# An Analytical Study on Technique against DDoS Attack on the Cloud Networks

**Pruthviraj R. Pawar[1]* Dr. Santosh Kumar Mishra[2]**

[1] Research Scholar, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow

[2] Assistant Professor, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow

***Abstract – Our study remains localized on DDoS attacks and aims to provide the prevention algorithms and suggest the solutions for data security against malware injection attacks. Further, our endeavor would be to bring out the reasons for reluctance for adoption of Cloud ‘toward achieving the third objective of the study. This study ‘s focus is on security threats on public cloud and the prominent cloud network security attacks including data security attacks on public cloud with proposed solution(s). The study also intends to bring out the facts about reluctant in adoption of cloud computing by consumers and their level of awareness among them.***

***Keywords: DDOS, Cloud Networks, Technique, Experimentation Algorithm;***

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - x - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## INTRODUCTION

DDoS (Distributed Denial of Service) attack is a form of DoS (Denial of Service) attack. It is one of the most significant threats to the security of the cloud network. DDoS attack targets an internet server with the aim to exhaust the server‘s resources to prevent the authorized access to cloud services or to degrade the quality of cloud services. In other words, DDoS attacks are launched to shut down the servers for a certain period or to make resources nonfunctional for a time period. DDoS attack is usually lunched by sending unlimited number of packets, application level flood and executing malwares to the server.

Launching DDoS attack is quite an easy task with availability of various prewritten tools on the internet. The attacker tries to find out large number of IP address blocks of those systems that have security vulnerability. This initial phase of mass intrusion employs automated tools (such as botnet) that remotely compromises several hosts (hundreds to thousands) and installs DDoS agents on those systems. Theses automated tools (toolkit) compromise the victim systems. These compromised systems are the initial victims of the DDoS attack. Then, the victims exploited systems will be loaded with the DDoS daemons that carry out the actual attack. Countering DDoS attacks is becoming more challenging with the vast resources and techniques available to the attackers. We have mainly focus on the flood attacks (main categories are SYS, ICMP and Slow Read) using DDoS. In this chapter, we propose an algorithm for preventing DDoS Attacks followed by detection techniques.

## DENIAL OF SERVICE ATTACKS

DDoS attacks are executed to disable networked circuits, server systems by limiting access to them and termed as DoS attacks. These deny users the access to applications like Email, Chat, Ecommerce or Banking, or hosted Cloud services like SaaS, PaaS or IaaS Cloud services and computing resources like Network or VoIP infrastructure. The attacks are performed from a single source address as described by Deshmukh, et al. (2015) and illustrated in Figure 1 below.

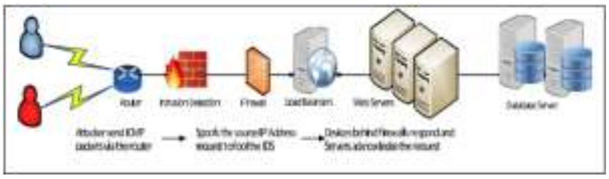

**Figure 1: Denial of Service attack process**

## DISTRIBUTED DENIAL OF SERVICE ATTACKS

DDoS based attacks started with cyber-attacks on Gaming and Gambling web sites, the new cyber-attacks are now used for political reasons, financial gains and even as diversionary tactics to steal intellectual property and data. These cyber-attacks then amplify the Denial of Service attack by launching a flood assault from several thousand nodes by bombarding the target with malformed information requests and data packets in order to overwhelm the infrastructure and disrupt normal operations. These attacks are termed as DDoS attack, as described by Mishra, et al. (2011) and Anwar, et al. (2014).

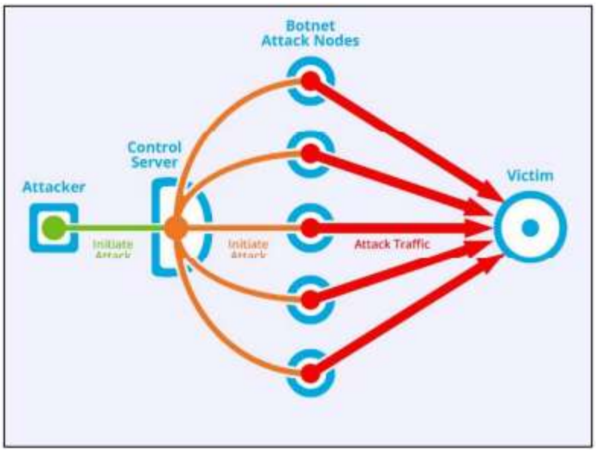

**Figure 2: DDoS attack using Botnet nodes**

Figure 2 above illustrates the use of Botnets sending amplified requests and turning the DoS attack into a DDoS flood attack. The attacker exploits vulnerable systems across geographies by compromising them with a malicious payload. This payload infects the end user systems with a malware application which enables the attacker to gain remote access with command and control capabilities. This is performed without the knowledge of the users with the intent to have the target services, hosted web applications unavailable to the authorized users as well as unavailability and security issues for Cloud computing services. This is presented by Zargar, et al. (2013) and Wong, et al. (2014) with DDoS attacks being performed by sending a flood of network packets, data or transaction requests over the Internet. These are sent from multiple locations and multiple systems at the same time. The infected and compromised user systems or nodes are referred to as Zombies or Bots which further compromise other user systems. The flood of compromised systems working as a group is known as Botnets and also controlled by a single attacker performing the attack sequence as shown in Figure 5. DDoS attacks present a high priority risk for Cloud service providers and Cloud service consumers with regards to the hosted infrastructure for managing the Service Level Agreements, Cloud service delivery, Cloud availability and avoiding any collateral damages.

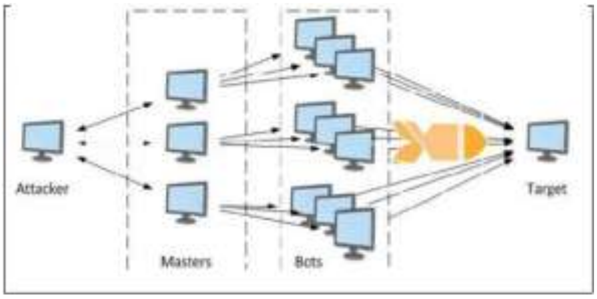

**Figure 3: Distributed Denial of Service attack process**

Besides having the cloud infrastructure and services being unreachable to actual consumers resulting in rental losses, increased delivery cost and harm to reputation, which further lead to legal and financial consequences, DDoS attack directly impacts the cloud providers and cloud service consumer in the below mentioned ways.

- Resource exhaustion like over whelming and consuming the bandwidth capacities of Internet pipes or making the Server CPU and memory to exceed the capacity
- Triggering Fallbacks to have the Intrusion Detection systems or Web Application Firewalls alter from their filer-mode to log-only-mode.
- Exploitation of end user accounts with lockouts by repeatedly attempting logon access with invalid credentials
- Cloud Portal access disruption by crashing the web application process by attacking vulnerabilities in the application code or altering user types to an invalid type and making it incorrect to input data for the legitimate user
- Camouflage the real attack motive by diverting the Security team attention acting as a smoke screen to steal data or hijack other services
- Pushing malware which affects the user access and data by opening up sockets and triggering errors in the micro codes

## TYPES OF DDoS ATTACKS

DDoS attacks are broadly categorized into three main types of attacks depending on the area of Cloud infrastructure on which the cyber-attack is focused. These attacks are described in the below section as

- Network or Volumetric DDoS attack
- Application layer DDoS attack

- Reflector DDoS attack

## ALGORITHM FOR PREVENTING DDoS ATTACKS

Our proposed algorithms uses three filtering techniques to prevent the DDoS attack. Combination of all three filtering techniques namely first filter, second filter and third filter, makes it strong enough to prevent the DDoS attacks. Our first filter is used for authentication purpose by third party. We have used third party authentication because we want secure connection for sending the authentication code. Using our second filter, we compare the current requests with predefined limits of requests from the table. Last one filter is used to filter out the spoofed packets.

### Model/Flow Diagram for Proposed Algorithms

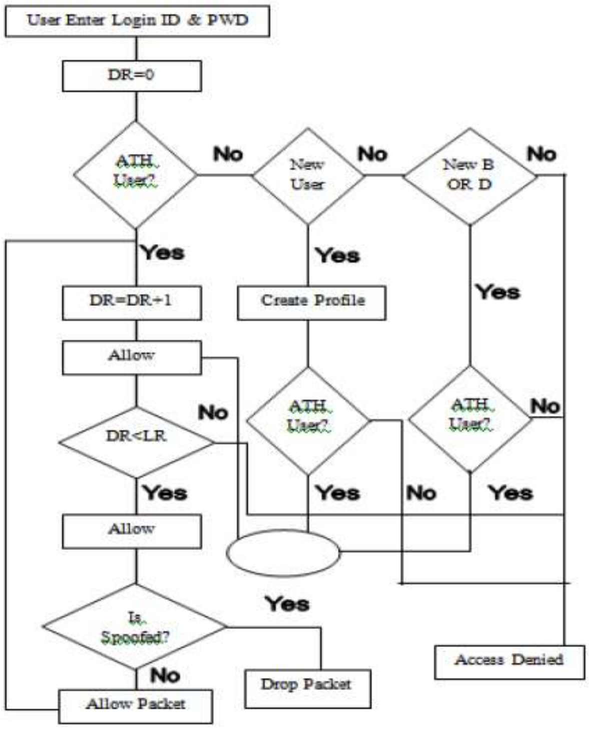

**Figure 4. Model/Flow Diagram of Proposed Algorithm**

In this flow diagram UID, PWD, ATH, D and B indicate user ID, password, authentication of the user, New Device and New Browser respectively used by the third party for authentication (Filter 1).DR and LR indicate demanded Requests by the user and predefined requests in the table respectively (Filter 2). Packet spoof is check by the Hop-Count filtering technique (Filter 3).

### Proposed Algorithm

LR= Predefined limit of requests

R= Number of requests demanding by User(s)

P= Packets in the network

S= Spoof packet in the network

**Step1:** Authenticate User ID& Password using Third Party (if required)**//1.**

**Authenticate the user**

**Step 2:** Compare R with LR **// 2. Compare the limit of requests**

If [R>LR]

Goto step 5;

Endif

**Step 3:** Check the Packets using Hop-Count Method **// 5. Check the spoofed packets**

If [P==S]

Discard packet;

Endif

**Step 4:** Repeat step 2& 3 while user logout

**Step 5:** Service Denied and Exit

### FIRST FILTER:

First filter is used to authenticate the user by using third party authentication at first or even if he/she uses deferent device or different browser (used previously) for accessing the service. This filter confirms that no hacker is using cloud services. In the whole process of authentication, every current value will be compares from history tables. We have used four tables in this process T1, T2, T3 and T4 to Store all the information of user when he/she creates the profile, Store all the information of device and browser which is used by the user, Store all users Login ID and Password and list of registered mobile numbers to send the unique number. Process will be as follows:

1. User enters the ID and Password and if it does not exist, he/she creates a new profile with his/her registered mobile number using third party authentication and if successful then all information is stored in table T1, T2, T5. Otherwise message access denied contact to CSP is displayed.

2. User enters the ID and Password and if successful than compare the current device and browser with history data base, if found new than again authenticate the user by third party, if found successful then add new

entry into T2.if not message access denied contact to CSP is displayed.

3. Jump to the 2nd Filter algorithms

**Detailed Algorithms for Step 1 (Authentication):**

**Tables used**

T1=Stored all information of user (whole profile)

T2=Store information of device and browser using by users

T3=Stored user ID and password

T4=Stored register mobile numbers Lists with user identification for send the

One Time Password (OTP)

**Variables used**

LID=User ID which is entered by the user using browser

TLID=User ID which is stored in Table 3 (T3)

PWD=User password which is entered by the user using browser

TPWD=User password which is stored in Table 3 (T3)

OTP=A number generate by third party on registered mobile numbers

UOTP=A number is entered by the user at the time of creation of profile

UDev=User device which is used by the currently

UBrow=Browser which is used by the user currently

TUDev=User device history which is stored in T2

TUBrow=Browser history which is stored in T2

**Step 1:** Input login ID & Password and compare from the T3

```
If [LID≠TLID AND PWD≠TPWD]
Print: “Does Not Exist”
Print: “Create New Profile”;
If [UOTP≠OTP] // third party authentication
Print: “Contact your Cloud Service Provider”
Go to step 4;
Endif
If [UOTP==OTP] //third party authentication
Add record T1, T2 and T3
Go to step 3;
Endif
Endif
```

**Step 2:**Compare device and browser used by the user currently

```
If [UDev≠TUDev OR UBrow≠TBrow]
//third party authentication on registered mobile number
If [UOTP≠OTP] // third party authentication by OTP
Print: “Contact your Cloud Service Provider”
Go to step 4;
Endif
If [UOTP==OTP] //third party authentication
Add new entry in T2
Go to step 3;
Endif
Endif
```

**Step 3:** Allow requests and compare the demanding request with pre-defined request with the help of **Algorithm 2**

**Step 4:** Exit

**SECOND FILTER:**

Second filter is used to restrict the user to gain access extra services/resources. This filter ensures that no one can send extra flood knowingly or unknowingly. Therefore, our server will be safe from the flood attacks like DDoS. Here we are using table T5 to store the information about limits of requests of all the users and updating the current request used field time to time. Here we are assuming that the limits of requests are defined on hourly basis.

1. If user crosses limit of requests as predefined limits then services would be denied until next session starts. Suppose a user has right to access 200 requests in one hour and he/she cross the limit of 200 within half an hour than his/her services would be denied until next hour start.

2. Jump to the 3rd Filter algorithms

**Detailed Algorithms for Step 2 (Limit):**

**Table(s) used**

T5= Used to store the information of requests limit and demanding requests

**Variables used**

LR= Predefined limit of requests //

R= Number of requests demanding by User(s)

**Step 1:** Count R and update Table T5 with R

**Step 2:** Compare R with LR

If [R>LR]

Print: " Access Denied"

Go to step 4;

Endif

**Step 3:** Allow packets and check the packets with the help of **Algorithm 3**

**Step 4:** Exit

**THIRD FILTER:**

In our third filter, we have proposed Hop-Count Filtering algorithm with some modifications. In the earlier proposed algorithm, there was a requirement of using four cases to recognize the legitimate packets however; we have used only two cases. This makes the algorithm stronger than the previously proposed algorithm. It ensures that no spoofed packet will be entertained. Hop-Count Filter (HCF) algorithms are used to filter the IP packets, which requires continuous monitoring during travel over the cloud networks and extract Synchronous Flag, TTL (Time to Live) and Source IP from these packets. It recognizes two cases (modified-4 cases with 2) for each captured packet in the entire process. In both cases, Synchronous Flag should be set (SF=1) otherwise packet will be discarded and no further processing would be required.

1. If Source IP addresses exist in the table T6 then calculate hop-count using TTL Value of IP packet, which is extracted from the packet. Compare hopcount (HC) matches with the stored hop count (HS), if not then update source HC field of table T6 for that Source IP address (SIP).

2. If source IP address does not exist in the table T6 then calculate hop-count (HC) and add a new entry for the TSIP (Source IP) address with the equivalent hop-count (HC) in the table T6.[84].

**Detailed Algorithm for Step 3 (TTL Based Hop-Count):**

**Tables used**

T6= Store Source IP Address (TSIP) of packets, stored Hop-Count for that IP

Address (HS) and Synchronous Flag (SF) [84].

**Variables use**

TTL=Time-to-Leave Value extract from TCP/IP packet

TF=Final value of TTL (Time to Live) extract from the TCP/IP Packet

TI=Initial value of TTL // by selecting the smallest value from set (modern OS has set values 30,32,60,64,128 and 255) that is larger than its final TTL

SF**=** Synchronous flag bit extracted from TCP/IP packet and stored in T6

HC=To get the Hop-Count Value //TI-TF

HS=Stored Hop-Count Value from T6

SIP=Source IP Address extract from TCP/IP packet

TSIP=Store Source IP Address (TSIP) of packets

**Step 1:** Set TTL and SIP values from TCP/IP packet

**Step 2:** Check Synchronous Flag is set

If [SF≠1]

Goto step 7;

**Step 3:**Check the value of SIP is set

If [SIP≠1] // 1 indicate that value is found

HC=TI-TF // Hop-Count Value

TSIP=HC

Endif

**Step 4:** Calculate Hop-Count

```
HC=TI-TF // Hop-Count Value

Step 5: Compare HC with HS (Stored Hop-Count value in T6)

If [HC≠HS]

TSIP=HC;

Endif

Step 6: Allow packets (Legitimate packets)

Step 7: Discard packet (spoofed packet)

Step 8: Repeat steps 1 to 6 until flow is continue
```

### Proposed Algorithm with Example

Suppose N users are using cloud services and their limits of cloud services are already define in the Service Level Agreement (SLA) on daily basis. Cloud Service Provider (CSP) has registered cloud user‘s mobile numbers with cloud services. These registered mobile numbers are maintained in the table by CSP and provide this table to third party also to authenticate the users with the help of registered mobile numbers by providing a secure connection. This process has two cases one is at the time of creating ID & Password third party authentication uses secure connection to provide the secret unique key to the cloud user based on their registered mobile numbers such that secret key is not tempered by malicious person. After successful completion of first process, third party also creates two tables first, to maintain the history of devices and browsers second, to save the profile information used by corresponding users and handover the copies of both the tables to the CSP. Second case at the time of using new device or browser if cloud user uses a new device or new browser for accessing cloud services then again it would be authenticated by the third party with the help of secret unique key or by asking security questions. If successful, than second table will be updated by adding new information of device or browser. In the whole process of authentication in both cases, if cloud user is unsuccessful then access will be denied. In other words, any un-authenticated person will not be able to use or access cloud services.

With our proposed third party authentication process, there is no possibility that an attacker will successful to send the flood. If an attacker is successful for make victim any legitimate user machine/system because of mistaken disclosure of the authentication credentials then also the attacker would be able to send only limited requests in the form of flood to the cloud network. This means our second filter for service limitation will stop the extra flood from entering the cloud network. Still if there are spoofed packets in the cloud networks, then our third filter Hop-count will discard these spoofed packets from the cloud networks and we would achieve our objective of keeping our server free from the DDoS attacks. This will ensure that the server is utilized only for providing the services to legitimate cloud users.

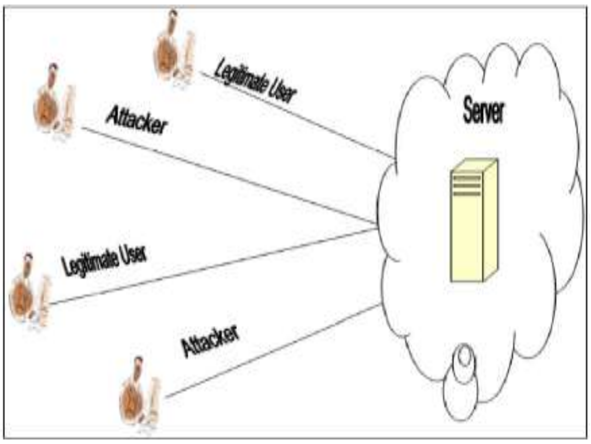

**Figure 5. With DDoS Attack without prevention**

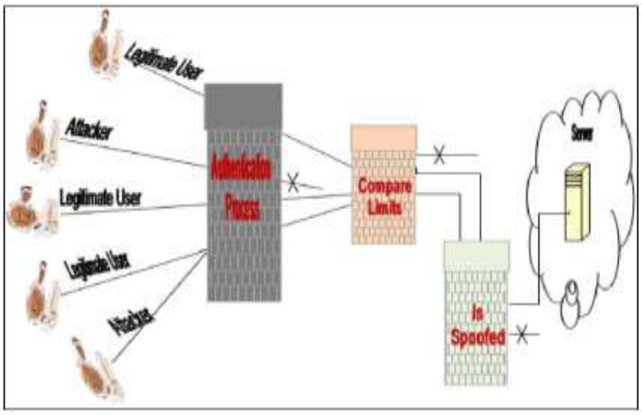

**Figure 6. With DDoS Attack and Prevention with proposed Algorithms**

## COMPARISON OF ALGORITHMS

Hop-count filtering algorithms are based on packet filtering techniques, which requires continuous monitoring of cloud network. It was proposed by Vikas Chauhan el. (2012) at and similar work by Biswa Ranjan el (2009) at Indian Institute of Technology, Roorkee to prevent the DoS attack in the cloud network. They ensured that no spoofed packet will passed to the cloud server. They performed packet filtering based on TTL (time to Live) from the IP Packet for Hop-count, than compared the Compute hop-count value with stored hop count value. This whole process is based on SYN Bit and Source IP address with four different cases such as SYN=1and sourceIP=1, SYN=1and sourceIP=0, SYN=0 and sourceIP=1 and SYN=0 and source IP=0 to filter the packet.

Our proposed algorithms is combination of three different techniques, which makes prevention stronger against the DDoS attack. Our proposed algorithms use three filters and every filter uses strong steps to stop the attacker from sending the flood to exhaust/exploit the server or compromise the resources. After passing through three level -

filtration, attacker would almost never be able to intrude the cloud server. Our authentication process limits the user to access the cloud services. If attacker pass the first filter than our second filter will put limit to send the flood and if attacker sends the malicious packets within the limit than our third filter which is based on Hop-count filter (modified version of existing algorithms) will discard the spoofed packets. Our hop-count filter is considers only two cases if Synchronous bit is set otherwise discard the packet.

## CONCLUSION

We are considering that all tools such as IDS (Intrusion Detection System), IPS (Intrusion Prevention System) and Firewall (ACL firewall category recommended) should also be installed at the gateway. It is a combination of three filtering techniques, which is considered more secure and reliable in the current scenario. In our proposed algorithms, we have considered and combined many different techniques, which are in vogue. The goal of these proposed algorithms is to maximize the system resources and system functionality. When a DDoS attack occurs, the proposed prevention algorithms ensure that, no unauthenticated person can use the systems. We conclude that if these algorithms are implemented in cloud environment with all the tools, it will definitely help to reduce the threats in the cloud.

## REFERENCES

[1]. Liang, K, Au, MH, Liu, JK, Susilo, W, Wong, D, Yang, G (2014). "A DFA based functional proxy re-encryption scheme for secure public cloud data sharing", IEEE Transactions on Parallel and Distributed Systems, vol. 9, no. 10, pp. 1667-1680.

[2]. Ruiping Lua and Kin Choong Yow (2011). Mitigating DDoS Attacks with transparent and Intelligent Fast-Flux Swarm Network‖, IEEE Network, Volume 25, Number 4, August 2011.

[3]. Neeta Sharma, Dr. Mayank Singh, Dr. Anuranjan Misra, Prevention against DDoS Attack on Cloud Systems Using Triple Filter: An Algorithmic Approach.

[4]. Aman Bakshi and Yogesh B. Dujodwala (2010). Securing Cloudfrom DDoS Attack using Intrusion Detection System Virtual Machine, ICCSN'10 Proceeding of the 2010

[5]. Zissis, D. & Lekkas, D. (2012). "Addressing cloud computing security issues", Future Generation computer systems, vol. 28, no. 3, pp. 583-592.

[6]. Zhou M, Zhang R, Xie W, Qian W & Zhou A (2010). 'Security and privacy in cloud computing: A survey', Sixth International Conference on Semantics Knowledge and Grid (SKG), 2010 pp. 105-112.

[7]. Tirthani N. & Ganesan R. (2014). 'Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography', IACR Cryptology e-Print Archive, vol. 2014, p. 49.

[8]. Xu G, Yu M, Cheng J, Li L & Shi Y (2013). 'Data Sampling Algorithms for Data Integrity Verification in Cloud Storage', International Journal of Advancements in Computing Technology, Vol. 5, No. 9.

[9]. Arockiam, L. & Monikandan, S. (2013). "Data security and privacy in cloud storage using hybrid symmetric encryption algorithm", International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 8, pp. 3064-3070.

[10]. Kaur, M. & Mahajan, M. (2013). "Using encryption algorithms to enhance the data security in cloud computing", International Journal of Communication and Computer Technologies, vol. 1, no. 12, pp. 56-59.

**Corresponding Author**

**Pruthviraj R. Pawar***

Research Scholar, Computer Science Engineering, Maharishi University of Information Technology University, Lucknow