IBM Project

Report

On

"Analysis and Detection of DDoS Attacks Using Machine Learning Techniques"

Developed By: -

Yashvi Amin (18162101001)
Manush Pandya (18162171010)
Henal Patel (18162121018)

Guided By:-

Prof. Aniket Patel (Internal)
Mr. Ashwin Thandani (External)

Submitted to
Department of Computer Science & Engineering Institute of Computer Technology

Year: 2022

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

CERTIFICATE

This is to certify that the **IBM** Project work entitled **"Analysis and Detection of DDoS Attacks Using Machine Learning Techniques"** by Yashvi Amin (Enrolment No.18162101001) , Manush Pandya(Enrolment No. 18162171010) and Henal Patel (EnrolmentNo.18162121018) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CBA/BDA/CS) Department. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

Place: ICT
Date:

ACKNOWLEDGEMENT

IBM project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Hemal Shah, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. My grateful thanks to Prof. Aniket Patel & Mr. Ashwin Thandani (Internal & External Guides) for their guidance in project work Analysis and Detection of DDoS Attacks Using Machine Learning Techniques, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE Department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

ABSTRACT

A look into the recent attacks all over the world reveals that DDoS attacks have been on the rise. For any service provider website, DDoS attack is the most dangerous of all. Loss of Service for any amount of time leads to huge losses in revenue, customer base and reputation. Considering the given resources and project timeline we are creating a research based project where we explain and demonstrate various methods of DDoS mitigation techniques using open source tools and how to pair them with machine learning techniques to provide the most optimum security for a company deploying a web server.

CHAPTER: 1 INTRODUCTION

CHAPTER 1 INTRODUCTION

With the increasing dependency on web applications in the upcoming future, server security is very important. The most common attacks faced by servers are DDoS attacks. DDoS (Distributed Denial of Service Attacks) use the help of bots that send fake responses to the server this slows down or sometimes crashes the server. This causes the user experience to be affected inturn damaging the webapps usability and reputatation of the company. Although readymade industry tools are available for DDoS detection, for small companies they can be very expensive.

By considering this above scenario, we decided to make a system which helps to detect and mitigate DDoS attacks by utilizing open source tools readily available on the market and IBM Watson. The mitigation tools fetch the server request data and IBM Watson uses said data to analyze DDoS attack patterns and help the tools in enabling quicker detection process.

The main focus of this project is to utilize server request data and make effective use of this data in the analysis and automation of mitigation using various open source tools. Below is the list of the tools and technologies which we have used in this project: -

- DDoS Detection & Mitigation tools
  - DDoS Deflate
- Proxy Server Load Balancer
  - HAProxy
- Log Collection and Management
  - Graylog2
  - Wireshark
  - Tcpdump
- DDoS Attack Tools
  - LOIC - windows
  - HOIC - windows
  - HULK - kali
  - PyLoris - windows
  - Tors hammer - kali
  - SLOWLORIS - kali
  - DDOSIM - windows
  - RUDY - windows(cmd)

- HYENAE - windows
- DAVOSET - windows
- GOLDENEYE - kali
- HPING3 - kali(built in)

CHAPTER: 2 PROJECT SCOPE

CHAPTER 2 PROJECT SCOPE

The project is limited to only local server environment setup for testing purposes but can be applied to any live server infrastructure.

CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

Minimum Hardware Requirements

Processor	2.0 GHz
RAM	4GB
HDD	50GB

Table 3.1 Minimum Hardware Requirements

Minimum Software Requirements

Operating System	Ubuntu 20.04
Other tools & tech	Internet browser, Apache, virtual box

Table 3.2 Minimum Software Requirements

Other Tools:

- Ubuntu 20.04 OS for server
- Apache for server environment
- Graylog2 for log management
- Wireshark & tcpdump for log collection
- HAProxy for load balancing
- DDOS Deflate for DDOS mitigation
- IBM Watson for Machine learning
- Jira for project management
- Virtual Box

CHAPTER: 5 PROJECT PLAN

CHAPTER 5 PROJECT PLAN

5.1 List of Major Activities

Task: - 1
Searching for Open Source DDoS mitigation tools

Task: - 2
Setup and Testing of DDoS Detection tools

Task: - 3
Setup Log Collection and management tools

Task: - 4
Setup Load Balancer and backup web servers

Task: - 5
Implementing IBM Watson Environment

Task: - 6
Creating machine learning models using Auto AI

Task: -7
Creating Deployment Space and deploying machine Learning model

Task: - 8
Training & Testing of model

Task: - 9
Creating API in IBM Watson

Task: - 10
Creating Python Script For Email Alerts.

Task: - 11
Creating Login, Registration and Dashboard

Task: - 12
Connecting UI with the machine learning Model

CHAPTER: 6 IMPLEMENTATION DETAILS

CHAPTER 6 IMPLEMENTATION DETAIL

6.1 Searching for Open Source DDoS mitigation tools

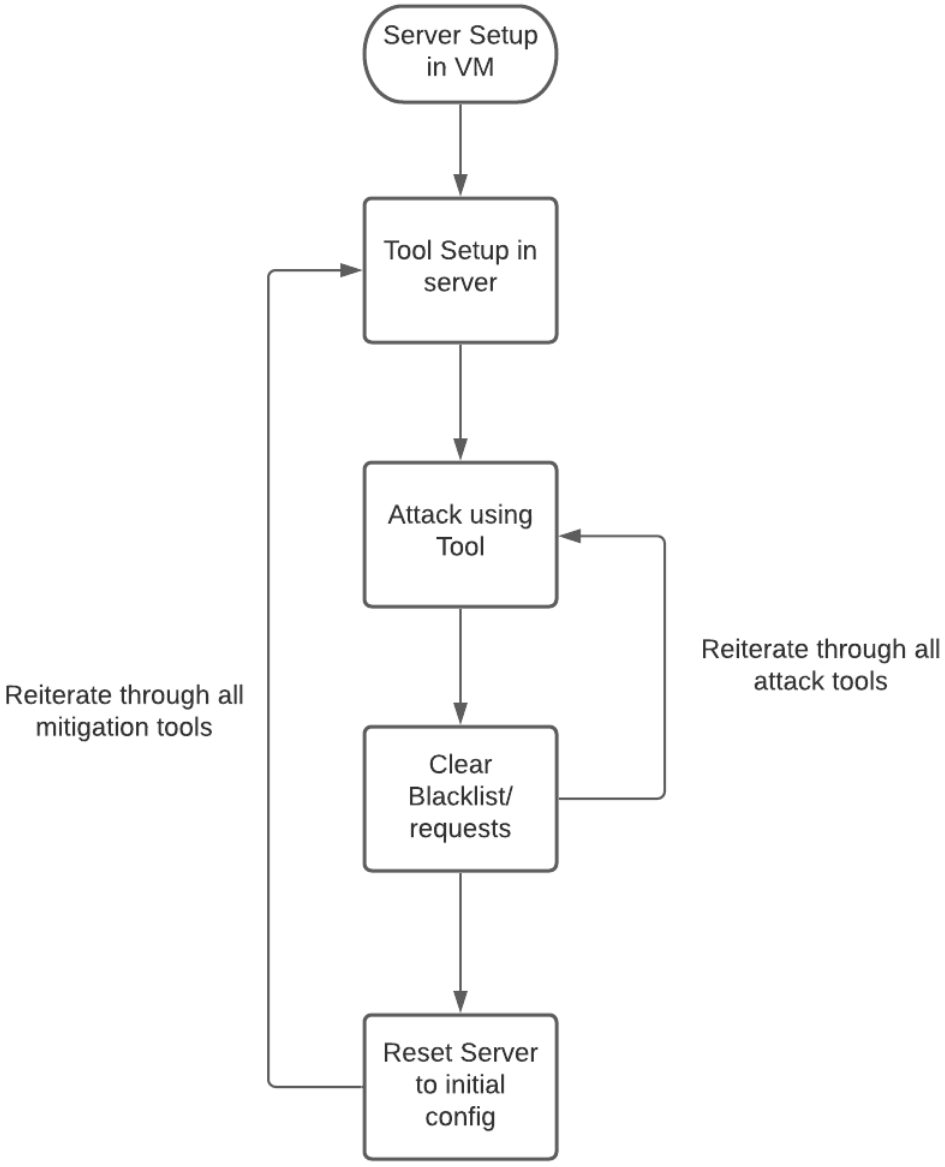
The following tools are the most reputed and utilized open source Ddos detection and mitigation tools.

- *DDoS Deflate*
- *DDoSMon*
- *Fail2ban*
- *Fastnetmon*
- *HAProxy*

6.2 Setup and Testing of DDoS mitigation tools

In this phase, we setup and test the above tools under various conditions and test how well they respond against Ddos attacks and which would be the best possible option.

Testing Procedure



6.3 Setup Log Collection and management tools

6.3.1 Wireshark

Wireshark is available in the Universe repository of Ubuntu. You can enable universe repository and then install it like this:

```
sudo add-apt-repository universe
sudo apt install wireshark
```

6.3.2 Graylog2

PREREQUISITES

Taking a minimal server setup as base will need these additional packages:

```
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install apt-transport-https openjdk-<version_number>-jre-headless uuid-runtime pwgen
```

Shell
Copy

If you get an error stating *Unable to locate the package*, you likely need to enable the universe repository which can be done by typing the below command, and subsequent commands as follows:

```
sudo add-apt-repository universe
sudo apt-get update && sudo apt-get upgrade
sudo apt-get install apt-transport-https openjdk-<version_number>-jre-headless uuid-runtime pwgen
```

Shell
Copy

MONGODB

The official MongoDB repository provides the most up-to-date version and is the recommended way of installing MongoDB:

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.0.list
sudo apt-get update
sudo apt-get install -y mongodb-org
```

Shell
Copy

For e.g. corporate proxies and other non-free environments you can use a keyserver approach via wget.

```
wget -qO- 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x9DA31620334BD75D9DCB49F368818C72E52529D4' | sudo apt-key add -
```

Shell
Copy

The last step is to enable MongoDB during the operating system's startup and verify it is running

```
sudo systemctl daemon-reload
sudo systemctl enable mongod.service
sudo systemctl restart mongod.service
sudo systemctl --type=service --state=active | grep mongod
```

Shell
Copy

ELASTICSEARCH

Graylog can be used with Elasticsearch 7.x, please follow the below instructions to install the open source version of Elasticsearch.

```
wget -q https://artifacts.elastic.co/GPG-KEY-elasticsearch -O myKey
sudo apt-key add myKey
echo "deb https://artifacts.elastic.co/packages/oss-7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
sudo apt-get update && sudo apt-get install elasticsearch-oss
```

Shell
Copy

The above instructions are a derivative from the Elasticsearch install page

Modify the Elasticsearch configuration file (/etc/elasticsearch/elasticsearch.yml) and set the cluster name to Graylog and uncomment action.auto_create_index: false to enable the action:

```
sudo tee -a /etc/elasticsearch/elasticsearch.yml > /dev/null <<EOT

action.auto_create_index: false
EOT
```

Shell
Copy

After you have modified the configuration, you can start Elasticsearch and verify it is running.

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart elasticsearch.service
sudo systemctl --type=service --state=active | grep elasticsearch
```

Shell
Copy

GRAYLOG

Now install the Graylog repository configuration and Graylog itself with the following commands:

```
wget https://packages.graylog2.org/repo/packages/graylog-4.2-repository_latest.deb
sudo dpkg -i graylog-4.2-repository_latest.deb
sudo apt-get update && sudo apt-get install graylog-server graylog-enterprise-plugins gra
ylog-integrations-plugins graylog-enterprise-integrations-plugins
```

Shell
Copy

Hint

If you do not want the **Integrations Plugins** or the **Enterprise Plugins** installed, then simply run sudo apt-get install graylog-server

Edit the Configuration File

Read the instructions within the configurations file and edit as needed, located at /etc/graylog/server/server.conf. Additionally add password_secret and root_password_sha2 as these are *mandatory* and **Graylog will not start without them**.

To create your root_password_sha2 run the following command:

```
echo -n "Enter Password: " && head -1 </dev/stdin | tr -d '\n' | sha256sum | cut -d" " -f1
```

Shell

Copy

To be able to connect to Graylog you should set http_bind_address to the public host name or a public IP address of the machine you can connect to. More information about these settings can be found in Configuring the web interface.

Note
If you're operating a single-node setup and would like to use HTTPS for the Graylog web interface and the Graylog REST API, it's possible to use **NGINX or Apache as a reverse proxy**.

The last step is to enable Graylog during the operating system's startup and verify it is running.

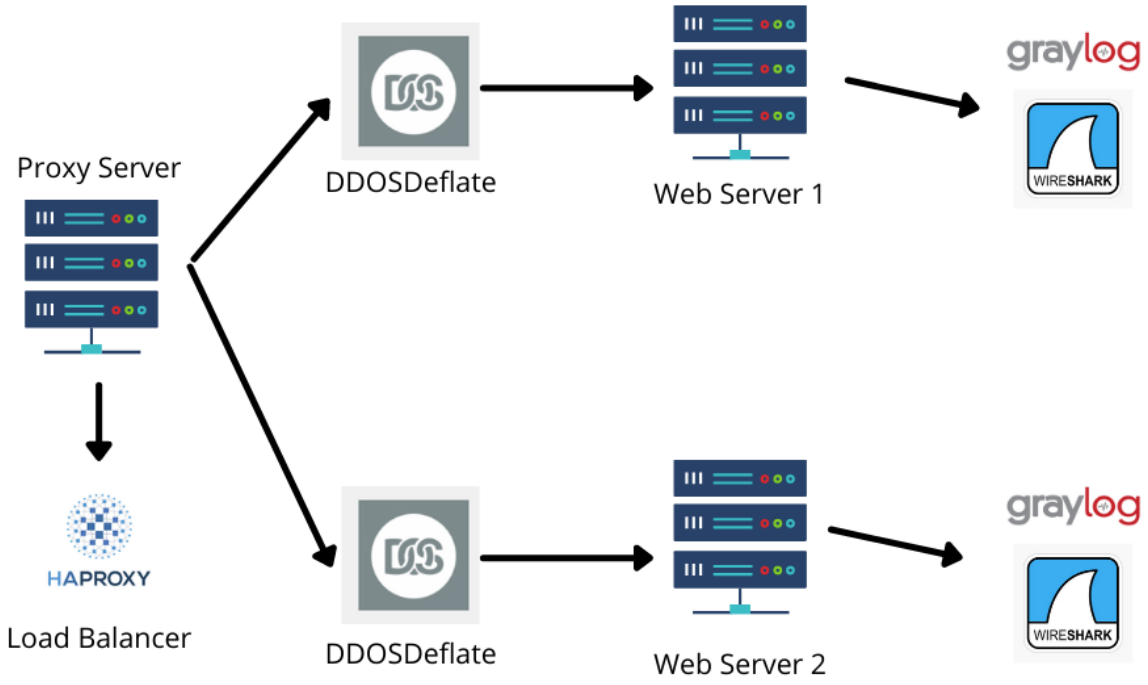
```
sudo systemctl daemon-reload
sudo systemctl enable graylog-server.service
sudo systemctl start graylog-server.service
sudo systemctl --type=service --state=active | grep graylog
```

6.4 Setup Proxy server and load balancer

For this purpose, we used HAProxy load balancer in addition with 2 web servers in VMs.

```
192.168.1.26 - HAPROXY server
192.168.1.46 - web server 1 (main server)
192.168.1.27 - web server 2 (backup server)
```

Completed network setup

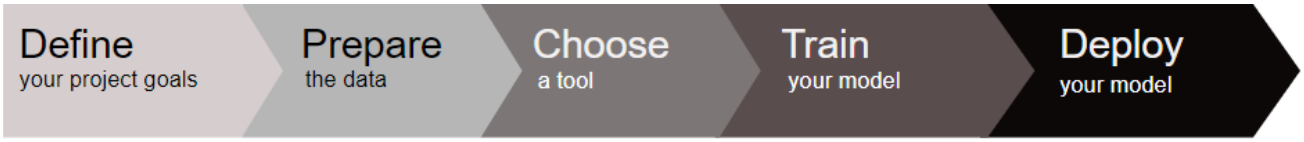


6.4.1 Data Generated

No.	Time	Source	Destination	Protocol	Length	Info
1	0	Giga-Byt_3c:7f:2f	Broadcast	ARP	60	Who has 192.168.1.34? Tell 192.168.1.4
2	0.502128482	Giga-Byt_3c:7f:2f	Broadcast	ARP	60	Who has 192.168.1.34? Tell 192.168.1.4
3	1.502746306	Giga-Byt_3c:7f:2f	Broadcast	ARP	60	Who has 192.168.1.34? Tell 192.168.1.4
4	3.346199613	PLUS_ea:85:eb	Broadcast	ARP	60	ARP Announcement for 192.168.1.10
5	7.77155044	192.168.1.15	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
6	8.067952612	192.168.1.15	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
7	8.380805595	192.168.1.15	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
8	13.37062286	PLUS_ea:85:eb	Broadcast	ARP	60	ARP Announcement for 192.168.1.10
9	16.01097717	Giga-Byt_3c:7f:2f	Broadcast	ARP	60	Who has 192.168.1.34? Tell 192.168.1.4
10	17.01047136	Giga-Byt_3c:7f:2f	Broadcast	ARP	60	Who has 192.168.1.34? Tell 192.168.1.4
11	17.43237815	192.168.1.46	192.168.1.1	DNS	90	Standard query 0x2288 SRV _http._tcp.security.ubuntu.com
12	17.43360019	192.168.1.46	192.168.1.1	DNS	92	Standard query 0x63a6 SRV _https._tcp.artifacts.elastic.co
13	17.43858152	192.168.1.46	192.168.1.1	DNS	88	Standard query 0x9544 SRV _https._tcp.repo.mongodb.org
14	17.44338099	192.168.1.46	192.168.1.1	DNS	89	Standard query 0x7438 SRV _http._tcp.archive.ubuntu.com
15	17.44600785	192.168.1.46	192.168.1.1	DNS	93	Standard query 0x7479 SRV _https._tcp.packages.graylog2.org
16	17.44879705	192.168.1.46	192.168.1.1	DNS	92	Standard query 0x8b65 SRV _http._tcp.in.archive.ubuntu.com
17	17.49313306	192.168.1.1	192.168.1.46	DNS	150	Standard query response 0x7438 SRV _http._tcp.archive.ubuntu.com SOA ns1.canonical.com
18	17.49620953	192.168.1.46	192.168.1.1	DNS	78	Standard query 0x914c A archive.ubuntu.com
19	17.49631261	192.168.1.46	192.168.1.1	DNS	78	Standard query 0xccc4 AAAA archive.ubuntu.com
20	17.49963775	192.168.1.1	192.168.1.46	DNS	142	Standard query response 0x914c A archive.ubuntu.com A 185.125.190.36 A 185.125.190.39 A 91.189.91.39 A 91.189.91.38
21	17.5004884	192.168.1.1	192.168.1.46	DNS	190	ry response 0xccc4 AAAA archive.ubuntu.com AAAA 2001:67c:1562::15 AAAA 2001:67c:1562::18 AAAA 2620:2d:4000:1::19 AAAA 2620:
22	17.50279012	192.168.1.46	185.125.190.36	TCP	74	55518 > 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1568773441 TSecr=0 WS=128
23	17.6607708	185.125.190.36	192.168.1.46	TCP	74	80 > 55518 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1452 SACK_PERM=1 TSval=2842362719 TSecr=1568773441 WS=128
24	17.66080677	192.168.1.46	185.125.190.36	TCP	66	55518 > 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1568773599 TSecr=2842362719
25	17.66106182	192.168.1.46	185.125.190.36	HTTP	271	GET /ubuntu/dists/focal/InRelease HTTP/1.1
26	17.78723245	192.168.1.15	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
27	17.81883814	185.125.190.36	192.168.1.46	TCP	66	80 > 55518 [ACK] Seq=1 Ack=206 Win=64640 Len=0 TSval=2842362922 TSecr=1568773599
28	17.81907665	185.125.190.36	192.168.1.46	HTTP	277	HTTP/1.1 304 Not Modified
29	17.81908528	192.168.1.46	185.125.190.36	TCP	66	55518 > 80 [ACK] Seq=206 Ack=212 Win=64128 Len=0 TSval=1568773757 TSecr=2842362922

6.5 IBM Watson

Using IBM Watson Machine Learning, we can build analytical models and neural networks, trained with your own data. We can also deploy models, scripts, and functions, manage our deployments, and prepare our assets to put into production to generate predictions and insights.

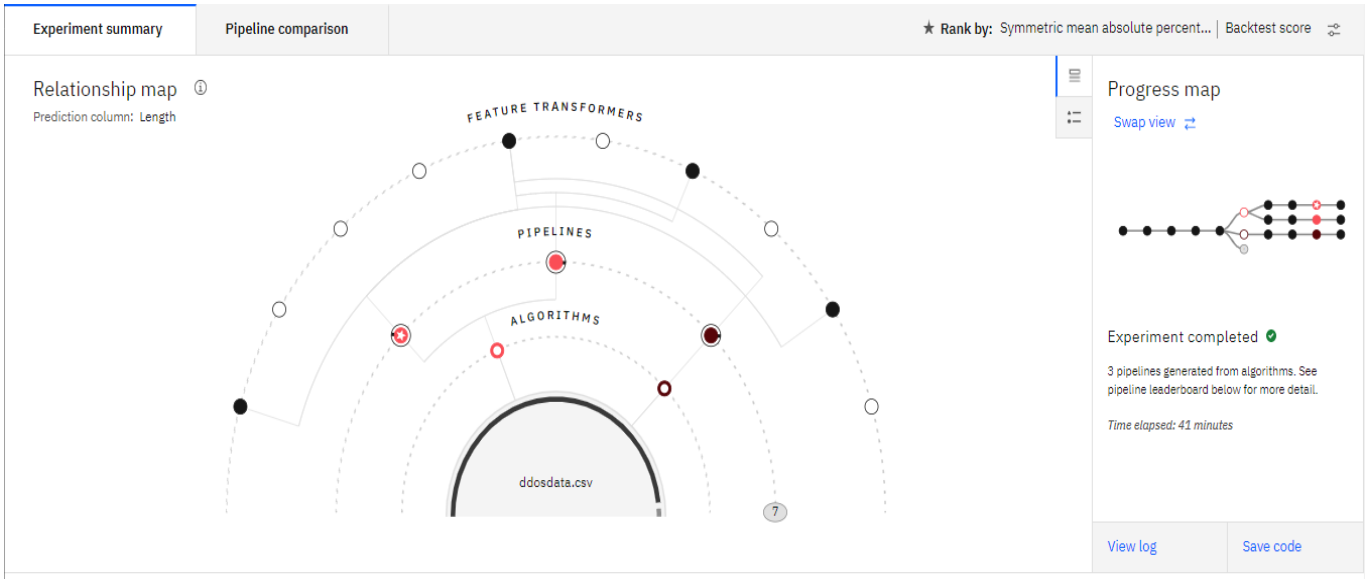


6.5.1 Steps For Detection of Ddos Attack

- Created Watson Studio Environment with machine Learning Service for the deployment
- Building and deploying a machine learning model with Auto AI

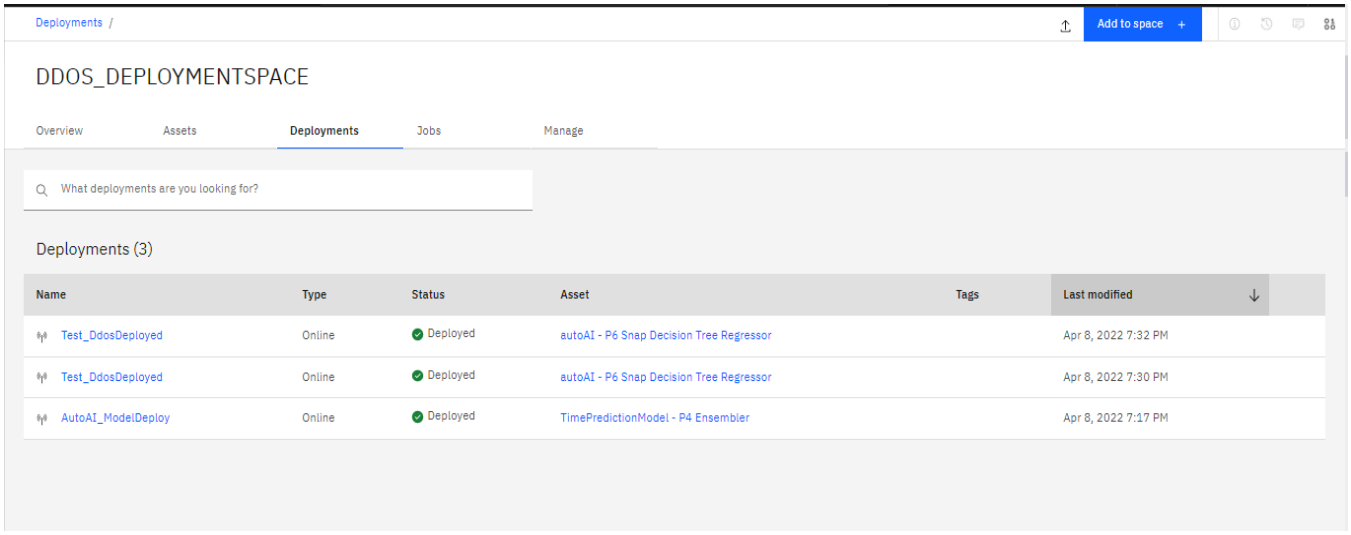
Basic workflow includes these tasks:

- Create a project
- Associate the Watson Machine Learning service with the project
- Create an Auto AI experiment in the project
- Build and train a model
- Save a pipeline as a model
- Deploy the model
- Test the model



Experiment Summary

Pipeline Comparision

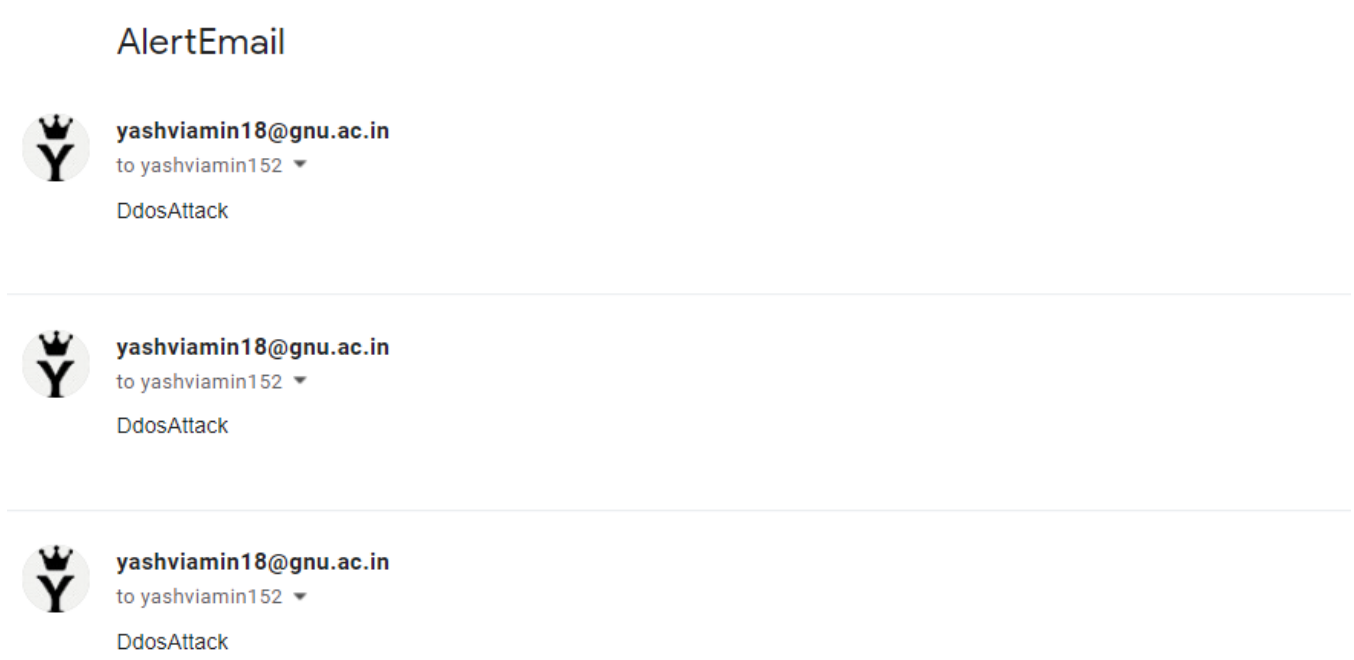


Machine Learning Models deployment

6.5.1 Email Alerts

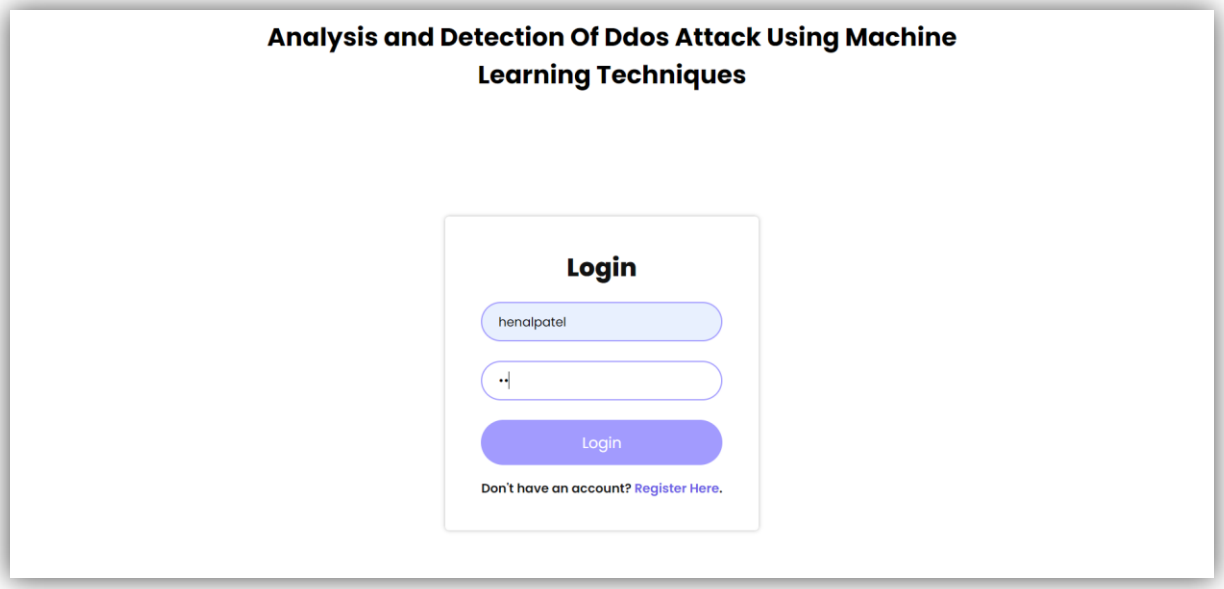
```
import smtplib
from email.message import EmailMessage
import requests
import json

API_KEY  = "eWKzaxa-HP0hgtj3e7OgK6ZXipW88MsA7f7pBxkSay4f"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',  data={"apikey": API_KEY,
"grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually  define and pass the array(s) of values to be scored in the next line
payload_scoring={"input_data": [
                {
                        "fields":  [
                                "No.",
                                "Time",
                                "Source",
                                "Destination",
                                "Protocol",
                                "Info"
                        ],
                        "values":  [
                                ["",0.8676,"185.125.190.36","192.168.1.46","HTTP",""]
                            ]
                }
        ]
   }

response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/9c300ed2-bf32-43e8-bdca-
9417e3b16e07/predictions?version=2022-04-08', json=payload_scoring,
 headers={'Authorization': 'Bearer ' + mltoken})
```

```
print("Scoring response")
predictions= response_scoring.json()
print(predictions)
pred= predictions['predictions'][0]['values'][0][0]

if (pred >= 200):
    def email_alert(subject,body,to):
        msg=EmailMessage()
        msg.set_content(body)
        msg['subject']=subject
        msg['to']=to

        user="yashviamin18@gnu.ac.in"
        msg['from']=user
        password="voxgxoducxnaeqpf"

        server=smtplib.SMTP("smtp.gmail.com",587)
        server.starttls()
        server.login(user,password)
        server.send_message(msg)
        server.quit()

    if __name__=='__main__':
        email_alert("AlertEmail",   "DdosAttack", "yashviamin152@gmail.com")


else:
   print('You are safe!!')
```

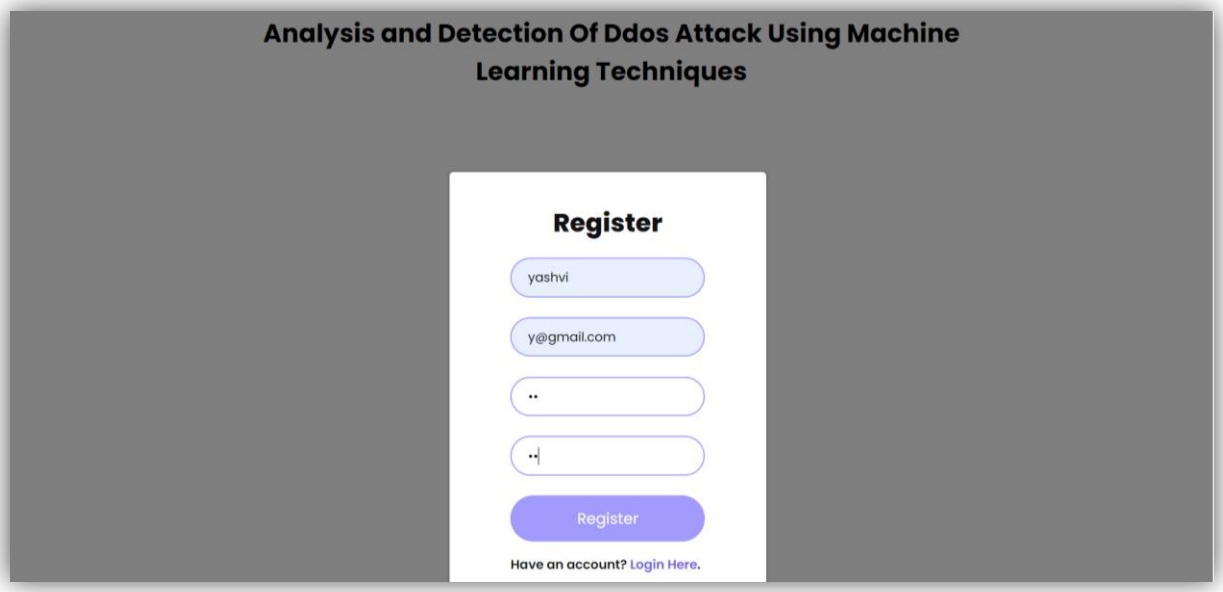


6.6 Frontend Side

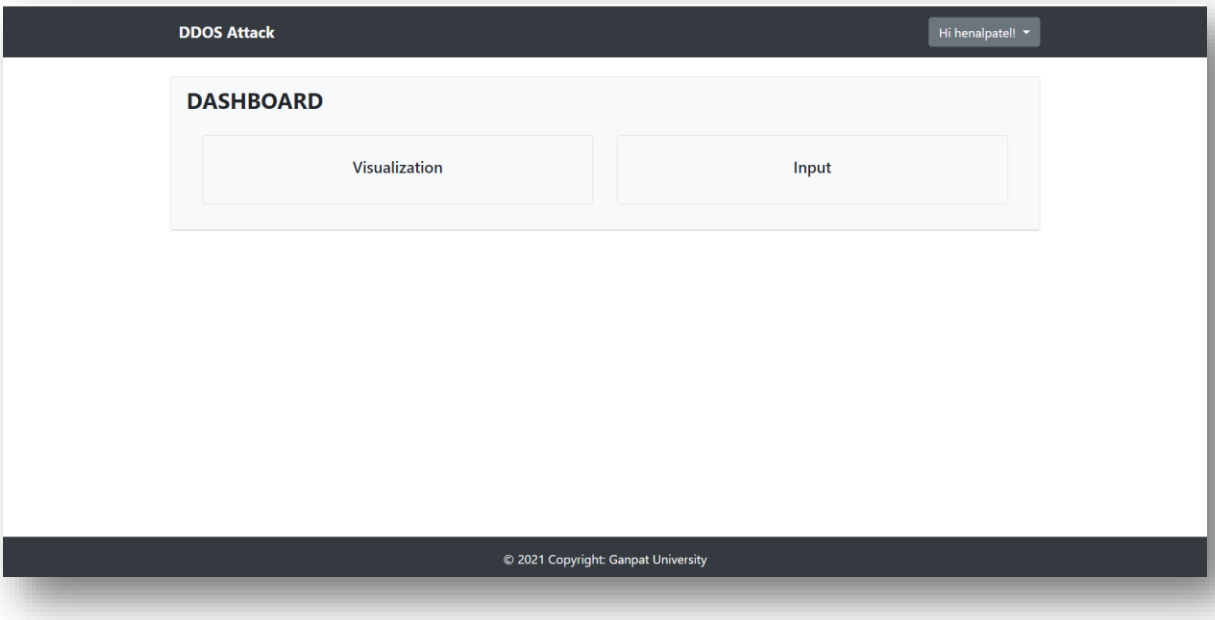
LOGIN PAGE

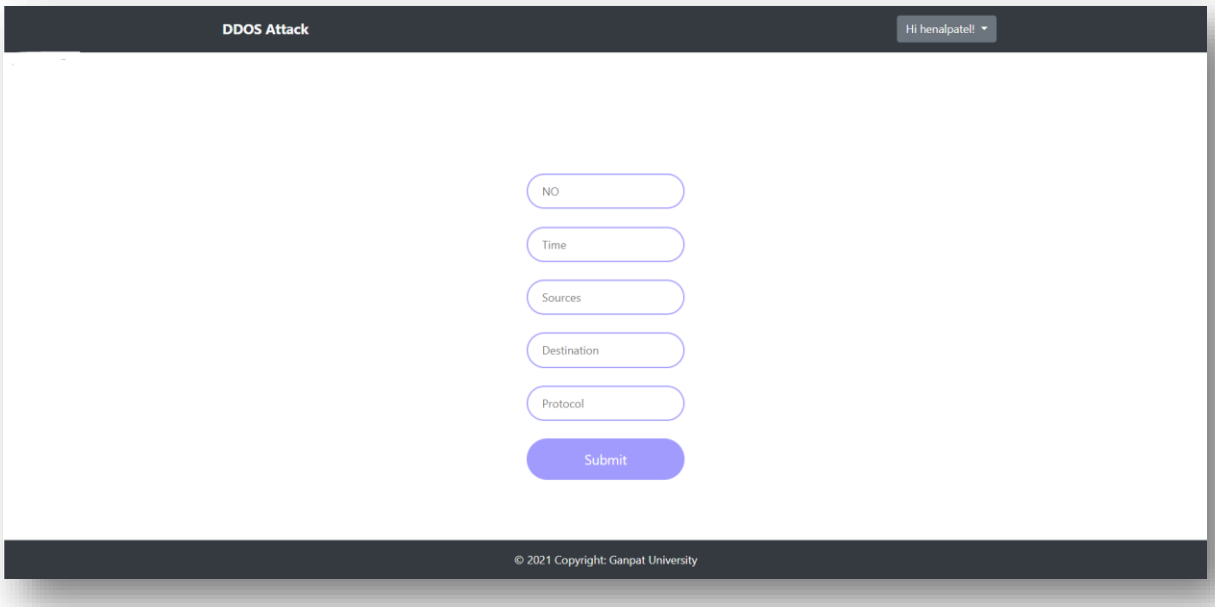


REGISTRATION

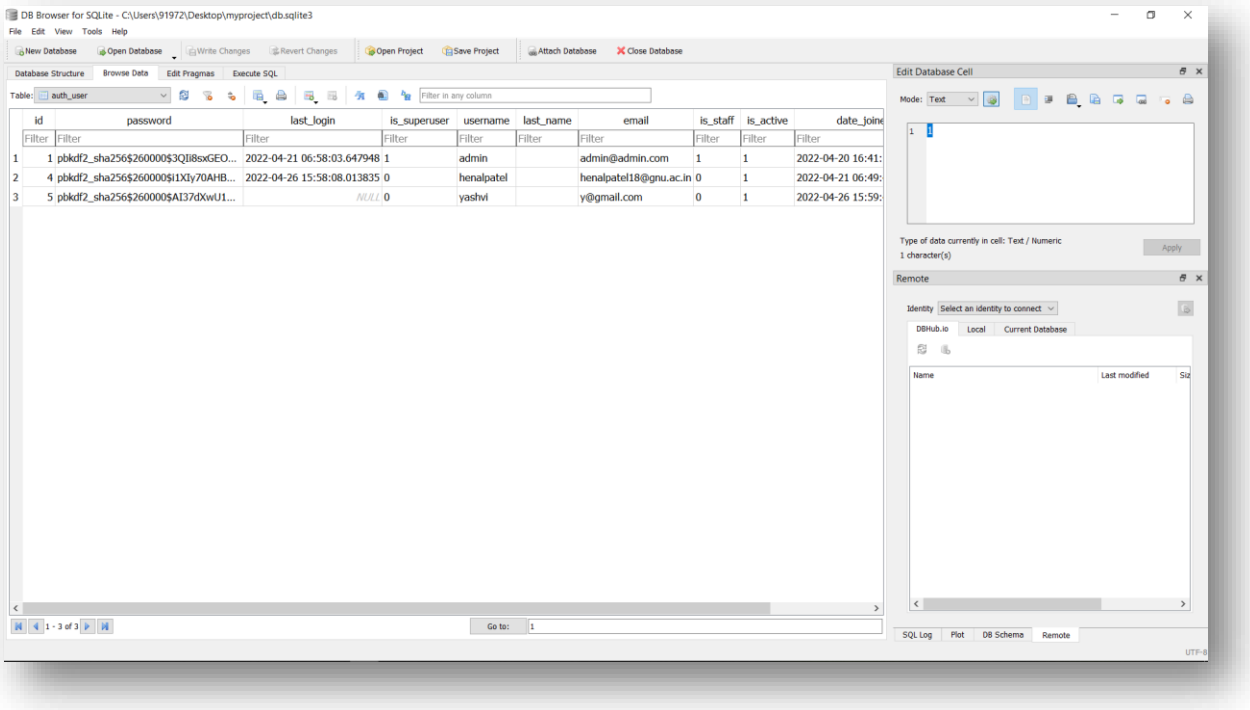


DASHBOARD PAGE



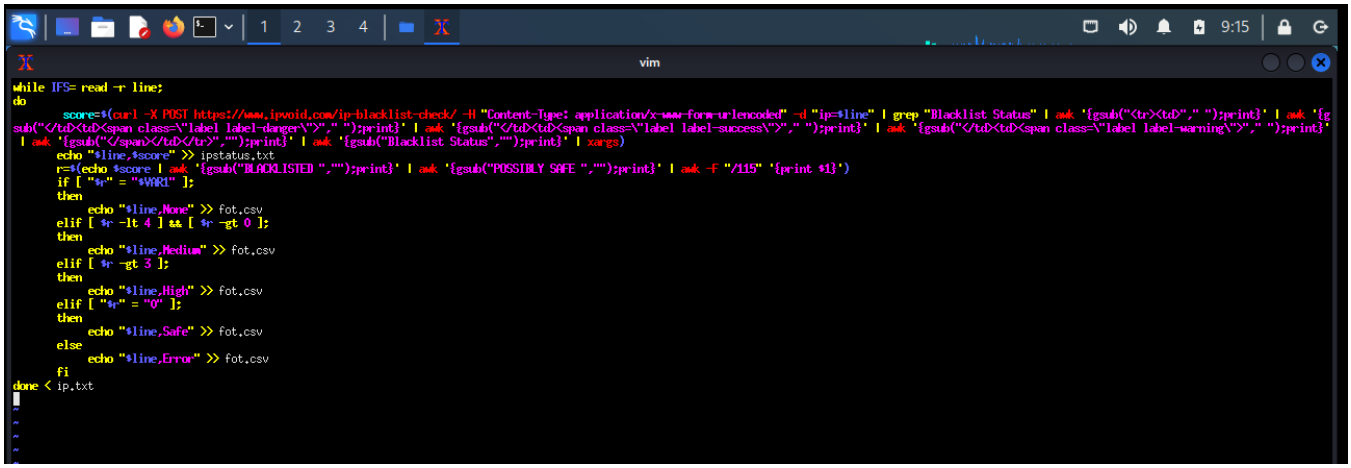


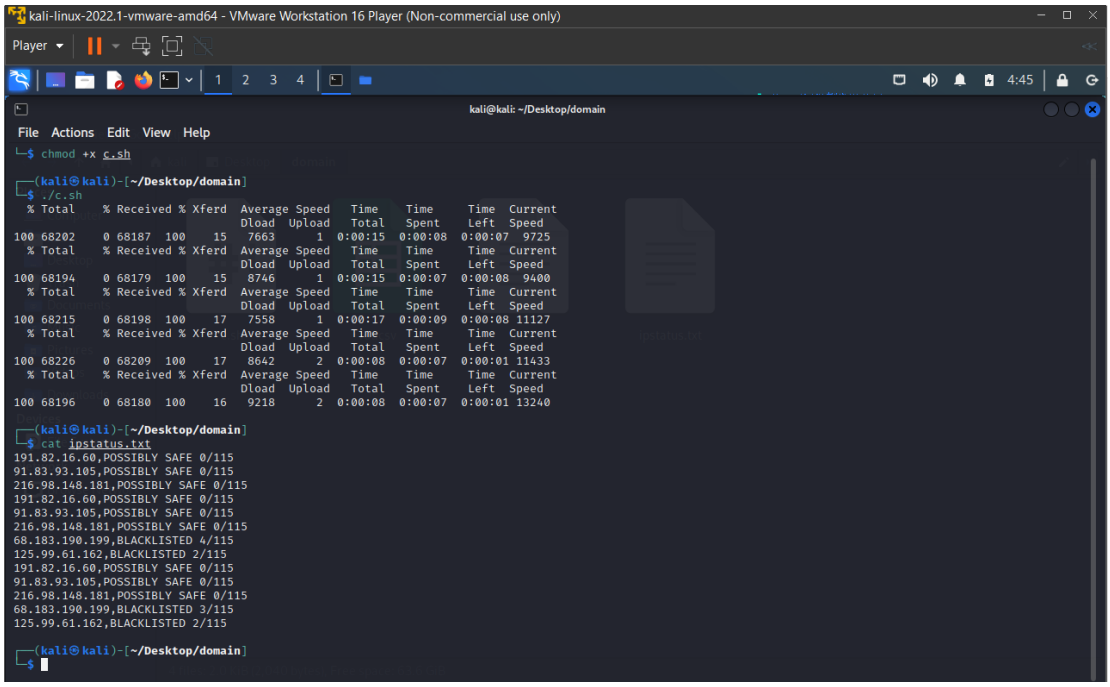
DATABSE DETAILS

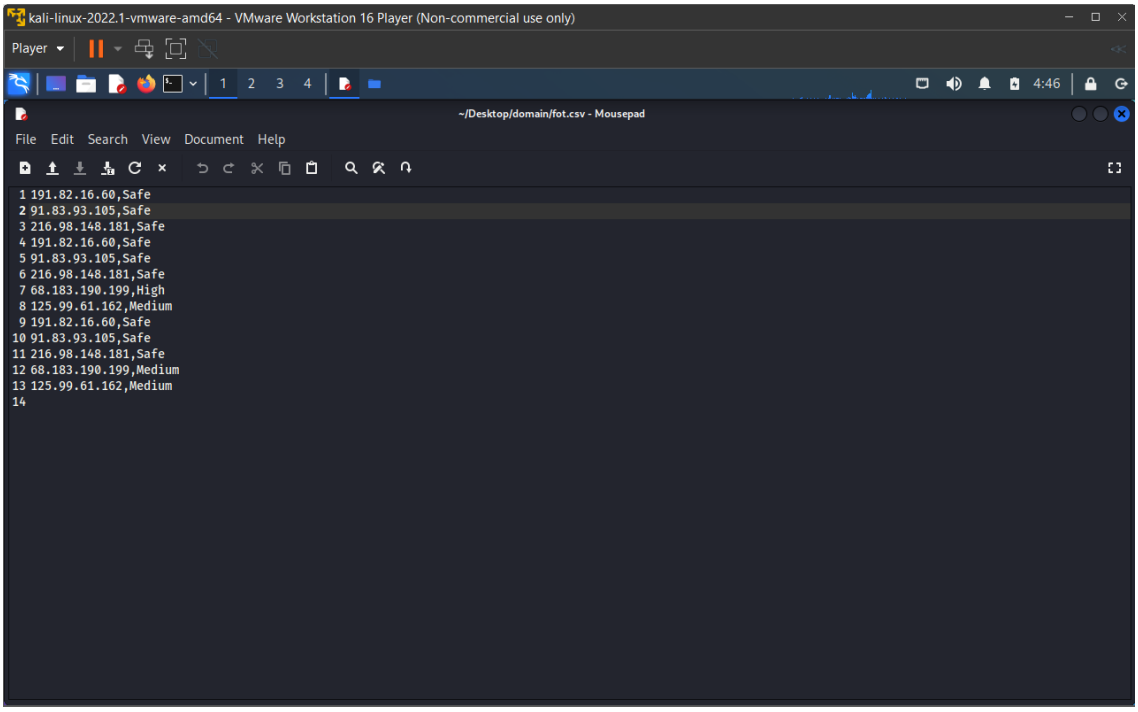


6.5 Domain checkup

- DDoS occurs due to plethora of request made by bots or infected system to the target using various different ips.
- Thus, IP reputation along with identifying and differentiating malicious IPs with the genuine IP plays an important role in the project.
- The task of segregating malicious IPs and classifying into 3 different categories i.e High Medium Low.
- The defined parameter for categories is set hypothetically. To successfully get this task done, IPVoid is been used to check the IP reputation score and provides reliable results.
- Moreover the script is designed to integrate the functionality and use it in broader case i.e identifying the ip reputation score for large dataset.

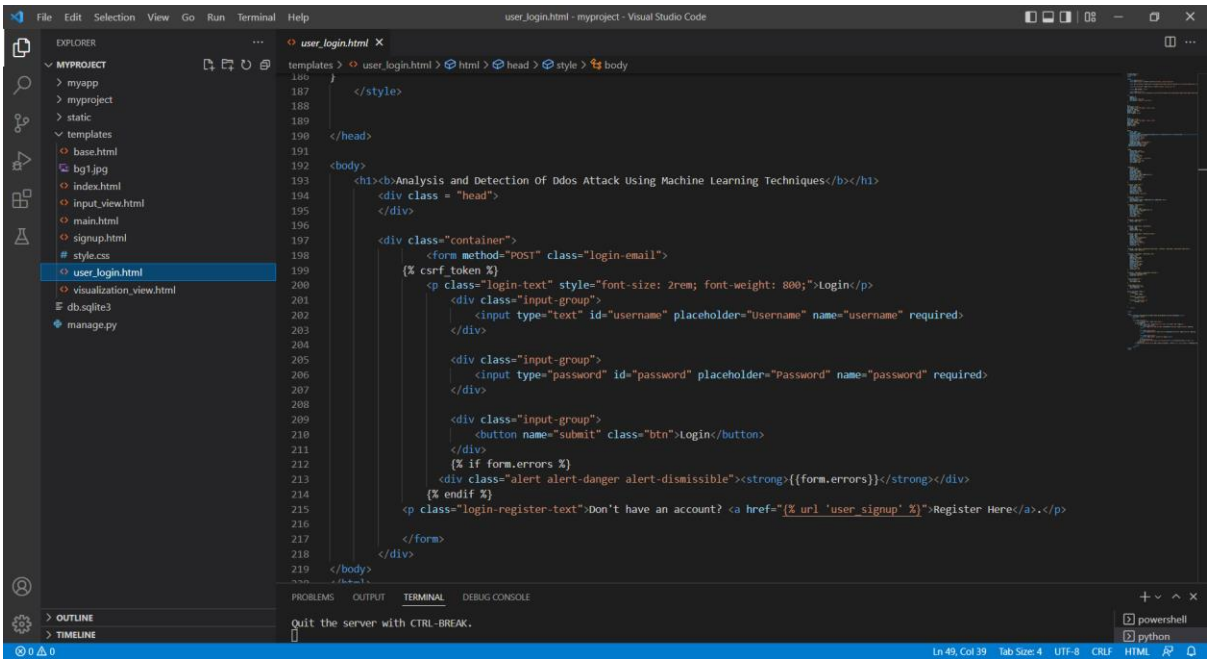

vim
while IFS= read -r line;
do
score=$(curl -X POST https://www.ipvoid.com/ip-blacklist-check/ -H "Content-Type: application/x-www-form-urlencoded" -d "ip=$line" | grep "Blacklist Status" | awk '{gsub("<tr><td>" ");print}' | awk '{gsub("</td><td><span class=\"label label-danger\">" ");print}' | awk '{gsub("</td><td><span class=\"label label-success\">" ");print}' | awk '{gsub("</td><td><span class=\"label label-warning\">" ");print}' | awk '{gsub("</td></tr>" "");print}' | awk '{gsub("Blacklist Status" "");print}' | xargs)
echo "$line,$score" >> ipstatus.txt
r=$(echo $score | awk '{gsub("BLACKLISTED " "");print}' | awk '{gsub("POSSIBLY SAFE " "");print}' | awk -F "/115" '{print $1}')
if ["$r" = "$VAR1"];
then
echo "$line,None" >> fot.csv
elif [$r -lt 4] && [$r -gt 0];
then
echo "$line,Medium" >> fot.csv
elif [$r -gt 3];
then
echo "$line,High" >> fot.csv
elif ["$r" = "0"];
then
echo "$line,Safe" >> fot.csv
else
echo "$line,Error" >> fot.csv
fi
done < ip.txt


kali-linux-2022.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
kali@kali: ~/Desktop/domain
File Actions Edit View Help
└─$ chmod +x c.sh
┌──(kali㉿kali)-[~/Desktop/domain]
└─$ ./c.sh
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68202 0 68187 100 15 7663 1 0:00:15 0:00:08 0:00:07 9725
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68194 0 68179 100 15 8746 1 0:00:15 0:00:07 0:00:08 9400
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68215 0 68198 100 17 7558 1 0:00:17 0:00:09 0:00:08 11127
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68226 0 68209 100 17 8642 2 0:00:08 0:00:07 0:00:01 11433
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 68196 0 68180 100 16 9218 2 0:00:08 0:00:07 0:00:01 13240
┌──(kali㉿kali)-[~/Desktop/domain]
└─$ cat ipstatus.txt
191.82.16.60,POSSIBLY SAFE 0/115
91.83.93.105,POSSIBLY SAFE 0/115
216.98.148.181,POSSIBLY SAFE 0/115
191.82.16.60,POSSIBLY SAFE 0/115
91.83.93.105,POSSIBLY SAFE 0/115
216.98.148.181,POSSIBLY SAFE 0/115
68.183.190.199,BLACKLISTED 4/115
125.99.61.162,BLACKLISTED 2/115
191.82.16.60,POSSIBLY SAFE 0/115
91.83.93.105,POSSIBLY SAFE 0/115
216.98.148.181,POSSIBLY SAFE 0/115
68.183.190.199,BLACKLISTED 3/115
125.99.61.162,BLACKLISTED 2/115
┌──(kali㉿kali)-[~/Desktop/domain]
└─$


kali-linux-2022.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
1 2 3 4
4:46
~/Desktop/domain/fot.csv - Mousepad
File Edit Search View Document Help
1 191.82.16.60,Safe
2 91.83.93.105,Safe
3 216.98.148.181,Safe
4 191.82.16.60,Safe
5 91.83.93.105,Safe
6 216.98.148.181,Safe
7 68.183.190.199,High
8 125.99.61.162,Medium
9 191.82.16.60,Safe
10 91.83.93.105,Safe
11 216.98.148.181,Safe
12 68.183.190.199,Medium
13 125.99.61.162,Medium
14

Working of script:

- The script starts with input of file with a list of ip address in a format of xlsx and then request IPVoid with the IPVoid using curl which in response shows the HTML response.
- Later script filters out the specific part which includes the IP Score and stores it into new file in specific format with the IP address and score, can be used as input for the next phase.



6.7 Challenges Faced & Mitigation

Integration of PHP with Python backend script
Mitigation: Though PHP supports python scripts still for better support , migrated to the Django from PHP

Responsiveness
Mitigation: Implemented responsive css to the application

Database Support
Mitigation: Intially, we were using MYSQL which was creating issue while testing on different devices, thus switched to portable database SQLite for the application

Dashboard Integration
Mitigation: Displaying the output in the graphical dashboard was quite tedious task to perform. Thus django library helped the integration very simple

Watson
limited source for train the model
limited data can be used for training and testing

CHAPTER: 7 CONCLUSION AND FUTURE WORK

CHAPTER 7 CONCLUSION AND FUTURE WORK

Conclusion

To reiterate my views, as we all know that the ratio of DDoS Attacks has rapidly increased in the last few years. A company needs to protect its service and server infrastructure from such attacks. There are industry tools available that provide optimum Ddos protection but more often than not they are very expensive for small buisness. Small business don't have the instrastructure, resources or technical know how to deploy these solutions. In this case open source tools are generally looked forward upon. But open source tools lack the appropriate analysis and detection mechanisms. We plan on improving these solutions with the help of IBM Watson and create a better and more cost effective system that provides moderate DDoS mitigation.

Future Enhancement

Till now we have setup testing servers and tested the Ddos detection tools and their features. We have setup a full server environment with load balancers and proxy servers that allows us to do real time Ddos monitoring and log generation as well as mitigation. We have also setup IBM Watson to take the data from the servers and generate alerts based on that. Email alerts have also been configured based on the IBM reports. Following are some of the suggestion for future enhancement of this project.

- Create a pipeline to send logs to IBM Watson in real time by sending data from the servers.
- Improve on the UI system by adding MFA, and a better reporting system for ease of use by system administrators.
- Reduce the turnaround time and improving accuracy for calculation of DDoS attack by training the model with more data.

CHAPTER: 8 BIBLIOGRAPHY

CHAPTER: 8 BIBLIOGRAPHY

- *https://www.ibm.com/*
- *https://stackoverflow.com/*
- *https://www.ijeat.org/*

PLAGARISM

DDos Attack

by Amin yashvi

Submission date: 27-Apr-2022 02:34PM (UTC+0530)
Submission ID: 1820738548
File name: ContentForPlagarism.docx (17.77K)
Word count: 966
Character count: 4812

DDos Attack

ORIGINALITY REPORT

4%	3%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dataplatform.cloud.ibm.com Internet Source	2%
2	www.itgovernanceusa.com Internet Source	1%
3	Bartosz Chrabski, Cezary Orłowski. "Chapter 9 A Reference Model for the Selection of Open Source Tools for Requirements Management", Springer Science and Business Media LLC, 2013	1%