# A Convenient way to mitigate DDoS TCP SYN Flood attack

Mahmud Hasan and Habibur Rahman

A Thesis in the Partial Fulfillment of the Requirements

for the Award of Bachelor of Computer Science and Engineering (BCSE)

Department of Computer Science and Engineering
College of Engineering and Technology
IUBAT – International University of Business Agriculture and Technology

Fall 2021

# A Convenient way to mitigate DDoS TCP SYN Flood Attack

Mahmud Hasan and Habibur Rahman

A Thesis in the Partial Fulfillment of the Requirements for the Award of Bachelor of Computer Science and Engineering (BCSE)

The thesis has been examined and approved,

________________________________

Prof. Dr. Utpal Kanti Das
Chairman

________________________________

Dr. Hasibur Rashid Chayon
Co-supervisor, Coordinator and Associate Professor

________________________________

Toyeer-E-Ferdoush
Supervisor and Senior Lecturer

Department of Computer Science and Engineering
College of Engineering and Technology
IUBAT – International University of Business Agriculture and Technology

Fall 2021

# Letter of Transmittal

3 September 2021
The Chair
Thesis Defense Committee
Department of Computer Science and Engineering
IUBAT–International University of Business Agriculture and Technology
4 Embankment Drive Road, Sector 10, Uttara Model Town
Dhaka 1230, Bangladesh

**Subject:** Letter of Transmittal.

Dear Sir,

With all due respect, I would like to let you know that it gives us great joy to present this report, "A Convenient technique to mitigate DDoS TCP SYN Flood attack," in order to finish our thesis course.

It was a great pleasure to work on this research topic. It helped us to make our theoretical knowledge more realistic. I'm now waiting for your thoughtful comments on this report.

We would appreciate it greatly if you would review this report and evaluate our performance.

Yours sincerely,

| | |
|---|---|
| Mahmud Hasan | Habibur Rahman |
| 19103111 | 19103119 |

# Student’s Declaration

The following report, titled "A Convenient way to mitigate DDoS TCP SYN Flood attack," has been created as part of the Bachelor of Computer Science and Engineering degree's thesis course by Mahmud Hasan and Habibur Rahman, students in the program at the International University of Business, Agriculture, and Technology (IUBATCollege )'s of Engineering (CEAT).

This report entitled “A Convenient way to mitigate DDoS TCP SYN Flood attack” was edited solely by us. All the framework has been proposed based on our testing, academic papers and online resources.

This research was accepted and presented on SUMCOM-2022 International Conference. The paper ID for the conference was ‘C2022096’.

Mahmud Hasan Habibur Rahman

19103111 19103119

## Supervisor’s Certification

This is to make sure that Mahmud Hasan, ID 19103111, and Habibur Rahman, ID 19103119, of IUBAT - International University of Business Agriculture and Technology, compile the thesis report on "A Convenient technique to mitigate DDoS TCP SYN Flood attack" as part of a successful thesis defense course. The report, which was created under my direction, serves as a record of the successfully performed task.

You are now allowed to submit a report. I wish your success in your every future endeavors.

Toyeer-E-Ferdoush
Senior Lecturer
Department of Computer Science and Engineering
IUBAT–International University of Business Agriculture and Technology

# Abstract

Sharing information from one device to another replaces hand-to-hand communication in this connected digital era. Modern technology is used to control data communication. Because of this, the pace of a device's cyber security is increasing rapidly. DDoS (Distributed Denial-of-Service) is one such threat that every startup business and major companies are concerned about, as it is not possible to predetermine such activities. The DDoS attack uses various protocols; TCP is one of them. In the TCP (Transmission Control Protocol), half-open states cause SYN(Synchronization) flood attacks. It is a distributed denial of service attack that seeks to block all valid communication to a server to access available server resources. This paper aims to protect the server from DDoS TCP SYN flood attacks. By detecting the attack before reaching the server, this model bypasses the connection through a proxy server with a newly designed firewall and verifies the connection. The firewall allows the users to access the server & the flood connections are dropped and do not reach the server. There are many research papers which can detect the attack after the attack occur, and the prevention percentage is low. In this research paper, this attack can be prevented much well than other models because a flood attack can detect before it affects the server and denies the valid connection attempt. Two cases including one of the future issues will be solved here: 1. SYN-ACK Lost and 2. SYN-ACK No Response.

# Acknowledgments

Many thanks to our supervisor, Toyeer-E-Ferdoush, for your tolerance, advice, and assistance. We have gotten a lot out of your extensive expertise and careful advice. We are incredibly appreciative that you accepted us as students and kept believing in us throughout the year.

Thank you to Prof. Dr. Utpal Kanti Das and Dr. Hasibur Rashid Chayon for encouraging words, thoughtful suggestions and detailed feedback. It has been very important for us.

Thank you to my thesis partner for putting up with my continual ranting, talking things out, proofreading repeatedly, cracking laughs when things became too serious, and making sacrifices so that we could finish this thesis.

# Table of Contents

# List of Figures

# List of Tables

# Chapter I. Introduction

Distributed Denial of Services is written as DDoS. This is considered as a form of Cyber Attack. In this attack botnets are used with spoofed IP addresses. Previously Denial of Service (DoS) was considered threat until DDoS was discovered.

DDoS attack is solely based on the connection from client to server. The connection might Use the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP) for communication.As the TCP can recognize and resend the lost package during communication and control the overflow of the segment so in almost every communication establishment TCP is applied.

TCP communication consists of 3 flag signals known as SYN, SYN-ACK & ACK. The initial state of TCP connection is SYN (Synchronization) flag. After the recipient receive the flag, The SYN and ACK bits are set on the TCP packet the recipient sends back (which identifies that it is a SYN packet and also that it is acknowledging the previous SYN packet). The sender device then sends an ACK in response to the SYN-ACK.

The system to create a flag with SYN flag is continuously sending the SYN flag to the receiver device and “not receiving” the SYN-ACK signal from counter-side device.

# Chapter II. Literature Review

The Internet Protocol Journal, 9(4), pp. 2–16, "Defenses against TCP SYN flooding attacks," by W.M. Eddy (2006). TCP SYN Flooding, a sort of Denial of Service (DoS) attack, is covered in this suggested paradigm. The attack takes advantage of a peculiarity of the TCP implementation and can be used to prevent server processes from responding to requests for new TCP connections from valid client applications. Any service that connects to and listens on a TCP socket is susceptible to TCP SYN flooding attacks. Practical network engineering is essential to preventing these assaults since it involves well-known server software like e-mail, the Web, and file storage services. Variations of the attack, which has been publicly covered for more than ten years, are still occurring. Although there are practical ways to handle SYN flooding, there is still no unified standard for TCP implementations. There are numerous solutions available with current operating systems and hardware, but each has its own effects on the applications and networks that are being attacked. This article explains the attack and why it succeeds before giving a summary and evaluation of the current strategies employed to combat it. SYN flooding attacks on both end hosts and network devices.

(Liu, P.E. and Sheng, Z.H., 2008, July. Defending Against TCP SYN Flooding with a new kind of SYN-Agent. In 2008 International Conference on Machine Learning and Cybernetics (Vol. 2, pp. 1218-1221). IEEE) In this paper, The agent shakes hands with the client instead of the server first, like in the enhanced syn-agent model. Once a complete connection has been established, the syn-agent notifies the server to connect directly to the client with the specified SEQ. Thus, for the actual server, the three-step TCP handshake is reduced to just one.

To cut down on the cost of communication between the server and syn-agent, the remark is included in the third-time handshake packet from the client by changing the TCP header reserved bit to "1". When the TCP header reserved bit was set to "1" in a packet, the server instantly established a connection with the state of ESTABLISHED. The strain on the server and syn-agent can both be greatly reduced by doing away with the requirement for the server to save any connection information for connections that are only partially open. End hosts and network administrators have observed this SYN flooding attack in action for a considerable time, and the community is well aware of it. To reduce the effectiveness of SYN-flooding numerous strategies have been developed and applied. Despite the attack's notoriety and the availability of defenses, the RFC series did not provide any recommendations for TCP implementations, instead describing the vulnerability as an example of the necessity for ingress filtering. There are no standards outlined in this document, despite the fact that it addresses both problems. The formal requirements for defense mechanisms are not covered here. Many defenses only impact the implementation of an end host's interoperability.

This paper deliberately concentrates on SYN flooding assaults from the viewpoint of a single to prevent service to that specific object from being provided by the end host or application. Operationally, high-rate attacks that target the capability of the network and capabilities for packet processing have been seen. Whether or not such attacks use TCP SYN segments, they are outside the scope of this paper since they target the system rather than a

TCP implementation, and the kind of packets utilized is unimportant compared to the packet rate in such assaults. (TCP SYN flooding attacks and typical mitigations (Eddy, W., 2007) (No. rfc4987).

The authors of this study introduced an SRL module to prevent one of the most common TCP SYN Flood DDoS attacks in an SDN system. SRL does not involve a temporary TCP handshake, in contrast to SLICOTS and OPERETTA. Instead, SRL sets up permanent forwarding for the requested connection, and after the handshake, SRL adds that user to the Whitelist. Additionally, SRL offers the user an opportunity if there is a problem with connection establishment. This method also identifies malicious users who initiate an attack after a full TCP handshake. Utilizing a hashing module to replace the flow rules in the flow table based on the priority of the hash value is another benefit SRL offers. Additionally, it makes use of flow aggregator to stop the nefarious connection requests. By restricting TCP connection requests, SRL additionally aids in the detection of slow-moving DDoS attacks. We are experimenting with this module in a heterogeneous context, such as a delocalized controller, as part of future work. Additionally, we intend to upgrade this technology to better protect SDN from additional disruptive forms of attacks.

The concept of a "Programmable Network," which is new thanks to Software Defined Networking (SDN), offers flexibility, simplicity, and quick implementation. The core idea behind SDN architecture is the division of the control plane from the data plane. The controller is given a complete perspective of the network by separating these planes, and the decision-making for packet forwarding. As a result, the controller abstracts away the network's complexity. Unfortunately, TCP SYN Flood Attacks, one of the most common type of DDoS attacks, target SDN because of this functionality.

TCP SYN Flood attacks is frequently used to deplete the server's resources. This SDN attack compelling the switch to transmit the packet to the controller and exhausting all of the controller's resources turns the controller into a single point of failure. Another result of it is a saturation attack on the data plane. In this study, we suggest SRL as a competent and practical framework for guarding against TCP SYN Flood attacks. in order to overcome this challenge. The controller has put SRL into effect. It uses the flow aggregator module and the hashing module as a defense against this attack. For the Floodlight Controller, we implemented SRL in a number of attack and typical traffic scenarios. According to the findings, SRL barely affects how SDN controllers operate. (SRL: A TCP SYNFLOOD DDoS mitigation technique in software-defined networks, Ubale, T. and Jain, A.K., March 2018. 2018 saw the second iteration of ICECA, the international conference on electronics, communications, and aerospace technology (pp. 956-962). IEEE.)

(Lemon, J., 2002. Resisting {SYN} Flood {DoS} Attacks with a {SYN} Cache. In 2002's BSDCon (BSDCon 2002) In this study, workstations that provide TCP services are frequently the targets of various types of DDoS attacks from external network hosts. An attempt by external hosts to overwhelm the server machine by delivering a continuous stream of TCP connection requests results in a "SYN flood" attack, which forces the server to allocate resources for each new connection until all resources are used up. The fatigue issue and numerous remedies, including SYN caches and SYN cookies, are covered in this study. The benefits and drawbacks of each approach are reviewed, and the chosen solution's precise implementation in FreeBSD is examined.

(Rahouti, M., Xiong, K., Ghani, N. and Shaikh, F., 2021. SYNGuard: Dynamic detection and mitigation of SYN flood attacks using thresholds in software-defined networks. 76–87, IET Networks, 10(2). , SYN flood attacks (half-open attacks), which have been explored in this work, have been shown to pose a serious risk to SDN-enabled infrastructures. To identify and stop these security risks, a number of intrusion detection and prevention systems (IDPS) have been developed, however they usually incur high performance overhead and reaction time. Because of this, current strategies for massive networks and real-time applications are rigid. As a result, we suggest a cutting-edge, flexible, threshold-based SDN-based intrusion detection and prevention system.

The proposed systems for detecting and mitigating the threats within an SDN is compared to Snort and Zeek, two widely used traditional IDPS technologies. The approach is evaluated on a real-world testbed using a combination of fundamental adverse attacks and SDN-specific threats. The results of the experiments show that the mechanism can detect and mitigate SYN flood attacks in an SDN environment.

(Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A. and Abduallah, W.M., 2019. Comprehensive review of artificial intelligence and statistical approaches in attacks using distributed denial of service and countermeasures. (51691–51713) IEEE Access, 7, p. In this study, security systems are examined for their lack of quick-acting protocols, operational procedures, or countermeasureorts. The bulk of DDoS defense techniques that have been suggested have various flaws and restrictions. Some of these techniques use defensive mechanisms based on signatures that are unable to identify fresh attacks, while others use defense mechanisms based on anomalies that are only applicable to certain kinds of DDoS attacks and have not yet been tested in open environments.

In order to recognize, mitigate, and avoid these attacks, a great deal of research has been done on the use of statistical and artificial intelligence techniques in protection strategies. The best and most efficient defense mechanisms, strategies, and approaches against such attacks are still unknown, nevertheless.

This review study focuses on the most popular statistical and artificial intelligence-based DDoS defense strategies. The review further classifies and exemplifies the attack kinds, testing parameters, evaluation techniques, and testing datasets used in the methodology of the suggested defense measures. The review concludes by offering a framework and potential areas of convergence for creating improved DDoS protection solution models.

(A.K. Soliman, C. Salama, and H.K. Mohamed, December 2018. identifying a DDoS assault that uses DNS reflection and amplification that comes from the cloud. The 13th International Conference on Computer Engineering and Systems (ICCES) will take place in 2018. (pp. 145-150). IEEE. ), this article examines The cloud is currently being adopted by companies of all sizes since it is viewed as a crucial business enabler thanks to features like reduced go-live times and associated resources. Concerns about availability, integrity, and secrecy are raised by moving to the cloud.

The availability of services is severely impacted by distributed denial of service (DDoS). A cloud may initiate such attacks or become their target. Compared to attacks on a single physical server, DDoS attacks on the cloud can cause much more harm. One of the most well-known DDoS attack types is DNS reflection amplification. We introduce a novel method for reducing the sources of DNS reflection amplification threats in this research. To control all network traffic in the virtualization environment, we deploy cloud hypervisors in particular. The suggested method is easier to deploy and can avoid situations that ISP edge router ingress screening cannot.

(Virupakshar, K.B., Asundi, M., Channal, K., Shettar, P., Patil, S. and Narayan, D.G., 2020. Distributed denial of service (DDoS) attacks detection system for OpenStack-based private cloud. Procedia Computer Science, 167, pp.2297-2307.) Cloud computing is a popular technique employed by Internet-based apps that are expanding quickly. The cost of operating and maintaining IT infrastructure is reduced by moving to the cloud. Networking protocols and standards are used by numerous enterprises to manage cloud resources online. Because of this, IT infrastructure is spread in nature but is managed centrally, leaving attacker infiltration exposed. One of the most frequent invasions in private clouds is the Distributed Denial of Service (DDoS) attack, which causes service degradation or denial.

This research focuses exclusively on DDoS assaults that are designed to identify bandwidth flooding and connection flooding. These attacks target the network layer of the cloud, which is configured to reject legitimate requests and accept illegitimate requests. The entire cloud infrastructure gets exposed as a result and is susceptible to DDoS attacks. To get around this, you need a cloud operating system with a built-in firewall and DDoS detection mechanism.

Here, a network traffic monitoring system with an integrated OpenStack firewall and raw socket programming is suggested. Using a dataset created in a controlled DDoS assault scenario, decision tree, K closest neighbor (KNN), Naive Bayes, and Deep Neural Network (DNN) algorithms are evaluated against the trained model. DDoS attacks are finally found, and the private cloud administrator is informed.

(Gupta, A. and Sharma, L.S., 2020. Detecting attacks in high-speed networks: Issues and solutions. Information Security Journal: A Global Perspective, 29(2), pp.51-61.)

One of the network needs for identifying network attacks is the use of intrusion detection systems. Attackers that find novel techniques to assault and damage network security typically pose a challenge to organizations trying to secure their data from hackers. Attacks using dispersed, high-speed botnets make detection more challenging. Early identification of network attacks is essential to preserving legitimate users' confidentiality as well as the network's architecture against these threats.

The detection of DoS and Port Scan network attacks in a high-speed network is addressed using Snort, an open-source Intrusion Detection System (IDS). It has been suggested to use Snort custom rules to identify DoS and port scan attacks in high-speed networks. With multiple attack generators, including Scapy, Hping3, LOIC, and Nmap, the rules are contrasted and tested. It has been empirically demonstrated that Snort's detection effectiveness for DoS and Port Scan attacks utilizing the new rules is around 99% for all assaults other than Ping of Death. The suggested method performs effectively for various attack generators in a high-speed network.

Singhal, P., Medeira, S., and Khorajiya, M., 2020. Big data technologies for the detection of application layer DDoS attacks, Journal of Discrete Mathematical Sciences and Cryptography, 23(2), pp. 563-571.)

The velocity of data generation and the sources of data generation are vast in the modern world, where data is essential. Because of the rise in data threats, data security has become more and more crucial. Traditional defense strategies are failing to work. This study presents a novel approach to network log analysis using R and big data.

The method examines a set of criteria to identify bogus IP addresses in network logs using the pig scripting language. The outcomes of applying this technique to identify the dataset assault are also shown. Because of the rise in data threats, data security has become more and more crucial. In this paper, a novel approach to network log analysis with R and big data technologies is presented. Based on a set of criteria, the method finds bogus IP addresses in network logs. The outcomes of applying this technique to identify the dataset assault are also shown.

Internet of Things security concerns and essential technologies (Fadhil, S.A., 2021). pp. 1951–1957 in Journal of Discrete Mathematical Sciences and Cryptography, 24(7).)

New security issues are emerging as the Internet of Things (IoT) grows quickly. The expansion of data sharing is critically dependent on security issues. Concerns over the security of data exchange are growing in many nations. The Internet of Things (IoT) is the next stage in the development of the internet. It enables us to collect, examine, and disseminate data that we might later use to create information, knowledge, and, eventually, wisdom. IoT assumes a crucial role in this scenario. This article addresses the major technologies that need to be prioritized in the Internet of Things security and analyzes the security position of the IoT, as well as the security threats of the IoT from three dimensions: physical security, computational security, and data security. The expansion of data sharing is critically dependent on security issues. This essay examines the Internet of Things' security condition. It examines the IoT security risks from three angles: physical, computer, and data security. The main technologies that should be prioritized in security are also covered in the paper.

# Chapter III. Research Methodology

The main goal of this research is to find out a convenient solution for DDoS TCP SYN Flood Attack. To find a new & better solution we need to design a new framework that will solve the existing issues such as slow detection, spoofed ip attack any many others. For designing our framework, we have upgraded an existing framework and added necessary steps to detect the attack early.

In this research paper the discussion will be based on preventing two different cases of attack form the attacker.

1. SYN – ACK lost (no destination)
2. SYN – ACK complete —no response for (ACK)

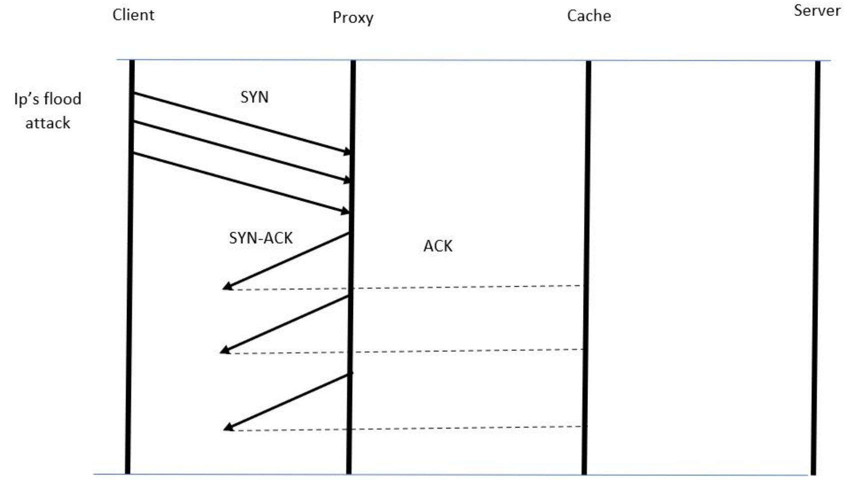


Figure 3.1 SYN – ACK Lost

In the figure 3.1, it can be seen that if any anonymous or someone IP try to connect with the server but before connecting with the server this anonymous IP have to go through the Proxy server and the Cache due to security purpose. When the client or anonymous user's only motive to make a flood of SYN request from the client PC (Device). This user does not wait for the response of the proxy. That's why the proxy server could not find the destination to send the ACK request to establish a network. This user is trying to send so much requests of SYN so that it creates a flood in the proxy server. If the user successes in his plan, then our system will be busy for other to use it might crash in some point. So, whenever the flood is happening the cache warns the proxy server about the irregularity of the IP address from where the connections are coming then proxy server halt the connection and stop it there to execute any command.

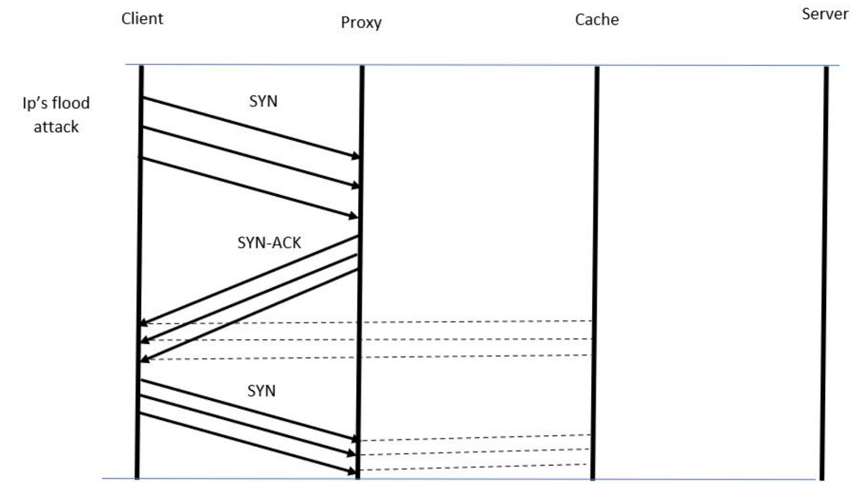


Figure 3.2 No ACK

In this case the user is trying to send SYN request as much as the user want but every time the user send the SYN request. In reply the Proxy is sending SYN-ACK (Syn-Acknowledgement) to the device but the client doesn't wish to create a secure connection to share data.

When the user is sending only the SYN request the proxy instantly sending it a recognition massage through ACK. But the user doesn't wish to create such a connection. So, the user doesn't response to the SYN-ACK of the proxy. Instead of that the user sends again SYN requests. This type of IPs will be saved in the cache, so that the user can't use the same ip again to attack or make a flood. When the next SYN request come Proxy will see the Cache first then give ACK to request. If the data is found in the Cache, then that ip will be block again this data will be saving in the Cache for future purpose. Though this type of attack is still not so common and usually doesn't take place and need very special type of software and hardware is required to create this attack. So because of the complexity and cost this attack is not taking place right now but for future cases this might happen.

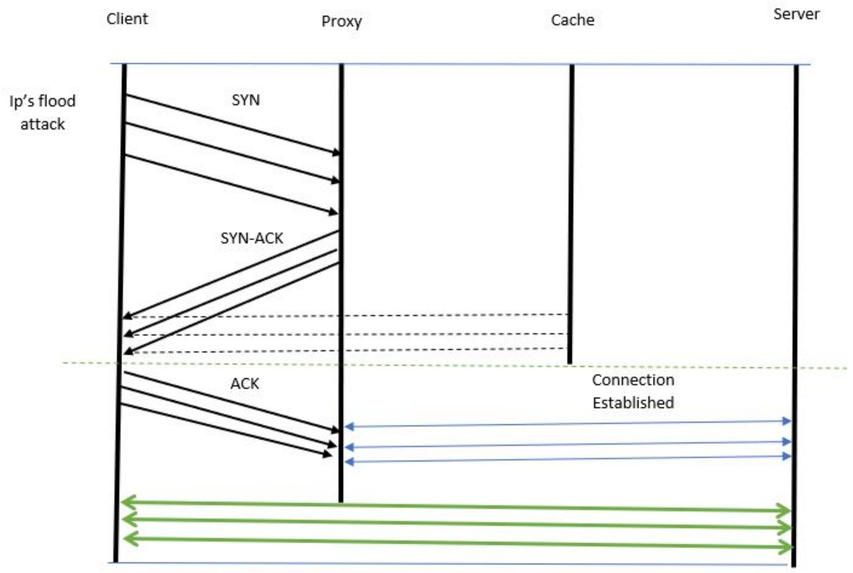
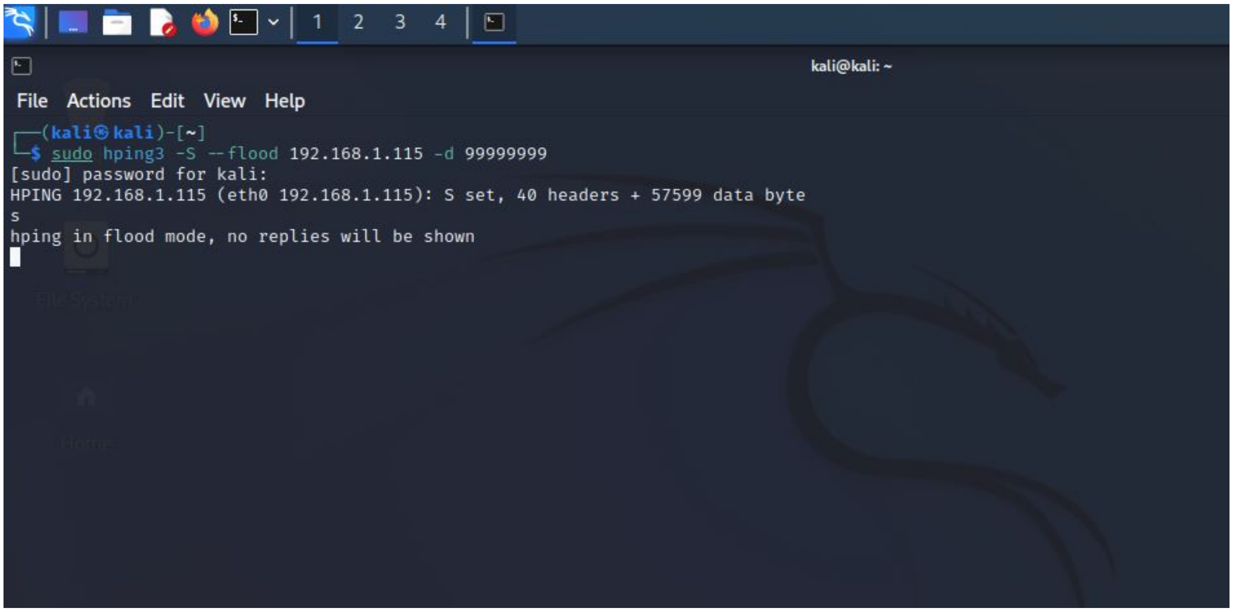


Figure 3.3 Establishing Secure Connection

Multiple SYN signal/request is coming from single/multiple IP. So rather than directly communicating with the server it is interacting with proxy. Proxy is sending a Syn-acknowledgement (SYN-ACK) to every request and the SYN-ACK is complete so they keep the record to the cache & tell the system to wait for Acknowledgement (ACK) from client. Client sends the ACK signal so the secure connection is established. After the connection is established the proxy & cache will not interact with the connection. Also, there will be an connection period fixed for every connection. So, after a particular time the device again needs to gives ACK signal to keep the connection established otherwise there is a possibility that attackers might first complete the 3-way-handshake of TCP Protocol and after completing the connection when it is directly connected with the server and the proxy & cache is overlooked, they will continue with the attack. So, it is also necessary to put some security feature even it proves itself as a legitimate user.

# Chapter IV. Result and Discussion

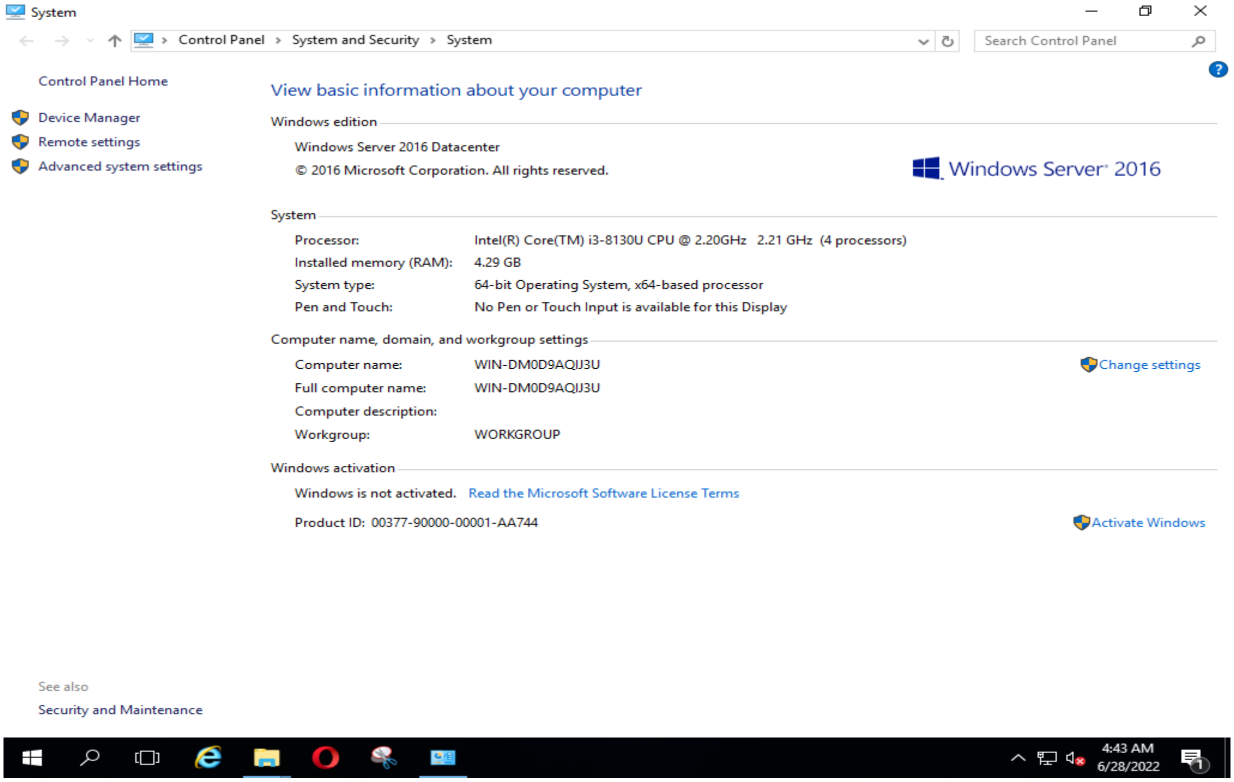
In this command we are using hping3 tool to generate an TCP SYN Flood attack. Here, -s set the signal to be a SYN signal & --flood set the signal to be in flood mode & -d determines the data pack size.



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S --flood 192.168.1.115 -d 99999999
[sudo] password for kali:
HPING 192.168.1.115 (eth0 192.168.1.115): S set, 40 headers + 57599 data byte
s
hping in flood mode, no replies will be shown
```

Figure 4.1 hping3 in Kali Linux

The hping3 tool allows to send manipulated packets including size, volume, and fragmentation of packets to overwhelm the target and get around or through firewalls. For security or capabilities testing, Hping3 may be helpful. [n1]. Now as the signal is set as SYN by using '-s' so we are generating SYN Type attack that we intended to test. Next, we have set the SYN signal in the flood mode using '--flood'. If it was not in the flood mode then the attacker server will start waiting for replies and start taking replies but that will not create the scenario as the ddos attack. So, we must need to set the SYN signal in the flood mode. After that we have given the IP number of our destination computer so that it can find which pc it needs to attack. Finally, by using the'-d' we have set the data size of the packets that we are generating. In our testing the highest possible packet size was set to generate the largest possible packet. By doing these tasks we have simulated our attack for testing purpose.



View basic information about your computer

Windows edition

Windows Server 2016 Datacenter

© 2016 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz 2.21 GHz (4 processors)

Installed memory (RAM): 4.29 GB

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name: WIN-DM0D9AQIJ3U

Full computer name: WIN-DM0D9AQIJ3U

Computer description:

Workgroup: WORKGROUP

Windows activation

Windows is not activated. Read the Microsoft Software License Terms

Product ID: 00377-90000-00001-AA744

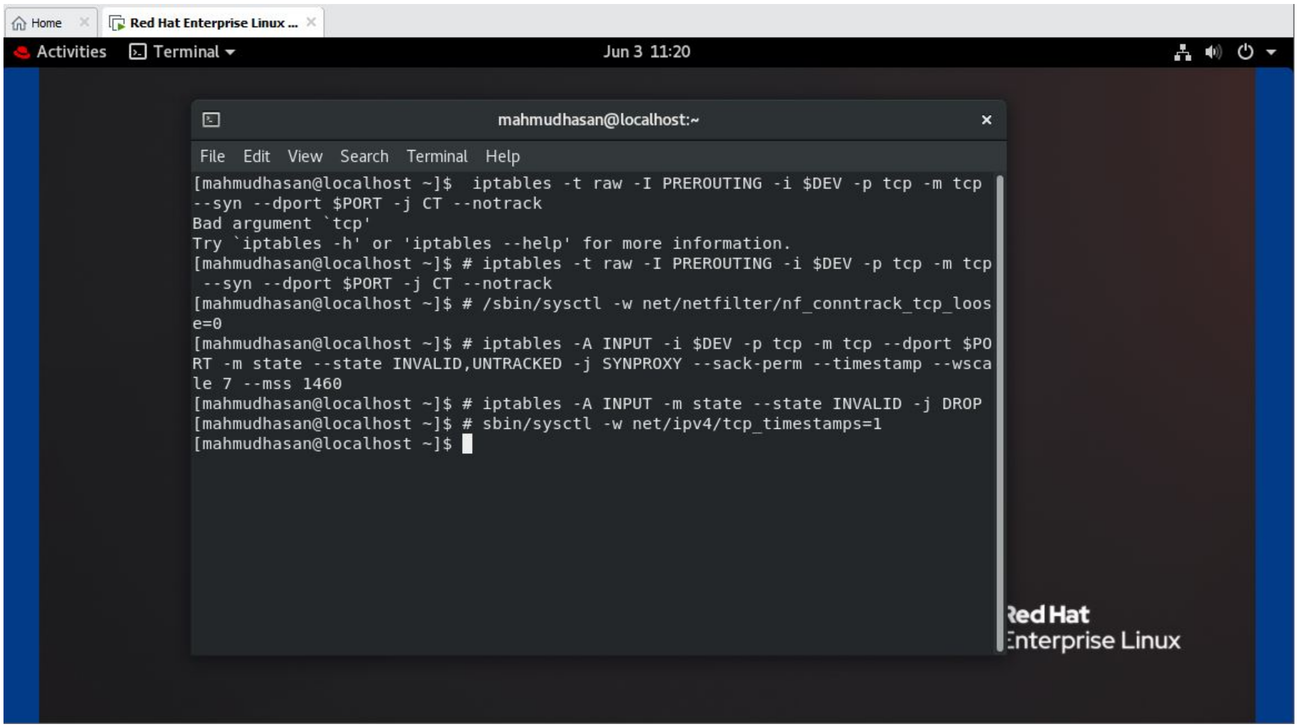
Figure 4.2 Our Attacked System Configuration

Specification:

Processor: Intel i3 8130U

Ram: 4.29 GB

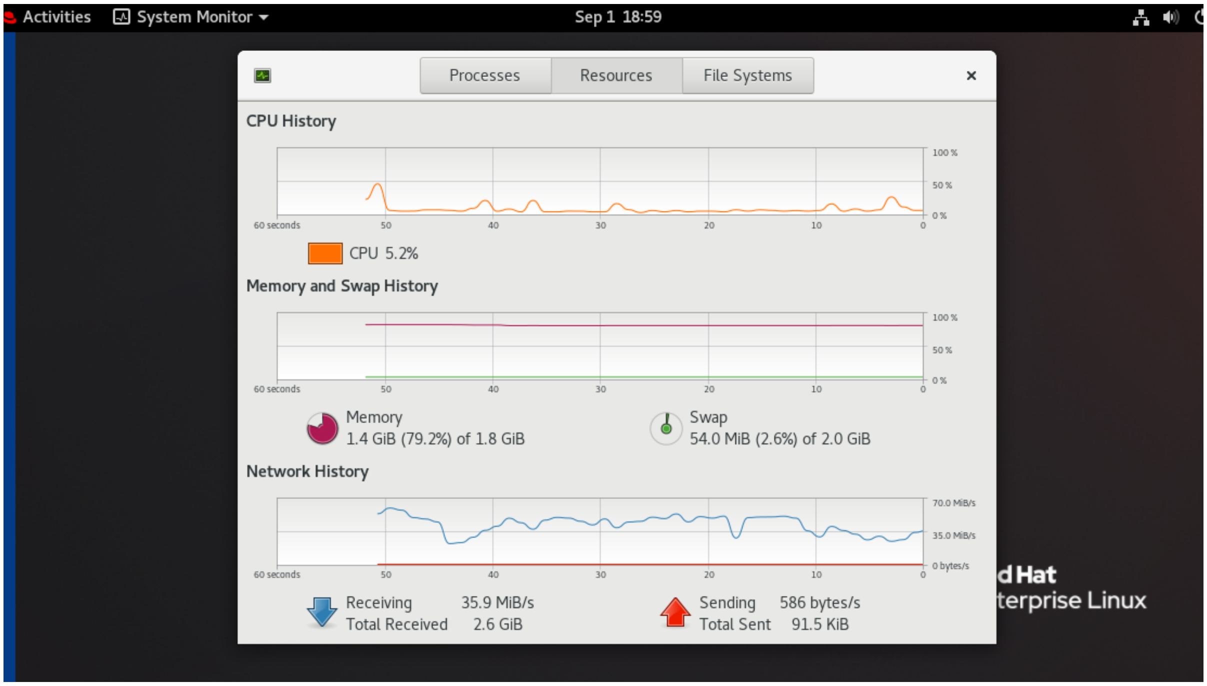
OS: Windows Server 2016 Datacenter



```
[mahmudhasan@localhost ~]$  iptables -t raw -I PREROUTING -i $DEV -p tcp -m tcp
--syn --dport $PORT -j CT --notrack
Bad argument `tcp'
Try `iptables -h' or 'iptables --help' for more information.
[mahmudhasan@localhost ~]$ # iptables -t raw -I PREROUTING -i $DEV -p tcp -m tcp
 --syn --dport $PORT -j CT --notrack
[mahmudhasan@localhost ~]$ # /sbin/sysctl -w net/netfilter/nf_conntrack_tcp_loos
e=0
[mahmudhasan@localhost ~]$ # iptables -A INPUT -i $DEV -p tcp -m tcp --dport $PO
RT -m state --state INVALID,UNTRACKED -j SYNPROXY --sack-perm --timestamp --wsca
le 7 --mss 1460
[mahmudhasan@localhost ~]$ # iptables -A INPUT -m state --state INVALID -j DROP
[mahmudhasan@localhost ~]$ # sbin/sysctl -w net/ipv4/tcp_timestamps=1
[mahmudhasan@localhost ~]$ 
```

Figure 4.3 Using Netfilter on Kali Linux

In the figure 4.2, we can see the code that we used to implement net filter on our Linux Machine. We have also tweaked the code so that it drops the connection with no destination. Also we have set the code the to drop connection if a connection is taking much longer than expected.



🟧 CPU 5.2%

Memory
1.4 GiB (79.2%) of 1.8 GiB

Swap
54.0 MiB (2.6%) of 2.0 GiB

Receiving 35.9 MiB/s
Total Received 2.6 GiB

Sending 586 bytes/s
Total Sent 91.5 KiB

Figure 4.4 Mitigating on Kali Linux

We can see that we are receiving upto 35.9 MiB/s which is close to 287mbps. Even in this 287mbps attack our cpu is only at 5% usage. So, it can be said that our system is working properly & it is blocking the attack and it is not allowing to make the server busy. As the server is not busy so it will be able to reply to the legitimate connection.

# Chapter V. Conclusion

We have discovered that DDoS TCP SYN Flood was one of the most severe attacks for small to big organization. So, we decided to mitigate that & found out that though some researches already used various techniques and developed various tools still the attack is not totally mitigatable. So, we choose the most efficient design till now existed and upgraded it with our own design. By doing so we have achieved not only much more efficiency but also we have attempted to detect earlier than all the models. While the testing phase we have found out a new type of attack which is can cause more severe damage than the current TCP SYN Flood but the bright side is that attacker's information gets leaked in this type of attack. So in future scope we plan to work with that attack. Still there are some flaws such as whenever the attack gets much bigger for few moments the protection system might not work but eventually it will restore itself as soon as possible. There are some future scopes to increase the attack detection efficiency, attack mitigation in windows system and reducing cost overall.

# References

Eddy, W.M., 2006. Defenses against TCP SYN flooding attacks. *The Internet Protocol Journal*, *9*(4), pp.2-16.

Liu, P.E. and Sheng, Z.H., 2008, July. Defending Against TCP SYN Flooding with a new kind of SYN-Agent. In *2008 International Conference on Machine Learning and Cybernetics* (Vol. 2, pp. 1218-1221). IEEE.

Eddy, W., 2007. *TCP SYN flooding attacks and common mitigations* (No. rfc4987).

Ubale, T. and Jain, A.K., 2018, March. SRL: An TCP SYNFLOOD DDoS mitigation approach in software-defined networks. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 956-962). IEEE.

Lemon, J., 2002. Resisting {SYN} Flood {DoS} Attacks with a {SYN} Cache. In *BSDCon 2002 (BSDCon 2002)*.

Rahouti, M., Xiong, K., Ghani, N. and Shaikh, F., 2021. SYNGuard: Dynamic threshold based SYN flood attack detection and mitigation in software defined networks. *IET Networks*, *10*(2), pp.76-87.

Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A. and Abduallah, W.M., 2019. Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods. *IEEE Access*, *7*, pp.51691-51713.

Soliman, A.K., Salama, C. and Mohamed, H.K., 2018, December. Detecting DNS reflection amplification DDoS attack originating from the cloud. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)* (pp. 145-150). IEEE.

Virupakshar, K.B., Asundi, M., Channal, K., Shettar, P., Patil, S. and Narayan, D.G., 2020. Distributed denial of service (DDoS) attacks detection system for OpenStack-based private cloud. *Procedia Computer Science*, *167*, pp.2297-2307.

Gupta, A. and Sharma, L.S., 2020. Detecting attacks in high-speed networks: Issues and solutions. *Information Security Journal: A Global Perspective*, *29*(2), pp.51-61.

Singhal, S., Medeira, P.A., Singhal, P. and Khorajiya, M., 2020. Detection of application layer DDoS attacks using big data technologies. *Journal of Discrete Mathematical Sciences and Cryptography*, *23*(2), pp.563-571.

Fadhil, S.A., 2021. Internet of Things security threats and key technologies. Journal of Discrete Mathematical Sciences and Cryptography, 24(7), pp.1951-1957.)