IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Shielding the Server from Furtive Distributed Denial of Service Attacks using Dominate and Release Policy

Akshitha Koluguri[1], Akshay Chotiya[2], Arindam Das[3], Aman Ajmani[4], Vishwajeet Singh[5]

Assistant Professor, Computer Science & Engineering, Guru Nanak Institutions Technical Campus, Hyderabad, India[1]

B.Tech Student, Computer Science & Engineering, Guru Nanak Institutions Technical Campus, Hyderabad, India[2,3,4,5]

Abstract: With the increase in threats in cyber security it is predictable that any service providing systems could be easily victimized by intruders. All the incoming requests are served by the servers attached to the systems and thus attackers can target the servers and generate illegitimate requests with the possibility of malfunctioning. The faux requests are practically imperceptible as the traffic they engender is suspected to be in low-rate. We represent respective intensities for legitimate traffic and fake traffic. The initial result is under the assumption of static routing by the attackers, followed by time-varying attacks. We confront the case like Join-the-Shortest-Queue is not a solid policy to defend from the time-varying attacks as the throughput region providing intensities for both legitimate and fake traffics are compromised. This study will provide you a clear idea on how to balance the traffic and defend the network from Distributed DoS attacks which lugs the possibility to degrade the system by using dominate and release(DaR) policy. We shall discuss this method using .NET framework on how we can identify the manual and automatic incoming requests to the server and provide protection.

Keywords: Service provisioning system, Distributed Denial of Service, Dominate and Release (DaR) policy, DDoS attack, illegitimate traffic.

I. INTRODUCTION

Denial of Service (DoS) [10] attack is a type of attack where the intruder targets a specific server or system and makes its attempt to penetrate the system. This type of attack will not endeavour to fetch any kind of personal data but their prime intent is to make the server or system unavailable for any kind of service. Distributed Denial of Service (DDoS) attack is similar to DoS attack, the only difference is that in DoS attack the attacker will attack from only one system and in DDoS attack the attacker will attack from numerous system, can also be called as Botnets. Botnets are not the only source of DDoS attacks [1]. Social media sites can coordinate large number of willing users to carry out DDoS attacks as illustrated by WikiLeaks inspired attacks in late 2010,[1]thereby increasing the traffic to the server or system and generate a heavy load of traffic, making the targeted system unavailable to respond to any service.

Reported DDoS attack traffic grows by 23 per-cent quarter-over-quarter, up by 75 per-cent from fourth quarter of 2012 [2].As in Service provisioning systems the services are provided to the authorized users, here the DDoS attacks try to create fake traffic with an attempt to penetrate the system, making the server slow enough to provide required services to the legitimate users. So, in order to provide safety and shield the server from such kind of menacing DDoS attack it becomes the elite priority to use a system which will be able to distinguish between the manual traffic and automatic traffic and thus keeping the server safe from malfunctioning.

This system defends the DoS and DDoS attacks by working together with several attack detection techniques, which detects the legitimate and illegitimate traffics and based on which the identification of the intruders can be processed. The malicious intruders might use a large number of computers which are clustered in botnets, responsible enough to create and generate faux requests to damage the servers. As the fake requests resemble enough with the legitimate ones it becomes hard to distinguish them. So, this system uses statistical processing of arriving requests where the system sends a random code to the user to detect if it is a manual or automatic process. After the confirmation is done and if it is a manual process then only it provides service to the system. We define the policy of "Dominate and Release" (DaR), which consists of mainly two phases. With the combination of all such detection technique we aim to sustain the service provisioning system from being getting affected by DoS (Denial of Service) or DDoS (Distributed Denial of Service) attacks, which can basically damage the server and make the server unavailable for use and disrupt it with the loss of data.

II. EXISTING SYSTEM

The existing system is a service provisioning system consisting of bank of servers, providing services to the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

incoming requests. As this system is not fashioned for shielding the furtive attacks, malicious intruders can generate fake requests, attempting to degrade theservice provisioning. One of the common methods of attacks isDoS and DDoS.

A. Denial of Service

Denial-of-Service [6] attacks aim at reducing services availability by exhausting the resources of the services host system, like memory, processing resources and network bandwidth.[6] The Denial of Service (DoS) attack is an attempt to cause a server to stop functioning properly. One common method of attack involves saturating the target machine with communications requests, such that it cannot respond to legitimate traffic, or it responds so slowly as to be rendered effectively unavailable. DoS attacks result in service downtime for corporations and organizations that use Internet services, which in turn are translated to significant financial costs.

B. Understanding the cost of an attack

Organizations observed a number of different business impacts as a direct result of DDoS attacks.[1]

About half cited operational expenses and nearly 40 per-cent indicated reputation or customer loss due to DDoS attacks. One-fifth indicated direct revenue loss, with other impacts including employee turnover and stock price fluctuation. The costs associated with DDoS attacks are multi-faceted, and organizations should factor all of these into their calculations when looking at their investment strategies for defensive solutions.[1]

C. ICMP (ping) Flood

ICMP flood overwhelms the target resource with ICMP Echo Request (ping) packets, generally sending packets as fast as possible without waiting for replies. This type of attack can consume both outgoing and incoming bandwidth, since the victim's servers will often attempt to respond with ICMP Echo Reply packets, resulting a significant overall system slowdown.

III.PROPOSED SYSTEM

We define the "Dominate and Release" (DaR) policy. The policy operates in periods, where period (i) has duration, which is a parameter chosen by the policy.

Each period has two phases. In the Dominate phase, the policy targets one by one all dominated servers causing their backlogs to increase. In this section we present simulations for a system with random arrivals and service times, and First-Come-First-Serve (FCFS) discipline in order to demonstrate that the presented results are not an artifact of our traffic modelor of our HLPPS assumption [5][8][9].

A. Dominate and Release policy

Dominate and Release policy is a type of policy that we shall demonstrate here. Initially when the attackers try to send illegitimate requests to penetrate through the server getting jumbled with the legitimate traffic the system tries to take control over those traffic. This is where the dominate policy works out where the system identifies the auto-generate traffic and the manual traffic by filtering them in the throughput region and thereby blocking the fake traffic and releasing only the genuine requests to the server.

We define the "Dominate and Release" (DaR) policy. The policy operates in periods, where period i has duration τ_i, which is a parameter chosen by the policy. Each period has two phases. In the Dominate phase, the policy targets one by one all dominated servers causing their backlogs to increase. The time spent on j^{th} dominated server t j i is designed so that at the end of Dominate phase all dominated servers have backlogs greater than a parameter B_i. The entire phase lasts for d_i. In the Release phase, the policy performs a static routing directed only to free servers for duration r_i.[3]

The Dominate phase is composed of J intervals, where in interval j the fake traffic is targeted to j^{th} dominated server. The duration of j^{th} interval t^j_i is chosen according to the recursive expression.[3].

B. First-Come-First-Serve (FCFS) discipline

First come, first served is a very common way of organizing access to a limited resource or service in the real world. It can be explained by saying that whenever the resource is available the person who has been waiting the longest is served.

In our system the random arrival of the requests are served in FCFS discipline to demonstrate that the presented results are not an artefact of our traffic model or of our HLPPS assumption.

C. Join the Shortest Queue (JSQ) policy

The JSQ policy routes the traffic to the less loaded server. JSQ is known to achieve throughput optimality in the bipartite routing problem [5][6], in the absence of the malicious attacker. Also, it is known to have good load balancing properties. [5], [7], [8]

D. Guaranteed Throughput Region

In a dynamic attack the servers with individual capacity less than the attack intensity become neutralized. Since an attacker can time-share between dynamic and static attacks, it may reach any intermediate performance degradation rate it wishes.[5]

If we make a crude assumption that an attack is detectable if more than a fraction of the system capacity is lost, then the attacker can incur the maximum possible damage subject to being undetectable, while using only a relatively small attack intensity (yet larger than individual server capacity) [7].

Consider the case of a system with a very large number of servers each with small capacity, i.e., each server can be a virtual machine. Then the attacker can destabilize this system with a very small attack intensity. In particular, it

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

can control exactly the volume of the inflicted damage. used.[5][10]

IV. SYSTEM ARCHITECTURE

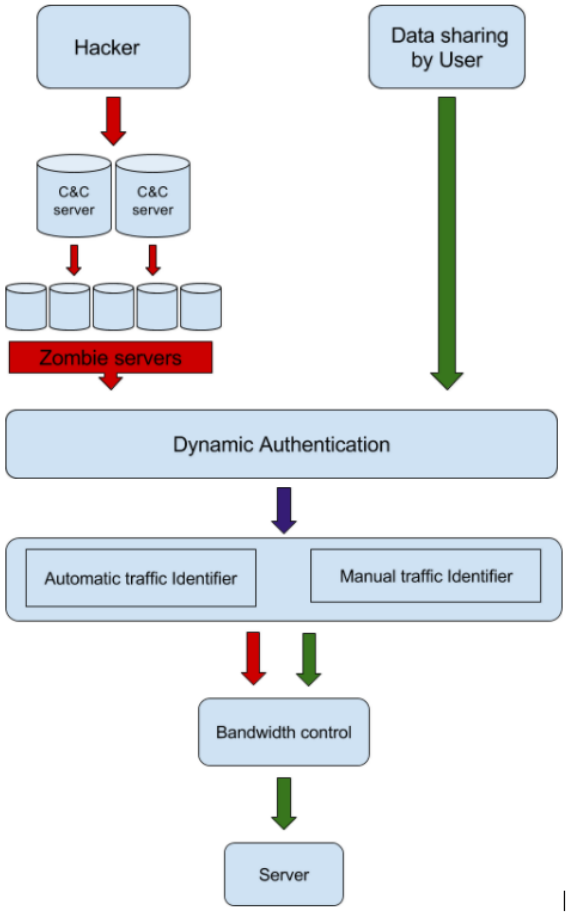

Fig. 1. System Architecture of the DaR Policy System

The System Architecture of the DaR policy system consists of the 4 modules.

1) Server
2) Data Sharing by User
3) Hacker
4) Unknown User

The users or nodes involved in our projects are Sender, Intermediate and Receiver. In order to send file, the sender has to find out the list of nodes which are connected with the sender. From that available list he can choose receiver. Then the sender has to analyse the performance of each and every node which is connected with the sender. The performance analysis list will return the priority based result so that sender can choose the intermediate to send the file. For that the User has to do the Authentication Login, which is also an attempt of the Attacker or Hacker to penetrate the server. The attacker will have few C&C Servers who will further have various Zombie Servers or Botnet, whose attempt is to create fake traffic and flood the server.

This system has a guaranteed throughput region where it identifies the Manual traffic and Auto-generated traffic. After it identifies the Auto-generated traffic, the system will filter them in the Bandwidth Control and only allow the Manual traffic to get through it and connect to the server, thereby shielding the server from the furtive attacks of Distributed Denial-of-Service.

V. RESULTS

The Hackers use the physical IP address of the authenticate user to send requests to the server and these requests become hard to notice as they use to be in low-rate. We shall show you the login and registration process of our system below which will be followed by how the system identifies the illegitimate requests even though they are in low-rate.

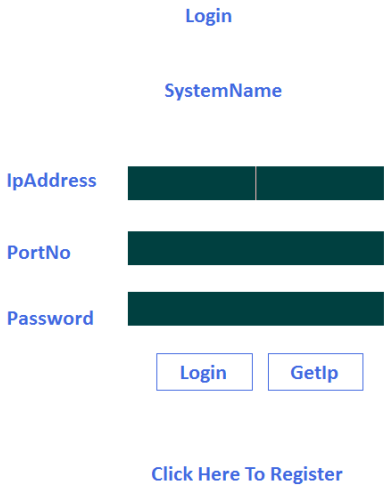

Fig. 2.Login Page

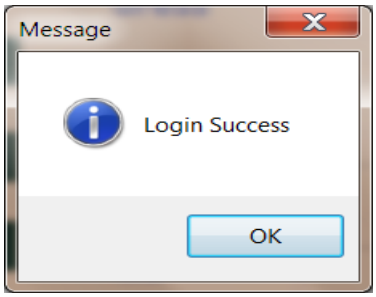

Fig. 3.On successful Login

For Login in the User has to provide the IP Address (which can be generated by click “GetIP”), Port number and an unique password, which is used while registering to the system.

Below is the demonstration on how this System will be able to provide shield to the server from DDoS attacks. So, we shall manifest the system from four perspectives i.e.

1) Server
2) DataShareView
3) Hacker
4) UnknownUser.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

And provide the glimpse of what actually happens.

Fig. 4.Main screen of the System

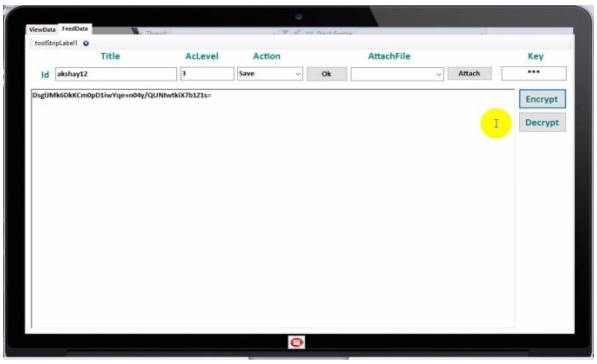

Fig. 5. Data encryption for better Security by the User

Fig. 6.When Attackers attempt to Hack.

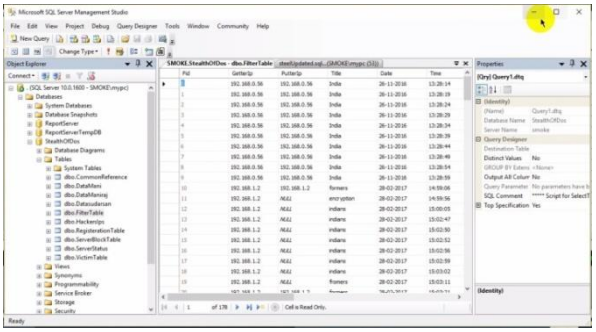

Fig. 7.DataBase

Fig. 8. Filtering the Automatic and Manual Traffic.

VI. CONCLUSION

We obtained adequate conditions for the guaranteed throughput region for a system under a furtiveDDoS attack. This system will provide a clear idea on how a system can be designed in such a way that no attackers can perform any kind of attack to degrade the system. In case where the malicious controller uses a simple static routing policy, JSQ is proven to be a desirable defense policy. We show that the damage can be severe if the malicious controller performs non-stationary dynamic routing. Moreover, we exemplify the interaction between JSQ and JLLQ policies .It is found by simulations that JSQ is not a maximally stable policy in the sense that depending on the attacking policy it can be strictly outperformed by other legitimate policies.

REFERENCES

[1] The Risk vs. Cost of Enterprise DDoS Protection. Arbor Networks, White paper.
[2] Akamai. The state of the Internet.Technical report, 2013.
[3] E. Altman and N. Shimkin.Worst-case and Nash routing policies in parallel queues with uncertain service allocations.University of Minnesota, IMA Preprint No. 1120, 1993.
[4] M. Bramson. Convergence to equilibria for fluid models of head-ofthe-line proportional processor sharing queuing networks. QueuingSystems, 23(1-4):1–26, 1996.
[5] Q. Duan, H. Jafarian, E. Al-Shaer, and J. Xu.ModelingDDoS attacks bygeneralized minimum cut problems. Technical report, arXiv: 1412.3359,2014.
[6] M. Ficco and M. Rak.Intrusion Tolerance of Stealth DoS Attacks toWeb Services. In Information Security and Privacy Research, volume376 of IFIP, pages 579–584. Springer, 2012.
[7] R. D. Foley and D. R. McDonald. Join the Shortest Queue: stabilityand exact asymptotics. Ann. Appl. Prob., 11(3):569–607, 2001.
[8] M. Guirguis, A. Bestavros, and I. Matta. Reduction of Quality (RoQ)Attacks on Dynamic Load Balancers: Vulnerability Assessment and DesignTradeoffs. In INFOCOM, pages 857–865, 2007.
[9] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang.Reduction of Quality (RoQ) Attacks on Internet End-Systems. In INFOCOM, pages 1362–1372, Mar. 2005.
[10] A. Hussain, J. Heidemann, and C. Papadopoulos.A Framework for Classifying Denial.