Detecting and Securing of IP Spoofing Attack by using SDN

[1]Shalini A, [2]Nivedha L, [3]Priyadharshini K, [4]Sridevi P, [5]Vinothini R

[1]Assistant Professor, Department of Computer Science and Engineering, Kingston Engineering College,
[2,3,4,5]UG Scholar, Department of Computer Science and Engineering, Kingston Engineering College, Katpadi, Vellore, Tamilnadu.

---***---

Abstract - *IP address spoofing or IP spoofing is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system. The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol (IP).The protocol specifies that each IP packet must have a header which contains (among other things) the IP address of the sender of the packet. Software defined networking (SDN) technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring making it more like cloud computing than traditional network management. In this we use SDN architecture for implementing control for IP spoofing by certain algorithm to maintain the flow of Non- used IP by shortest path.*

Key Words: **IP address, spoofing, IP packets, SDN, dynamic, shortest path.**

1. INTRODUCTION

Software-defined networking (SDN) emerged as an attempt to introduce network Innovations faster, and to radically simplify and automate the management of large networks. SDN traditionally leverages Open Flow as device-level abstraction. Since Open flow permits the programmer to "just" abstract a static flow-table, any stateful control and processing intelligence is necessarily delegated to the network controller. Motivated by the latency and signaling overhead that comes along with such a two-tiered SDN programming model, in the last couple of years several works have proposed innovative switch-level (data plane) programming abstractions capable to deploy some smartness directly inside the network switches, e.g., in the form of localized stateful flow processing. Furthermore, the possible inclusion of states and state maintenance primitives inside the switches is currently being debated in the Open Flow standardization community itself. In this paper, after having provided the reader with a background on such emerging stateful SDN data plane proposals, we focus our attention on the security implications that data plane programmability brings about. Also via the identification of potential attack scenarios, we specifically highlight possible vulnerabilities specific to stateful in-switch processing (including denial of service and saturation attacks), which we believe should be carefully taken into consideration in the ongoing design of current and future proposals for stateful SDN data planes. Collaborative intrusion detection systems (IDSs) have a great potential for addressing the challenges posed by the increasing aggressiveness of current Internet attacks. However, one of the major concerns with the proposed collaborative IDSs is their vulnerability to the insider threat. Malicious intruders, infiltrating such a system, could poison the collaborative detectors with false alarms, disrupting the intrusion detection functionality and placing at risk the whole system. In this paper, we propose a P2P-based overlay for intrusion detection (overlay IDS) that addresses the insider threat by means of a trust-aware engine for correlating alerts and an adaptive scheme for managing trust. In current stream processing frameworks, the share of network bandwidth has left to the mercy of the underlying transport mechanisms (e.g., TCP, DCTCP). Nonetheless, such mechanisms are designed mainly for end-to-end data delivery in an application agnostic manner, i.e., flows traversing the bottleneck links sharing equal portion of the bandwidth. This, with high probability, will lead to sub-optimality in the overall application-level performance because some flows can be of paramount importance than other flows of the same application.

In existing, we the design space of the bandwidth allocation, formulate it as a utility maximization problem, and propose a heuristic algorithm to derive the close-to-optimal solutions. This whole procedure is encapsulated into a cross-layer framework which utilizes the additional information measured from the running applications and quickly deploys the new allocation decisions to the physical network layer. The latter is enabled by the rapid development of the Software-Defined Networking (SDN) techniques and toolkits and realized through plugging in a control plane module. The main contributions we have made in this paper are listed as follows:

1. We formulate the bandwidth allocation among flows belonging to a stream processing application as an optimization problem and design a heuristic algorithm to seeking for the optimal allocation solution.

2. Leveraging the SDN capabilities, we develop a native SDN control plane application that deploys and updates the bandwidth allocation results derived by our optimization algorithm.

3. We develop an automated cross-layer bandwidth allocation framework and implement a prototype of it based on a popular open-source stream processing

platform, Apache Storm, integrating with the Open Daylight SDN controller.

4. We carry out comprehensive performance evaluation through running stream data applications with real-world workloads in a local cluster composed of 10 workstations interconnected by a hardware SDN-enabled switch.

5. We introduce an exemplary mechanism for bandwidth sharing and reasoning of performance among multiple active applications and present a case on how to approximate application level fairness.

6. We built a fat-tree like test bed to carry out the evaluation of optimization particulars in a more general setting with a multi-hop network.

In proposed system, in this paper we explore SPM defense at the traffic's destination associates a source autonomous system (AS) with a secret it exchanged with the defense. The source marks packets with this secret. Unique temporal key K(S, D) associated with each pair ordered air of source destination networks. Router closer to the destination verifies authenticity of the source address of the packet. Effective and provides incentive to ISP's implementing SPM.IP address spoofing or IP spoofing is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system. The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol (IP). The protocol specifies that each IP packet must have a header which contains (among other things) the IP address of the sender of the packet. Software-defined networking (SDN) technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring making it more like cloud computing than traditional network management. In this we use SDN architecture for implementing control for IP spoofing by certain algorithm to maintain the flow of Non- used IP by shortest path.

2. Analysis of IP Spoofing Attack

IP address spoofing or IP spoofing is the creation of Internet Protocol (IP) packets with a false source IP address, for the purpose of impersonating another computing system. The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol (IP). The protocol specifies that each IP packet must have a header which contains (among other things) the IP address of the sender of the packet. Software-defined networking (SDN) technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance and monitoring making it more like cloud computing than traditional network management. In this we use SDN architecture for implementing control for ip spoofing by certain algorithm to maintain the flow of Non- used IP by shortest path.

In the existing, IP source address spoofing or IP spoofing attack, it refers to attackers release packets with forged IP source addresses so that they can conceal their real identities and launch attacks, e.g., reflect network traffics to flood victim hosts. Once suffering such attack, it is hard for victim to trace back to perpetrators and identify their real identities, which severely compromises Internet accountability indeed. From the perspective of technique, IP spoofing threat is derived from the design that Internet packet forwarding in routers only relies on packet's destination IP address, but neglects the validation of packet's IP source address to verify sender authenticity. Taking this vulnerability, attackers can launch serious attacks against specified targets, and as a matter of fact, most of attack directly related with this volubility, i.e., TCP-SYN flooding DDOS. It doesn't support bandwidth allocation among flows will be static, defined Software Protocol was not implemented so lack of IP address consumption occur.

In the proposed approach we explore SPM defense at the traffic's destination associates a source autonomous system (AS) with a secret it exchanged with the defense. The source marks packets with this secret. Unique temporal key K(S,D) associated with each pair ordered air of source destination networks. Router closer to the destination verifies authenticity of the source address of the packet. Effective and provides incentive to ISP's implementing SPM. It supports Packet leaving a source network S tagged with Key K(S,D).Destination network upon reception of packet verifies the packet using the key & then removes the key. Keys are changed periodically.

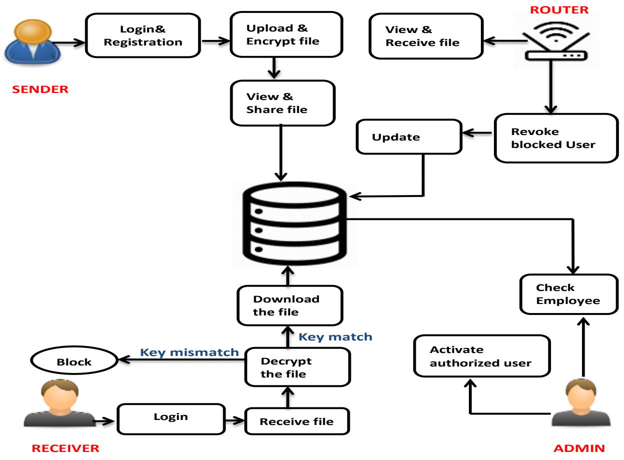


Figure. Architectural Diagram

3. MODULES

Let's us illustrate a sample of the model of a detection of spoofing attack.

- User registration.
- File sharing and uploading.
- Check incoming file and download a file.
- Block and revoke a file.

USER REGISTRATION

User:

The Owner associated items are viewed by the users and the results can be summed up from the users, because they are using this system and according to their response the association technique works.

Registration:

A registered user is a user of a website program or other system who has previously registered. Registered users normally provide some sort of credentials (such as a username or e-mail address, and a password) to the system in order to prove their identity: this is known as LOGGING IN. Systems intended for use by the general public often allow any user to register simply by selecting a register or signup function and providing these credentials for the first time. Registered users may be granted privileges beyond those granted to unregistered users.

Login:

Logging in is usually used to enter a specific page, which trespassers cannot see. Once the user is logged in, the login token may be used to track what actions the user has taken while connected to the site.

FILE SHARING AND UPLOADING

Share the File:

File sharing is the practice of sharing or offering access to digital information or resources, including documents, multimedia(audio/video), graphics, computer programs, images and e-books.it is the private or public distribution of data or resources in a network with different levels of sharing privileges.

Upload the File:

Uploading is the transmission of a file from one computer system to another, usually larger computer system. From a network user's point-o-view, to upload a file is to send it to another computer that is set up to receive it.

CHECK INCOMING FILE AND DOWNLOAD A FILE

Incoming file:

In this module the receiver can view the sender file using IP address. The receiver can get the private key or the shared file when the receiver is authorized.

In this module the receiver can view the sender file using IP address. The receiver can get the private key or the shared file when the receiver is authorized.

Download a File:

The Authorized receiver can decrypt the file using the key and download the file.

BLOCK AND REVOKE A FILE

Block:

In this module, the router will block the authorized user if the key was mismatched and also the unauthorized user.

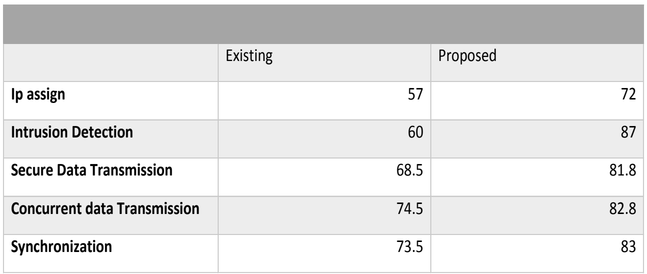
Revoke:

The Router will revoke the user if the user has valid login details.

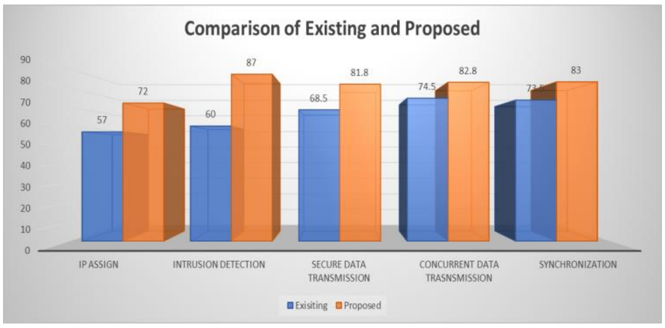
SPOOFING PREVENTION METHOD

We describe the spoofing attack by SPM ,it defense at the traffic's destination associates a source autonomous system (AS) with a secret it exchanged with the defense. The source marks packets with this secret. Unique temporal key K(S,D) associated with each pair ordered air of source destination networks. Router closer to the destination verifies authenticity of the source address of the packet.

Packet leaving a source network S tagged with Key K(S,D).Destination network upon reception of packet verifies the packet using the key & then removes the key. Keys are changed periodically.



	Existing	Proposed
Ip assign	57	72
Intrusion Detection	60	87
Secure Data Transmission	68.5	81.8
Concurrent data Transmission	74.5	82.8
Synchronization	73.5	83



CONCLUSION

The current system can be implemented based on Software Defined Network (SDN), which provides security for data transmission between end users. This is achieved by verifying the key in every data packet that is passing through the router. Router is maintained by separate admin. If any kind of spoofing (or) hacking monitored by the admin, they are immediately blocked from using this service. There are chances, that legitimate users key may have some error, which as identified as spoofing by admin and immediately blocked. In this situation, the user could request the admin for revocation of the service. We expect that our proposed scheme be very useful in protecting the data sharing over the networks.

FUTURE ENHANCEMENT

This system can be improved by increasing the volatility of IP address, which means increasing the frequent changes in the dynamic IP address. The range of users can be increased by increasing the bandwidth. The system security and data sharing capabilities can be improved by implementing the peer-to-peer network connections.

REFERENCES

1. Mohamed Aslan, Ashraf Matrawy.,'On the Impact of Network State Collection on the Performance of SDN Applications'., 2016.

2. Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, Mauro Conti.,'A Software Engineering Perspective on SDN Programmability'.,2016.

3. TooskaDargahi, Alberto Caponi, Moreno Ambrosin, Giuseppe Bianchi, Mauro Contil.,'A Survey on the Security of Stateful SDN Data Planes'.,2017.

4. Duma. C, Karres. M and, Shahmehri.N and Caronni. G.,'A Trust-Aware, P2P-Based Overlay for Intrusion Detection'., 2018.

5. Cho.J, Chang.H, S.Mukherjee, T.Lakshman, and J.Van der Merwe., 'Typhoon:An SDN enhanced real-time big data streaming framework'., in Proc. ACM Context., 2017, pp. 310–322.

6. W.Aljoby, X.Wang, T.Fu, and R.Ma.,'On SDN-enabled online and dynamic bandwidth allocation for stream analytics'., in Proc. IEEE ICNP, Sep. 2018, pp. 209–219.

7. A.Toshniwal et al., 'StormTwitter'., in Proc. ACM SIGMOD, 2014,pp. 147–156.

8. S.Kulkarni et al., 'Twitter heron: Stream processing at scale'.,inProc.ACM SIGMOD, 2015, pp. 239–250.

9. S.A.Noghabi et al., 'Samza: Stateful scalable stream processing at linkedin'., in Proc. VLDB, 2017, pp. 1634–1645.

10. P.Carbone, A.Katsifodimos, S.Ewen, V.Markl, S.Haridi, and K.Tzoumas, 'Apache flink: Stream and batch processing in a single engine'., IEEE Data Eng. Bull., vol. 38, no. 4, pp. 28–38, Dec. 2015.

11. T.Akidau et al., "MillWheel:'Fault-tolerant stream processing at Internetscale'., in Proc. VLDB, 2013, pp. 1033–1

12. C.Lin, J. Zhan, H.Chen, J.Tan, and H.Jin., 'Ares: A high performanceand fault-tolerant distributed stream processing system'., in Proc. IEEE ICNP, Sep. 2018, pp. 176–186.

13. M.Stonebraker, U.�tintemel, and S.Zdonik., 'The 8 requirements of real-time stream processing'., ACM SIGMOD Rec., vol. 34, no. 4,pp. 42–47, 2005.

14. M.Kleppmann, Designing Data-Intensive Applications:'The Big Ideas Behind Reliable, Scalable, and Maintainable Systems'.,O'ReillyMedia,Apr. 2017.

    A. Bremler-Barr and Y. Koral.,'Accelerating multipattern matching on compressed HTTP traffic'.,Mar.2017.

    B. El-Atawy, E.Al-Shaer, T. Tran, and R. Boutaba.,'Adaptive Early Packet Filtering for Defending Firewalls against DOS Attacks'.,Vol.24,no.4.,2005.

15. C.J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba.,'Trust Management for Host-Based Collaborative Intrusion Detection'., Mar.2005.

16. C.J. Fung, Q. Zhu, R. Boutaba, and T. Basar.,'Bayesian Decision Aggregation in Collaborative Intrusion Detection Networks'.,Vol.24,no.4.,2005.

17. A.K. Ghosh, J. Wanken, and F. Charron.,'Detecting Anomalous and Unknown Intrusions Against Programs'., Apr.2016.

18. S. Ioannis, D. Vasilis, P. Dionisios, and V. Stamatis., ’Packet Pre-filtering for Network Intrusion Detection’., vol. 34, no. 4,pp. 42–47, 2005.

19. H. Kim, H.-S. Kim, and S. Kang., ’A memory-efficient bit-split parallel string matching using pattern dividing for intrusion detection systems’., 2017.

20. Y. Meng, L.-F. Kwok, and W. Li., ’Towards Designing Packet Filter with a Trust-based Approach using Bayesian Inference in Network Intrusion Detection’., in Proc. IEEE ICNP, Sep. 2018, pp. 176–186.

21. Y. Meng and L.-F.Kwok., ’Adaptive Blacklist-based Packet Filter with A Statistic-based Approach in Network Intrusion Detection’., vol. 34, no. 4,2003.

22. D. Quercia, S. Hailes, and L. Capra., ’B-Trust: Bayesian Trust Framework for Pervasive Computing’., in Proc. VLDB, 2017, pp. 1634–1645.

23. D. Pao and X. Wang., ’Multi-stride string searching for high-speed content inspection’., vol. 38, no. 4, pp. 28–38, Dec. 2015.