Impact factor: 4.046 e-ISSN: 2456-3463
National Conference on Quality Up-gradation in Engineering, Science and Technology (NC-QUEST018)
Organized by College of Engineering and Technology, Dhanamgaon Rly-444709
International Journal of Innovations in Engineering and Science, Vol. 3, No.6, 2018
www.ijies.net

Distributed Denial of Service Attack and Its Prevention

Swapnil Moreshwarrao Thakre, CSE 3rd Year DESCOET Dhamangaon [1]

Abstract – *DoS, or denial of service attack, is an attempt to suspend the services of an online host by flooding the target with excessive and unnecessary requests, causing them to overload and prevent legitimate requests from being fulfilled. DDoS, or distributed denial of service, is where the flooding occurs from multiple sources of attack, rather than just a single computer or IP address.*

Keywords- DoS, Mitigation, ACK, Email Bombing,slowloris

INTRODUCTION

In a DoS attack, an attacker with malicious intent prevents users from accessing a service. He does so by either targeting your computer and its network connection, or the computers and network of the website that you are trying to use. He can thus prevent you from accessing your email or online accounts. Imagine a situation, where you are trying to log into your Internet Banking account for online transaction activity. However, as strange as it may seem, you are denied of an access to the bank's website, in spite of having a swift internet connection. Now there could be two possibilities – either your internet service provider is down or you're under a DoS attack!

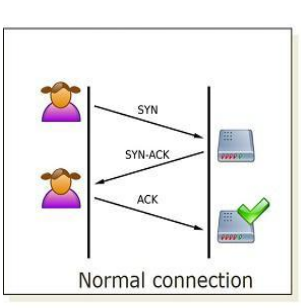
In a DoS attack, the attacker sends out a flood of superfluous requests to the main server of the website in question, which basically overloads it and blocks out any further requests before the capacity is retained back. This causes a denial of the incoming legitimate requests for this website and consequentially, you're the victim.

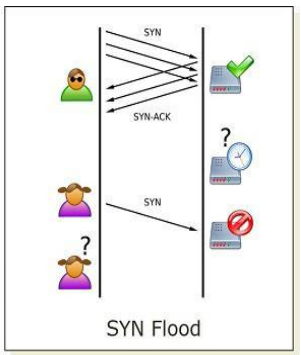
Other ways of attacking may involve preventing a particular person from accessing a certain website, obstructing the connection between two machines at the server ends, therefore, disrupting the service etc. Some attackers also act on an another kind of DoS attack – **Email bombing** in which a lot of spam emails are generated and flooded into one's Inbox so that any further request to the mail server are debarred. This can happen widely, even on the email account provided to you by your employers, not to mention the public mail services like Yahoo, Outlook etc. You can even get deprived of receiving any further legitimate emails as your allotted storage quota will be filled up.

A. Types of Dos Attack:

1. SYN FLOOD:

SYN Flood takes undue advantage of the standard way to open a TCP connection. When a client wants to open a TCP connection with the server's open port, it sends out a SYN packet. The server receives the packets, processes it and then sends back a SYN-ACK packet which includes the source client's information stored in Transmission Control Block (TCB) table. Under normal circumstances, the client would send back an ACK packet acknowledging the server's response and hence opening a TCP connection. However, under a potential SYN flood attack, the attacker sends out an army of connection requests using a parody IP address which are treated as legitimate requests by the target machine. Subsequently, it gets busy processing each one of these





and makes an attempt to open connection for all of these malevolent requests.

Fig 1: SYN Flood

Under normal circumstances, the client would send back an ACK packet acknowledging the server's response and hence opening a TCP connection. However, under a potential SYN flood attack, the attacker sends out an army of connection requests using a parody IP address which are treated as legitimate requests by the target machine. Subsequently, it gets busy processing each one of these and makes an attempt to open connection for all of these malevolent requests. This causes the server to keep waiting for an ACK packet for each connection request which actually never arrives. These requests quickly fill up the server's TCB table before it can time any connection out and thus any further legitimate connection requests are pushed into the waiting queue.

2. HTTP FLOOD:

This is most commonly used for attacking web services and applications. Without putting much emphasis on high-rate network traffic, this attack sends out a complete and seemingly valid HTTP POST requests. Designed specifically to exhaust the target server's resources, the attacker sends out a number of these requests to make sure the further legitimate requests are not pulled through by the target server while it is busy processing the fake requests. Yet so simple but it is very difficult to distinguish these HTTP requests from the valid ones as the content of Header seems admissible in both the cases.

2. DISTRIBUTED DENIAL OF SERVICE ATTACK (DDOS):

Distributed Denial of Service or DDoS attack is like the decorated officer in this gang. Much sophisticated by levels above normal DoS attack, DDoS generates the traffic on the target machine via more than one computer. The attacker controls several compromised computers and other devices at once and distribute the task of flooding the target server with traffic, heavily eating on its resources and bandwidth. The attacker can also use your computer to launch an attack on another computer if there are lingering security issues. Now, as obvious as it is, a DDoS attack can be much more effective and real when comparing to DoS. Some websites which can easily handle multiple connections can be brought down easily by sending numerous simultaneous spam requests. Botnets are used to recruit all sort of vulnerable devices whose security can be compromised by injecting a virus into them and signing them up for Zombie army which the attacker can control and use them for a DDoS attack. Might end up doing somebody's dirty work and never know about it.

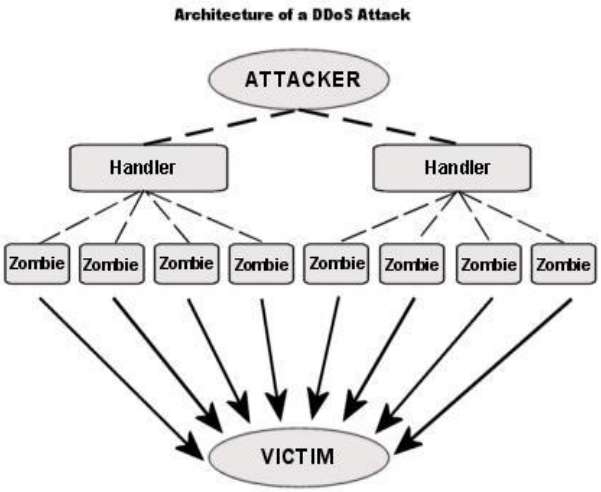


Fig2: DDoS Attack

B.Motives of a DoS attack:

Not all cases of mitigated or failed connection is due to DoS; however, it's a distinct possibility if the affected service has garnered a great deal of publicity lately. This is because perpetrators frequently use DoS software to temporarily disable websites, network providers, web-based services and video game hosts in order to make a statement, gain publicity, exact revenge, or simply as a show of power. In the case of the ABS being attacked, the Census was a matter of high publicity. As with other high-profile web-based situations, such as the launch of an anticipated site or video game, the attention of unsavory individuals was captured. Attacks were administered for what was guessed to be no other reason than "because they can". Denial of service attacks have been used for benevolent causes as well, shutting down criminal enterprises or even singular IP addresses of criminal perpetrators. Hacktivist group Anonymous has gained traction for administering DoS attacks against organizations and people that are thought to have engaged in illicit activities.

C. common DDoS attack tools:

1. TYPES OF GRAPHICS:

i. Low Orbit Ion Cannon (LOIC) - This is a DoS script that disrupts a target server by sending a large number of TCP requests or through a UDP food. It is very user friendly and doesn't require extensive knowledge. LOIC has been used in a number of notable attacks, including those

targeting the Church of Scientology and the Recording Industry Association of America.

ii. High Orbit Ion Cannon (HOIC) : Created as a LOIC replacement, this script was designed to launch a DDoS attack using a minimal amount of perpetrators. It works by executing a HTTP flood against a target server until it crashes.

iii. Slowloris: This is an attack script designed as a simple way for a single computer to take down a server. It works by continuously sending partial HTTP GET requests to its target. The server opens more and more connections in anticipation of receiving the completed requests, which never occur.

iv. Torshammer: This is a slow-rate, application layer DoS attack script that uses the TOR network to mask its origin. TOR is a network of servers that routes user traffic through a series of tunnels instead of establishing direct connections. It's used to increase online privacy.

.

D. Server Affected:

1. Apache (1.x & 2.x)
2. dhttpd
3. Goahead web server

Web server's that work on an event based architecture like nginx are not affected by a slowloris attack.
It seems that IIS is also is not affected by a slowloris attack.

E. Working of Slowloris Http Dos:

An in depth understanding of http request and response is very much necessary to comprehend this attack tool. Because it exploits a vulnerability in the web server(which was purposely made by the authors for different advantages like serving requests for a slow connection) which wait for a complete header to be received. Apache & some other web server's have a mechanism of timeout. An Apache web server will wait for this specified timeout duration for the completion of a request(if the request was incomplete).This timeout value is by default 300 seconds, but is modifiable. This timeout value is very much useful if a website serve's large files for download through http(because it maintains an active http connection of a slow client without breaking the download).Another important fact to note here is that the timeout counter is reset every time the client sends some more data(so the timeout count will start again from 1).But imagine a situation if somebody purposely send partial http requests and reset the timeout counter of each request by sending some bogus data very frequently. That's exactly what slowloris does. It sends partial http request with bogus header's. Once all connections are consumed by sending partial requests, it keeps on maintaining the connection's by sending request data and resetting the timeout counter. A complete GET request looks like something below.

1 GET / HTTP/1.0[CRLF]
2 User-Agent: Wget/1.10.2 (Red Hat modifed)[CRLF]
3 Accept: */*[CRLF]
4 Host: 192.168.0.103[CRLF]
5 Connection: Keep-Alive[CRLF][CRLF]

F. CRLF Request

CRLF stands for CR (Carriage Return) and LF (Line Feed). This character is an entity which is non printable, used to denote end of the line. Even when you are typing on a text editor the editor puts a CRLF at the end of a line when you want a new line after that. And two CRLF characters together is used to denote a blank line. In the above shown GET request there are two CRLF characters at the end of the "Connection" header(which means a blank line). In http protocol, a blank line after the header's is used to represent the completion of the header. Slowloris tool takes advantage of this in implementing its attack. It does not send a finishing blank line, which indicates the end of the http header. Some web server's give higher priority to those requests which are complete in its header's. This is the reason why IIS is not affected by a slowloris attack. An incomplete request send by the slowloris script is shown below. This below snippet is taken from the slowloris script.

"GET /$rand HTTP/1.1\r\n"

"Host: $sendhost\r\n"

"User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;

1 Trident/4.0; .NET CLR 1.1.4322; .NET
2 CLR 2.0.503l3; .NET CLR

3 3.0.4506.2152; .NET CLR 3.5.30729;

4 MSOffice 12)\r\n". "Content-Length: 42\r\n";

In the above snippet shown \r\n is used to denote carriage return and newline in perl. Two consecutive "\r\n\r\n", should be there to denote a blank line, which is not there. So that an incomplete header in HTTP. Slowloris is mostly not noticed by IDS(Intrusion Detection system's), because it does not send a malformed request, but a legitimate request to the web server. Hence it bypasses most of the IDS system's out there. Slowloris works by the principle of consuming all available http connections on the server. Hence it takes time if its a high traffic web site, and are already connected by a number of clients. Because in that case slowloris needs to wait, for http connections to become available(because other clients are connected to it and are being served) An important funny thing with slowloris attack is that, as soon as the attacker stops running the script, the website will become back online. Because the connections will automatically be closed by the web server after some time(after the timeout interval).

G. DDoS Mitigation

1. Use Hardware Load Balencers that accepts only full http connections.

Using hardware load balencer's with an http profile configured will be the best method to stop such an attack. Because the loadbalencer will inspect the packet's and will forward only those http request to the web server which are complete.If you are using a F5 based BIG-IP Load Balencer i recommend reading the below link for mitigating slowloris attacks. Mitigate Slowloris on BIG-IP F5 Load Balencer. Other Load balencer's like the below ones also can be configured with http profile to mitigate such an attack.

Citrix NetScaler

Cisco CSS

2. **Protect your web server by using IPtables by limiting connections from a particular host**

You can certainly limit the number of connections with the help of iptables to port 80. For example if suppose i want to block

iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 30 -j DROP

3. **Configure the timeout directive in apache**

Although this is not at all a good solution, you can still increase the rate with which your web server will reap inactive connections. You can simply modify the timeout directive in /etc/httpd/conf/httpd.conf file.

Reducing it to a lower value will at least make the attack difficult(but still the attack can take down the server, by increasing the number of requests).

4. mod_antiloris apache module

Another good solution that i tested is an apache module called as mod_antiloris. This module can be installed using the below steps

1

2[root@localhost ~]# wget http://sourceforge.net/projects/mod-

3 antiloris/fles/mod_antiloris-0.4.tar.bz2/download

4[root@localhost ~]# tar -xvjf mod_antiloris-0.4.tar.bz2 mod_antiloris-0.4/ChangeLog

5 mod_antiloris-0.4/mod_antiloris.c

6 [root@localhost ~]# cd mod_antiloris-0.4

7 [root@localhost mod_antiloris-0.4]# apxs -a -i -c mod_antiloris.c

5. **Kernel configuration**

Beyond this shell script, the number of SYN_ACK retries can also be lowered, so that the kernel closes SYN_RECV state connections earlier. The parameter net.ipv4.tcp_synack_retries defaults to 5 SYN_ACK retries, which leaves SYN_RECV-state connections open for 3 minutes. You can reduce this so that these hanging connections will close sooner. I have used a value of 1, just for demonstration. This allows a time interval of 10 seconds before the connection is closed. Set the parameter using the following method. Add or edit the following line in

/etc/sysctl.conf:

net.ipv4.tcp_synack_retries = 1

Commit the changes made using the sysctl command as root as follows:

sysctl –p /etc/sysctl.conf

Verify the changes in effect using the following command: root@ubiserv:~# cat /proc/sys/net/ipv4/tcp_synack_retries 1

CONCLUSION

From the review of the above papers and completely different options, it will be all over that a lot of completely different techniques will be wont to detect Distributed Denial of Service(DDoS) attack completely with different options. DDoS may be a reasonably DOS attack during which multiple compromised systems, that area unit oftentimes infected with a Trojan, area unit wont to goal one machine inflicting a Denial of service (DoS) attack. Hence, the detection must be wiped out its earlier stages. There is a constant research happening in this field. right here, an strive is done to research and apprehend a number of the strategies used until now for the detection and classification of DDoS assault through the usage of proposed algorithms and the methods proposed within the research papers.

REFERENCES

[1] *Qiao Yan, F Richard Yu, Qingxiang Gong and Jianqiang Li, "Software Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges" IEEE 2015*

[2] *I Gde Dharma N.,M. Fiqri Muthohar, AlvinPrayuda J. D., Priagung K., Deokjai Choi, Tim based DDoS Detection and Mitigation for SDN Controller", IEEE 2015.*

[3] *Bharti Nagpal, Pratima Sharma, Naresh Chauhan and Angel Panesar, "DDoS Tools: Classification, Analysis and Comparison" IEEE 2015*

[4] *Bharat Rawal, Anthony Tsetse and Harold Ramcharan, "Emergence of DDoS Resistant Augmented SplitArchitecture", IEEE 2013.*