Distributed Denial of Service(DDoS) and DHCP Masquerading Attack Vectors for Software-Defined Networks

Dogukan Yalcin
Middle East Technical University
Computer Engineering Department
E-mail: e194246@metu.edu.tr

Tugca Eker
Middle East Technical University
Computer Engineering Department
E-mail: e194201@metu.edu.tr

***Abstract*—Software-Defined Network is a state of art technology that constructed over the idea of control and data plane separation in the network. While this property provides many advantages like flexibility, centralization of management and simplification it also results in some security challenges. This report presents application of two attack vectors on SDN by using mininet virtual network environment; DDoS and DHCP Masquerading.**

1. Introduction

Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of todays applications [1]. SDN separates the network forwarding and control functions so that the related engineers or administrators will be able to solve or update the network from a distance.

The aim of this work is to simulate some SDN Specific network attacks and show their potential impacts by working on two specific attack vector; Distributed Denial of Service(DDoS) and DHCP Masquerading.

2. Attack Vector: Distributed Denial of Service (DDoS)

Denial-of-Service (DoS) attacks flood networks with an high-traffic in order to overwhelm the target resources and make it unavailable for non-malicious users. It means that DoS attack aims breaking of the availability component of the CIA triad. If more than one sources are used by the attackers, it becomes ”Distributed” and identified as DDoS. According to Kalkan, Gur and Alagoz considering its design pillars and characteristics, DDoS attacks are also intrinsically viable of Software Defined Networks [2].

SDN allows the decoupling of data and control planes in network devices to centralize control features by limiting the passive capabilities of switches. Since the controller becomes central property, DDoS attacks can cause fatal problems on the network if the controller penetrated with high & malicious traffic.

2.1. Impact and Real-life Examples

IoT (Internet of Things) devices become more popular day by day and they’re increasing in number. In Aug 2016, IoT devices pose enormous security thread against the well-known security journalist Brian Kreb’s web site and French Web Host company OVH (Mirai Botnet). Nearly 400.000 hijacked devices shows impressive peak of 620 Gbps traffic against targets which causes OVH to stop their services. [3]

Likewise Mirai Botnet attack, well-known DDoS attacks such as UDP, ICMP, TCP SYN flooding, NTP amplification, and ping of death are also viable in Software-Defined Networks.

2.2. Design & Implementation

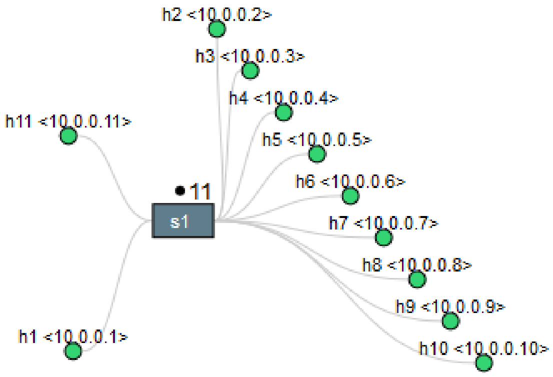

Figure 1. Topology - Distributed Denial of Service.

Requirements:

- Floodlight
- Sflow-RT
- SimpleHTTPServer (Python)
- hping3

2.2.1. Topology. As a network simulation environment we create a linear topology with 11 hosts, a single switch and a single controller. All hosts are directly connected to switch. In our case, h1 is a web server and h11 is a non-malicious user which is always accessing to web server. All other hosts are (namely) IoT devices and they are members of the BotNET which is controlled by malicious attacker.

2.2.2. Setup. To initialize our network, we started by deploying HTTP Server on h1 node, which runs on port 80. To deploy server we used Python's SimpleHTTPServer library and created example index.html file as an example.

As a second step we configure the controller and the switch in our network. For the controller, we used OpenFlow - Floodlight. Also for tracing and monitoring the our network flow we setup a sFlow-RT instance in our environment and add our assets to it. After this step, we are able to manage and monitor our network via web interfaces of OpenFlow and sFlow.

2.3. Launching the Attack

As a simulation of normal network traffic before the attack begins, the host h11 (non-malicious / non-attacker host), continuously requests the content of the index.html page and gets responses from the HTTP Server (host h1).

Launching the attack is done by using hping3 command with flood option to the HTTP server installed on h1. From h2 to h10 all hosts execute this command in order to make the HTTP server unable to respond other non-malicious requests. With each new execution of hping3 the HTTP server deals with increasingly more requests and higher network traffic. After a threshold, HTTP server cannot respond to any message from h11 (non-malicious) since it is busy with others' requests sent from h2-h10 which are unending. Hence h11 is unable to reach to the HTTP server.

2.4. Results & Reasons of Success

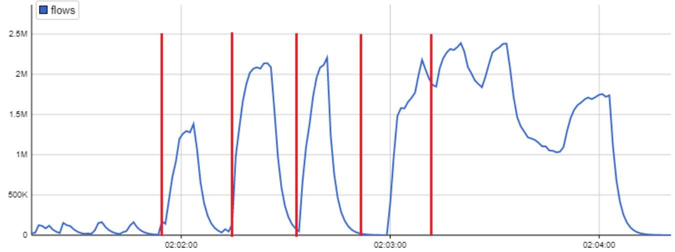

Figure 2. Graphic - Network Flow during DDoS Attack.

The figure above shows the flow-time relationship in the switch(s1). Initially only non-malicious user(h11) sends request to the server and gets related responses for 1 minute(until the first vertical red line in the figure). Afterwards, malicious users started the attack one by one(at each vertical red line a new attacker is joined the attack). The non-malicious user starts getting responses slower and slower with each new attacker. After the 4th attacker the server starts to struggle and eventually becomes unable to respond any requests. Starting from this time, the non-malicious user cannot get any response from the server.

The main reason for this attack's succession is the fact that for both data and control plane limited resources make SDN defenseless against the DDOS attacks. As a consequence, the connection between servers and end users can be blocked in SDNs by cause of the presence of weak points in SDN architecture(Central Controller and separation of control planes).

In conclusion, centralization of management feature of the SDN makes it vulnerable against DDoS attacks and attackers may target controller instruments on the network. This causes congestion on the network and non-malicious users may not be able to use network as they should.

2.5. Solutions & Possible Future Work

In order to deal with different sub-types of the DDoS Attack in Software-Defined Networks, there are several solutions in the literature. These solutions can be categorized as follows [5];

- Table-entry based solutions
- Scheduling based solutions
- Architectural solutions
- Statistical Solutions
- Machine Learning based solutions

If we focus on the Machine Learning based DDoS security solution; this mechanism trains itself with non-malicious / attack-free network packets and then classifies malicious network packets by machine learning algorithms. So, it gives detection ability to network; then different mitigation solutions can be applied on the network. As stated by Gopinat et al; dynamically scalable load balancers may be deployed and controlled by the SDN controller. [6]

3. Attack Vector: DHCP Masquerading

Dynamic Host Configuration Protocol (DHCP) servers are utilized by network providers to provide Internet Protocol (IP) addresses, subnet masks, and gateway information. Hosts which are connected to network have an ability to pop-up rogue DHCP server which is not under the administrative control. Then they may become man-in-the-middle (MitM) and other users may be unaware of this malicious action.

As non-malicious clients connect to network, rogue(masquerading) and legal DHCP servers will propagate the connection information packet which includes IP addresses, default gateway, DNS servers etc. If client accepts the packet which is created by rogue DHCP Server, it creates vulnerability that can be exploited by attacker if we assume that network doesnt have precautionary switches and local DNS cache servers.

3.1. Impact and Sophistication

Deploying a Rogue DHCP Server is an easy task for an attacker who already connected to the network. Clients may not have a chance to understand that there is a man-in-the-middle on the connection between themselves and the destination. Additionally, detecting & stopping the Rogue DHCP server is not an easy task to do. Network needs intrusion detection system with appropriate signatures or multilayer switch structure to stop it.

3.2. Design & Implementation

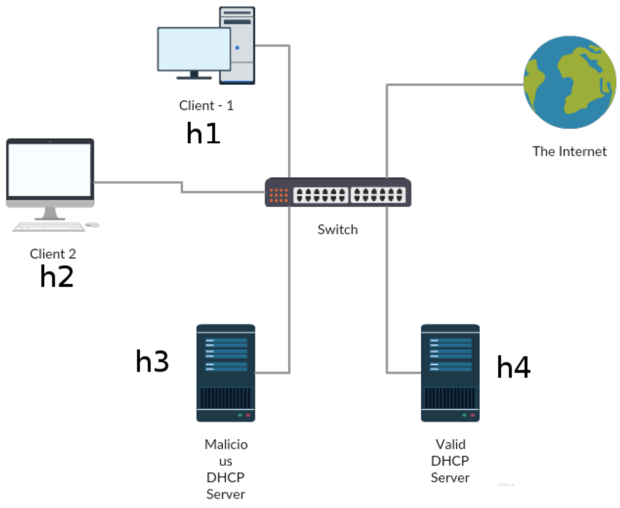

Figure 3. Topology - DHCP Masquerading.

3.2.1. Topology. As a network simulation environment we create a linear topology including 2 host, single good DHCP Server, single rouge (masquerading) DHCP server, single switch and single controller. Additionally in some way switch is connected to the Internet which allows hosts (Client-1, Client-2) to reach any web site like google.com or metu.edu.tr.

To allow Malicious DHCP Server to respond earlier than good one, 500ms to 1s delay added to link between switch and the good DHCP Server (slower link).

To sophisticate, attacker may add ”Browser Mining Snippet” to the requested page’s main page code and send it to victim as a response. Most probably, victim won’t realize that a malicious entity is using it’s system resources to mine cryptocoins.

3.2.2. Setup. After construction of the topology, we need to connect switch (s1) to the Internet. This is done by using mininet’s NAT library. The library connects mininet topology to the Internet via NAT through eth0 on the host (mininet) machine. Then at this point, all hosts are able to connect to the Internet and request the content of the google.com or metu.edu.tr.

Then; DHCP server deployed on the h4 which has slower connection rate, h1 or/and h2 also starts DHCP client using ”dhclient” executable binary of the Linux.

3.3. Launching the Attack

As a simulation of normal network traffic before the attack begins, the clients (h1 and h2) continuously requests web content. At some point attacker gets the control of the host h3 and does following;

1) Deploys malicious DHCP Server to masquerade the request coming from clients (udhcpd)
2) Deploys fake DNS server to redirect clients to fake web-server (dnsmasq)
3) Deploys web server to eavesdrop data communication, sniff or sending malicious responses (Python SimpleHTTPServer)

After that, whenever clients make DHCP Request, request packet forwarded to both good DHCP server and malicious DHCP server. Since good DHCP server has a delayed connection; malicious DHCP server responds first and its response is accepted by the victim client. Malicious DHCP server provides its own network address as the DNS server address and deceives the victim. Then whenever victim make an DNS request (request to google.com initiates a DNS request call to get IP address of google.com), fake DNS server provides its own IP address instead of google.com’s correct IP address.

At this point, victim is connected to the malicious web server thinking it is connecting to google.com. So attacker may request victims credentials by imitating the design of google.com’s main page or do whatever he/she want.

As stated on subsection 3.1, attacker may add ”Browser Mining Snippet” to the Google’s main page code and send it to victim as a response. Most probably, victim won’t realize that a malicious entity is using it’s system resources to mine cryptocoins and earn some ”malicious” money.

3.4. Solutions

There are several methodologies to block attacker to successfully deploy rouge DHCP and become man-in-the-middle. Duangphasuk et al. suggest using digital certificates to identify the legal DHCP communication [7]. Also there is authentication based solution [8]. But still these solutions makes hard the DHCP server or client implementation [9].

Also, there is Software-Defined Network based defensive security solution which constantly monitors flows and resists against DHCP Rouge or DHCP Starvation attacks. It dynamically analyses all DHCP-related information on the packets using packet inspection. Then machine learning algorithm analyzes the data-set so it matches DHCP Server information, gateway and DNS information in the packet to avoid misconfiguration, roguing and any other malicious intention and informs DHCP clients about malicious entities.

3.5. Reasons of Success & Conclusion

Dynamic Host Configuration Protocol (DHCP) is an automated mechanism to distribute IP addresses, gateways, DNS addresses and other network configuration information. By its nature, when DHCP clients propagates DHCP request; the first DHCP response received by host determines the network configuration which will be used by the host. This is the case for every single network system without exception if network does not have any additional security instrument. On this attack vector, we will make use of this commonality.

The fundamental ignorance is letting this attack to success which is the speed of the link(channel) between h4(the non-malicious DHCP Server) and the switch. The attacker observed the slowness of this link and used this information to become a man-in-the-middle with the help of a faster channel between his/her DHCP server and the switch.

4. Conclusion

In this report, we presented two different attack vector applications on SDN domain which is simulated on the Mininet Virtual Network. On DDoS attack vector example, we tried to show the idea of ”Security for SDN” by explaining the vulnerabilities of the SDN against DDoS attacks. And for the DHCP Masquerading attack vector, we stated our arguments to explain ”SDN for Security” idea.

References

[1] Software-Defined Networking (SDN) Definition, *What is SDN?* (2017), Retrived from https://www.opennetworking.org/sdn-definition/

[2] Kalkan, K., Gur, G., & Alagoz, F. (2017). *Defense Mechanisms against DDoS Attacks in SDN Environment.* IEEE Communications Magazine, 55(9), 175-179.

[3] G. Kambourakis, C. Kolias and A. Stavrou, *”The Mirai botnet and the IoT Zombie Armies”* MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 2017, pp. 267-272.

[4] Project Floodlight, *Open Source Software for Building Software-Defined Networks*, http://www.projectfloodlight.org/floodlight/

[5] R. Kokila et al., *”DDoS Detection and Analysis in SDN-Based Environment Using Support Vector Machine Classifier”*, Proc. 2014 IEEE Sixth Int’l. Conf. Advanced Computing, pp. 205-10, 2014.

[6] Y, N., Gopinath, M., & L, G. (2015). *DDoS Mitigation using Software Defined Network.* International Journal of Engineering Trends and Technology, 24(5), 258-264.

[7] Duangphasuk, S., Kungpisdan, S., & Hankla, S. (2011). Design and implementation of improved security protocols for DHCP using digital certificates. *2011 17th IEEE International Conference on Networks.*

[8] T. Aur & M. Roe & S. Murdoch, ”Dynamic host configuration protocol” U.S. Patent 8239549, Aug 7, 2012.

[9] Wang, J., & Chen, Y. (2017). An SDN-based defensive solution against DHCP attacks in the virtualization environment. 2017 IEEE Conference on Dependable and Secure Computing.