DOS Attack on the Availability of Cloud Network and its Avoidance Mechanism

Syed Muhammad Asad Zaidi, Waleed Akram Baig, Hassan Redwan, Ki-Hyung Kim
Ajou University, Suwon, Korea
{asad, waleedbaig, hassanredwan, kkim86}@ajou.ac.kr

ABSTRACT

Data centers are usually under provisioned. This is not a problem in corporate data networks, but it can be a problem in cloud data networks. If an application is being served by that cloud infrastructure, the application owner must know the infrastructure limitations and should take some special measures to ensure QoS/availability of that application and to prevent against possible threats. In this paper we have discussed a new form of DoS that could take place in a cloud data network using the vulnerability caused by under provisioned network. We have also proposed a solution for this DoS attack, by which not only this attack will be detected, but can also be avoided in a very short time.

I. INTRODUCTION

Data centers are usually under-provisioned by a factor of 2.5:1 to 8:1. 4:1 of under-provisioning ratio means that user can only send data at one-fourth of its total interface speed. There are several reasons of under provisioning. Firstly, it is very expensive to build a full 1:1 ratio bandwidth network. Secondly, although the network supports multi-path routing, the number of paths supported is typically small. [1]

Under provisioning is typically not a problem in traditional data-centers, Servers of an application are placed in the same subnet and the data-center managers have the full control over the architecture and to take preventive measurements against suspected attacks, Also it is very easy to localize any active attack.

However, under provisioning can be a problem in a cloud data-center. A cloud data-center is drastically under-provisioned by atleast a factor of 45:1. This can also result in many security concerns for cloud architecture. There are few vulnerabilities in cloud data centers. Firstly, size of cloud data-center is very large, and because of limitation in number of multipath. Secondly, a cloud data-center is accessed by many people, this opens a door for adversaries to attack and perform DoS on other applications running in that cloud data-center. Thirdly, the application owner has very little control over the underlying data-center architecture to put counter measures against attacks by adversaries. [3]

We have verified that this new form of DoS can badly affect many cloud users simultaneously with a minimal cost, and since the manager of cloud data-center is unaware that his data center is under new type of DoS attack, no measurement is taken to detect and avoid such attacks.

II. A NEW WAY OF DOS ATTACK

Two main vulnerabilities found in cloud data-centers are the under-provisioning of upload link and its public nature. Attacker can saturate the limited bandwidth of uplink and can perform DoS attack against other applications running in the cloud network. [4] To do so, the attacker must find a bottleneck link in the network first.

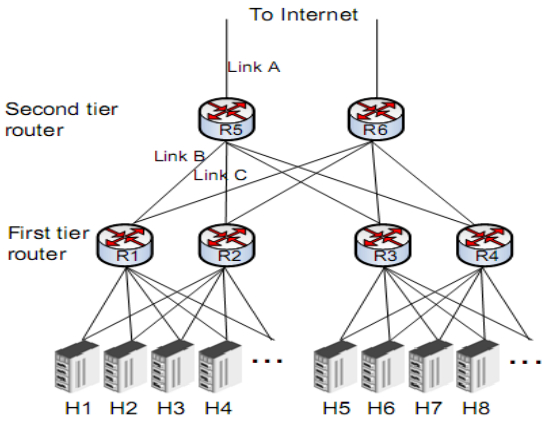

Figure 1: A typical data center network architecture.

In a data center, a router is typically connected to a large number of hosts and their combined link bandwidth is quite large as compared to the uplink capacity. In figure 1, Link A, B and C are uplinks of router R5, R1 and R2 respectively. We consider the hosts (e.g. H1….H4) connected to the Routers 1 & 2 to be a part of the subnet of the router. To attack, an adversary simply transmits enough traffic from a few hosts of one subnet to other hosts in a different subnet. By symmetry, the uplink will be saturated in both directions. As an example, Let us consider the target link to be 'Link B'. Link B is the active link and Link C is the fail over link. [6] Normally, uplink bandwidth is 1Gbps and hosts interface cards also have 1Gbps data-rate. So, one host will be enough to saturate the uplink. When say host H1 will transmit enough traffic to a host in a different subnet (say host H5) the uplink B will be saturated, and no other legitimate traffic can reach other hosts in that subnet connected to R1, and hence denying services to the hosts in R1s subnet.

A. ACCESSING HOSTS & TOPOLOGICAL IDENTIFICATION

To carry out an attack, the adversary must have access to few hosts inside the cloud network (e.g. by launching virtual machines using the cloud API), then learn the topology and to look for any bottleneck. If none is found, the adversary should gain access to more hosts and repeat the same process.

To initiate this DoS attack, the attacker must gain some topological information. Because if he doesn't have topology information, there is a possibility that he blindly transmits high speed data between hosts in the same subnet, hence no uplink will be used between data transfer and normal communication will occur uninterrupted and without causing the uplink to get saturated. However, if he knows the topology, he can easily identify the uplink between two routers as bottleneck, and can saturate the uplink by transmitting data between hosts of different subnets.

B. METHODS FOR TOPOLOGICAL IDENTIFICATION

Now we shall discuss two ways by which we can find out the topology provided we have access to some of the hosts. In the first approach, we can found the network architecture simply by running traceroute command from end hosts; it gives all the hop-info between the end hosts. [7] For ease of installation and management, IP ranges are defined following a regular structure and convention, so by using traceroute, IP addresses are known and layout of the whole topology can be deducted. But there are two disadvantages of it, firstly, we can't get info about the Layer 2 switches involved and if router does not support traceroute, we can't get the IP addresses. [8]

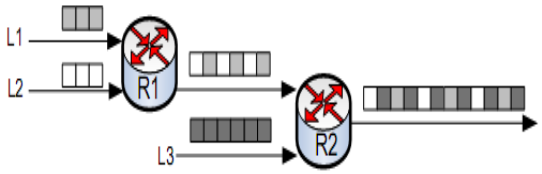

Figure 2: Routers multiplex incoming packets.

Second approach uses the multiplexing nature of routers. As shown in figure 2, packets from the two incoming links L1 & L2 of Router R1 are multiplexed and then sent to the output link and again the output link of R1 is multiplexed with the other input link (Link 3) of Router R2. In the final output link, half of bandwidth is available for Links 'L1 & L2'. So in general, the farther away a link is from the final output link, the lesser bandwidth it will get. Given a set of hosts, assign one host as sink and the other as sources; from each source, we send a sequence of packets to the sink and based on the traffic pattern in the output link, we can know that how many routers/switches are located in between the sink and sources. Repeat the process after changing the sink host to get more accurate/complete network topology.

Note that we can have compression affect; if two routers are attached back to back with a single link in between, there will be no multiplexing and the two routers will be misinterpreted as only one as shown in the figure 3. But this won't have any important impact on our target.

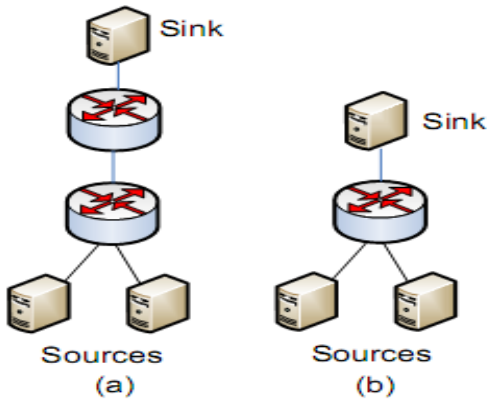

Figure 3: The compression affect (a) The real topology (b) The detected topology.

C. CARRYING OUT THE ATTACK

Once an attacker has gained enough access to multiple hosts and is aware of the network topology, all he need to do is to send a large amount of traffic from hosts residing in one subnet to hosts residing in any other subnet, and hence could saturate the uplink using the under-provisioning affect. The easiest way is to send a large amount of UDP packets, because it will deprive TCP packets of uplink bandwidth.

Totally saturating the uplink is not a good approach because some applications have active standbys which periodically communicate with the main application via heart beat mechanism, and if no heart beat is received, standby will assume that the main application has died and it will become active. Hence our attack would be in-effective if the standby resides in a different subnet. So if the uplink is not fully saturated, heartbeat can pass through the link as it requires very less bandwidth, however the application users will still experience much degraded level of service.

III. DOS PREVENTION TECHNIQUES

Traditional DOS and DDOS attacks and their counter measure techniques are well known. Once a DOS attack is detected, traceback mechanism can be used to trace and stop the attack at the source. However, all these techniques ensure that the packets are sent directly to the application, but in this form of DOS, packets are sent indirectly and the application doesn't know that DOS attack is in progress. There are many ways to prevent this new type of DOS Attack which we have explained in this paper, but none of them is attractive and do not ensures full protection. Now we shall discuss few of them briefly. First, a cloud provider can provide more bandwidth upto the full bisection bandwidth, but it is very costly solution. Second prevention mechanism is that the cloud provider should be given a percentage of total bandwidth available to a particular bandwidth but, cloud data centers are already under provisioned; assigning them a percentage of available bandwidth would decrease their throughput even more. Third solution can be that a cloud provider should charge according to the bandwidth consumed; unfortunately, cloud providers can not charge a large amount because it would make their business unattractive then, and if they charge less amount, it won't

ensure complete protection since an attacker can pay the amount in order to attack. Forth proposed solution is that the high bandwidth usage should be detected and respective application should be shutdown immediately, but it is very difficult to detect either it is legitimate bandwidth hungry application or a DOS attack.

IV. OUR PROPOSED SOLUTION

Figure 4 shows our solution architecture, there is a monitoring agent which resides either in a different subnet or outside the cloud data-center. The monitoring agent and the application periodically contact each other to see the available bandwidth in both directions. When the application notices that the available bandwidth decreases below certain threshold, the application sends multiple UDP packets to the monitoring agent to ask for help. Even in case of full starvation, few of these UDP packets are able to compete and reach the monitoring agent. Upon receiving these UDP packets, the agent starts the process for migrating the application to a different subnet. It first looks for an active standby application, but if no active standby exists, it launches a new active standby in a different subnet behind a different router and then measures the available bi-directional bandwidth between itself and the new standby. If less bandwidth is calculated, it repeats the process (with the new standby in a different subnet) unless it found a perfect subnet where there is enough available bandwidth.

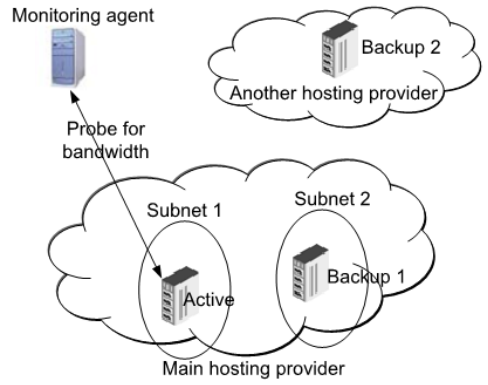

Figure 4: Application migration under DoS attack

As shown in the figure, the application standby/backup can be in a different subnet within the same cloud data center or in the data center of another provider. If the attacker attacks multiple subnets, it may not be possible to find the appropriate subnet (Usually the monitoring agent gives up after a threshold of say 3 trials). It then finds a good subnet in the data center of a different provider. After finding the required subnet in that cloud data-center, it then converts the standby to the main application, and hence this type of attack can be eliminated. Another solution we can implement is called application hopping, where the application shifts to a different subnet periodically after few minutes, so that the adversary can not launch DoS attacks in that subnet again.

V.CONCLUSION

The large size and public nature of cloud data-centers have bring some vulnerabilities in its architecture. Also, these data-centers are highly under-provisioned, even below the recommended under-provisioning ratio. All these factors can be exploited to initiate DoS attacks and to have a targeted attack which can degrade QoS of target application with a very minimal cost. We proposed dynamic migration architecture to detect and avoid this new type of DOS attacks, and hence cloud data-center can be made secured, protected and pro-active against DoS attacks.

VI. ACKNOWLEDGEMENT

This work (Grants No.00035521) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2010

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ″program for CITG″ support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-0004)

VII. REFERENCES

[1]. Al-Fares, M., Loukissas, A., and Vahdat, A. A scalable, commodity data center network architecture. In Proc.SIGCOMM (2008).

[2]. Carter, R.L., and Crovella, M.E. Measuring bottleneck link speed in packet-switched networks. Tech.rep, Performance Evaluation, 1996.

[3]. Greenberg, A., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D.A., Patel, P., and Sengupta, S. Vl2: A scalable and flexible datacenter network. In Proc.SIGCOMM (2009).

[4]. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., and Lu, S. Bcube: A high performance, server-centric network architecture for Modular datacenters. In Proc. SIGCOMM (2009).

[5]. Guo, C., Wu, H., Tan, K., Shiy, L., Zhang, Y., and Luz, S. Dcell: A scalable and fault-tolerant network structure for datacenters. In Proc. SIGCOMM (2008).

[6]. Hopps, C. Analysis of an Equal-Cost Multi-Path Algorithm. RFC2992 (Informational), Nov.2000.

[7]. Yaar, A., Perrig, A., and Song, D. Fit:Fast internet traceback. In Proc.IEEE Infocom (March 2005).

[8]. Savage, S., Wetherall, D., Karlin, A., and Anderson, T. Practical network support for ip traceback. In Proc.SIGCOMM (2000).