www.ijiarec.com

ISSN:2348-2079

Volume-7 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

End-to-end detection of cloud network IP spoofing attacks on SAASS

R. Anand[1], D.Kavipriya[2], P. Poomani[2], P. Ravisivaselvakumar[2], A. Sachin[2]

[1]Assistant Professor, Department of Information Technology, Nandha Engineering College (Autonomous)

[2]UG Students, Department of Information Technology, Nandha Engineering College (Autonomous)

ABSTRACT

Software Defined Networking is a new networking paradigm thus enabling new innovations in network protocols and applications. Our new attacks are somewhat similar in spirit to spoofing attacks in legacy networks however with significant differences in exploiting unique vulnerabilities how current Software Defined Network operates differently from legacy networks. According to our study, all current major Software Defined Network controllers we find in the market are affected i.e., they are subject to the Network Topology Poisoning Attacks. We then investigate the mitigation methods against the Network Topology Poisoning Attacks and present Tope Guard, a new security extension to SDN controllers, which provides automatic and real-time detection of Network Topology Poisoning Attacks. Our evaluation on a prototype implementation of Tope Guard in the Floodlight controller shows that the defense solution can effectively secure network topology while introducing only a minor impact on normal operations of Open Flow controllers.

Keywords: IP Spoofing Attacks, Software Defined Network, Secure Computing Networks, Tope Guard.

INTRODUCTION

Software-Defined Networking has emerged as a new network paradigm to innovate the ossified network infrastructure by separating the control plane from the data plane (e.g., switches), as well as providing holistic network visibility and flexible programmability. As the brain of the network, a SDN controller grants users a great tool to design and control. The first two authors contribute equally to the project. In real-world production networks, SDN, particularly its popular realization OpenFlow1, has been increasingly employed.

Many application scenarios have been studied and deployed. Since the controller is the core of the Software Defined Network architecture, if the Open Flow controller suffers from any serious vulnerability in its design/implementation. Identify the new attacks that an attacker can exploit to poison the network topology information in Open Flow networks.

The whole network-wide visibility is one of the key innovations provided by Software Defined Network compared to legacy networking technologies. As a fundamental building block for network management, the topology information is adopted to most controller core services and upper-layer apps, e.g., those related to packet routing, mobility tracking, and network virtualization and optimization.

However, if such fundamental network topology information is poisoned, all the dependent network services will become immediately affected, causing catastrophic problems. For example, the routing services/apps inside the Open Flow controller can be manipulated to incur a black hole route or man-in-the middle attack.

Residential broadband consumption is growing rapidly, increasing the gap between Internet service provider (Internet Service Provider) costs

Author for correspondence:
Department of Information Technology , Nandha Engineering College (Autonomous)

and revenues. Meanwhile, proliferation of Internet-enabled devices is congesting access networks. In this paper, we propose a new model content provider explicitly signals fast lane and slow-lane requirements to the Internet Service Provider on a per-flow basis, using open APIs supported through software defined networking (Software Defined Network).

Our first contribution is to develop an architecture that supports this model, presenting arguments on why this benefits consumers (better user experience), Internet Service Providers (two-sided revenue), and content providers (fine-grained control over peering arrangement). Our second contribution is to evaluate our proposal using a real trace of over 10 million flows to show that video flow quality degradation can be nearly cancelled by the use of dynamic fast-lanes, and web-page load times can be improved by the use of slow-lanes for bulk transfers. Our third contribution is to develop a fully functional prototype of our system using open-source Software Defined Network components (Open flow switches and controller modules) and instrumented video/file-transfer servers to demonstrate the feasibility and performance benefits of our approach. Our proposal is a first step to the long-term goal of realizing open and access network service quality management that is acceptable to users, Internet Service Providers, and content providers alike.

FIXED-LINE Internet Service Providers (ISPs) are increasingly confronting a business problem residential data consumption continues to grow at 40% per annum, increasing the cost of the infrastructure to transport the growing traffic volume. However, revenues are growing at less than four percent per annum, attributable mainly to “flat-rate” pricing. To narrow this widening gap between cost and revenue, Internet Service Providers have attempted throttling selected services which sparked public outcry (resulting in “net neutrality” legislation), and now impose usage quotas, which can stifle delivery of innovative services. It is increasingly being recognized that ensuring sustainable growth of the Internet ecosystem requires a rethink of the business model, that allows Internet Service Providers to exploit the service dimension to differentiate their offerings and tap into new revenue opportunities .[1-5]

PROBLEM DEFINITION

Several verification approaches were often used to debug and check network invariants. Very Flow presents a layer between the control plane and the data plane that monitors network state updates and verifies the violations of invariants dynamically at real time. The previous introduces a real-time network-wide policy checking tool using Header Space Analysis (HSA).

In existing work uses model checking and symbolic execution to find network software bugs in Open Flow applications. This work approach for testing the interoperability of Open Flow switches with reference implementations. It designs and presents the first machine-verified SDN controller based on Net Core. It introduces a verification tool that takes the software program of a data plane as input and check target properties.

These verification solutions only verify the logic correctness of the control plane and data plane, however fail to locate the network topology exploitations discussed in this project. One insight behind Network Topology Poisoning Attacks stems from the centralized network visibility that Open Flow Controller offers to lessen onerous network management tasks. Unfortunately, our study in this project shows that this function could be exploited if not carefully designed, thereby incurring serious security threats. [6-10]

SYSTEM ANALYSIS

Feasibility study

Preliminary investigation examine project feasibility, the likelihood the system would be useful to the organization. The objective of the feasibility study was to test the Technical, Operational and Economical feasibility for including new modules and debugging old running system. All system is feasible if they are unlimited resources and many time. There are aspects in the feasibility study portion of the investigation:

- Technical Feasibility
- Operation Feasibility

- Economical Feasibility

Technical feasibility

The technical issue usually raised to during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required for use the new system?
- Will the proposed system provide response to inquiries, regardless of the number (or) location of users?
- Can the system be upgraded is developed?
- Are there technical guarantees for accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to needs of 'Secure Infrastructure Implementation System'. The current system developed one is technically feasible. It is a web based user interface for audit workflow at DB2 Database. Thus it is provides an easy access to the users. The database purpose was to created, an establish and maintain a work among various entities in order to facilitate all concerned users in their various roles. Permission to the users will be granted based on the roles specified.

Therefore, it provides the technical of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at Network Interface Card or are available as free as open source. The work for the project is done with the equipment and existing software technology. Necessary bandwidth exists for providing a feedback to the users irrespective of the number of users using the system.

Operational feasibility

Proposed projects are beneficial only if they can be turned out to information system. That will meet the organization is operating requirements. Operational feasibility aspects for the project are to be taken as an important part of the project implementation. Some of the important issues raised is to test the operational feasibility of a project includes the following: -

- Is there sufficient support is the management from the users?
- Will the system be used and work properly if it is developed and implemented?
- Will there are any resistances from the user that will undermine the possible application benefits?

This system is targeted to be accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken to consideration. So there are no questions of resistance from the users that can undermine the possible application benefits.

The well-planned design will ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Economic feasibility

A system can be developed technically and would be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost is creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The systems are economically feasible. It doesn't require any addition hardware or software. Since the interface for this systems are developed using the existing resources and technologies available at Network Interface Card, There is nominal expenditure and economic feasibility for certain. [11-15]

EXISTING SYSTEM

In Existing System Software-Defined Networking (SDN) is a new programmable network framework that decouples the control plane from the data plane. An SDN application in the control plane generates complicated network functions such as computing a routing path, monitoring network behavior, and managing network access control.

PROPOSED SYSTEM

In order to mitigate such attacks, we investigate Tope Guard (Topology Guard) possible defense strategies. We note that it was difficult to simply use static configuration to solve the problem (similar to using static ARP entry for hosts or the port security features for solved ARP poisoning attacks), because it requires tedious and error-

prone manual effort and is not suitable for handling network dynamics, which is a valuable innovation of SDN. To better balance the security and usability, in this project, we propose Tope Guard, a new security extension to the existing Open Flow controllers to provide automatic and real-time detection of network topology exploitation.

By utilizing SDN-specific features, Tope Guard checks precondition and post condition to verify the legitimacy of host migration and switch port property to prevent the Host Location Hijacking Attack and the Link Fabrication Attack.

IP SPOOFING ATTACKS

In networking, Internet Protocol Internet Protocol spoofing is the creation of Internet Protocol (IP) packets with a source Internet Protocol address, for the purpose of another computing system. The protocol specifies that each Internet Protocol packet must have a header which contains the IP address of the sender of the packet. The source IP address is normally the address the packet went sent from, but the sender's address in the header can be altered, so the recipient it appears that the packet came from another source. The protocol requires that receiving computer to send back a response to the source IP address, so that spoofing was mainly use to when the sender can anticipate the network response or doesn't care about the response. It may provide information on the region, city and town when on the packet was sent. It doesn't provide information on the identity of the sender or the computer being used.

EXPERIMENTAL SETUP

Programs have been written that execute the functions of the network owner, user and sensor node. To implement Tope guard SDN with the data hash chain method. The following functionalities are added to the user side program of Tope guard: construction of data hash chain of a round of dissemination data, generation of the signature packet and all data packets. Based on the design of Tope guard, SDN.

We implement the verification function for signature and data packets based on the verify function and Link Fabrication attack hash function. Also, in our experiment, when a network user that is laptop user disseminates data items, it first sends them to the serial port of a specific sensor node in the network which is referred to as repeater. Then, the repeater carries out the dissemination on behalf of the user using Tope guard SDN.

The following metrics are used to evaluate Tope guard SDN; IP Attacks overhead, execution time of spoofing operations and propagation delay, and energy overhead. The IP Attacks overhead measures the required data space in the implementation. The propagation delay is defined as the time from construction of a data hash chain until the parameters on all users corresponding to a round of disseminated data items are updated.

Secure data transmission is the need of the hour in any wireless network due to the broadcasting facilities used in such networks. Due to many special features and constraints wireless sensor networks differ in many ways from ad-hoc networks. Some of the security goals to be achieved in general in wireless sensor networks include: Confidentiality of data, Integrity of data, Authentication for data, Access control, Data availability, Non-repudiation, Authorization Some of the specific security goals to be achieved in WSN include: Efficiency, Scalability, Freshness of data, Survivability of network, Forward and backward secrecy of data When a dissemination protocol is designed and developed to be used in wireless sensor networks all these security issues must be considered otherwise the attackers can easily get into the system and steal out the critical data. Security breaches in WSN are of various types and their effect can be catastrophic in nature if not dealt with carefully.

Wireless Sensor Networks are a wide and open area in networking research, which is increasingly being deployed for monitoring applications. This are need for quick and efficient disseminating data and code to users to reprogram to suite the current needs of the application. This was achieved by making use of data dissemination protocols.

It was administrated by the owner and accessible by many users. The users are usually resource constrained with respect to IP Attacks space, computation capability, band-width, and power supply. Thus, a sensor node can only perform a limited number of public key spoofing

operations during the lifetime of its battery. The network users use to some mobiles to disseminate data items into the network. The network owners are responsible for generating keying materials. It can be offline and it was assumed to be un comprisable.

Networks users are assigned dissemination privileges by the trusted authority in a PKI on behalf of the network owner. However, the network owner may be, for various reasons, impersonate network users to disseminate data items.

Table 4.1: Running Time for each phase of the basic protocol of Tope guard SDN (Except the sensor node verification phase)

	System Initialization	**IP-Link Fabrication Attacks**	**The certificate generation (i.e., signing a 20 byte message)**
Time(CPU = 1.8GHz) (µs)	1608.0	1576.31	634.8
Time (CPU = 2.6 GHz (µs)	1111.3	1092.12	435.4
Time (CPU = 3.1 GHz) (µs)	931.1	915.18	372.3

Table 4.1 shows the execution times of some important operations in Tope guard SDN. For example, the execution times for the system initialization phase and signing a random 20-byte message (i.e., the output of proposed algorithm) are 1.608 and 0.6348 MS on a 1.8-GHz Laptop PC, respectively. Thus, if proposed Topogaurd SDN is used, generating a user certificate or signing a message takes 0.6348 MS on a 1.8-GHz Laptop PC.

CONCLUSION

The Poisoning Network routing has been developed in such a structured manner which is reducing the traffic further development. The coding was done in simplified manner as they are more understandable and flexible. The evaluate the effect of varying network and protocol parameters in order to observe the performance trends using the poison -aware traffic allocation formulation. In particular, we are interested in the effect of the update relay period and the maximum number of routing paths on the performance of the flow allocation. In order to compare trials with different update times or numbers of paths, we average the simulated results over each simulation run, yielding a single. We simulate a small-scale network similar to that in while varying network and protocol parameters in order to observe performance trends are made for further developments.

REFERENCES

[1]. Y. Yiakoumis et al., “Putting home users in charge of their network,” in Proc. ACM UbiComp, 2012, 1114–1119.

[2]. Y. Yiakoumis, K. Yap, S. Katti, G. Parulkar, and N. McKeown, “Slicing home networks,” in Proc. SIGCOMM HomeNets Workshop, 2011, 1–6.

[3]. J. Matias, E. Jacob, N. Katti, and J. Astorga, “Towards neutrality in access networks: A NANDO deployment with OpenFlow,” in Proc. Int. Conf. Access Netw., 2011, 7–12.

[4]. A.Kumar et al., “BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing,” in Proc. ACM SIGCOMM, 2015, 1–14.

[5]. C.Joe-Wong, S. Ha, and M. Chiang, “Time-dependent broadband pricing: Feasibility and benefits,” in Proc. IEEE ICDCS, 2011, 288–298.

[6]. P. Danphitsanuphan, “Dynamic bandwidth shaping algorithm for Internet traffic sharing environments,” in Proc. World Congr. Eng., 2011, 1–4.

[7]. Nikolaos Laoutaris, Michael Sirivianos, Xiaoyuan Yang, and Pablo Rodriguez “Inter-Datacenter Bulk Transfers with NetStitcher” Telefonica Research Barcelona, Spain nikos@tid.es, irivi@tid.es, yxiao@tid.s,

pablorr@tid.es

[8]. A.Mahimkar et al., "Bandwidth on demand for inter-data center communication," in Proc. ACM HotNets Workshop, 2011, 24.

[9]. R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. 2011, 485–492.

[10]. H. H. Gharakheili, A. Vishwanath, and V. Sivaraman, "Pricing usersanctioned dynamic fast-lanes driven by content providers," in Proc. IEEE INFOCOM Workshop Smart Data Pricing (SDP), 2015, 528–533.

[11]. S. Sundaresan et al., "Broadband Internet performance: A view from the gateway," in Proc. ACM SIGCOMM, 2011, 134–145.

[12]. V. Sivaraman, T. Moors, H. H. Gharakheili, D. Ong, J. Matthews, and C. Russell, "Virtualizing the access network via open APIs," in Proc. ACM ConNEXT, 2013, 31–42.

[13]. Kok-Kiong Yap Te-Yuan Huang Ben Dodson Monica S. Lam Nick McKeown "Towards Software-Friendly Networks" Stanford University {yapkke,huangty,bjdodson,lam,nickm}@stanford.edu

[14]. Barbara van Schewick "Network Neutrality: What A Non-Discrimination Rule Should Look Like" Paper presented at the 38th Research Conference on Communication, Information and Internet Policy, Arlington, VA.2010.

[15]. Erik Nordstr¨om, David Shue, Prem Gopalan, Robert Kiefer Matvey Arye, Steven Y. Ko, Jennifer Rexford, Michael J. Freedman "Serval: An End-Host Stack for Service-Centric Networking" Princeton University.