Google Cloud Platform Security

Nicholas J. Mitchell and Kazi Zunnurhain Ph.D

School of Computer Science, Kean University

Abstract

In modern days, use of service-oriented architecture (SoA) is becoming popular due to the use of public cloud. A cloud can provide storage services, computing services and infrastructure without any capital or operational expenditure. There are numerous cloud providers in the market. Among the prominent ones, such as Amazon AWS, Google Cloud Platform (GCP), and Microsoft Azure etc. GCP offers a wide array of services for many organizations and individuals around the world like other cloud providers. However, the question is, all those data, confidential information, intellectual property of individual or industries safe enough? Can we trust a public service provider? Hence, our initiative aims at investigating that GCP is secure from inside attackers. Why we chose inside attackers? Well the idea is to check if GCP can detect and ignore an attack launched internally. In this case, it should be secure enough to prevent from outsiders as well. The research investigation would involve Distributed Denial of Service (DDoS) attacks against a local virtual machine (VM) launched from GCP. There is a broad spectrum of VMs available in GCP compute engine service.

Introduction

As of recently, cloud computing has become a growing industry. More tech companies have started offering cloud services such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, Rackspace, Verizon wireless, VMware and many more. Cloud computing services are provided in complete disguise and independent from the user's current location. The most vital advantage of cloud could be the most vulnerable cause of security threats. By vital advantage, we are referring the cloud services provided from remote servers in unknown locations. Hence, the question arise: "*Is cloud safe?*"

For this project, we have received an educational grant from Google for their GCP Program. The project was motivated from a class project. This class is about World Wide Web programming. The GCP grant allowed about 25 student licenses to host their virtual machines in GCP in the beginning. Then later we applied in 2019 and received a grant for 60 licenses. This project involved focusing on the vulnerability of GCP. We decided to investigate in GCP with operating systems Windows Server 2016 Data Center and Ubuntu 18.04 LTS. Windows Server has a Graphical User Interface (GUI) and Ubuntu 18.04 LTS is a Command line interface (CLI). In the following experiments the Pen-Testing tool Low Orbit Ion Cannon (LOIC) was used to launch Distributed Denial of Service attacks (DDoS) via Transmission Control Protocol (TCP).

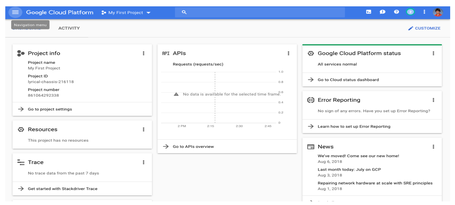

Figure 1: GCP Interface

Tools

The cloud provider that was used in the following experiment was Google Cloud Platform (GCP). There are many sub-services offered within GCP such as App Engine, Compute Engine, Kubertenes Engine, and many other tools and programs.

The sub-service of GCP used in this project was Compute Engine. Compute Engine allows a user to create and deploy instances of Virtual Machines within GCP's infrastructure [1]. When creating a virtual machine, GCP has sixty Operating System (OS) images to choose.

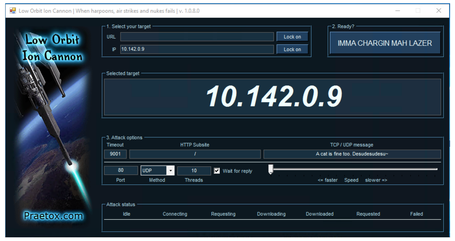

Figure 2: LOIC Diagram

The test bed in this research involves launching a DoS attack from a running VM with Windows on GCP to another running VM in GCP. The attack scenario will certainly test the DoS attack handling capability of GCP. Why DoS attack? GCP is a service-oriented architecture (SoA).

The penetration-testing tool used in this research experiment was Low Orbit Ion Cannon (LOIC) [3]. LOIC offers Distributed Denial of Service (DDoS) attack using three different types of Protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Hypertext Transfer Protocol (HTTP). Another feature offered on LOIC is Threads or simulated users in the attack. By default, it is set to ten threads. LOIC is very user friendly and easy to launch attacks with either of those protocols. A user can provide the target machine URL or IP address to launch attack. In our experiment, it was the IPv4 address of the victim virtual machine.

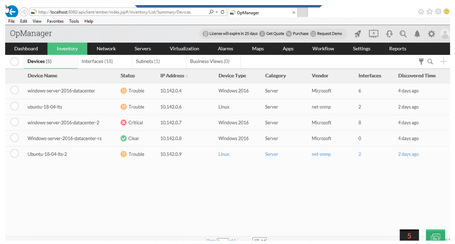

Figure 3: OP Manager List of VM's

An intelligent Monitoring tool used in this project was ManageEngine OP Manager. It tracks the performance of the VM's using Simple Network Managing Protocol (SNMP). It is accessed through Microsoft Internet Explorer using a localhost (*https://localhost8060/*). OP Manager can also set threshold alarms that give the user notifications via email, text messaging, and custom scripts [5].

Referring to the inventory tab in Figure 3, there is a list of VMs listed with various statuses. Each tuple in the figure below OP Manager is providing the status, IP address, device types, category and vendor types of each running VM. A VM with status *Trouble* means that one of the performance dials is at a high percentage. These dials consist of availability, CPU Utilization, Disk Utilization, Packet Loss, and Memory Utilization.

Experiment

When creating an instance, it is important to know which OS the VM is going to use. In another word, which platform the VM will be launched from. With a diverse community of users, we had to consider multiple OS platforms to conduct the experiment in order to observe the experimental results in a diverse manner. For the experiment, two OS's were used: Windows Server 2016 Datacenter (Windows Server) .

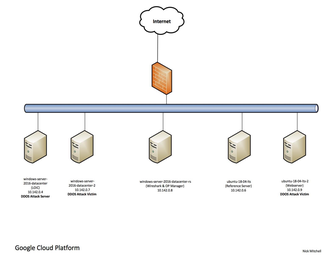

Figure 3: OP Manager List of VM's

To facilitate a DDoS attack, it was essential to turn off all the security features in windows. This also similar for Internet Explorer Enhanced Security Configuration, as keeping all the ports open would create a honeypot environment for the adversaries. It also prevents anything from being downloaded onto the VM.

For this experiment, five VMs were launched. Refer to the topology in Figure 4. Three of them had the Windows Server 2016 OS. The other Two VMs had the Ubuntu 18.04 LTS OS. Two trials were performed. Both were conducted at four different time intervals as seen in Figures 4 and 5.

In the first trial, LOIC was running on a Windows Server VM (10.142.0.4). In the topology shown in Figure 4, it is labeled as the DDoS Attack Server. The DDoS Attack Victim was also a Windows Server VM with IP address 10.142.0.7. An additional Windows Server VM ran Wireshark and OP manager to track the activity and measure the parameters of the victim. Monitoring was done through this VM, 10.142.0.8. We called it the Wireshark and OP Manager server.

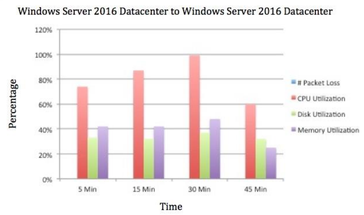

Figure 4: Windows to Windows Attack

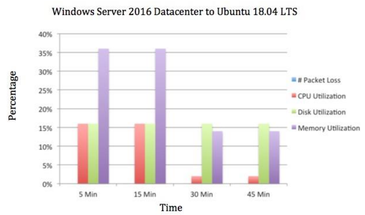

Figure 5: Windows to Ubuntu Attack

Future Directions and Conclusion

In summary, it is clear that GCP has a strong security system to distinguish regular protocol from spoofed ones. Even though it took about 45 minutes to detect but GCP could start discarding DoS attack to re-establish efficient utilization of CPU and memory in the virtual machines. But GCP's response to amplified pen testing with multiple virtual machines raise an eyebrow. Even after 45 minutes, 5 virtual machines have managed to keep the CPU utilization over 80%, which is a very alarming indicator. Now, in the future we intend to extend the research investigation with a large botnet and running for longer interval to see how GCP respond. We also plan to involve several types of protocol to see if the GCP respond is different.

Another observation is, during most of the cases, the DDoS Attack Server started to slow down, which was observed by checking the task manager running in the VM. LOIC was using most of the CPU power. It even got to a point where the Remote Desktop access of Windows Server went to a black screen. Occasionally the VM shutdown automatically. Hence, it can also be inferred that VM running on GCP platform with all open ports and firewall has bare minimum infrastructure to defend against a DoS tool. Certainly, GCP did not run any vulnerability testing on the client VMs, which could certainly cost their reputation if GCP has an inside, attack.

From some trials, it was anticipated that everything including packet loss, CPU utilization, Disk Utilization and Memory Utilization were to increase in a linear pattern. Instead, there was no packet loss, and the CPU Utilization started fluctuating.

References

[1] Legorie Rajan. 2018. "Google Cloud Platform cookbook: implement, deploy, maintain, and migrate applications on Google Cloud Platform", Birmingham, UK: Packt Publishing.

[2] Ted Hunter and Steven Porter. 2018. "Google Cloud Platform for developers: building highly scalable, resilient web services with the power of Google Cloud Platform", Birmingham, U.K.: Packt Publishing Ltd.

[3] Michael Annor-Asante and Bernardi Pranggono. 2018. Development of Smart Grid Testbed with Low-Cost Hardware and Software for Cybersecurity Research and Education. (April 2018). Retrieved April 20, 2019 from https://link.springer.com/article/10.1007/s11277-018-5766-6.

[4] Molly Sauter. 2013. "LOIC Will Tear Us Apart": The Impact of Tool Design and Media Portrayals in the Success of Activist DDOS Attacks. (March 2013). Retrieved April 20, 2019 from https://doi.org/10.1177/0002764213479370.

[5] Josune Hernantes, Gorka Gallardo, and Nicolás Serrano. 2015. IT Infrastructure-Monitoring Tools. (2015). Retrieved April 20, 2019 from https://ieeexplore.ieee.org/document/7140697

[6] Kazi Zunnurhain and Susan V. Vrbsky. 2010. Security Attacks and Solutions in Clouds. (December 2010). Retrieved April 30, 2019 from https://pdfs.semanticscholar.org/3f29/bcf67ddaa3991ad2c15046c51f6a309d01f8.pdf

[7] Kazi Zunnurhain and Susan V. Vrbsky. 2011. Security in Cloud Computing. (2011). Retrieved April 30, 2019 from https://pdfs.semanticscholar.org/d8b8/8c941fe400161123fd04300f721f6bdf26fb.pdf

[8] Kazi Zunnurhain and Susan V. Vrbsky. 2012. FAPA: A Model to Prevent Flooding Attacks in Clouds. (March 2012). Retrieved May 6, 2019 from http://worldcomp-proceedings.com/proc/p2013/SAM9705.pdf

[9] Kazi Zunnurhain. 2014. FAPA: flooding attack protection architecture in a cloud system. (2014). Retrieved April 30, 2019 from https://ir.ua.edu/handle/123456789/2128