Poster: Denial of Service Defence using Router Migration

Jonathan Weekes, Shishir Nagaraja
Security Lancaster Research Centre
Lancaster University
United Kingdom, LA1 4WA
Email: {j.weekes,s.nagaraja}@lancaster.ac.uk

***Abstract*—Software Defined Networking (SDN) is a new network paradigm which separates the data plane from the control plane in a network[1], [2], [3]. It presents radical new capabilities for networking, increasing performance and security among other properties. This work will look at improving network security, particularly against Link-Flooding Denial of Service (DoS) attacks which are more critical, but less explored than server based Denial of Service attacks. This work proposes to implement a system which increases network security against DoS attacks using the new facilities made available through SDN. The system seeks to move the network state of a router to one or more different routers by migrating flows so that under attack, a link router can transfer some or all of its flows to another router to reduce its load and mitigate the attack**

I. Introduction

Denial of Service attacks are one of the most prevalent attacks against computer networks. One method is by flooding a resource with such high volumes of traffic that legitimate users are unable to receive the intended service [4], [5], [6], [**?**]. The most common (but not only) form of Denial of Service attacks is Distributed Denial of Service attacks (DDoS). Attackers carry out this attack by using a number of different sources for the purpose of flooding. No conclusive and foolproof method has been found to deal with these attacks. Most methods attempt to filter out the attack flows from the legitimate flows using some sort of detection method in order to restrict the attack flows while still providing service to the legitimate flows. Unfortunately, many times, legitimate flows are also filtered out in the attempt to stop the attack (collateral damage), therefore the goal of the attack is still being achieved. Link-Flooding attacks such as the Coremelt attack described in [7] and the Crossfire attack described in [8] are a particularly potent type of DDoS attack which floods a link in the network in an attempt to disconnect the computers on either side of the link. This packet flooding is done until the router can handle no more traffic and is too congested to deliver reasonable service to the target area. This attack is extremely difficult to detect as most DDoS detection schemes work at the point of the server rather than the link and often the attack flows do not raise suspicion because they maintain a low intensity which does not trigger any alarms at the servers. Attacks such as these can be shown to disconnect entire countries or enterprises from the Internet making it particularly devastating. Two reasons it is so difficult to defend against these attacks is because attack traffic often looks no different from legitimate traffic [9], [**?**] and DoS solutions are often created to be deployed at the server. Legitimate traffic which suddenly increases is called a flash crowd and is a perfectly legitimate internet phenomenon (versus DoS attacks which have malicious intent). It is often caused by an event of some sort (such as natural disaster or death of a famous person) which causes users to collectively become interested in information on a particular website. Many solutions involve attempting to filter out the DDoS traffic or analysing patterns to tell which traffic is a DDoS attempt, however attackers are able to find ways around this. Solutions also attempt to deal with the traffic at the server being attacked by which time it would be too late to do anything about a Link-Flooding attack.

II. Proposed Methodology

The solution which is the subject of this project ignores this issue by treating all traffic as legitimate traffic. It will allow a network switch/router to increase its size as a form of defence. The solution put forward in this project transfers a data flow from one switch to another within a network without affecting the end hosts using the network using flow migration. It should be noted that this solution is specific to link-flooding attacks and is not necessarily a general purpose solution. This effectively creates a form of virtual router which is able to increase and decrease its size on demand. It is the first step toward creating a dynamic network topology whose movements are entirely abstracted from the applications running on the network edge. This project will attempt to leverage the key characteristics of the new networking concept- Software Defined Networking- to move the network state of a switch/router over to a different one in the event of a link-flooding attack. Characteristics such as having a centralized controller and the OpenFlow protocol interface make this an ideal tool for this solution. Once the state of the network has been moved, all subsequent packets are then rerouted through to the second switch, thus effectively ending the attack. If all this can be done without affecting the transactions between the end-hosts, i.e. completely transparent to the network edge, an effective defence has been created.

III. Challenges

The concept of router migration brings with it several challenges. Because migration brings about changes in the forwarding topology, operators no longer have the luxury of applying the changes and waiting for the network to converge. This strategy does not cater for topology changes particularly through the re-configurations of several switches at a time. Because of this, the router migration mechanism must ensure consistency. Topology changes must cater for packets already in the network when the changes occur. They must ensure that packets not only reach their destination but do so in a timely manner. Applications running on the network edges may not be able to cope with packets arriving out of order or far later than they should. Also, if not properly done, topology and configuration changes can cause packets to take more than one path to their destination. Packets may be duplicated and may even take all possible paths. Besides the wasted resources caused by this, network security mechanisms may not react well to a single flow using multiple paths. Finally, because of these changes, it is can be difficult to determine what path a packet took for debugging purposes. The research will therefore take mitigation of these challenges into consideration.

References

[1] M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, P. Weissmann, and N. Mckeown, "Maturing of openflow and software-defined networking through deployments," *Comput. Netw.*, vol. 61, pp. 151–175, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bjp.2013.10.011

[2] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "A demonstration of fast failure recovery in software defined networking," in *Testbeds and Research Infrastructure. Development of Networks and Communities*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, T. Korakis, M. Zink, and M. Ott, Eds. Springer Berlin Heidelberg, 2012, vol. 44, pp. 411–414. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35576-9_46

[3] A. Valdivieso Caraguay, L. Barona Lopez, and L. Garcia Villalba, "Evolution and challenges of software defined networking," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, Nov 2013, pp. 1–7.

[4] L. Garber, "Denial-of-service attacks rip the internet," *Computer*, vol. 33, no. 4, pp. 12–17, Apr 2000.

[5] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, May 2006. [Online]. Available: http://doi.acm.org/10.1145/1132026.1132027

[6] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, Apr. 2004. [Online]. Available: http://doi.acm.org/10.1145/997150.997156

[7] A. Studer and A. Perrig, "The coremelt attack," in *Proceedings of the 14th European Conference on Research in Computer Security*, ser. ESORICS'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 37–52. [Online]. Available: http://dl.acm.org/citation.cfm?id=1813084.1813088

[8] M. S. Kang, S. B. Lee, and V. Gligor, "The crossfire attack," in *Security and Privacy (SP), 2013 IEEE Symposium on*, May 2013, pp. 127–141.

[9] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 3, pp. 62–73, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/571697.571724