Proceedings of the Pakistan Academy of Sciences:
A. Physical and Computational Sciences: 57 (2): 97-108 (2020)

ISSN: 2518-4245 (print), 2518-4253 (online)

Pakistan Academy of Sciences

Research Article

Distributed Denial of Service Attacks Analysis, Detection, and Mitigation for the Space Control Ground Network

Ahmed Ramzy Shaaban*, Essam Abdelwanees, and Mohamed Hussein

Department of Communication, Military Technical College, Cairo, Egypt

Abstract: After launching any satellite, it must be controlled from the ground by the mission control center (MCC) by receiving the health state telemetry and issuing telecommand to control it or to execute its mission so, the network of MCC should be kept safe from any kind of malicious attacks such as Distributed Denial of Service (DDoS). The DDoS attacks could be launched or deployed either by external or internal attackers. DDoS can be defined generally as follow: it is an attempt to exhaust target server resources or consume the available bandwidth to make the target server unavailable to the normal clients. MCC network was simulated using virtual machines – 8 virtual machines. More than 5 types of DDoS tried to attack the simulated MCC network but 2 types were chosen – HTTP and TCP flood- to be designed because of its effectiveness. The analysis was done before and after the attacks by analyzing the captured traffic by Wireshark software. According to the deep analysis results, the detection algorithm was designed to detect the applied attacks. Now the attacker machines are known, so mitigation of theses attacked machines was done by adding blocking rules in the windows firewall automatically. Mitigation was done simply and in a straightforward way but with some instability. Consequently, a new mitigation technique will be developed to block DDoS attacks.

Keywords: Mission control center (MCC), Distributed Denial of Service (DDoS), TCP flood, DDoS detection, DDoS mitigation.

1. INTRODUCTION

After launching any satellite, you have to control it by receiving health state telemetry and sending telecommand so, the ground control network is required to observe the satellite and control it. Thus, the MCC network is very critical and it should be kept safe from any kind of malicious attacks such that DDoS attack .the victim server was flooded with the incoming traffic which originates from many different sources by overwhelming [1] the server with a massive amount of traffic, causing the server to be crashed or work very slowly and it will be extremely hard to differentiate between traffic from normal users and malicious traffic from attackers.

Thus, it is impossible to prevent the attack simply by blocking a single source. The DDoS attack can be either just game played for fun by internal user attackers, by expert hackers as a part of the cyberwar or for financial purpose. DDoS flooding attacks [2] are one of the considerable concerns for security administrators. Some example of DDoS, An Iranian hackers involved in conspiracies to conduct a coordinated series of distributed denial of service attacks against the United States financial sector and other United States companies from 2011 through 2013, at 2019 the Ministry of Education exam server in Egypt was attacked by DDoS during the exam, at 2018 DDoS attacks targeted a popular online code management server -GitHub- used by a lot of developers. At its peak, this attack saw incoming traffic at a rate of 1.3 terabytes per second (Tbsp.), sending packets at a rate of 126.9 million per second, DDoS attack can be categorized generally into 3 types:

- Application layer attack: known as a layer 7 DDoS attack, the objective of these attacks is to consume the resources of the target. The attack targets the layer where web pages are created

Received: October 2019: Accepted: June 2020
*Corresponding Author: Ahmed Ramzy Shaaban <ramboramzy@yahoo.com>

on the server and supplied in response to HTTP requests, an example of this attack is HTTP flood [4].

- Protocol attack: known as a state-exhaustion attack, cause a service to be unavailable by consuming the available bandwidth of the application servers or intermediate resources like load balancers and firewalls. Protocol attacks exploit weaknesses in layer 3 and layer 4 of the protocol stack to make the target server unavailable, example this attack is an SYN flood [5].
- Volumetric attacks: This type of attack attempts to generate congestion by consuming the available bandwidth between the target server and allowed users in this network.

Secondly, DDoS attack can be divided into [3,6] the following two categories from the connection point of view:

- Connection-based: the attack takes place once a connection between a server and a client has been established via certain protocols.
- Connectionless: the attack does not require a session to be properly established before a source can send "data packets" to the receiver.

Huge amounts of packets are sent to the target server by using amplification software or another way of creating massive traffic, example of this attack is TCP flood. The main concern of the defense algorithm ensures that receiving the expected service for normal clients even during DDoS attacks without any interruption in the service. To diminish the effect of DDoS attack, detection mechanisms [7] should be used during the attack to detect unknown behavior of malicious packets so, necessary procedures should be taken to mitigate this attack, detection techniques can be classified into 4 categories:

- statistical-based method
- Knowledge-Based method
- Soft-Computing method
- Machine-Learning method

Some of these techniques focus on software-defined networks, cloud computing web traffic, and big data strategies. For example, applying filter [8] by source IP address to all ingress (incoming traffic to the local network) and egress (outgoing traffic from the local network) is the primitive technique to detect DDoS attack. In this way, we can avoid IP spoofing [9] which has been stimulated by attackers in their packet.

This technique supposes that the IP address of the attack traffic is spoofed i.e. attacker needs to hide his IP and exploit protocol vulnerabilities. However, the filtering technique is ineffective unless executed by completely ingress routers. So Many techniques [10] have been developed to detect and isolate the impact of DDoS attacks. It is useful to choose DDoS mitigation techniques that keep engineers and network administrators on-site monitoring traffic continuously.

This enables a faster response time [11] for detection and mitigation so faster decisions will be taken. The three primary components of DDoS attack detection that jointly define all the elements of an attack that effect on the network infrastructure and analysis of legitimate traffic and malicious traffic Hence an effective simulation of the DDoS attack requires a combination of traffic generation software, software for statistical analysis and detection algorithm, therefore the simulation of DDoS divided into three main sections:

- DDoS attack methodology: in this section, the DDoS attack types used to attack our simulated network were described and explained.
- DDoS attack analysis: in this section, Wireshark and the designed sniffer program were used to analyze transferred packets between the target server and other peripherals before deploying a DDoS attack (TCP &HTTP flood) and after it.
- DDoS attack detection and mitigation: it is the last section in the paper, explains the proposed detection algorithms to detect the used DDoS attack and how can we mitigate this attack.

2. RELATED WORKS

Many detections and isolation mechanisms for mitigating DDoS flood attacks have been developed in the last few years. Effective DDoS TCP flood attack detection in a cloud environment [12] was proposed and present a classification system for detecting and preventing DDoS TCP flood attacks in public clouds. The proposed system provides a solution to secure stored records by classifying the incoming packets and taking a decision based on the classification result. But at the detection phase, the author relied on the number of requests connections from one source to the target server while the attacker can establish one connection and begin flooding the system as a part of the DDoS bots army also the attacker IP might be blacklisted but DDoS attacker can change his IP and deceive blacklist table. Analysis of ping-of-death attacks [13] shows how straightforward DDoS attack can effect on the behavior of the network - average response time, traffic received, traffic sent, upload and download response times- By analyzing single and multiple machine attacks, the great influence of the DOS and the DDoS attacks is observed. The simulation shows how DDoS attacks disrupt the normal operation of the network and how the attack can affect the productivity of the network. Proactive DDoS attack discovery and isolation [14] an early detection and mitigation technique was designed to mitigate against insider DDoS attacks. This technique detects inside attack among all authenticated normal clients present in the system at the proxy level and isolates it by converting its traffic to attack proxy. But the basic concept of a DDoS attack is to deny the users from accessing the service or the server so, if the attacker is capable of attacking or down the management server the users cannot access the application server i.e. attacker succeeded to deny the services even though he has not attacked the application server. DDoS detection and prevention relied on artificial intelligence techniques [15]. Because traffic of DDoS attack is similar to normal traffic some artificial intelligence techniques and algorithms like machine learning algorithms have been used to classify malicious traffic generated by DDoS and detect it, such as Naïve-Bayes and random forest tree, but Multi-machine algorithms can be combined to detect DDoS attacks. Blacklist-based malicious IP traffic detection methodology [16] for discovering any connection from a malicious IP address which is expected to be control and command server. This detection method is based on a blacklist technique. But spoofed IP could be used to deceive the Blacklist technique. Mitigation and detection algorithms of different types of DDoS attacks [17] presented an overview of detection and isolation algorithms to diminish the effect of four types of DDoS attacks: ping-of-death, UDP flood, TCP SYN flood, and smurf attack. The used algorithm for detection made deep analysis and check the incoming packets to differentiate which is normal and which is malicious. DDoS attacks at the application layer, challenges, and research [18] from point of view for protecting web applications discuss a detailed description of the application layer distributed denial of service and review the existing defense mechanisms to know different features used to detect these attacks. Detection of DDoS attack via deep packet analysis in real time systems [19], firstly packets are captured by listening to network traffic. Packet filtering was achieved at certain threshold. The sniffed packets are recorded to database to be analyzed and average values are compared by known DDoS attack patterns and will be determined if a DDoS attack attempts to attack the network in real time but if the database attacked or down for any reason the attack will be successfully done.

3. DDOS ATTACK METHODOLOGY

The DDoS attack may deplete resources of the target server by simply sending a huger volume of traffic than the victim is not able to handle (Flooding Attacks) i.e. overwhelm the target server by high volume.

This method is more difficult to mitigate, as malicious packets can be of any type of content and the high volume prevents traffic to be analyzed. An appropriate environmental internal network is created to simulate space ground control stations as shown in Fig. 1. Note that this network is isolated from being connected to the internet and from connecting any other subnets, so it is a very critical network because it is responsible for receiving telemetry from the spacecraft to analyze the health status of its subsystems then sending commands to control it. So, Target Server designed to be both layer 4 application server and webserver to simulate spacecraft and cortex module that receives telemetry from spacecraft then broadcast it to the network, connection-based attack technique was chosen to attack the simulated network actually 2 different types were chosen TCP flood and HTTP flood attacks,

The attacker was an insider, the attack script was deployed on internal computers bots to attack the target server i.e. certain client or user compromised all workstations in the network to be controlled by his machine, and at a certain time, he launched his DDoS attack script.

3.1 TCP Flood

After the Three-Way Handshake process was completed TCP packets were sent at a very high rate – this rate is varying according to the capability of the used server- and it seems to be normal packets at the beginning so, the target server was flooded with these packets to exhaust the server resources and consume bandwidth, it is a very fast attack that makes the server unavailable in a very short time, easy in implementation anyone can download open-source tool or design his script, but it is a very powerful technique. TCP flood was designed to attack layer 4 application running in the target server and make it unavailable as fas as possible it took (2) seconds to bring the service down.

3.2 HTTP Flood

It was designed to attack layer 7- application layer- the web server and make it unreachable as fast as possible so a high volume of HTTP get requests were sent to the target server that cannot handle these volumes of requests and leads to down the server and make it unavailable. It took (3) second to bring the server down. A TCP and HTTP flood attacks were carried out using our designed software as shown in Fig. 2. and successfully down the target server.

4. DDoS ATTACK ANALYSIS

The analysis was made to the simulated network to understand the behavior of it during normal operation and after overwhelming the network with the malicious traffic.

Wireshark network analyzer and the designed software were used to capture and analyze the captured traffic before and during the attack.

4.1 Analysis of TCP Flood

Before the TCP flood attack, all applications are running and telemetry from the satellite model is available during the session, and ensure that the network is up and work normally use the ping command to the server IP (10.10.33.10) and by using Wireshark and our program we captured TCP packets and the captured data was normal according to server behavior and statistics from sniffing programs.

When the TCP traffic was normal, statistics show the following: captured packets displayed in Fig 3 shows the total number of TCP packets in approximately 60 sec. And TCP packets percentage from Total captured packets is displayed in Fig. 4. I-O Graph is normal as shown in Fig. 5. Note that the number of TCP packets approximately was less than 8 packets per 1 msec.

During a TCP flood attack: TCP flood attack was deployed using our designed software which

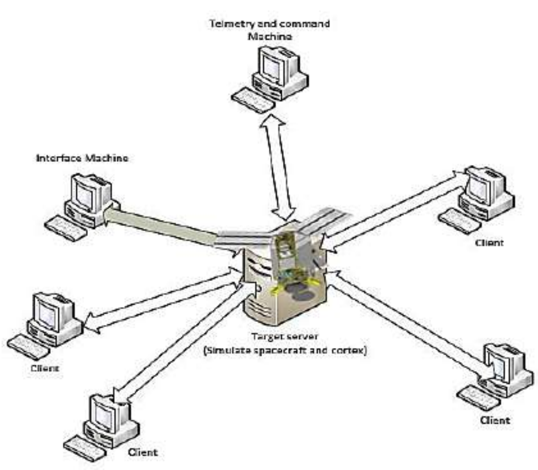


Fig 1. Simulated Network

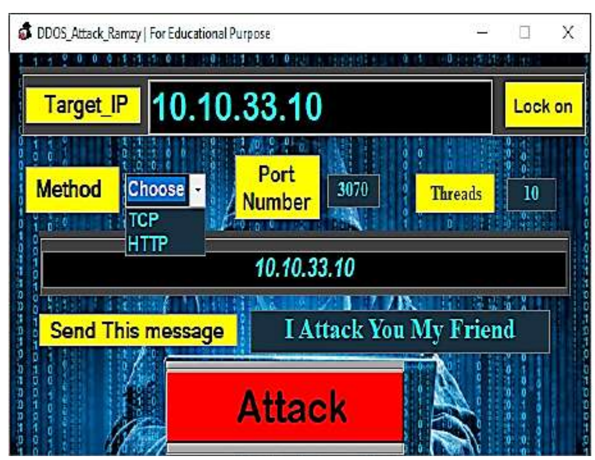


Fig 2. DDoS Attack Software

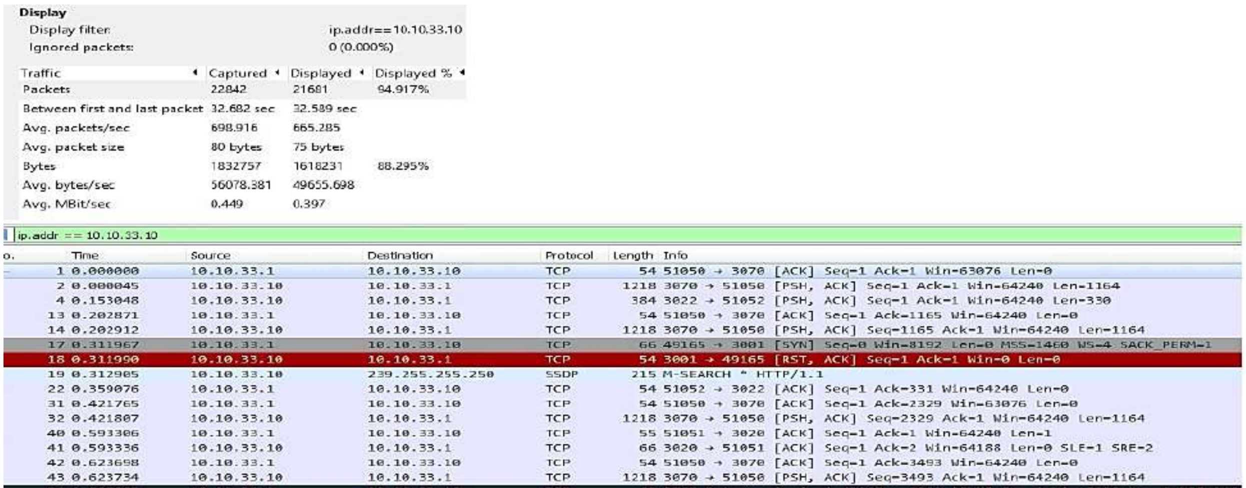


Fig 3. Captured Normal TCP Packets

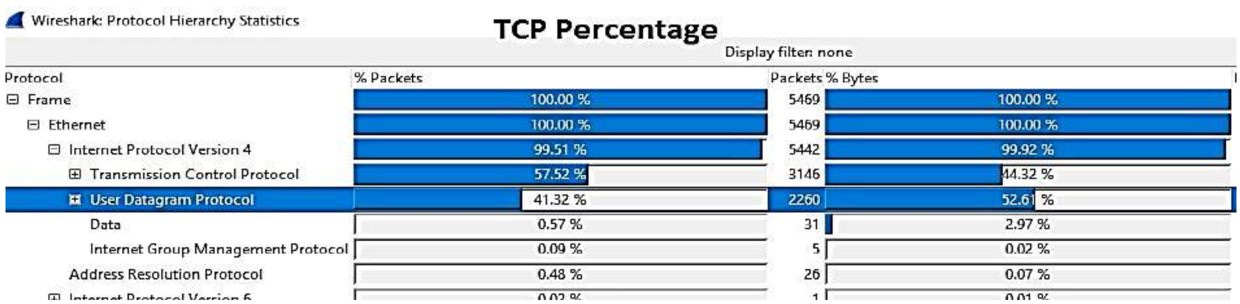


Fig 4. Normal TCP Packets Percentage

performed a DDoS TCP flood attack on a target server (IP: 10.10.33.10). When the attack started on the target server it could not respond to all the requesting normal or malicious packets and the application on the target server didn't respond and telemetry was off during the session.

Statistics show the following: (Fig. 6) shows captured packets, a total number of TCP packets, and percentage displayed in Fig. 7. very high TCP packet rate W.R.T normal one, I-O Graph is shown in Fig. 8. Note that the number of TCP packets approximately 100 packets per 1 msec.

DDoS TCP flood attack effected on the target server within a short time (2 sec), slowing down the response, and then stop the service completely. So, efficient and effective detection and isolation/ mitigation technique are required.

4.2 Analysis of HTTP Flood

Before HTTP Attack, as mentioned before the target server (10.10.33.10) also is a spacecraft simulator web server and clients or users can connect to display telemetry from this web server as shown in Fig. 9, By using Wireshark and our program we captured HTTP packets as shown in Fig. 10. and the captured data was normal according to server behavior-telemetry available- and statistics from sniffing programs. When the HTTP traffic was normal, statistics show the following: Total number of HTTP packets and percentage displayed in (Fig. 11). I-O Graph is normal as shown in Fig. 12. Note that the number of HTTP packets approximately was less than 100 packets per sec. During HTTP Attack: When the target Server attacked by DDoS HTTP flood, normal clients try to host the webserver but message 503 services unavailable were issued

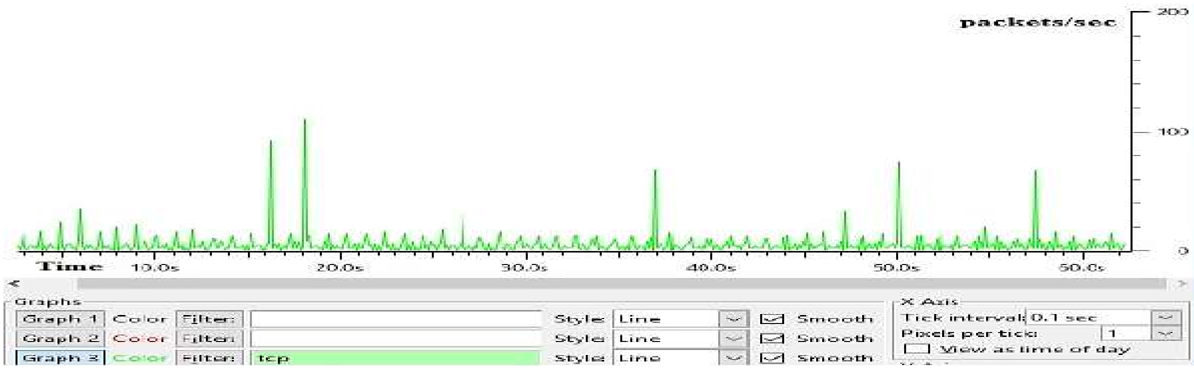
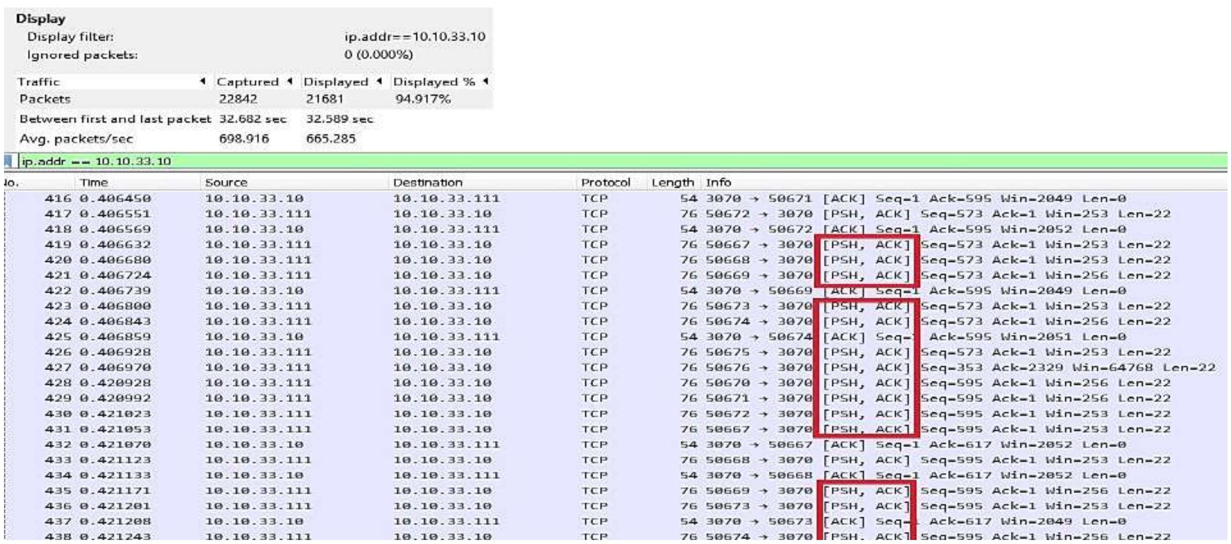


Fig 5. I-O TCP Normal Graph



Display

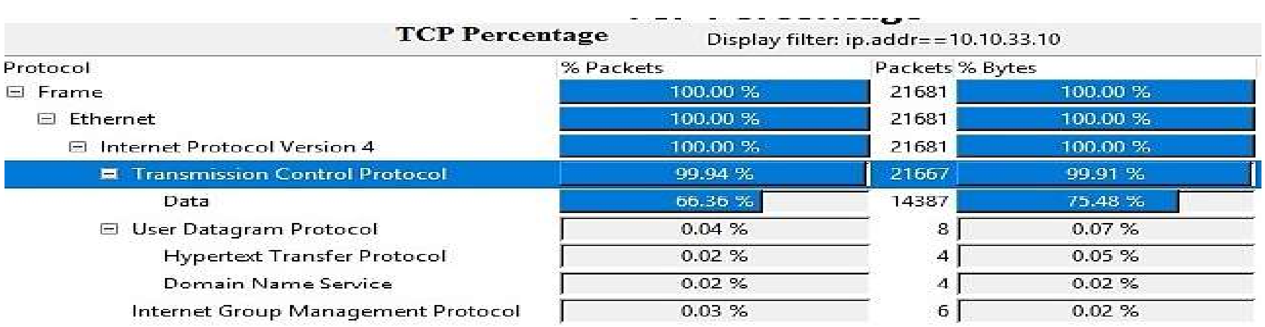
Display filter:	ip.addr==10.10.33.10
Ignored packets:	0 (0.000%)

Traffic	Captured	Displayed	Displayed %
Packets	22842	21681	94.917%
Between first and last packet	32.682 sec	32.589 sec	
Avg. packets/sec	698.916	665.285	

ip.addr == 10.10.33.10

No.	Time	Source	Destination	Protocol	Length	Info
416	0.406450	10.10.33.10	10.10.33.111	TCP	54	3070 → 50671 [ACK] Seq=1 Ack=595 Win=2049 Len=0
417	0.406551	10.10.33.111	10.10.33.10	TCP	76	50672 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=253 Len=22
418	0.406569	10.10.33.10	10.10.33.111	TCP	54	3070 → 50672 [ACK] Seq=1 Ack=595 Win=2052 Len=0
419	0.406632	10.10.33.111	10.10.33.10	TCP	76	50667 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=253 Len=22
420	0.406680	10.10.33.111	10.10.33.10	TCP	76	50668 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=253 Len=22
421	0.406724	10.10.33.111	10.10.33.10	TCP	76	50669 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=256 Len=22
422	0.406739	10.10.33.10	10.10.33.111	TCP	54	3070 → 50669 [ACK] Seq=1 Ack=595 Win=2049 Len=0
423	0.406800	10.10.33.111	10.10.33.10	TCP	76	50673 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=253 Len=22
424	0.406843	10.10.33.111	10.10.33.10	TCP	76	50674 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=256 Len=22
425	0.406859	10.10.33.10	10.10.33.111	TCP	54	3070 → 50674 [ACK] Seq=1 Ack=595 Win=2051 Len=0
426	0.406928	10.10.33.111	10.10.33.10	TCP	76	50675 → 3070 [PSH, ACK] Seq=573 Ack=1 Win=253 Len=22
427	0.406970	10.10.33.111	10.10.33.10	TCP	76	50676 → 3070 [PSH, ACK] Seq=353 Ack=2329 Win=64768 Len=22
428	0.420928	10.10.33.111	10.10.33.10	TCP	76	50670 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=256 Len=22
429	0.420992	10.10.33.111	10.10.33.10	TCP	76	50671 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=256 Len=22
430	0.421023	10.10.33.111	10.10.33.10	TCP	76	50672 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=253 Len=22
431	0.421053	10.10.33.111	10.10.33.10	TCP	76	50667 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=253 Len=22
432	0.421070	10.10.33.10	10.10.33.111	TCP	54	3070 → 50667 [ACK] Seq=1 Ack=617 Win=2052 Len=0
433	0.421123	10.10.33.111	10.10.33.10	TCP	76	50668 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=253 Len=22
434	0.421133	10.10.33.10	10.10.33.111	TCP	54	3070 → 50668 [ACK] Seq=1 Ack=617 Win=2052 Len=0
435	0.421171	10.10.33.111	10.10.33.10	TCP	76	50669 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=256 Len=22
436	0.421201	10.10.33.111	10.10.33.10	TCP	76	50673 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=253 Len=22
437	0.421208	10.10.33.10	10.10.33.111	TCP	54	3070 → 50673 [ACK] Seq=1 Ack=617 Win=2049 Len=0
438	0.421243	10.10.33.111	10.10.33.10	TCP	76	50674 → 3070 [PSH, ACK] Seq=595 Ack=1 Win=256 Len=22

Fig 6. Captured Malicious TCP Packets



TCP Percentage Display filter: ip.addr==10.10.33.10

Protocol	% Packets	Packets	% Bytes
Frame	100.00 %	21681	100.00 %
Ethernet	100.00 %	21681	100.00 %
Internet Protocol Version 4	100.00 %	21681	100.00 %
Transmission Control Protocol	99.94 %	21667	99.91 %
Data	66.36 %	14387	75.48 %
User Datagram Protocol	0.04 %	8	0.07 %
Hypertext Transfer Protocol	0.02 %	4	0.05 %
Domain Name Service	0.02 %	4	0.02 %
Internet Group Management Protocol	0.03 %	6	0.02 %

Fig 7. Malicious TCP Packets Percentage

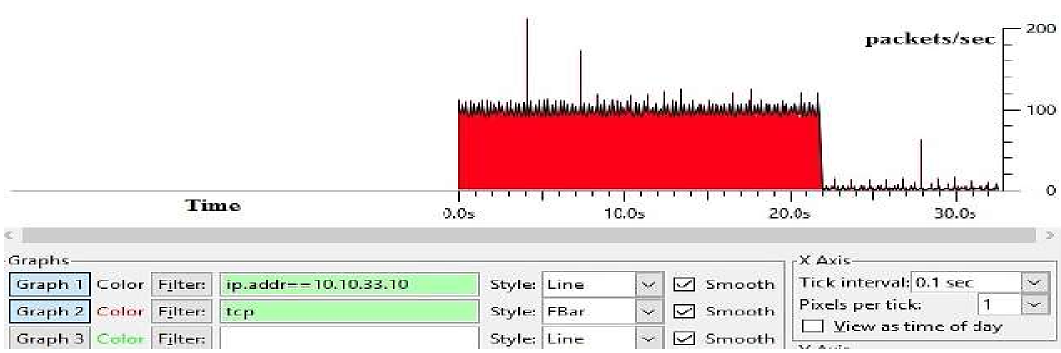


Fig 8. I-O TCP malicious graph

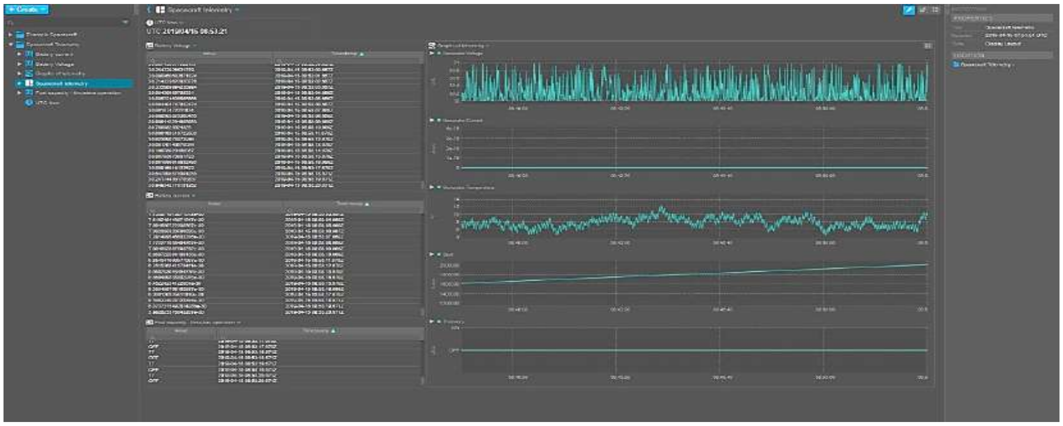


Fig 9. Spacecraft Telemetry

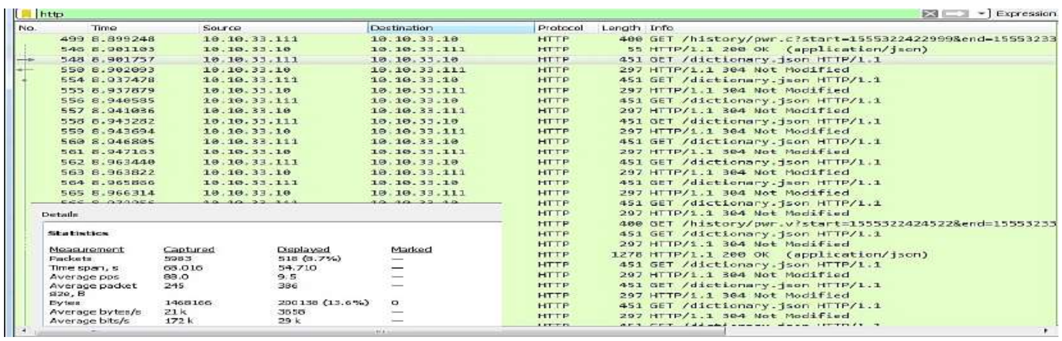


Fig 10. Captured Normal HTTP Packets

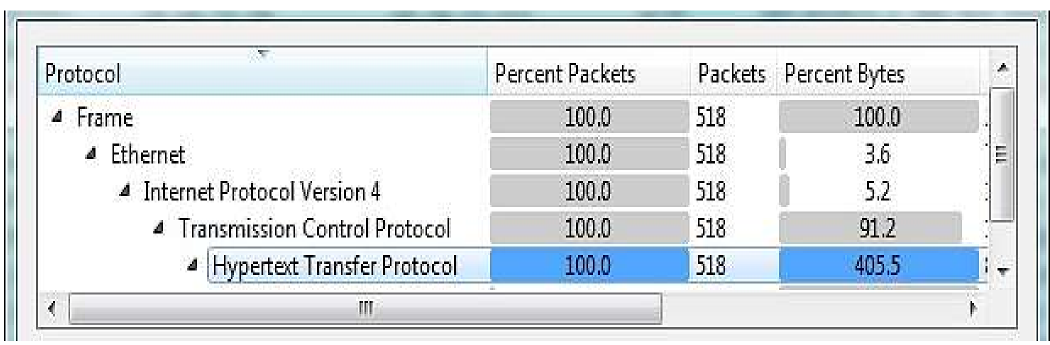


Fig 11. Normal HTTP Packet Percentage

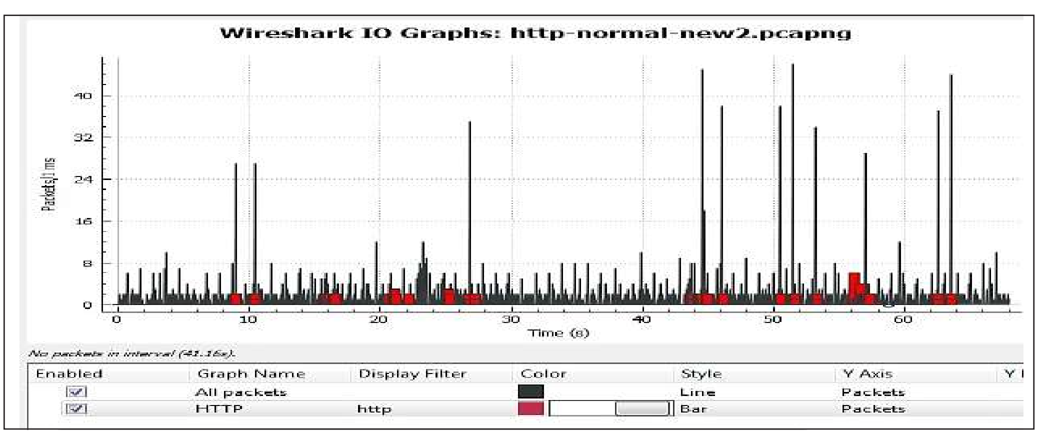
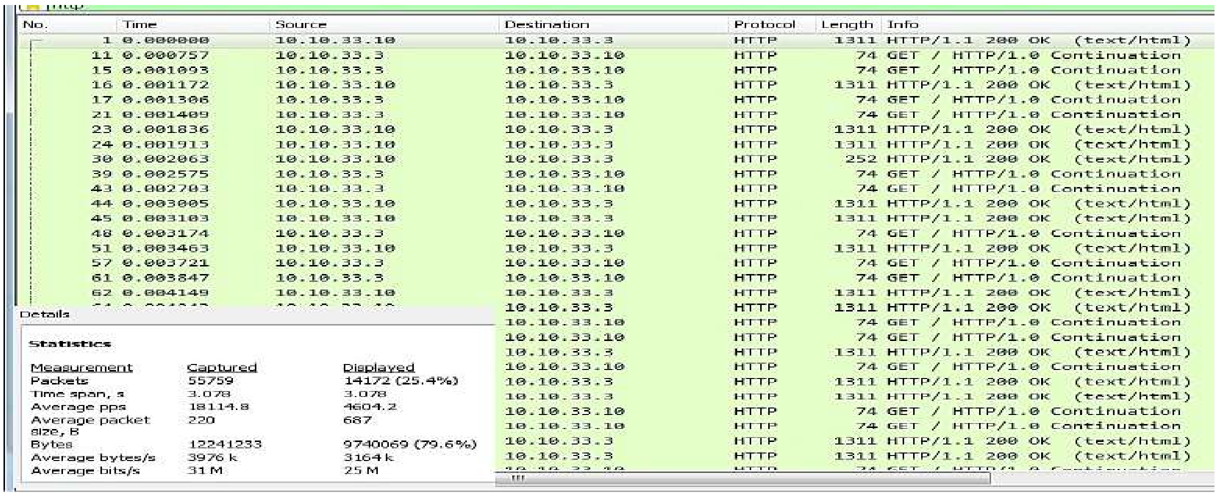


Fig 12. I-O HTTP Normal Graph

by the server and the service is completely down, captured HTTP packets as shown in Fig. 13. The total number of HTTP packets and percentage displayed in (Fig. 14). I-O Graph as shown in Fig.15. Note that the number of HTTP packets approximately was less than 4500 packet per sec.



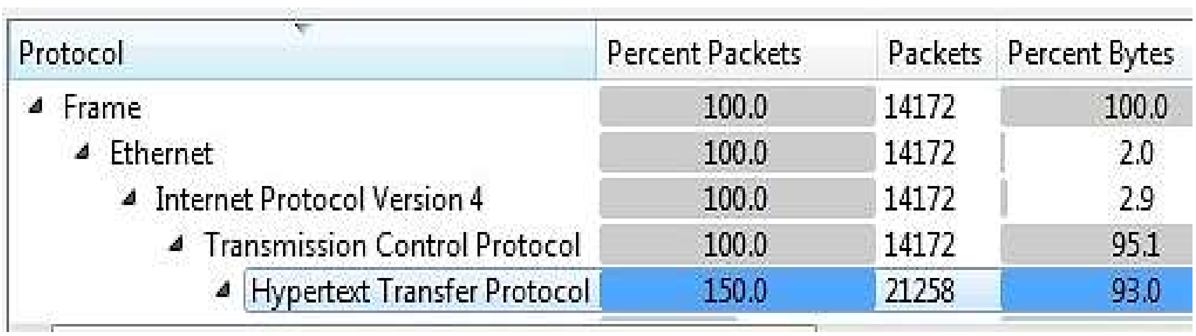
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
11	0.000757	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
15	0.001093	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
16	0.001172	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
17	0.001306	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
21	0.001409	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
23	0.001836	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
24	0.001913	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
30	0.002063	10.10.33.10	10.10.33.3	HTTP	252	HTTP/1.1 200 OK (text/html)
39	0.002575	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
43	0.002703	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
44	0.003005	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
45	0.003103	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
48	0.003174	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
51	0.003463	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
57	0.003721	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
61	0.003847	10.10.33.3	10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
62	0.004149	10.10.33.10	10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
			10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
			10.10.33.10	HTTP	74	GET / HTTP/1.0 Continuation
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)
			10.10.33.3	HTTP	1311	HTTP/1.1 200 OK (text/html)

Details

Statistics

Measurement	Captured	Displayed
Packets	55759	14172 (25.4%)
Time span, s	3.078	3.078
Average pps	18114.8	4604.2
Average packet size, B	220	687
Bytes	12241233	9740069 (79.6%)
Average bytes/s	3976 k	3164 k
Average bits/s	31 M	25 M

Fig 13. Captured Malicious HTTP Packets



Protocol	Percent Packets	Packets	Percent Bytes
Frame	100.0	14172	100.0
Ethernet	100.0	14172	2.0
Internet Protocol Version 4	100.0	14172	2.9
Transmission Control Protocol	100.0	14172	95.1
Hypertext Transfer Protocol	150.0	21258	93.0

Fig 14. Malicious HTTP Packets Percentage

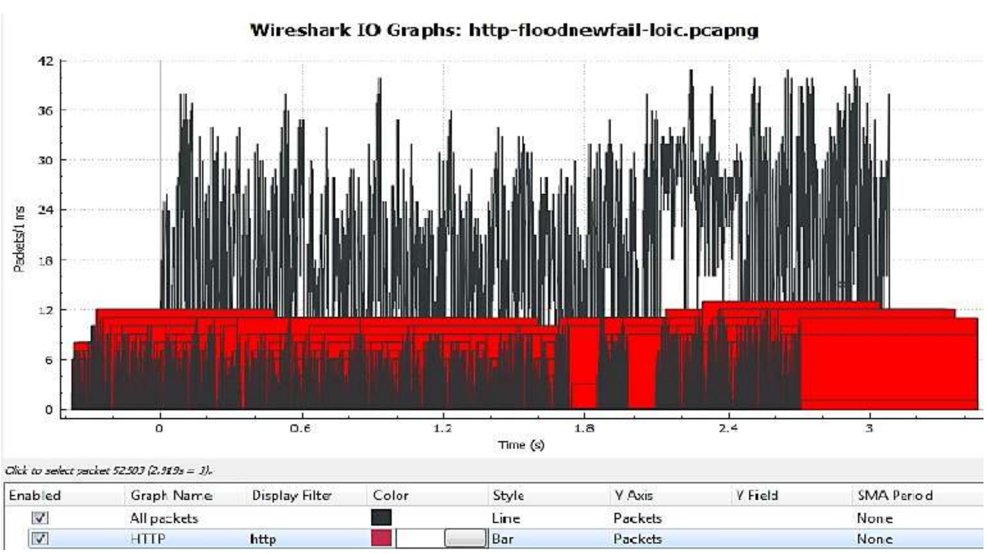


Fig 15. I-O HTTP Malicious Graph

5. DDoS ATTACK DETECTION & MITIGATION

TCP protocol uses several flags to manage connection establishment status and data transfer so in our proposed analysis and detection algorithm we focus on 3 flags:

- ➢ SYN: is initially sent when establishing a three-way handshake i.e. responsible for the initiation of the connection
- ➢ ACK: used to acknowledge the successful receipt of packets. Every packet you send or receive is followed by ACK.
- ➢ PSH: this flag is used to ensure that the data is given the priority it deserves and is processed at the sending or receiving end. When connecting with a server, the client can ask for confirmation that the information was received by setting the ACK flag, or it can force the server to process the information in the packet by setting the PUSH flag.

In the analysis phase, we notice that before the DDoS attack started PSH & ACK flags set to be 1 while transferring data between the target server and the clients with a low rate while after TCP and HTTP flood this combination was sent with abnormal very high rate. So, we have 2 proposed detection Algorithm:

Algorithm1 can be used to detect TCP flood – packets are normal or malicious- by counting the number of PSH & ACK flags (PSH=1& ACK=1) if this counting exceeds the predefined threshold within a certain time which can be adjusted by the security administrator. C# program was used to implement these algorithms so; the attack will be detected as shown in Fig 16.

Also, HTTP flood attack could be detected using algorithm1 and by using Algorithm2 which depends on counting get requests from certain IP address and if the counter exceeds the predefined threshold within certain time attack will be detected as well this algorithm use another counter to count several Three-Way-Handshake processes from a certain IP address if exceed the counter-attack detected.

The output from the detection phase is fed to a blocking algorithm to block the traffic from the attacker IP as shown in DDoS detection and mitigation flowchart in Fig 17. So, we already have the source IP pool of all attacker machines which participated in the attack. And by using the same designed c# program firewall class was added to create rules which block the list of attackers IP also as shown in Fig 16.

➢ <u>Algorithm 1</u>

1- Online capture packet
2- X=0
3- For (i=1: N)
4- SIP(i) = source_ip
5- Insert SIP(i) into the list
6- For (j=1:M)
7- If (time<Z& & tcp.flg.ack==1& & tcp.flag.psh==1)
8- X ++
9- If X>T
10- Flood Attack detected

Where,

- o N: Number of packets.
- o SIP: Source IP address from network header.
- o X: Counter increased when PSH &ACK flags =1.
- o T: Predefined threshold for the packets to be Considered a DDoS attack.
- o M: Total number of source IP in the list.
- o Z: Threshold Time.

➢ <u>Algorithm 2</u>

1- Online capture packet
2- X=0
3- For (i=1: N)
4- SIP(i) = source_ip
5- Insert SIP(i) into list
6- For (j=1:M)
7- If (time < Z && tcp.flg.syn==1)
8- X ++
9- If X>T
10- Flood Attack detected
11- If (time<Z && http.req== "Get")
12- Y ++
13- If Y>R
14- Flood Attack detected

Where,

- o N: Number of packets

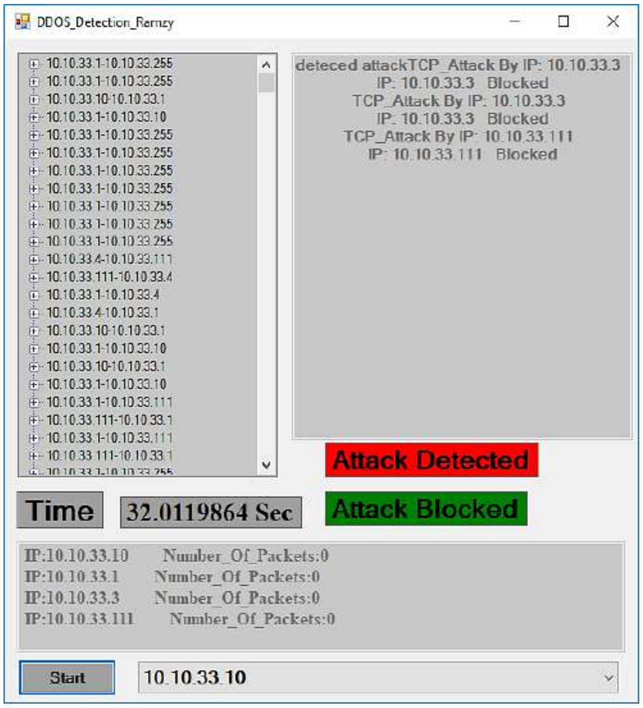


Fig 16. Detection and Blocking of DDoS

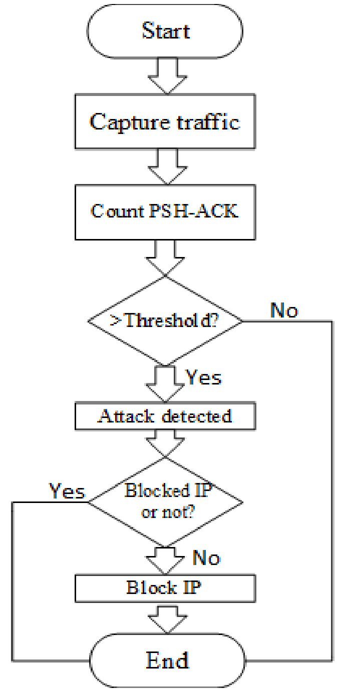


Fig 17. DDoS Detection and Mitigation Flowchart

- o SIP: Source IP address from network header
- o X: Counter increased when SYN flags =1
- o T, R: Predefined threshold for the packets to be Considered a DDoS attack.
- o M: Total number of source IP in the list.
- o Z: Threshold Time
- o Y: Counter increased when Get =1

6. CONCLUSIONS

The usage of DDoS attack detection and mitigation techniques became essential to detect insider attackers and, it is very important to use these techniques in the critical networks like ground control station networks that control the satellite. A new technique was proposed to analyze and detect TCP and HTTP flood DDoS attacks for the simulated space ground network during receiving telemetry from simulated spacecraft. Our simulation shows that insider attack is detected after the attacker launched his attack script within 3 secs and blocked within 2 secs after the detection then the results of the detection algorithms are fed to the mitigation algorithm to block the attacker IP.

In the future, we aim to design a new technique to stop or mitigate TCP and HTTP flood completely and the normal clients can receive their services normally.

7. REFERENCES

1. N. Yasmin, and M.U.D. Junjua, Some Derivative-1. A.Girma, K. Abayomi, and M. Garuba, The Design, Data Flow Architecture, and Methodologies for a Newly Researched Comprehensive Hybrid Model for the Detection of DDoS Attacks on Cloud Computing Environment, Information Technology: New Generations, *Springer*, 377-387 (2016).
2. D. Gillman, Y. Lin, B. Maggs, and R.K.Sitaraman, Protecting websites from attack with secure delivery networks, *IEEE*. 48: 26-34 (2015).
3. N. Hoque, D. K. Bhattacharyya, J.K.Kalita. Botnet in DDoS attacks: trends and challenges, *IEEE Communications Surveys & Tutorials* 17: 2242-2270 (2015).
4. C. Wang., T.T.N. Miu., X. Luo, J. Wang. SkyShield: A sketch-based defense system against application-layer DDoS attacks, IEEE Transactions on Information Forensics and Security. 13(3)559-573 (2018).
5. S. Acharya, and N.Tiwari, Survey of DDoS attacks based on TCP/IP protocol vulnerabilities, *IOSR Journal of Computer Engineering (IOSR-JCE)*

18(3) 68-76 (2016).

6. C.Rocky and Chang, Defending against flooding-based distributed denial-of-service attacks: a tutorial, *IEEE communications magazine*, 42-51, 40 (2002).
7. P. Kamboj, M. C. Trivedi, V. K. Yadav, and V. K. Singh, Detection techniques of DDoS attacks: A survey, *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), IEEE,* pp. 675-679 (2017).
8. A. Yusof, A. Riza'ain, N.L Udzir, and A. Selamat, Systematic literature review and taxonomy for DDoS attack detection and prediction, *International Journal of Digital Enterprise Technology,* pp. 292-315, 1 (2019).
9. Y-J. Lee, N-K. Baik, C. Kim, and C-N. Yang, Study of detection method for spoofed IP against DDoS attacks, *Personal and Ubiquitous Computing,* pp. 35-44, 22 (2018).
10. S. T. Zargar., J. Joshi, and D. Tipper. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks, *IEEE Communications Surveys & Tutorials.* 15: 2046-2069 (2013).
11. A. Stavrou., D. Fleck, and C. Kolias, On the Move: Evading Distributed Denial-of-Service Attacks, *IEEE Annals of the History of Computing,* 49: 104-107 (2016).
12. A. SahI, D. Lai, Y. Li and M. Diykh, An efficient DDoS TCP flood attack detection and prevention system in a cloud environment, *IEEE Access 5-2017*, pp. 6036-6048, 5 (2017).
13. F. Yihunie, E. Abdelfattah, and A. Odeh, Analysis of ping of death DoS and DDoS attacks, 2*018 IEEE Long Island Systems, Applications, and Technology Conference (LISAT), IEEE,* pp. 1-4 (2018).
14. V. Kansal and M. Dave, Proactive DDoS attack detection and isolation, *2017 International Conference on Computer, Communications, and Electronics (Comptelix), IEEE,* pp. 334-338 (2017).
15. B. Zhang, T. Zhang and Z. Yu, DDoS detection and prevention based on artificial intelligence techniques, *2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE,* pp. 1276-1280 (2017).
16. I. Ghafir and V. Prenosil, Blacklist-based malicious IP traffic detection, *2015 Global Conference on Communication Technologies (GCCT), IEEE,* pp. 229-233 (2015).
17. M. Yusof, M.A. Mohd, M.Y. Draus and F.Ali, Detection and defense algorithms of different types of DDoS attacks, *International Journal of Engineering and Technology 9,* pp. 410, 9 (2017).
18. A. Praseed, Amet, and P.Thilagam, DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications, *IEEE Communications Surveys & Tutorials 21,* pp. 661-685, 21 (2019).
19. E. Özer and M. İskefiyeli, Detection of DDoS attack via deep packet analysis in real-time systems, *2017 International Conference on Computer Science and Engineering (UBMK), IEEE,* pp. 1137-1140 (2017).