DDOS Mitigation Analysis of AWS Cloud Network

by

Waseem Ullah Khan
B.Sc,International Islamic University Islamabad, 2012

A Masters Project Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

DDOS Mitigation Analysis of AWS Cloud Network

by

Waseem Ullah Khan
B.Sc,International Islamic University Islamabad, 2012

Supervisory Committee

Dr. Daniela Damian, Co-Supervisor
(Department of Computer Science)

Dr. Sudhakar Ganti, Co-Supervisor
(Department of Computer Science)

Supervisory Committee

Dr. Daniela Damian, Co-Supervisor
(Department of Computer Science)

Dr. Sudhakar Ganti, Co-Supervisor
(Department of Computer Science)

ABSTRACT

Cloud computing is growing technology [10] which is adopted and used by many companies but it is facing a huge number of threats and Distributed Denial of Service(DDOS) tops the list. It can seriously affect the companies depending on cloud services for their business.This project is an analysis of DDOS mitigation on Amazon Web Services (AWS). AWS is currently one of the leading cloud service provider that powers thousands of businesses in more than 190 countries. The capabilities of AWS infrastructure for DDOS attacks resiliency and its prevention will be analyzed in this report. Some common DDOS attacks for which the scripts are easily available to everyone on internet has been carried out in the project against instances hosted on AWS to analyze its capability of dealing such attacks. This project report outlines the best practices for DDoS Mitigation with practical implementation to prevent common DDOS attacks and its mitigation.

Contents

List of Figures

Abbreviations

Denial of Service (DoS)

Distributed Denial of Service (DDoS)

Amazon Web Services (AWS)

Elastic Load Balancer (ELB)

Web Application Firewall (WAF)

Access Control List (ACL)

Transmission Control Protocol (TCP)

User Datagram Protocol (UDP)

Synchronize (SYN)

Acknowlede (ACK)

Virtual Private Cloud (VPC)

Hypertext Transfer Protocol (HTTP)

Low Orbit Ion Cannon (LOIC)

HTTP Unbearable Load King (HULK)

ACKNOWLEDGEMENTS

All praises belong to Allah the merciful for his guidance and blessings

All praises belong to Allah the merciful for his guidance and blessings to enable me complete this project.I would like to thank:

My Beloved Parents, who always supported me on every step of my life.

My Supervisors Dr.Daniela Damian and Dr.Sudhakar Ganti, for their support and mentoring throughout degree.

Friends, who always encouraged me where it was really difficult to move forward.

Department of Computer Science, whose friendly support and guidance made it possible.

Knock, And He'll open the door Vanish, And He'll make you shine like the sun Fall, And He'll raise you to the heavens Become nothing, And He'll turn you into everything.

Jalaluddin Rumi

DEDICATION

To my father **Muhammad Darwaish Khan** and my **Mother** whose life has been a perfect example for me.

Chapter 1

Introduction

In the world of computing, there have been different kinds of cyber attacks which kept threatening organizations regardless of their size and services since last decade. Denial of Service (DOS) and Distributed Denial of Service (DDOS) are one of the most common cyber attacks which have been a threat since long. DoS attacks are of different kinds, with some targeting the underlying server infrastructure and some exploit vulnerabilities in applications and communicating protocols. Unlike other cyber attacks which target valuable information or breaching data, DOS attacks target the availability of the service by consuming maximum resources with fake requests or packets which make the service unavailable to legitimate users. A successful DoS attack profoundly impacts the resources and is a notable event. DoS attacks can continue to target resources for days, weeks or even for longer periods of time making them destructive enough to any online organization and hence this makes it a popular weapon of choice for hackers, cybervandals, and extortionists.

Amazon Web Services(AWS) provide reliable, scalable, and inexpensive cloud computing services and solutions. Its infrastructure gives it an edge over other cloud services available in the market due to its low on demand prices and easy deployment. AWS infrastructure enables convenient, on-demand network access to many configurable computing resources that can be provisioned and released with very less effort or service provider interaction. Its Pay per usage, Virtualization, on demand access, flexibility and reduced hardware and maintenance cost are some of the factors contributing to the popularity of cloud computing. AWS comes with its firewall called Web Application Firewall (WAF) which can be used to filter traffic directed to resources.

1.1 Motivation

The increase in the number of attacks and the damage caused by such attacks in the world of cloud computing [5] has invoked my interest. This threat has been present for a while in the world of computing, and significant progress has been made on it, but in the cloud computing, the progress is yet to be made. The advancing of attack strategies and the damage caused by DDoS pushed me to learn more about this topic and urged me to make my contribution. These attacks have affected all sort of businesses from small to large enterprises like Visa and MasterCard.

1.2 Structure of the Project Report

This section provides the structure of this project report and a short headline of what each chapter will explain

Chapter 1 gives an overview of the project, an introduction of what this project is about.

Chapter 2 talks about the problem being worked upon in this project and the related work.

Chapter 3 discusses the infrastructure which is used in the project and the attacks which will be carried out for testing purpose

Chapter 4 explains the solution to the problems and different mitigation methods which is handy in dealing the most common attacks

Chapter 5 evaluates the results and future work that can be done to optimize cloud computing security.

Chapter 6 concludes the purpose and implementation of the project.

Chapter 2

Problem to be Solved

DoS attack is one of the major threat to the world of computing these days. With DoS attack, the attacker can significantly degrade the performance and quality of the targets network connectivity or can use the target's resources to maximum[7]. DDOS (Distributed Denial of Service) is, however, a bit different from DOS. In DDOS the attacker uses multiple agents or hosts to attack the victim from various locations and there can be no limit to the number of hosts which can be used to attack. The attacker initially compromises a number of hosts which can be in thousands and then use them to launch the attack by depleting the target resources or network. The primary objective of a DDoS attack is to make the victim unable to use the resources as it keeps the resources occupied or keep the resources busy enough so it can't handle any legit requests. In most of the cases, victims could be web servers, CPU, storage devices, or other network resources. In cloud environment DDoS can degrade the performance of cloud services significantly by overloading the virtual instances and consuming the bandwidth to its maximum.

There are several types of DoS Attack which ultimately target the following resources

- Congesting of network resources
- Completely draining CPU memory
- Decreasing computing power
- Exploiting timers

Chapter 3

Denial of Service In Detail

Distributed Denial of Service attack is usually carried out using a high number of systems attacking one target. The hosts utilized in the target are commonly controlled by a remote attacker who uses it for its attacks. Most of the times the real owner of the resource is not aware of the malicious activities carried out from their system by the remote attacker. Most attackers use the same mechanism allowing them to amplify their attack across the size of the botnet. One attacker can control 1,000 systems or more which can then be used to attack the victim. The greater the number of compromised systems the more powerful the attack will be, and this effect is called amplification effect.

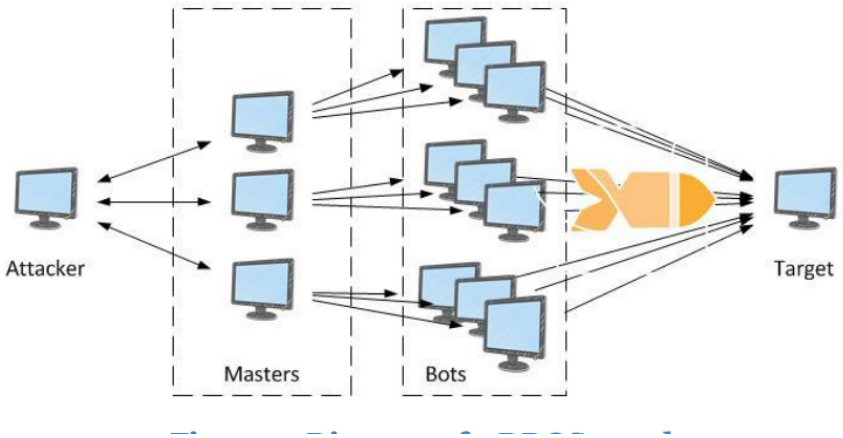

Figure 3.1: DDoS Attack

Figure 3.1 shows a number of bots are controlled by one attacker which attack the target

The DDoS attacks can be classified in two broad categories that is

- Network Layer Attacks
- Application Layer Attacks

3.1 Network-Layer Attacks

Network Layer Attacks are also known as Layer 3/4 attacks.The majority of DDoS attacks target the network and transport layers. Such attacks occur when incoming data packets overload the network and consumer server resources to its maximum. Some of the common attacks in this category cover UDP flood, SYN flood, NTP amplification and DNS amplification attacks, and more. These attacks can be used to restrict access to servers, while also causing severe operational losses, such as account suspension and massive over-consumption charges.

3.1.1 Network-Layer Attack Types

Network Layer Attacks can be further categorized as follows:

TCP SYN Flooding Attack

Transmission Control Protocol(TCP) is a connection oriented transport layer protocol of TCP/IP model that keeps a track of packet exchange between client and host. It also manages flow control of packets and provides error-free data transmission. During the packet exchange process, SYN message is transmitted by the connecting host and is acknowledged by the remote host by sending a SYN+ACK message. To complete the handshake process, the connecting host responds with a final ACK and the connection is established. An attacker exploit this connection mechanism, and they initiate a half-open connection by not sending the last SYN-ACK. Such connections are kept open by host, and it waits for the final SYN-ACK message. When thousands of such half-open connections are initiated the server runs out of memory and hence collapses. DDoS attacks using TCP SYN flooding can also be launced using spoofed IP addresses. During a spoofed attack, the final ACK required to complete the connection process will not be received, as the host whose IP address was spoofed will respond with TCP reset RST flag or the host not exist flag. This attack can also affect the cloud infrastructure by finding vulnerable nodes.

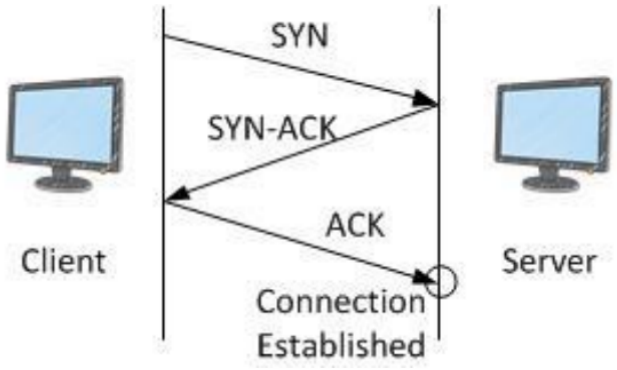

Figure 3.2: TCP-SYN flood attack mechanism

UDP flooding attack

UDP protocol is a connectionless transport layer protocol and is used when the reliability of the packet is not compulsory. Most of the times it is used for Audio/Video communication applications and instant messaging applications. A UDP attack can be launched by bombing target with a large number of data packets to random ports. The victim then responds with ICMP not reachable packet if no application is listening on that port. Thus, when a large number of UDP packets are received by the victim, the system will be forced to respond to many ICMP requests which leads to making the system unavailable for other clients.

ICMP flooding attack

ICMP is an IP protocol that checks the connectivity of host's network. Attackers have exploited this protocol in the form of smurf and ping flood attacks by sending a large number of ICMP packets to victim which consumes bandwidth and results in crashing the server. Consequently, the target will not be able to respond to an incoming request from legitimate users.

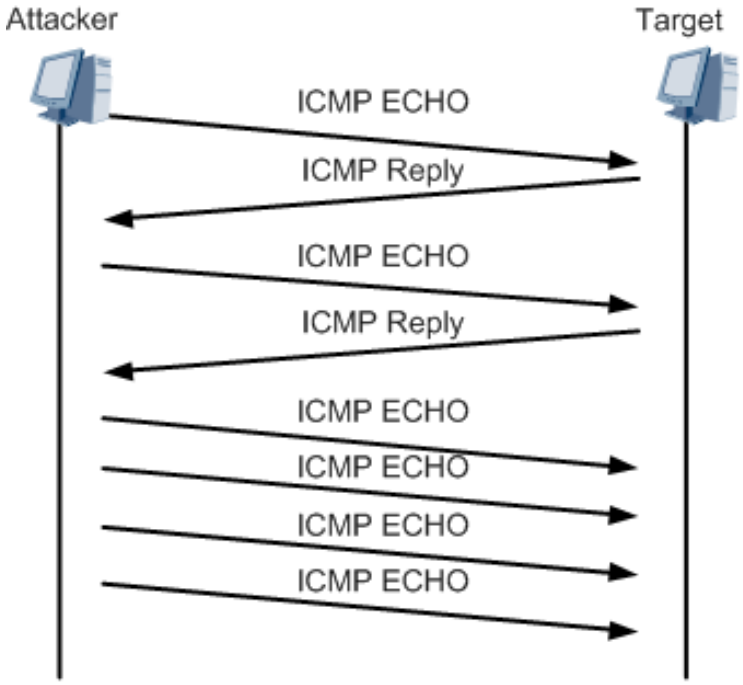

Figure 3.3: ICMP flood attack

Figure 3.3 shows the ICMP flood attack mechanism.

NTP Amplification Attack

NTP amplification is a type of Distributed Denial of Service (DDoS) attack in which the attacker exploits publically-accessible Network Time Protocol (NTP) servers to overwhelm the targeted with User Datagram Protocol (UDP) traffic.In the most basic type of NTP amplification attack, an attacker repeatedly sends the get monlist request to an NTP server, while spoofing the requesting servers IP address to that of the victim server.The NTP server responds by sending the list to the spoofed IP address. NTP amplification is essentially a type of reflection attack. Reflection attacks involve eliciting a response from a server to a spoofed IP address. The attacker sends a packet with a forged IP address (the victims) and the server replies to this address.

DNS Amplification attack

A Domain Name Server (DNS) Amplification attack is a popular form of Distributed Denial of Service (DDoS), in which attackers use publically accessible open DNS servers to flood a target system with DNS response traffic. The primary technique

consists of an attacker sending a DNS name lookup request to an open DNS server with the source address spoofed to be the targets address. When the DNS server sends the DNS record response, it is sent instead to the target. Attackers will typically submit a request for as much zone information as possible to maximize the amplification effect.

3.2 Application-Layer Attacks

Application Layer Attack is also called Layer 7 attack that seeks to overload resources by sending a large number of requests requiring intensive processing and handling. Application-layer DDoS attacks are increasing over the years, both in complexity and in volume. These attacks severely impact the quality of service, quality of experience and productivity of the cloud provider. Such attacks are carried out using flood packets by sending thousands of HTTP requests from different sources, and this makes it more lethal. Application-layer attacks are more challenging in mitigation as these attacks consume less bandwidth and is more stealthier than other DDoS attacks. Such flooding requests appear as legitimate requests which make them difficult to identify. The volume and size of application layer attacks can be measured in Requests-per-Second (RPS) and usually 100 to 150 RPS takes down a mid-sized server[4].

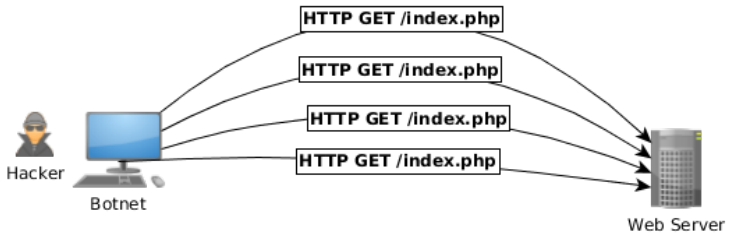

Figure 3.4: Application Layer Attack

Figure 3.4 shows the mechanism of Application layer attack in which an attacker request for a specific page after exploiting a flaw in an application.

3.2.1 Application Attack Types

Application Layer attacks can be further categorized into four categories:

Basic HTTP Floods

HTTP flood attacks are the most common Application layer attack in which a web server is flooded with HTTP GET or POST requests. Such attacks do not necessarily require a high rate of traffic. For example, HTTP GET attack can be carried out by compromising several nodes on the Internet to create several request sessions to the victim to disable the victim. A recent report on global DDoS attack reveals that close to a quarter of current DDoS attacks target the application layer, and one-fifth of the HTTP DDoS attacks are HTTP GET floods.

XML Flood Attack

Simple Object Access Protocol(SOAP) [11] message is a way to communicate between applications running on different operating systems, with different technologies and programming languages. When requesting for resources, cloud providers use SOAP message to start the communication. SOAP messages work with HTTP and are written in XML as XML is a universally acceptable language for any platform. X-DoS, an Extensible Markup Language DoS attack which can be carried out with less sophisticated tools due to its ease of implementation. Wordpress RPC pingback attack is an example of XML DoS attack in which WordPress brand content management software is used to generate a flood of HTTP requests. The ping-back feature allows a website hosted on WordPress (Blog A) to notify a different WordPress site (Blog B) that Blog A has created a link to Blog B. After which Blog A will fetch the Blog B to verify the existence of the link, which results in pingback flood. The attackers exploited this mechanism to use it against the target using WordPress blogs. This type of attack has a clear signature of WordPress present in the User-Agent of the HTTP request header.

Chapter 4

Environment and Tools

This project is based on the Amazon white paper best practices for DDoS Resiliency [2] and the best practices through which the environment can be deployed on Amazon Infrastructure. The cloud environment is set up and tested for two different scenarios. One scenario is testing DDoS directly against the servers and the other scenario is testing the server with DDOS mitigation infrastructure in place.

4.1 Scenarios

All the attacks will be carried out on two test cases as described below:

4.1.1 Scenario One

In this scenario the attacks will be carried out directly on the web server and statistics will be collected against each attack.

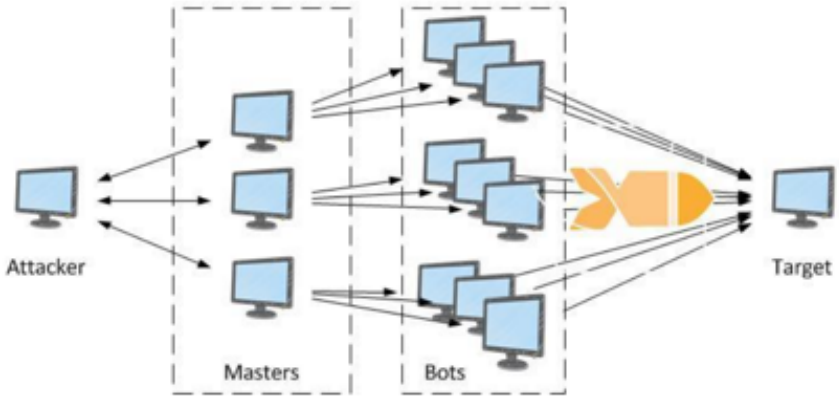

Figure 4.1: Scenario One of attacking strategy

Figure 4.1 shows a hacker targets the server directly when no security infrastructure is in place.

4.1.2 Scenario Two

In this scenario the attacks will be carried out against servers with deployed mitigation infrastructure.

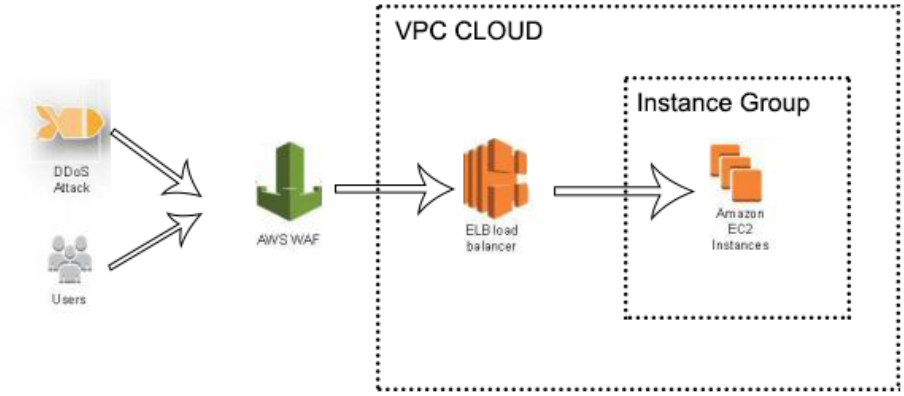

Figure 4.2: Scenario two of attacking strategy

Figure 4.2 shows an attack is carried out against infrastructure with AWS WAF and Elastic load balancer deployed. Following resources are deployed on Amazon Infrastructure.

4.2 EC2 Instances

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. We will be using Amazon two EC2 t2.micro Instances for our experiments. EC2 Instances can be upgraded with no downtime as AWS provides on-demand deployment of resources with which resources can be added of specific template in runtime. Each of the instances has its own IP-address and accessible from the internet. Both the Instances have been launched in west-2b region so we can have equal performance. We will be testing each ec2-instance with different DDOS attack strategies and will collect the results through Linux commands and AWS resource statistics. Both the instances have Ubuntu 16.04.1 LTS (Xenial Xerus) installed from the image provided by AWS.

4.3 Elastic Load Balancer

Elastic Load Balancing [1] automatically distributes incoming application traffic across multiple Amazon EC2 instances.In this project, we have two instances so that we will be distributing the traffic between them through Elastic Load Balancer. We will be using the Classic Load Balancer as its an ideal fit for simple load balancing of traffic across multiple EC2 instances.

4.4 NGINX WebServer

NGINX is free and open-source high-performance HTTP server. We will be using it on both of our instances as it is famous for its low resource consumption, high performance, stability, rich features and simple configuration.

4.5 AWS WAF

WAF [3] is an application layer defense firewall provided by AWS, however, its not free. Web application firewalls (WAFs) are often used to defend web applications against attacks that attempt to exploit a vulnerability in the application. AWS Web Application Firewall work as filter that applies rules and monitors HTTP and HTTPs requests based on source IP or more nuanced comparison to the values in several HTTP headers. This allows blocking cross-site scripting (XSS), SQL injection (SQLi)

and other common attack patterns. We will be testing it against DDoS mitigation with specific parameters and configurations on AWS infrastructure. AWS WAF can block IP's automatically according to the rules specified and also have a manual list where IP's can be added.

4.6 AWS CloudFormation

CloudFormation enables us to create and manage AWS infrastructure deployments predictably and repeatedly. A template of resources and dependencies is defined, and it will be deployed if resources are consumed to its maximum. However, we will not be using this service as it is out of scope regarding our project work.

4.7 Attacking Tools

Following are the tools that are used for different kind of attacks. All these tools are widely available on the internet and can be downloaded easily. The usage of all these tools in this project are only for educational and research purpose.

4.7.1 Low Orbit Ion Cannon (LOIC) DoS Tool

Low Orbit Ion Cannon (LOIC) [6] is an open source network stress testing and DoS attack tool, written in C-Sharp. LOIC was initially developed by Praetox Technologies, but was later released into the public domain and now is hosted on several open source platform for educational use. LOIC can be used for a DoS attack on a target site by flooding the server with TCP or UDP packets with the intention of disrupting the service of a particular host.

4.7.2 HTTP Unbearable Load King (HULK) DoS Tool

HULK [9] is a web server stress testing and DoS tool written in python. It is capable of generating different user-agent HTTP requests which bypass the server caching engines and hence every request it makes is processed by the server as a different request.

4.8 NGINX (Engine-x) Amplify

NGINX Amplify [8] is a statistics and graph collection module for NGINX web server. It will be used for collection of statistics and web server utilization. It also provides security and configuration recommendation for NGINX Webserver.

Chapter 5

Experiments and Attacks

5.1 TCP Flood Attack

In this attack, LOIC tool was used to flood the server. As discussed before in the report TCP handshake is completed in three steps. In the first step, client requests connection by transmitting SYN (synchronize) message to the server. In second phase server acknowledges by transmitting SYN-ACK (synchronize-acknowledge) message back to the client. In the third step, the client responds with an ACK (acknowledge) message, and the connection is established. However, in the case of attack, the server never receives ACK message from the attacker and waits for it.

5.1.1 Scenario One

Figure 5.1: TCP SYN flood attack with Low Orbit Ion Cannon

Figure 5.1 shows the LOIC tool has been used to launch a TCP-SYN attack on the independent server. Initially, the attack started with 60 MB/s of data flooding. It can be seen in figure 5.1 that after few seconds the graph began to drop which means that server couldn't take the load and hence not accepting further connections which is a clear indication of vulnerability to TCP-SYN attacks. Similarly, the attack was replicated on the second server. The attack time for each server was 60 seconds.

5.1.2 Detection of Attack

The general symptom of SYN Flood attack on the server is that it takes a long time to load. To detect TCP-SYN attack on server netstat command can be used to check connection requests that are in SYN-RECEIVED state.

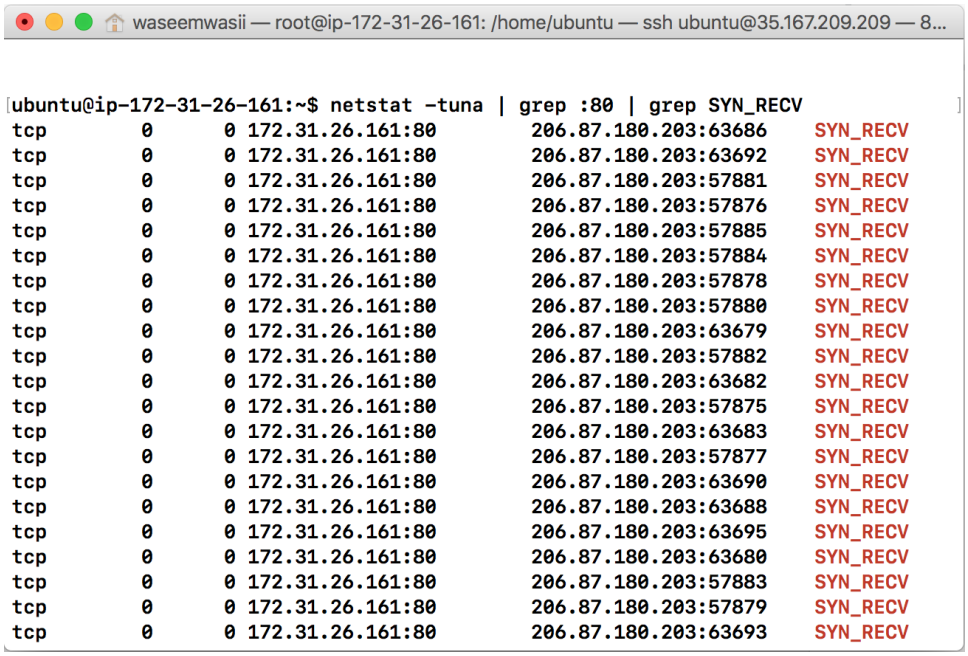

Figure 5.2: TCP SYN flood attack detection on server

In figure 5.2 above, it can be seen that there are many connection requests in SYN-RECV state which indicate the attack is affecting the server. The IP address 206.87.180.203 is the attacking source IP, while 172.31.26.161 is server's internal IP address.

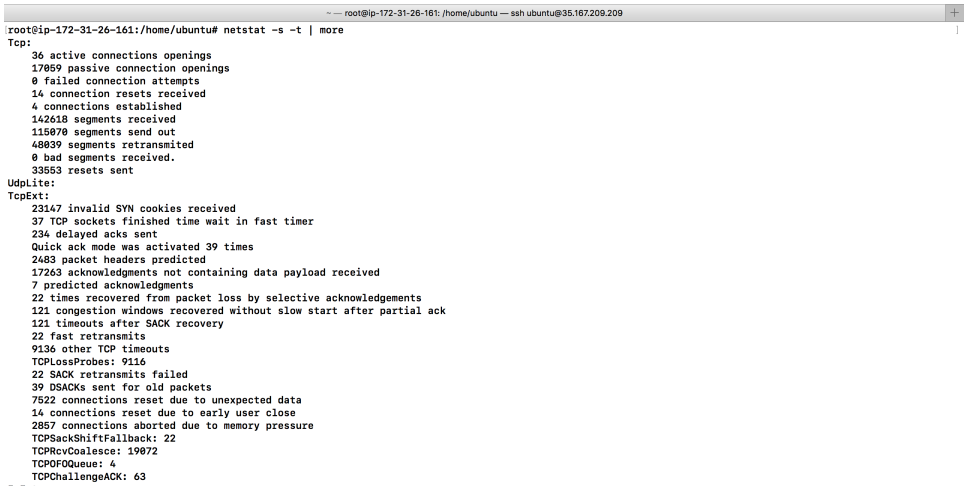

Figure 5.3: Netstat command statistics

In figure above 5.3 the netstat command has been executed to collect detailed statistics of TCP.

5.1.3 Mitigation

According to AWS whitepaper [2] Amazon's Elastic Load Balancer blocks TCP SYN floods and does not route it to the underlying servers. To follow the best practices, Elastic Load balancer is deployed. Furthermore following rules to block SYN packets are added to each server's IP tables. In addition to ELB, The Amazon's WAF is also deployed to filter out traffic and forward only legitimate requests to the servers.

```
# Drop Bogus Packets

iptables -A INPUT -m state --state INVALID -j DROP
iptables -A FORWARD -m state --state INVALID -j DROP
iptables -A OUTPUT -m state --state INVALID -j DROP

iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN,PSH SYN,FIN,PSH -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN,RST SYN,FIN,RST -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN,RST,PSH SYN,FIN,RST,PSH -j DROP
iptables -A INPUT -p tcp --tcp-flags FIN FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP

# Drop SYN Packets

iptables -N syn_flood
iptables -A INPUT -p tcp --syn -j syn_flood
iptables -A syn_flood -m limit --limit 90/s --limit-burst 150 -j RETURN
iptables -A syn_flood -j DROP
```

Figure 5.4: Ip-table rules for blocking SYN packets

In above Figure 5.4, the ip-table rules drop unwanted packets which have signatures of TCP-SYN attack. These rules blocks a proportion of TCP-SYN packets which escape the ELB and WAF.

5.1.4 Scenario Two

In this scenario, the attack will be carried out with Web Access Firewall and Elastic load balancer deployed which will equally divide the traffic to both servers. Both the servers were rebooted to reset the netstat counters for precise statistics collection. Same LOIC tool will be used to attack the infrastructure for 60 seconds.

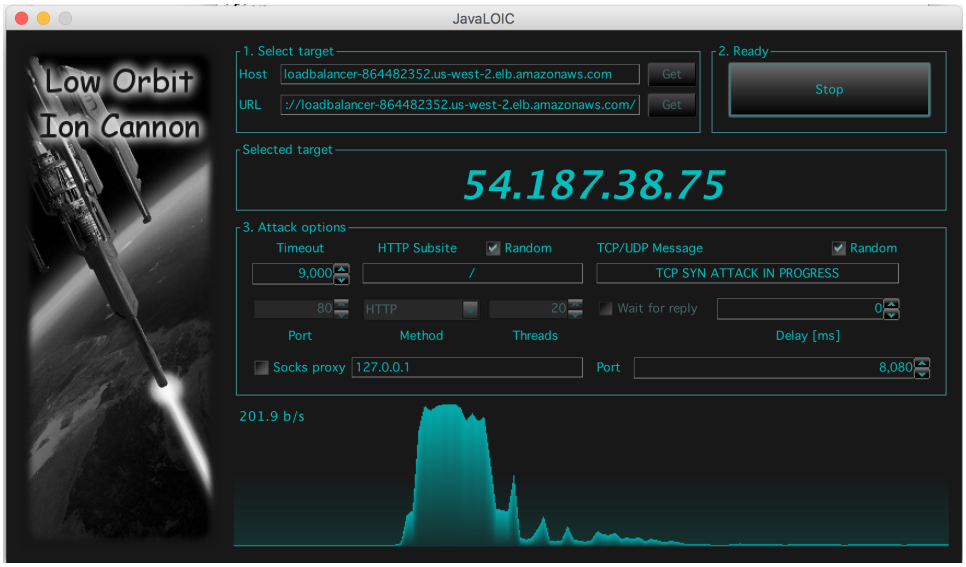

Figure 5.5: Attack on Infrastructure with ELB and WAF deployed

In above Figure 5.5, the attack started with 2 mb/s of flooding initially that is decreased gradually.

5.1.5 Detection

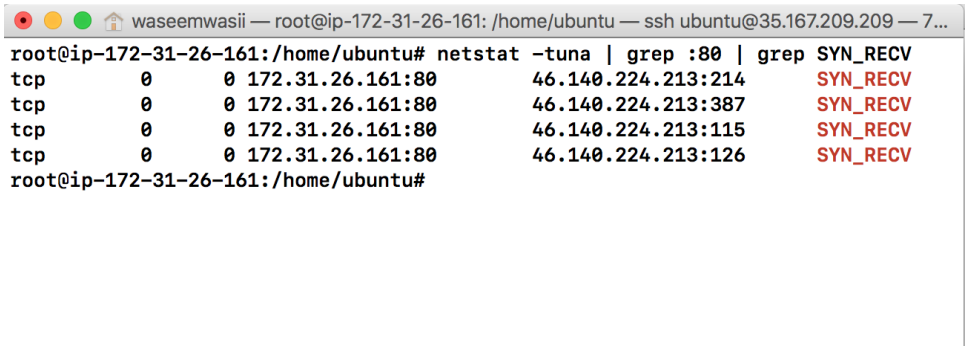

Figure 5.6: TCP-SYN flood connections statistics

Figure 5.6 shows SYN-RECV connections which is far less than the attack without infrastructure.

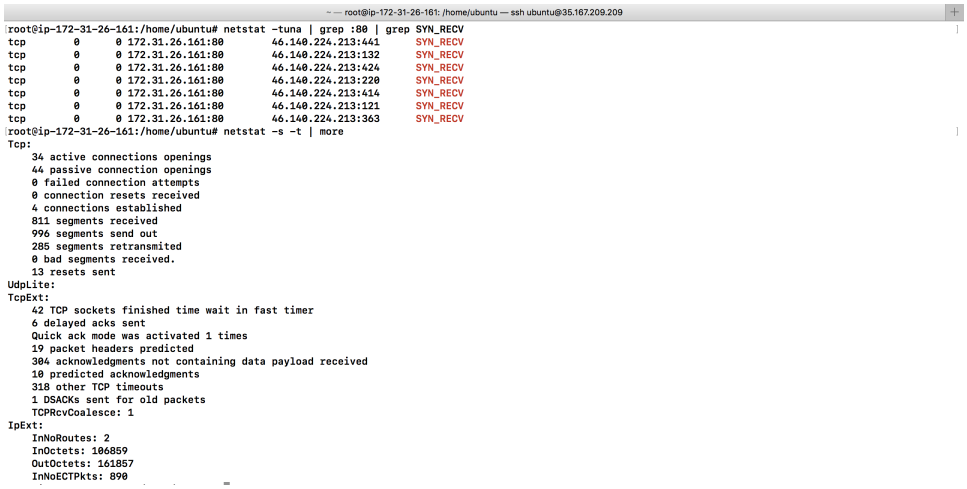

Figure 5.7: Netstat command statistics

In above Figure 5.7, netstat command has been executed from SSH to collect the detailed statistics of TCP which will be further analyzed in next section.

5.1.6 Analysis

From the above figure 5.3 and figure 5.7 we can see that the number of passive connections opened in scenario one is greater than the number of passive connections opened in scenario two. Passive connections are the number of connections opened in response to the active connection by the server. When a client sends a request to open connection, the server responds by opening a passive connection. From this, we can clearly infer that the WAF and ELB are blocking invalid SYN packets which result in a drop of passive connections.

The second noticeable metric is SYN-Cookies. The number of invalid SYN-Cookies in scenario one is much greater than the number of Invalid SYN-Cookies in Scenario two which shows the effectiveness of Infrastructure. The number of connections aborted due to memory pressure is those connections which were aborted by the server when it was drained out. The number of aborted connections in scenario one is 2857 however in scenario two the server wasn't drained out, so it is 0.

5.2 UDP Flood Attack

A UDP flood is a DoS attack that can be initiated when an attacker sends a large number of UDP packets to random ports on a remote host. In response to that, the victim will check for any application listening on the port and will respond with ICMP Destination unreachable packet. This makes the system unavailable to legitimate users. To launch UDP flood attack, LOIC tool is used with UDP mode selected. The tool will flood the servers with 1000 byte size datagrams.

5.2.1 Scenario One

Figure 5.8: LOIC tool UDP flood attack

In above figure 5.8 LOIC tool is flooding the server one with UDP 1000 byte size datagram.

5.2.2 Detection

UDP attack can be detected through multiple methods using Linux commands to see the UDP traffic. We will use Netstat command for detecting the attack.

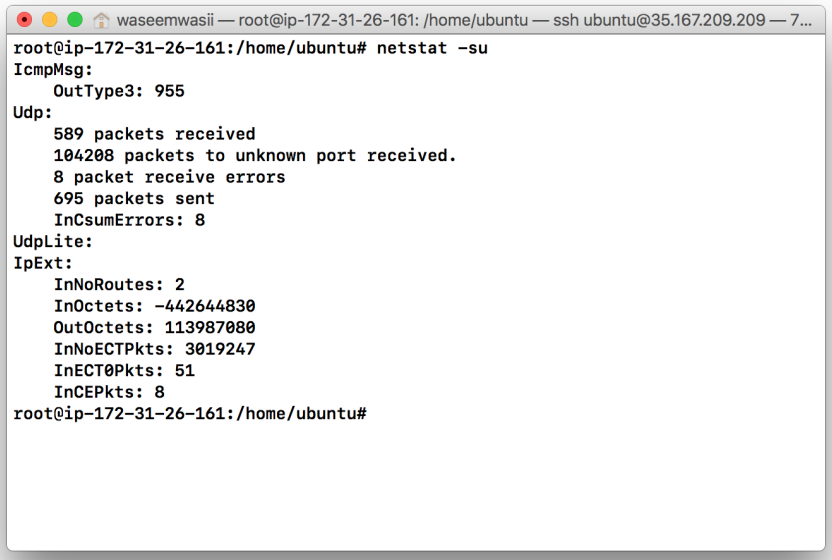

Figure 5.9: Netstat udp statistics

In above figure 5.9, we can see the netstat UDP statistics which shows that a total of 104208 packets were received on unknown ports and 955 ICMP unreachable destination messages were sent by the server. It is clear from the statistics that the server is under UDP flood attack and hence vulnerable to it.

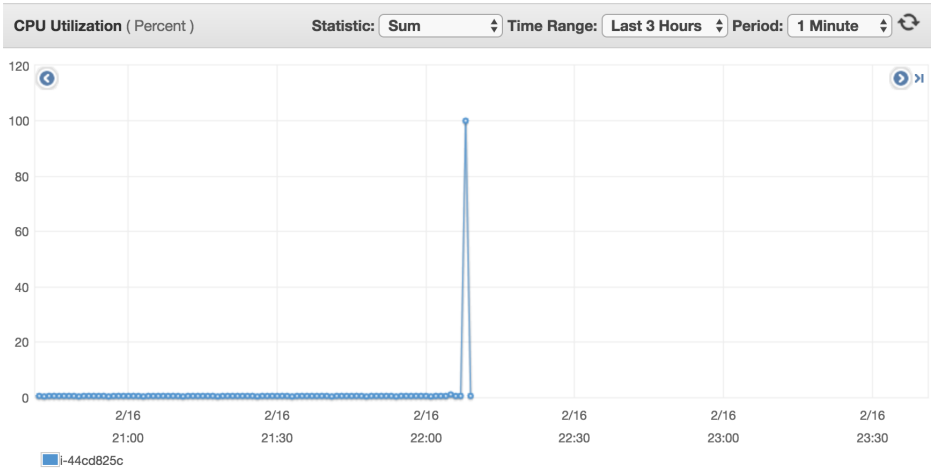

Figure 5.10: CPU Utilzation of UDP flood attack

In above Figure 5.10 we can see the CPU utilization from AWS Cloud Monitoring.

It shows clearly that CPU has been exhausted after flood attack and in result, the server was unavailable for legitimate clients.

5.2.3 Mitigation

To mitigate the attack, we will follow some best practices provided in AWS whitepaper for DDOS resiliency along with that we will also deploy some IP-table rules to further harden the server security for the attack.

```
1 iptables -I INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --set
2 iptables -I INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --update --seconds 60 --hitcount 10 -j
  DROP
3 iptables-save >/etc/iptables.up.rules
```

Figure 5.11: Iptables rules for UDP flood connection limitation

In above Figure 5.11, the IP-table rules have been applied which limits the number of connections per Ip address.

AWS also provide autoscaling option to absorb more attacks, but we won't be doing that here as we are only analyzing it for the attack. In addition to the IP-table rules, the WAF and ELB are also deployed to isolate the resources. This helps the servers to avoid direct traffic and ELB only forward legitimate requests.

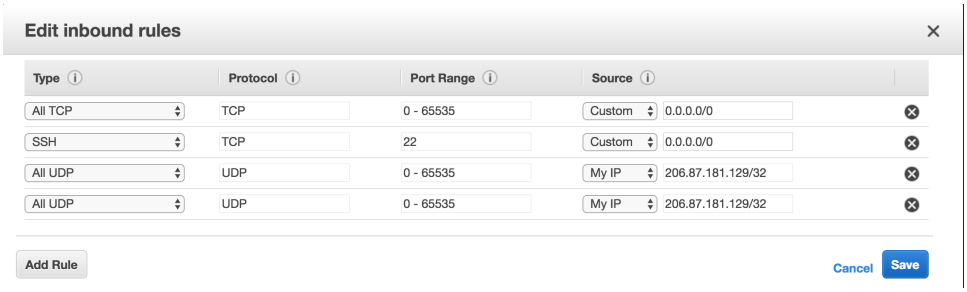

Figure 5.12: ACL rules for UDP

[?] Figure 5.12 the Access control list rules are shown. UDP incoming packets can be managed from here as well and only the ports on which applications are running can be opened. In our case the ports are only open for our own ip address.

5.2.4 Scenario Two

In this scenario, the attack will be carried out against the recommended infrastructure. Both the servers were restarted so that fresh UDP statistics can be collected for the attack.

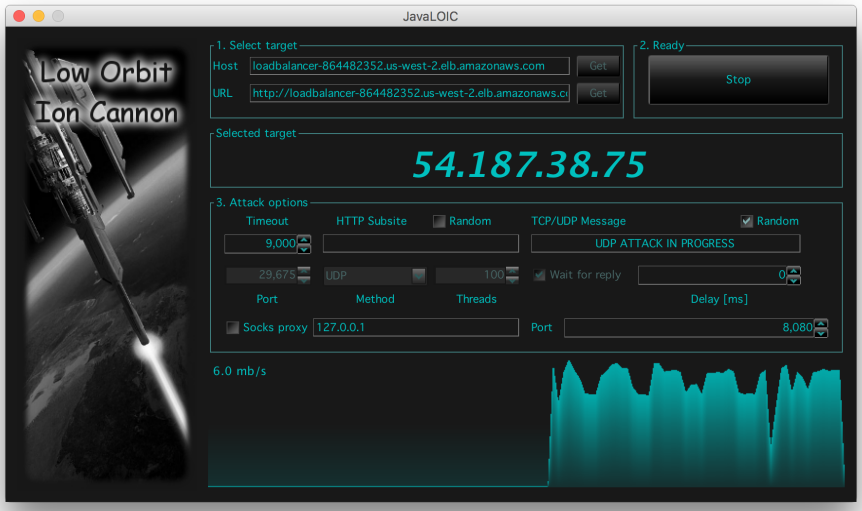

Figure 5.13: UDP attack on Infrastructure using LOIC

[?] Figure 5.13 the attack is launched against the infrastructure with 6 mb/s data.

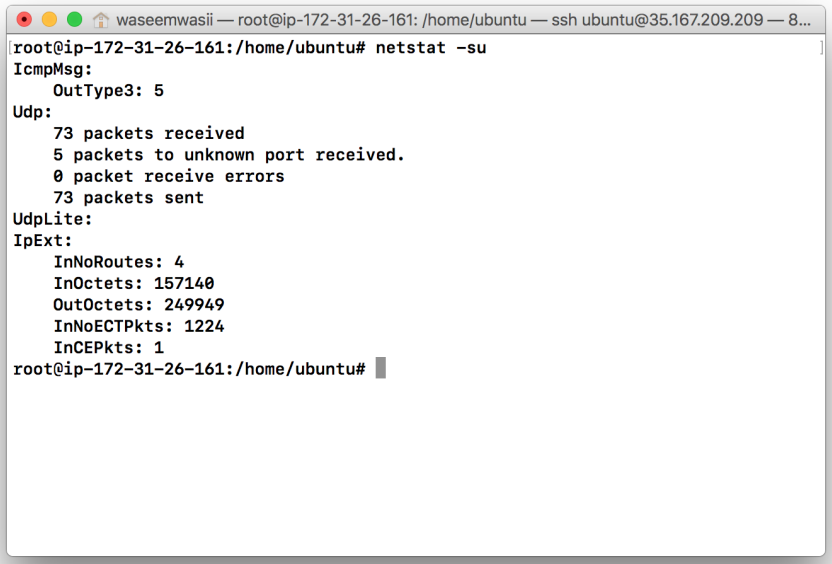

Figure 5.14: UDP netstat statistics during attack

[?] Figure 5.13 it can be clearly noticed that only 73 packets were received and five packets received on unknown ports in response to which the server sent out 73 packets as ICMP messages. This shows the efficiency of Infrastructure.

5.2.5 Analysis

From the above Figure 5.13 and Figure 5.9 we can see the difference. Independent servers went down due to the UDP flood attack however the result was opposite in scenario two, and the infrastructure successfully mitigated the UDP attack.

The AWS ELB played a vital role in blocking unwanted UDP packets. Hence it proves that UDP attacks can be addressed quickly as they contain clear signatures that make their detection easier. In order to effectively combat UDP floods, network resources must have the ability to both absorb the entire attack and exceed the volume thats generated by the attacker.

5.3 HTTP Flood Attack

Layer 7 or Application Layer Attacks are the most common attacks these days as it does not require high bandwidth or complex tools. As discussed before in the report the application layer attacks are more stealthier than Layer 4 attacks as they mix up with legitimate requests and sometimes it's really difficult to differentiate between legit requests and spoofed HTTP requests. There can be different browser agents which can be cache busting. To launch the HTTP flood attack, we will use both the tools i.e. LOIC and HULK to have a deep impact on servers and then analyze its effect.

5.3.1 Scenario One

In the first scenario similar to other experiments the attack will be launched on independent servers with both tools and maximum threads.

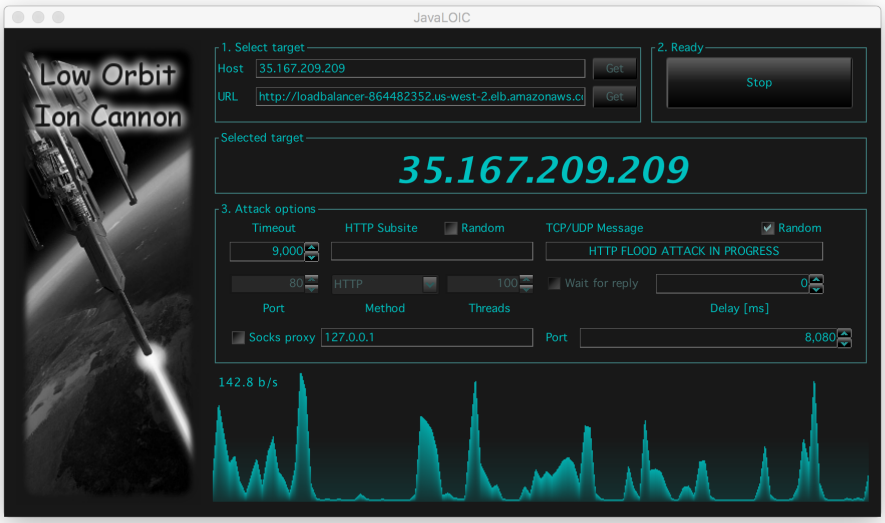

Figure 5.15: LOIC Http Flood Attack

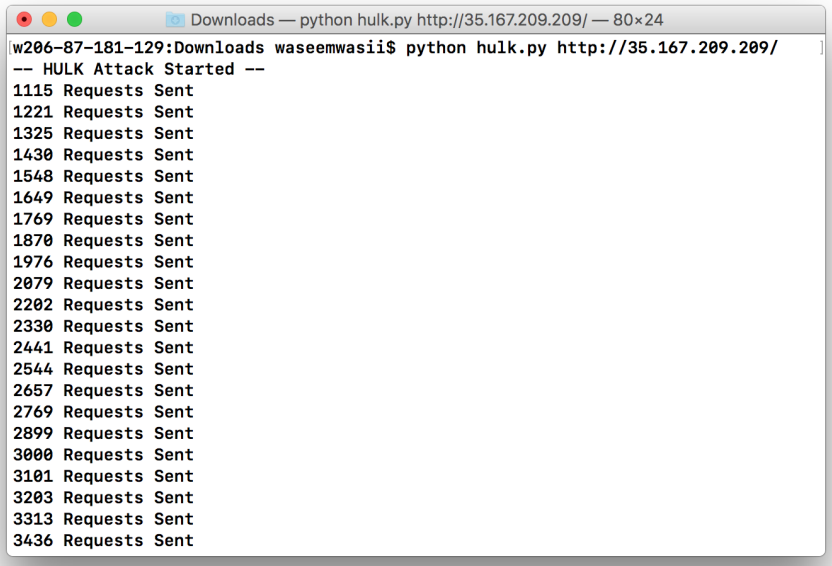

Figure 5.16: Requests sent using HULK tool

[?] Figures 5.15 and Figure 5.16 the HTTP flood attack has been launched on the server.

5.3.2 Detection

To detect the HTTP flood attack, we will use the tcpflow command. Tcpflow is a program that captures data transmitted as part of TCP connections, and stores the data in a way that is convenient for protocol analysis or debugging. In our case we used the tcpflow for capturing port 80 HTTP requests.

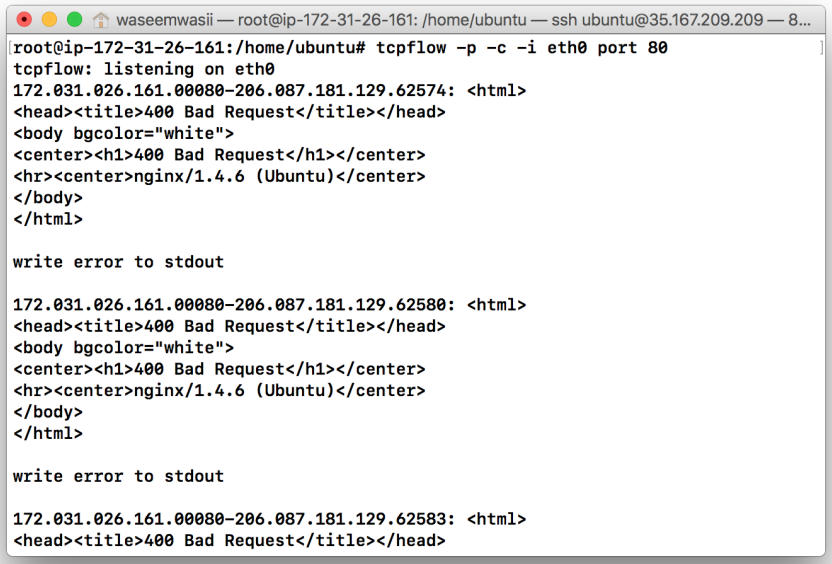

Figure 5.17: Hulk HTTP Flood attack

[?] Figure 5.17, HTTPFlood request can be seen as it looks like a legitmate request but its not. The sample request is from LOIC tool. However to figure out fake or spoofed requests, we can use netstat -an command which shows the number of connections made from a single IP.

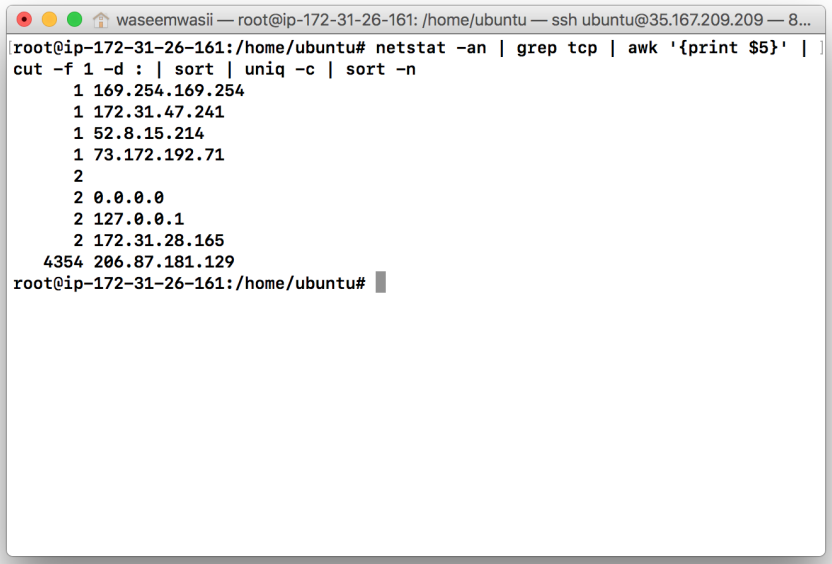

Figure 5.18: HTTP flood attack connections

[?] Figure 5.18 the number of connections from attacker ip which is 4384 can be clearly noticed which indicates that its not a legit client.

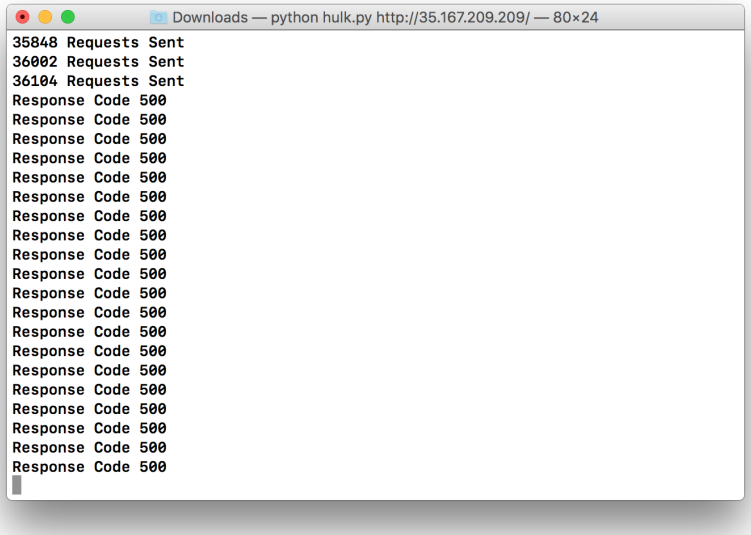

Figure 5.19: Server response 500 after attack

[?] Figure 5.19 the server went offline due to a high number of unserved requests. This indicates that the server is highly vulnerable to Layer 7 HTTP flood attack.

5.3.3 Mitigation

Application layer attacks can be mitigated with deploying AWS Web Access Firewall and by optimizing Nginx configurations for the number of requests. The AWS WAF is a powerfull firewall to mitigate such attacks.

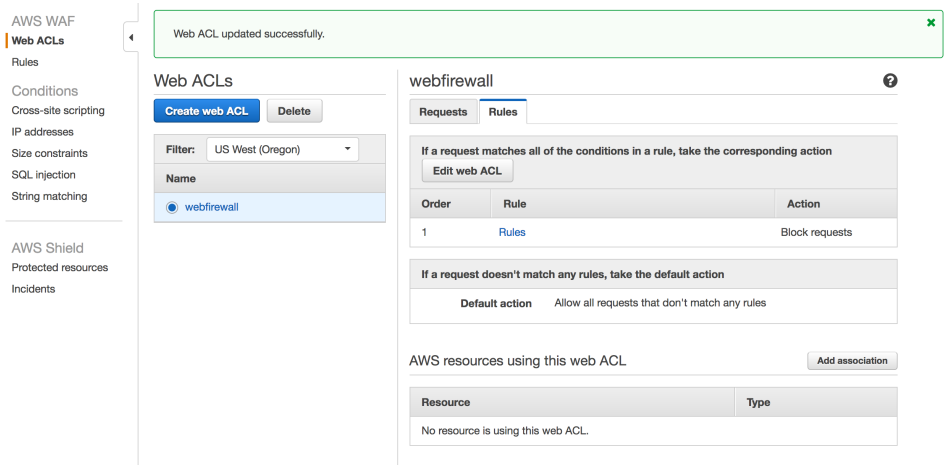

Figure 5.20: AWS Web Access Firewall

[?] Figure 5.20 the AWS WAF is shown in which rules for blocking requests are specified. To analyse the efficiency of firewall the rules are based on query string which is randomly generated by the attacking source.

```
## Allowing only 30 Requests per Minute
    limit_req_zone $binary_remote_addr zone=one:10m rate=30r/m;

    server {
        ...
        location /index.html {
            limit_req zone=one;
        ...
        }
    }

# Limiting Number of Connections

limit_conn_zone $binary_remote_addr zone=addr:10m;

    server {
        ...
        location /* {
            limit_conn addr 10;
            ...
        }
    }

# Closing Slow Connections

    server {
        client_body_timeout 5s;
        client_header_timeout 5s;
        ...
    }
```

Figure 5.21: Nginx Configuration for DDOS Prevention

[?] Figure 5.21 parameters have been added to limit the number of requests, number of connections and closing slow connections.

5.3.4 Scenario Two

In this scenario the attack will be carried out against protected infrastructure and statistics will be collected.

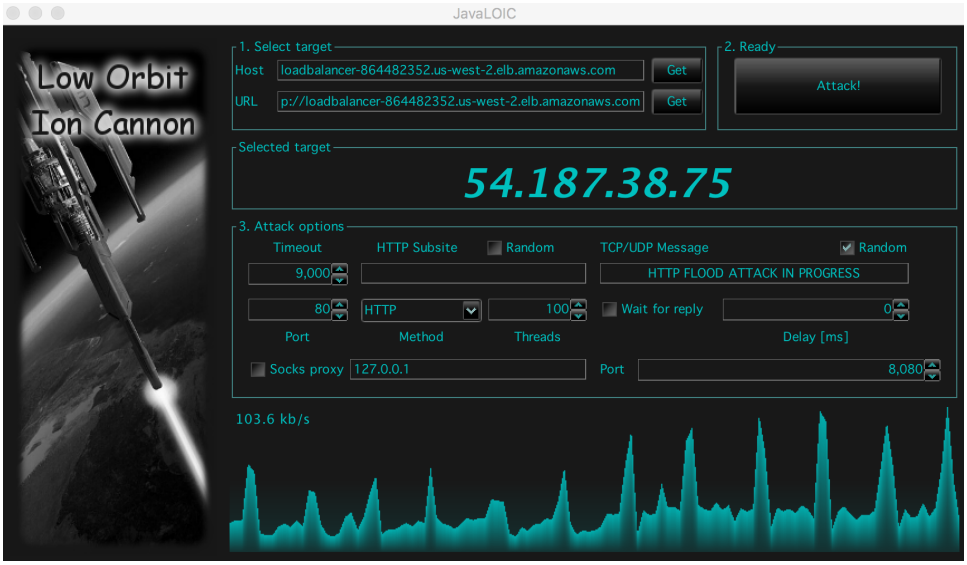

Figure 5.22: Http flood attack against protected infrastructure

[?] Figure 5.22 the firewall protected infrastructure is attacked with LOIC tool.

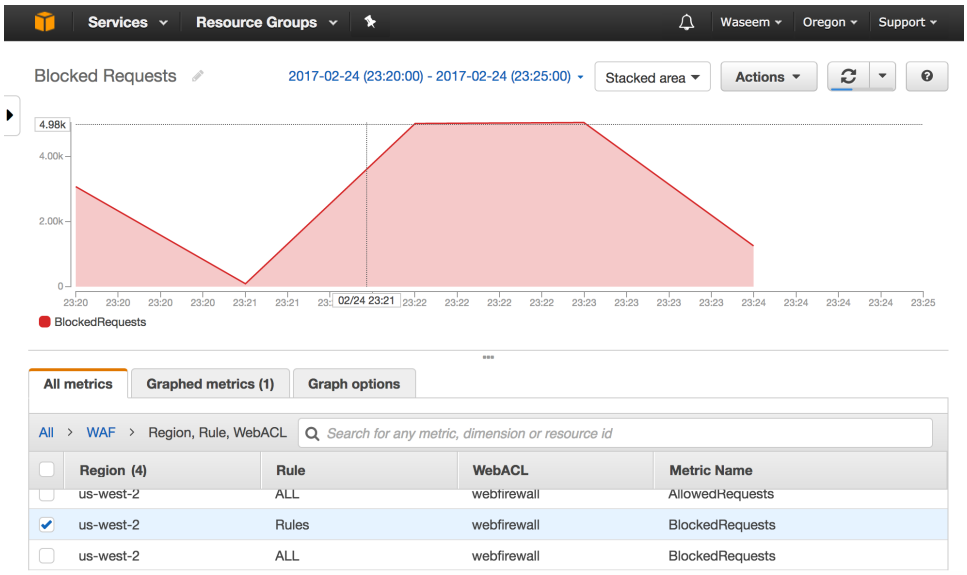

Figure 5.23: AWS CloudWatch Graph for WAF

[?] Figure 5.22 the screenshot is taken from AWS CloudMetrics which shows the number of requests blocked by the WAF. This shows the effectiveness of Web Access Firewall.

5.3.5 Analysis

From the above experiments, we can see that AWS firewall is an effective tool to mitigate application layer attacks as stated in the AWS whitepaper. However it can be challenging to identify the signatures of attacking sources as they seem to be legitimate requests but by looking at other parameters it can be determined. The AWS infrastructure provides scalability and mitigation of huge attacks in thousands of Gigabytes.

AWS WAF is also effective for other types of attacks such as Wordpress Pingback flood and XML-RPC floods. With customized mitigation rules infrastructure can be protected against common and advanced threats.

Chapter 6

Conclusion

In this report we studied the effect of DDOS attacks on AWS Cloud Infrastructure. From the results in the report it can be concluded that an independent server can be highly vulnerable to simple DDoS attacks however following the best practices results in reduction of common attacks. Some of the points which can be concluded from the report are :

- Isolating network elements in path keep the server safe from many threats.
- AWS Elastic load balancer keep the load balanced and keep the server safe from Layer 4 attacks.
- AWS WAF is effective for Layer 7 Application layer attacks and many other such attacks which doesn't need huge bandwidth to overload the server.

The best practices outlined in AWS Whitepaper gives a brief knowledge about building DDoS-resilient architecture. Some of the most most important best practices are implemented in this project with the results and analysis.

Appendix A

Additional Information

```
user www-data;
worker_processes 4;
pid /run/nginx.pid;

events {
    worker_connections 8096;
  multi_accept on;
    use epoll;
}
worker_rlimit_nofile 40000;
http {
geoip_country  /etc/nginx/geoip/GeoIP.dat; # the country IP database
geoip_city     /etc/nginx/geoip/GeoLiteCity.dat; # the city IP database

 limit_conn_zone $binary_remote_addr zone=alpha:10m;
 limit_req_zone $binary_remote_addr zone=one:10m rate=15r/s;
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 15;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # Logging Settings
    ##

    #access_log /var/log/nginx/access.log;
    access_log off;
    error_log /var/log/nginx/error.log;

```

Figure A.1: Nginx Configuration for Servers

```
    ##
    # Gzip Settings
    ##

    gzip  on;
    gzip_vary on;
    gzip_min_length 10240;
    gzip_proxied expired no-cache no-store private auth;
    gzip_types text/plain text/css text/xml text/javascript application/x-javascript application/xml;
    gzip_disable "MSIE [1-6]\.";

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;

    ##
    # nginx-naxsi config
    ##
    # Uncomment it if you installed nginx-naxsi
    ##

    #include /etc/nginx/naxsi_core.rules;

    ##
    # nginx-passenger config
    ##
    # Uncomment it if you installed nginx-passenger
    ##

    #passenger_root /usr;
    #passenger_ruby /usr/bin/ruby;

    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;


}
```

Figure A.2: Nginx Configuration for Servers

Bibliography

[1] Amazon. Elastic load balancer. https://aws.amazon.com/elasticloadbalancing/, 2014. [Online; accessed 6-Jan-2017].

[2] Jeffrey Lyons Andrew Kiggins. AWS Best Practices for DDoS Resiliency. https://d0.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf/, 2016. [Online; accessed 4-Jan-2017].

[3] Vineet Badola. AWS WAF (Web Application Firewall) and application security. http://cloudacademy.com/blog/aws-waf-web-application-firewall, 2015. [Online; accessed 12-Jan-2017].

[4] Wayne Booth. *The Craft of Research*. University of Chicago Press, 2003.

[5] M. SqalliK. Salah F. Al-Haidari. Evaluation of the impact of edos attacks against cloud computing services.

[6] infosecinstitute. Low Orbit Ion Cannon (LOIC) DoS Tool. http://resources.infosecinstitute.com/loic-dos-attacking-tool/, 2016. [Online; accessed 05-Jan-2017].

[7] Patel D. Borisaniya B. et al. Modi, C. A survey on security issues and solutions at different layers of cloud computing.

[8] NGINX. NGINX Amplify. https://www.nginx.com/amplify/, 2016. [Online; accessed 02-Jan-2017].

[9] Sectorix. HTTP Unbearable Load King (HULK) DoS Tool. http://www.sectorix.com/2012/05/17/hulk-web-server-dos-tool/, 2016. [Online; accessed 02-Jan-2017].

[10] S Srinivasan. Is security realistic in cloud computing?

[11] W3. Simple Object Access Protocol (SOAP) 1.1. https://www.w3.org/TR/soap/, 2007. [Online; accessed 02-Jan-2017].