Dynamically packet trajectory tracing algorithms for Software-Defined Networks

Banumathy. S
Department of CSE,
Banumathy195@gmail.com

Maharasi. M ,
Department of CSE,
rosesweet1703@gmail.com

Beslin Pajila .P.J,
Department of CSE,
Beslin.kits@gmail.com

1.Abstract Today's routing protocols analytically rely on the assumption that the primary hardware is reliable. Certain the increasing number of attacks on network devices, and recent reports on hardware backdoors this assumption has become questionable. Certainly, with the critical role computer networks play today, the difference between our security assumptions and reality is problematic. In this section, we describe our Adversarial Trajectory Sampling (ATS) scheme system which provably detects a wide range of routing attacks in SDNs. This paper presents Software-Defined Adversarial Trajectory Sampling (SOFTATS), an Open Flow-based mechanism to efficiently monitor packet trajectories, also in the presence of non-cooperating or even adversarial controls or routers, e.g., containing hardware backdoors. Our approach is based on a secure, redundant and adaptive sample distribution scheme which allows us to detect adversarial switches or routers trying to redirect, mirror, drop, inject, or modify packets (i.e., header and/or payload). We calculate the efficiency of our approach in different adversarial settings, report on a proof-of-concept implementation, and provide a first evaluation of the performance overheads of such a scheme. Using the flow information, UNIROPE dynamically selects one of the two proposed packet trajectory tracing algorithms to achieve a better trade-off between accuracy and efficiency.

Keywords GUPA, RFID, Soft ATS

2.INTRODUCTION

RADIO frequency identification (RFID) as an emerging sensor technique has been developed in various applications. Due to the limited communication resources and computation capabilities, several problems restrict its extensive development. Conventional cryptographic primitives have low portability on low-cost tags with inadequate power and storage, which may make security issue more formidable Software-Defined Networking (SDN) has emerged as a key technology to make datacenter network management easier and more fine-grained. SDN allows network operators to express the desired functionality using high-level abstractions at the control plane, that are automatically translated into low-level functionality at the data plane. However, debugging SDN-enabled networks is challenging. In addition to network misconfiguration errors and failures network operators need to ensure that operations at the data plane conform to the high-level policies expressed at the control plane. Noting that traditional tools (e.g., Net Flows, Flow, SNMP, traceroute) are simply insufficient to debug SDN-enabled networks, a number of tools have been developed recently.

A particularly interesting problem in SDN debugging is to be able to reason about flow of traffic (e.g., tracing individual packet trajectories) through the network [4,3]. Such a functionality enables measuring network traffic matrix [2], detecting traffic anomalies caused by congestion, localizing network failures [3,4,5], or simply ensuring that forwarding behaviour at the data plane matches the policies at the control plane [1]. We discuss related work in depth in §5, but note that existing tools for tracing packet trajectories can use one of the two broad approaches. On the one hand, tools like Net Sight [1] support a wide range of queries using after-the fact analysis, but also incur large "out-of-band" data collection overhead. In contrast, "in-band" tools (e.g., Path Query [5] and Path let Tracer significantly reduce data collection overhead at the cost of supporting a narrower range of queries. We present Cherry Pick, a scalable, yet simple "in-band" technique for tracing packet trajectories in SDN-enabled data centre networks. Cherry Pick is designed with the goal of minimizing two data plane resources: the number of switch flow rules and the packet header space. Indeed, existing approaches to tracing packet trajectories in SDN trade off one of these resources to minimize the other. At one end of the spectrum is the most naïve approach of assigning each network link a unique identifier and switches embedding the identifier into the packet header during the forwarding process.

This minimizes the number of switch flow rules required, but has high packet header space overhead especially when the packets traverse along non-shortest paths (e.g., due to failures along the shortest path). At the other end are techniques like Path let Tracers that aim to minimize the packet header space, but end up requiring a large number of switch flow rules Path Query [4] acknowledges a similar limitation in terms of switch resources. Cherry Pick minimizes the number of switch flow rules required to trace packet trajectories by building upon the naïve approach — each network link is assigned a unique identifier and switches simply embed the identifier into the packet header during the forwarding process. However, in contrast to the naïve approach, Cherry Pick minimizes the packet header space by selectively picking a minimum number of essential links to represent an end-to-end path. By exploiting the fact that data centre network topologies are often well-structured, Cherry Pick requires packet header space comparable to state-of-the-art solutions [2], while retaining the minimal switch flow rule requirement of the naïve approach. For instance, Table 1 compares the number of switch flow rules and the packet header space required by Cherry Pick against the above two approaches for a 48-ary fat-tree topology.

3.Existing System

Many researchers worked on proposes a novel Cherry Pick: Tracing Packet Trajectory in Software-Defined Datacenter Networks [5]. Enabling Layer 2 Path let Tracing through Context Encoding in Software defined networking [3]. Leveraging SDN layering to systematically troubleshoot networks [2]. Dynamically packet trajectory tracing algorithms for software-defined networking**.** Due to the limited communication resources and computation capabilities, several problems restrict its extensive development. Particularly, security issues are increasingly concerned in recent studies, and are also opposing with severe challenges. Conventional cryptographic primitives have low portability on low-cost tags with inadequate power and storage, which may make security issue more difficult.

4.PROPOSED SYSTEM

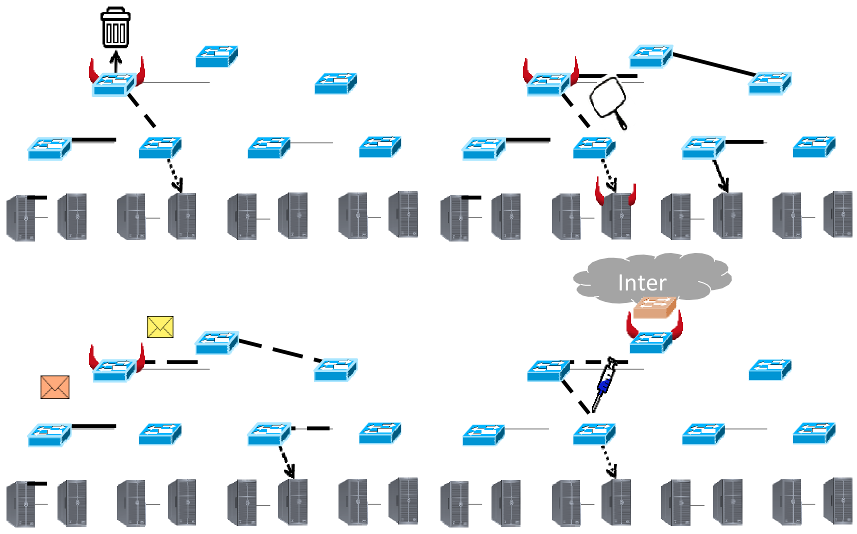

Fig.1 Malicious Switch attacks.

Overview of possible malicious switch attacks(fig1). As an example, a Clos (“fat-tree”) topology is depicted (for ease of representation, we aggregate links in this figure, for a full representation): servers are organized into racks, and are interconnected via so-called *Top-of-Rack (TOR)* switches. Racks are connected by aggregation switches to form *pods*. Finally, pods are connected by core switches, which may also connect the data centre to the Internet. *Top left* Denial-of-service attack resp. packet drop: instead of forwarding the packet to the server in the second rack (*dashed path*), the malicious switch drops the packet. *Top right:* The malicious switch injects a copy of the packet to the rack (*dashed path*), in addition to sending it along the regular path (*solid path*). In the rack where the packet is mirrored to, a malicious server may filtrate confidential information. *Bottom left* A malicious switch modifies the packet along the route (man-in-the-middle attack). *Bottom right:* A malicious core switch injects a harmful packet to attack an internal server (an insider attack).

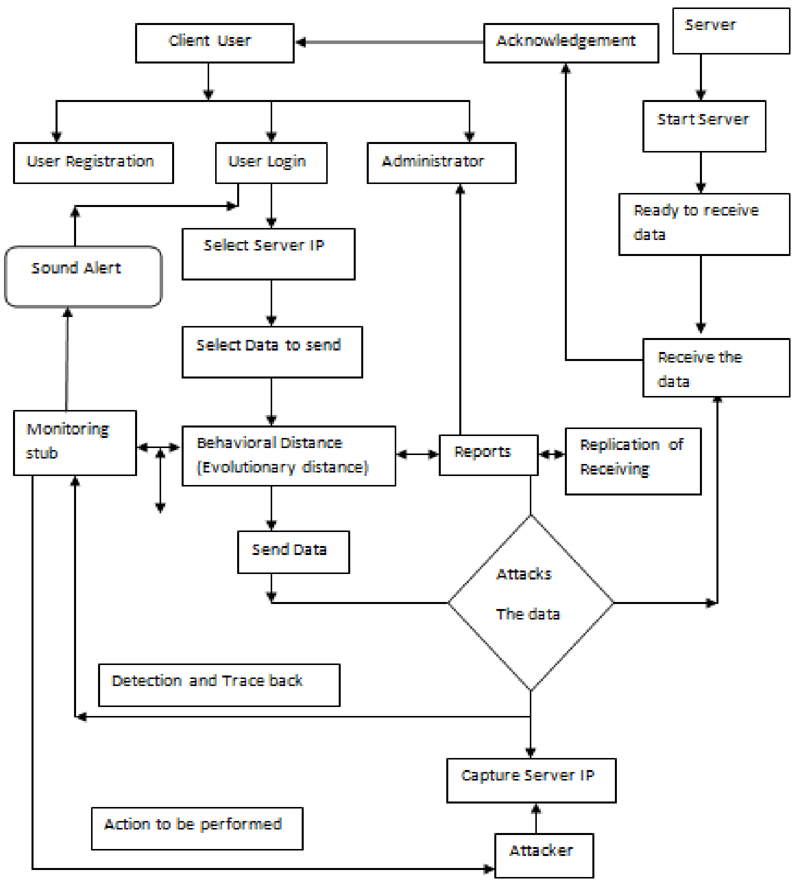

Figure.2 System Architecture

Many modules are Authentication module, Connected Network**,** Secure data Transfer, Reports**.** This authentication module holds the user and the administrator authentications. The admin will have permission to view the whole processes done by the user. The user can only view the real page after getting registered to the approach. User can view their personal information and the data which sent by him. In the server module have the static and secure login to enter and starts the server to receive the data. The connected network system has divided by workgroups. This component will help us to linked and the active systems in the network. After login to our process, this module will connected to systems and shows to the users. The user can select the system to deliver their data by file transfer. The detached and the power failure systems are not visible in the file. The secure data transfer has to select the system to transfer the data and the file to be transferred. The selected file will be encrypted for secured transfer. When the data received by the desired path of destination, the key automatically enabled and decrypted. When the user starts the process, the monitoring stub will initiate automatically to find behavioural distance and the evolutionary distances. All the data transactions and intruder information are forward to the administrator. The administrator can view all the reports and monitor the network paths. The whole histories of data are maintained by the administrator. So that, the administrator can able to make the denial of service of the intruder from the reports module.

5.Conclusion

In this paper, they considered the joint node and link attacks because it poses the major threat to the network. They proposed the performance to assess the vulnerability of the network. Further, have to solve the resource provision to the network and have to construct the efficient route path to attain the minimum network flow between each node. The process is experimentally evaluated in terms of real time transmission between the source and destination which are considered from the nodes those are considered as the whole network then reconstructed by the process which have satisfies the criteria in the process as the small cut ratio. In the proposed system we have improve the transmission performance by avoiding the attacked link and node. But here we need to improve the resource allowance process between the deployed in the reconstructed network. That work has to be investigating in the future work hence we can enhance the whole transmission performance based on the user priority in the deployed arrangement.

References

1. M. Yu, L. Jose, and R. Miao, "Software defined traffic measurement with open sketch," in *Proc. USENIX NSDI*, 2013, pp. 29–42.

2. B. Heller *et al.*, "Leveraging SDN layering to systematically troubleshoot networks," in *Proc. ACM Hot SDN*, 2013, pp. 37–42.

3. H. Zhang *et al.*, "Enabling layer 2 path let tracing through context encoding in software-defined networking," in *Proc. ACM Hot SDN*, 2014,

pp. 169–174.

4. K. Bu *et al.*, "Is every flow on the right track Inspect SDN forwarding with rule scope," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

5. P. Tam mana, R. Agarwal, and M. Lee, "Cherry Pick: Tracing packet trajectory in software-defined datacenter networks," in *Proc. ACM SOSR*,

2015, pp. 23:1–23:7.