NIMBUS: Cloud-scale Attack Detection and Mitigation

Rui Miao
Univ. of Southern California
rmiao@usc.edu

Minlan Yu
Univ. of Southern California
minlanyu@usc.edu

Navendu Jain
Microsoft Research
navendu@microsoft.com

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Experimentation

Keywords

Cloud attack; Software-Defined Networking

1. INTRODUCTION

Cloud services are growing rapidly and their market is expected to reach $180 billion by 2015 [4]. Today, large cloud providers host tens of thousands of different services, so *inbound* attacks targeting the cloud can cause significant, and sometimes spectacular, collateral damage. A recent survey of datacenter operators indicates that half of them experienced DDoS attacks, with 94% of those experiencing regular attacks [2].

We highlight several unique features and evolution trend among those attacks [6, 1]: *(1) Large-scale.* These attacks have the volume up to hundreds of gigabits per second against a single cloud service. *(2) Diverse attacks.* The attacks range from network-layer (e.g. SYN flood, UDP flood) to application-layer (e.g. HTTP GET, SQL injection) with different characteristics on volume, number of connections, and packet header signatures (e.g., TCP flag, port). *(3) Fast ramp-up rate.* The attack traffic ramps up quickly and affects the target cloud service usually within one minute.

In response to the challenges above, the attack detection and mitigation system needs to 1) have sufficient capacity to accommodate attack volume, 2) support the detection of diverse range of attacks, and 3) have accurate and fast attack detection with low collateral damage to legitimate traffic.

To detect attacks, cloud operators commonly adopt commercial hardware boxes such as Firewalls, IDS, DDoS-protection appliances) in the network. There are three problems in these hardware boxes. First, these hardware boxes cannot scale up to handle attacks in cloud scale. For example, Firewall and IDS examine the detailed signatures and states of the traffic. So they cannot handle attacks with high volume. The DDoS-protection appliances check only significant traffic in network-layer. So they can handle larger attack volume but are still unable to accommodate extreme cloud-scale attacks with up to hundreds of gigabits per second. Second, these hardware boxes introduce unfavorable cost vs. capacity tradeoffs. For example, the DDoS-protection appliance typically cost hundreds of thousands up to a few million dollars per box in each year [8]. Third, since these devices run proprietary software, they limit how operators can configure them to handle the increasing diversity of attacks today.

There have been some commercial attack prevention services (CloudFlare [3], Prolexic [1]) to redirect web service and enterprise traffic through a dedicated high-capacity scrubbing network for attack detection and mitigation. However, the cloud operator does not want tenant traffic to be re-routed given the private concerns.

In this paper, we propose a new paradigm of *attack-prevention-as-a-service* that leverages commodity VMs for attack detection and mitigation. We propose the NIMBUS service, which combines the elasticity of cloud computing resources with the kinds of programmability seen in software-defined networks (SDN). NIMBUS scales resource usage with traffic demands, is flexible to handle diverse attacks, and is relatively cheap and without the exposure of private tenant traffic.

2. NIMBUS DESIGN

There are several key challenges in designing and implementing NIMBUS:

1. Scaling to match datacenter traffic capacity at the order of hundreds of gigabits per second. The service should auto-scale to enable agility and cost-effectiveness.

2. Programmability to handle new and diverse types of network-based attacks, and flexibility to allow tenants/operators to configure policies specific to the traffic patterns and attack characteristics.

3. Fast and accurate detection and mitigation for attacks that ramp-up within one minute; once the attack subsides, we should revert the mitigation to avoid blocking legitimate traffic.

4. Robustness to prevent sophisticated attacks from a wise adversary, who knows our system well and want to attack either the NIMBUS service or individual cloud service.

In response to those design challenges, we design NIMBUS as an SDN-inspired framework, comprising a set of VM instances that analyze traffic for attack detection and an *auto-scale controller* that (a) does scale-out/in of VM instances to avoid overloading, (b) manages routing to traffic flows to them, and (c) dynamically instantiates anomaly detector and mitigation modules on them, as shown in Figure 2.

SIGCOMM'14, August 17–22, 2014, Chicago, IL, USA.
ACM 978-1-4503-2836-4/14/08.
http://dx.doi.org/10.1145/2619239.2631446 .

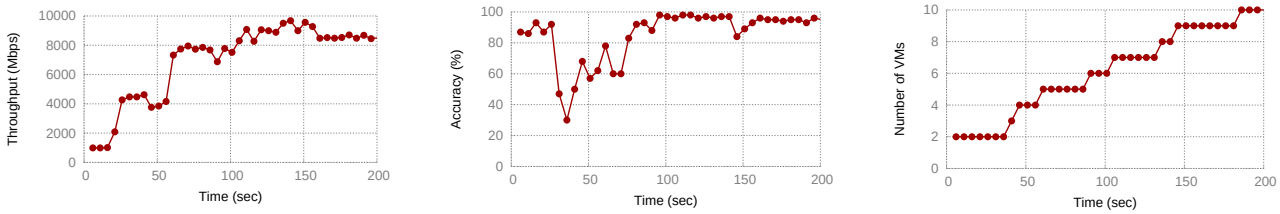

Figure 1: Timeline of attack burst. Starting with 1 Gbps and 2 VMs, and adding 9 Gbps attack traffic from 15th second.

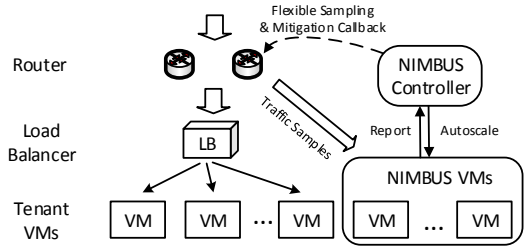

Figure 2: NIMBUS Architecture

Flexible sampling. NIMBUS supports flexible sampling configuration on different traffic flows, for the better accuracy of different attack detection applications. Since traffic monitoring in cloud data center typically employs uniform sampling with very low sample rate (e.g. 1 in 4000 packets), it is highly likely to miss median size DDoS attacks and other types of attacks (e.g. scan, Bruteforce). Biased sampling strategies have been proposed to have different sampling rate for different flow groups [9] or flow size [7], based on statistics over all packets in each flow group (e.g. sketch). However, those approaches need to change the hardware to support per-packet update for sampling rate. Here, we propose a flexible sampling strategy without hardware changes, where NIMBUS estimates traffic statistics from only traffic samples, and then adjusts the sampling rate for different flows accordingly. Finally, we would investigate an algorithm to allocate sampling resources to different flows to reduce the attack detection error for different attack types.

Autoscale. NIMBUS's central controller communicates with routers to direct traffic samples to different VMs, each of which detects attacks destined to different sets of cloud services. The VMs are allocated from available resources shared with other cloud tenant applications. When a VM is close to resource exhaustion under attack scenario, the controller can divert some of its traffic to other, possibly newly instantiated, VMs. By doing so, NIMBUS can maintain accurate attack detection with sufficient capacity provisioning.

One challenging is NIMBUS needs to manage the states of VMs in scale-up/down for consistent and responsive attack detection. Recent work [5] proposes to halt the packets of particular flows until state migration has completed. However, in an attack scenario, the traffic ramps up very fast and thus the VM is overloaded during the state migration. In this sense, NIMBUS needs to balance the traffic load immediately, even before the states have completely migrated. We need to investigate how to merge the new states and migrated states, and how to quantify the accuracy drop during the state migration. Further, in order to balance the cost (e.g. CPU, memory, bandwidth) of state migration and detection responsiveness, we need to investigate which flows to migrate and what associated states to migrate. In addition, we need to make NIMBUS robust to the wise adversary who may generate sophisticated attacks to degrade the effectiveness of autoscale system. For example, the adversary may generate on/off attack aligned with the auto-scale mechanism to waste NIMBUS resources and cause fluctuations of state migration.

Detection and Mitigation. NIMBUS supports flexible programmability for diverse and new attack detection using commodity VMs. For example, we may count throughput towards individual service to detect DoS-like attacks. In addition, we can also track the number of distinct IP sources to identify bot-like scan. Moreover, NIMBUS needs to be generic to support existing detection tools and algorithms (e.g., Deep Packet Inspection (DPI) [10]). One challenge is a wise adversary may generate attacks in a short burst that are hard to detect from the aggregated traffic statistics in a measurement epoch. Therefore, NIMBUS should dynamically change the aggregation time period to make it harder for the adversary to get around its detection while still saving cost. Once the attack characteristics (e.g., IP sources, header signature, payload) have been identified, NIMBUS controller can callback mitigation strategies to blacklist or rate-limit the attack traffic at the routers, prevent the attacks from hitting cloud infrastructure and tenant applications.

3. PRELIMINARY RESULTS

We build our prototype using three servers acting as the traffic generator, the host of NIMBUS VMs, and the NIMBUS controller, respectively. We connect them to a switch using 10G links. Our preliminary experiment is to detect flows with significant volume. The results are shown in Figure 1, where we add 9 Gbps attack burst from 15th second. As we can see, the accuracy decreases rapidly as the system drops lots of packets. As more VMs get started, the accuracy gradually recovers and the system throughput also increases to accommodate the attack traffic. In this experiment, The system has scaled-out to 10 VMs. With increasing number of active VMs, the system takes around 55 seconds to recover its measurement accuracy and 100 seconds to accommodate the 9 Gbps traffic burst.

4. REFERENCES

[1] Q4 2013 global ddos attack report. http://goo.gl/lIyRmK, 2013.

[2] Arbor Networks. Insight Into the Global Threat Landscape. http://goo.gl/15oOx3, February 2013.

[3] CloudFlare. https://www.cloudflare.com/.

[4] Gartner. http://www.gartner.com/newsroom/id/2562415. 2013.

[5] A. Gember, R. Viswanathan, C. Prakash, R. Grandl, J. Khalid, S. Das, and A. Akella. OpenNF: Enabling Innovation in Network Function Control. In *Sigcomm*, 2014.

[6] Google. Malware Distribution by Autonomous System. http://goo.gl/mZQeG4, 2013.

[7] A. Kumar and J. Xu. Sketch guided sampling - using on-line estimates of flow size for adaptive data collection. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006.

[8] F. Networks. 2011 adc security survey global findings. In *http://goo.gl/A3b2Q*, 2011.

[9] A. Ramachandran, S. Seetharaman, N. Feamster, and V. Vazirani. Fast monitoring of traffic subpopulations. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, 2008.

[10] Snort. http://www.snort.org/.