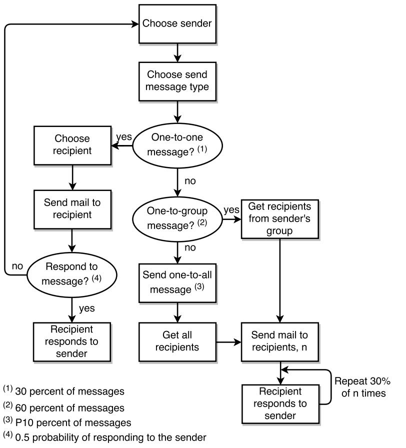
Choose sender
Choose send message type
One-to-one message? (1)
yes
Choose recipient
no
Send mail to recipient
One-to-group message? (2)
yes
Get recipients from sender's group
no
Respond to message? (4)
no
Send one-to-all message (3)
yes
Recipient responds to sender
Get all recipients
Send mail to recipients, n
Repeat 30% of n times
Recipient responds to sender
(1) 30 percent of messages
(2) 60 percent of messages
(3) P10 percent of messages
(4) 0.5 probability of responding to the sender

Fig. 2. Flow chart of actions in SMTP agent.

therefore set our goal as to derive a network topology which is "functionally" identical to the reference one, but not necessary exactly the same. Moreover, as it sniffs a production environment, privacy protection have to be taken into account.

3) Implement a function for automatically constructing an environment, with inputs such as a topology. Another module is needed for installing and configuring the environment (installing packages, services, firewalls, etc.). This module can be built from automation tools, such as `ansible`. Certain research is needed for applying network virtualization technology into the function, for the purpose of virtualizing over any substrate network with an arbitrary topology, especially multi-domain ones.
4) Implement agents for producing legitimate traffic for different protocols. They, as mentioned before, should implement various actions for reflecting user activity patterns. Moreover, as they have to reproduce the characteristics extracted from the model traffic, a controller would be required to monitor and manipulate their actions accordingly throughout the generation process.

IV. Current Progress and Preliminary Results

We have conducted a preliminary research on the subject of modeling network traffic, and have chosen several characteristics from the macroscopic level, including the flow abstraction, the arrival rate, and the protocol volumes distribution. These are listed as the main traffic characteristics in both the previous works ([3], [6], [7]), and other networking research projects[1]. However, a network sniffer has not been implemented yet, and the task of extracting these characteristics is rather done manually at the moment.

We currently focus on 3) and 4) tasks (Section III). An integration of OpenStack, as a virtualization platform, and OpenDaylight, as a SDN-enabled cloud deployment controller, has been deployed in our experimental testbed. An extension of this integration has been implemented, for the function of constructing a network with an arbitrary topology. The extension however does not include yet the ability of installing services and configurations in the environment.

Regarding the 4) feature, we have been implementing a set of agents for producing the traffic for several protocols, including HTTP(S), IMAP, SMTP, FTP, and SMB. We inherit from DARPA project the approach of modeling user actions in each agent, with several necessary modifications for adapting the current Internet usage trends. Figure 2 shows the algorithm to produce SMTP traffic as an example. SMTP traffic is generated based on several actions, including sending emails and responding. The emails' content are taken randomly from a corpus, which is prepared in advance. Three types of email posting are available currently, consisting of (i) "one-to-one" (a message is sent from a random client to another one), (ii) "one-to-group" (a message is sent from a random client to its group), and (iii) "one-to-many" (a message is sent from a random client to everyone). When a sending email session is initialized, one of the three types is chosen. In total, 60 percent of the messages are of type "one-to-group", 30 percent are of type "one-to-one", and 10 percent are of type "one-to-many".

Currently, the system uses a statistical approach to control these agents. The traffic data generated from one action by each agent has been measured and imported into the system as configuration files in advance. Based on the inputs as the traffic characteristics that are manually imported, the system calculates the number of necessary agents for each protocol, prepares their schedules, and deploys them.

References

[1] A. S. Raj, B. Alangot, S. Prabhu, and K. Achuthan, "Scalable and lightweight ctf infrastructures using application containers," in *Proceedings of the 2016 USENIX Workshop on Advances in Security Education (ASE'16)*, 2016.

[2] R. Beuran, C. Pham, D. Tang, K.-I. Chinen, Y. Tan, and Y. Shinoda, "Cybersecurity education and training support system: Cyris," *IEICE Transactions on Information and Systems*, vol. 101, no. 3, pp. 740–749, 2018.

[3] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[4] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 712–725, 2009.

[5] S. Behal and K. Kumar, "Characterization and comparison of ddos attack tools and traffic generators: A review.," *IJ Network Security*, vol. 19, no. 3, pp. 383–393, 2017.

[6] J. W. Haines, L. M. Rossey, R. P. Lippmann, and R. K. Cunningham, "Extending the darpa off-line intrusion detection evaluations," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, vol. 1, pp. 35–45, IEEE, 2001.

[7] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[8] X. Wang, C. Xu, W. Jin, J. Wang, Q. Wang, and G. Zhao, "A scalable parallel architecture based on many-core processors for generating http traffic," *Applied Sciences*, vol. 7, no. 2, p. 154, 2017.

[9] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 305–316, IEEE, 2010.

[10] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Inferring tcp connection characteristics through passive measurements," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1582–1592, IEEE, 2004.

[1] http://mawi.wide.ad.jp/mawi/.