Available online at www.elixirpublishers.com (Elixir International Journal)

Adoc Network

Elixir Adoc Network 38 (2011) 4169-4173

ISSN: 2229-712X

CoDe – an collaborative detection algorithm for DDoS attacks

M.Karthikeyan

Department of Computer Science, Thiagarajar College of Engineering, Madurai.

ARTICLE INFO

Article history:

Received: 28 June 2011;

Received in revised form:

22 August 2011;

Accepted: 27 August 2011;

Keywords

DDoS,

Distributed Change Point Detection,

Statistical methods,

UDP Flood attack,

Ranking scheme.

ABSTRACT

Security threats for the network services have been constantly increasing day by day. Distributed denial of service (DDoS) attack is one such kind of security threat which involves multiple systems generating a large amount of traffic towards a target machine and thereby making any service from that target machine or server unavailable to its clients. This threat by nature needs no control over the target system. Traditional methods of detecting DDoS attacks are mostly centralized in nature and highly disadvantageous. To overcome the disadvantages of those schemes, we propose a distributed methodology which involves installing the attack detectors at various parts of the network. Each router in the network will monitor the traffic flowing through it and if any anomaly in the traffic pattern is detected, it will raise an alarm to the nearby routers. The alarm propagates to all the routers through which the attack flows. By this way a tree like construct is made, which will have information about number of alarms raised and the path of the attack flow. If the construct shows any converging pattern then it is declared as DDoS attack.

Introduction

Network resources are more vulnerable to various types of attacks. Denial of service attack is one among the varied types of security threats identified as of now[9]. In general, out of the many types of attacks identified, denial of service is the one that exploits inter connectivity of the computer systems and is also the one which can be easily deployed. It purely concentrates on overloading a computer which is busy serving an essential service. By this, the targeted machine is made to do less useful things and thereby the essential service may not get CPU cycles and this may be interrupted. Thus it becomes essential for the network administrators to safe guard their systems and to ensure that a smooth and uninterrupted service is available for its clients. This job however cannot be manually ensured, since a quick and immediate response from the victim side is required [10]. Thus the automation of detection and mitigation of such type of attacks has become essential to effectively stop the perpetrators from causing any major damage to our system environment.

Background

Distributed denial of service attack (DDoS) is an enhanced and distributed version of denial of service attack [3]. DDoS attacks involve a large number of attacker machines which targets a single victim machine and generate a surge of traffic towards it. Since the evolution of Internet, this type of DDoS has been frequently prominent, just to stop any competitor's essential cum popular service, so that the other party may get an edge over them.

The prominence of the DDoS in Internet may raise a question that how come an attacker will be able to acquire some 100,000 computers to probe an attack? The answer is they will not actually acquire that computer by purchasing, but they enslave some 100,000 computers already connected to the internet. These enslaved computers are generally called as 'Zombies'. Thus a master attacker commands umpteen numbers of zombies which in turn target a victim. The command over zombies does not involves any type of hacking into those systems, although there are methods for performing that, but without having any knowledge about the zombies the master attacker can claim control. There are several types of attacks which accomplishes it [8].

The DDoS attacks are not the one which lasts for days or months. In real time the DDoS attacks generally lasts for 10 to 15 minutes. Within this period the target is severely damaged and so the recovery time is usually longer. Given this short duration, it is highly essential to automate the detection and also the mitigation process; since it is highly unlike that the network administrator can manually control the situation.

There are many detection schemes available in the market. Among those most popular types of detection are signature based detection and statistics based detection.

In the Signature based detection a database of signatures will be maintained for each type of attack, developed based on the history [12].

If any attack is found to match any signature then the system identifies that an attack has been established. In the statistics based approach [2], [5] the traffic statistics will be constantly monitored. Based on the history, a threshold is developed and at any particular time if any flow exceeds the threshold then it will be established as attack.

DDoS mitigation techniques are plenty [7], [11], [13]. It ranges from simple packet dropping [6] to more sophisticated techniques [14].

Some of the simple techniques like changing the IP address of the target system is also used [16]. By this the packets are diverted to old IP address which does not exist. We can also configure firewall such that only authentic and most reliable IP can access the system.

But this method involves identifying reliable sources. Sometimes a service level agreement will be established with the client and the IP addresses from that clients are alone considered reliable and the other sources are either blocked or rate limited.

Tele:

E-mail addresses: affablekarthik@gmail.com

A collaborative detection algorithm for DDOS attacks

Collection of statistics about the traffic pattern is an essential pattern to identify any anomaly in the traffic. DDoS detection cannot be done without the history of the normal traffic pattern. Decisions have to be made on the basis of this historical database. Hence IP Traffic information are collected and stored in a database using Net flow collector. The details about the traffic are stored per flow basis.

The proposed DDoS detection algorithm has a stastical analyzer program called 'stats analyzer' which will query the historical database. Based on the historical information and the current flow information, the statistical analyzer program will provide a rank for each flow. Based on the rank information and the current traffic information, each flow is classified into different categories. If any flow is found to be abnormal, then an alert is generated to 'detection program'. The alert raised by the 'statistical analyzer program' module to the 'detection program' module consists of the parameters as shown in Table I.

A flow is defined by a (src, dest) pair. Packet count is the number of packets routed by the router within a particular time interval. The mean packet count is updated constantly for each time interval. Rank is the present rank assigned to the particular flow. The mean rank is also updated constantly for each time interval. The time interval is preset as 2 minutes. Once in 2 minutes the statistical analyzer queries the database and fetches the new Netflow details.

Ranking Scheme

The flows are obtained for a particular time interval. Then the flows defined by (src, dest) pair, are sorted in the ascending order based on packet count. Insertion sorting method is adopted for sorting, since the number of flows is generally too few to consider in quick sort or merge sort. Then the flows ranked appropriately with the lowest rank assigned to flow with lowest packet count and the highest rank assigned to flow with highest packet count. This value is preset by the network administrator based on the previous analysis of traffic surge, after determining the lowest rank the rank is incrementally assigned to other flows based on ascending order of packet count.

Threshold based Flow Classification

At each interval, the mean rank of the flows will be updated. Based on the mean rank of the flow, current packet count and mean packet count any one of the three threshold levels is assigned. Based on these threshold levels the flows are classified into four types as Normal flow, Suspicious flow, Critical flow, DoS flow as shown in Table II.

For each interval period, after assigning ranks and based on the threshold classification of the flows are made, the statistical analyzer searches for flows other than 'normal'. If any of the flows has been termed other than normal, then an alert has to be raised to the local DDoS program. For this the program opens a UDP socket connection with the DDoS detection program and sends a packet indicating the details about the flow.

As the mean rank and mean packet count is considered till the $(i-1)^{th}$ interval a question may arise such that what if no statistics is available for a flow, i.e., what if a new flow arrives? For a new flow it is necessary to consider all flows with same destination ip as that of the Netflow. The overall average of their mean rank and overall average of their mean packet count is calculated and then the classification is made. By this it can be ensured that if spoofing of the source ip is done, then it is highly likely that at the first instance itself a surge of packets will arrive. It cannot be assumed to be legitimate just because there is no information about this source. So it will be wise to consider other normal flows with same destination ip and how their statistics are and determine whether this new flow exceeds the number of packets which usually flows through this router for this particular destination ip address.

If no flow has the same destination ip as that of the new flow, then classification cannot be made, since no assumptions can be made. In this case we are made to assume that this flow is legitimate however high the traffic may be, since there is no information about the traffic to this destination ipaddress. This is in no way a disadvantage since no analysis can be done without statistics or at the least a miniscule of information. So, the flow will be termed as normal and mean rank is set to current rank and mean packet count is set to current packet count. In the subsequent time intervals these values will be considered for analyzing the flow.

Methodology

An overall view of the proposed system architecture is provided in the fig.1. Here each network is collaborated with each other. Typically a network will have router which will be an entry point to the network. A dedicated server has to be setup for traffic information collection. Netflow protocol is used to collect traffic statistics.

The router is first configured to export Netflow details to a dedicated server. Port information of the receiving server is also given while configuring. Next step is to deploy a Netflow collector in the server which will look out for Netflow details from the router, in the particular port. A collector called "neye"is used [17]. The deployed Netflow collector will transfer details into human readable form and stores it in database. MySQL database has been used. This setup has to be established in each and every network so that statistics collection is made possible.

The stats analyzer program will query the database and check for any abnormalities in any flow. If it finds any abnormalities, then it will alert its local DDoS detection program. The DDoS detection program will communicate with neighbouring network's detection program and come to a conclusion of any attack. The communication is through UDP protocol. The details of the suspicious routers are used to form tree representing attack scenario.

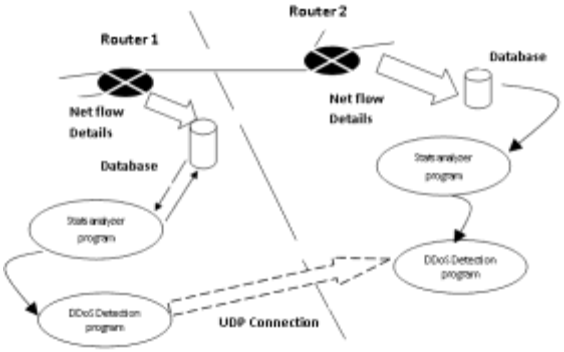

Fig. 1 System Architecture

UDP Attack

UDP Flood Attack is one of the attacks causing host based Denial of Service [4][15]. UDP is a connectionless protocol and it does not require any connection setup procedure to transfer data. A UDP Flood Attack is possible when an attacker sends a UDP packet, it will determine what application is waiting on the destination port. When it realizes that there is no destination unreachable to the forged source address. Then enough UDP

packets are delivered to ports on victim, the system will go down.

Attack generator

The UDP flood program is written in C language and forwards the given number of packets continuously on the target victim. As the real attacking scenario consists of normal and flood packets, a python script is written to control the number of attack packets and normal packets to be sent to victim. The script also maintains the time delay within the normal traffic. The attacking physical machines must be running Unix type OS and requires GCC compiler. The machine should also have python to run the script.

Attack Environment

Though the above attacking tools are well suited for real time environment, they cannot be used in the virtual network created. This is because the emulated routers can support only a bandwidth of 1 kbps. Thus the UDP flooding program can be used where the traffic can be controlled to give similar effects as the attacking tools.

A UDP socket client program is written in C. The target address and any opened UDP port address of the victim machine are given. This UDP port on victim act as a server providing all services. The client program sends bogus messages to the victim port for the specified number of times. Thus the victim providing service on the UDP port is attacked and performance of the victim is degraded.

The attack environment consist of 4 virtual routers as shown in fig. 2.The routers are established by Dynamips which is a router emulation software for establishing virtual connections between routers and different networks[18].

Fig. 2 Network Topology

Statistics Analyzer Program Module

A proper analysis of the traffic pattern should be made before implementing a detection scheme. The analysis should be made based upon the present and past flow of traffic across the network. To get the data from routers, there are two ways:

SNMP only provides the device based statistics like temperature, power, etc [1]. It is impossible to get interface based statistics like amount of packet flowing in each interface, source, destination IPs, traffic aggregation, etc. It requires additional MIBs to be installed in the manager as well as in the agents. RMON is one kind of MIB that specifically collects interface based statistics. RMON2 provides more detailed information on the traffic flow. But installing this RMON in the SNMP manager and including it in the IOS Software is extremely difficult [19]. Also, SNMP involves the usage of traps and information for sending the statistics. This takes a heavy traffic load by itself, needs the CPU intensive calculations at the routers and hence is disadvantageous to send the details of traffic flow when an attack is taking place on the network. Whereas, netflow overcomes this disadvantage by making it easy to configure and the traffic details are sent as UDP datagrams to the netflow collector, when properly configured. Netflow provides more detailed information of the traffic pattern. Being integrated with Cisco IOS, it provides better performance and takes lesser CPU computations at the routers. It sends details only as bursts and fixed intervals, thus making it ideal for the traffic conditions during the time of DDoS attack. Hence Netflow is used rather than the traditional SNMP as the statistical collection protocol.

DDoS Detection Program Module

The DDoS detection program is the one which communicates with the neighbouring routers. 'DDoS' detection programs conclude if any attack has been established. This program will keep track of routers which have raised alerts and determine whether a considerable amount of routers have raised concern. Based on these information available it will come to a conclusion that whether a DDoS attack has been established to one of its internal network or not.

Alert Specification

The DDoS detection program may get alerts from 2 sources. One is from local 'stats analyzer program' and another is from neighbouring 'DDoS detection programs'. The alerts are defined in the form of a class. The components of the class are mentioned as in Table III.

Since the detection is done from a number of routers raising alerts, the router list keeps track of all the routers which raised alerts. The traffic type tells about the flow type. 'Destination ip' specifies the ipaddress to the flow. 'Source' specifies the source of the packet. It may be local stats analyzer or name of a neighbouring router.

Communication between Routers

The information cannot be sent over a network as such. Serialization has to be done to make it compatible to be sent over a network. Serialization converts the class into a sequence of characters defining clearly all the attributes of the class. A python package called 'cpickle' is used to serialize and de-serialize the object to make the communication possible.

The communication is through UDP packets. Suspicion may arise whether the UDP packet will reliably reach the neighbouring routers during the attack period. To have the communication reliable, a separate VPN can be established between the communicating units. But to keep the process simple there is no need to consider about establishing tunnels. UDP communication has been chosen for simplicity. The detection is done as shown in Table IV

Experimental results

The detection algorithm should be relevant and robust for different scenario of attack. Generally DDOS are careful extension of highly sophisticated attack plans. It does not generally consist of careless flooding of packets, as it may turn out to be a criminal offense. So, consider two cases by which the attack may proceed.

The first case is that, the attacker may have only small number of systems acting as zombies. This is because he may have had less time to gather zombies or fewer sources to do that. So generally he will try to plan an attack with this small number of systems each generating large number of packets. In this case, our algorithm detects those abnormal flows as critical and DoS and collaboratively concludes that there is an attack.

Suppose assume that the attacker has enough time and resource to plan and execute the attack. So for a long period of time he will do the job of gathering zombies (say for a period of one year or so gathering some 100,000 systems). So, when the time has ripened to generate attack he may try to generate less

traffic from all zombies so that they may appear as normal and legitimate traffic. At this juncture all the intermediate routers, which are deployed with our system may not notice those attack as abnormal. But at our end router, due to large aggregate traffic the alerting system may now show up an alert as DOS flow and so immediately an alarm is sent to the network admin to look into the situation. So even though those attack traffic escapes the intermediate router, it will get caught at the end router. This situation may seem analogous to a centralized system. Thus the proposed system can adapt in different way for different situations.

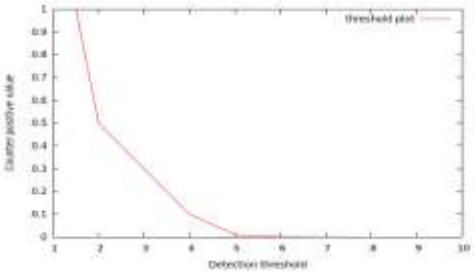

Fig.3 Effects on the threshold on false-positive rate in detecting UDP Flood attacks

Fig.3 shows the false positive alarm rate against the 'DDoS detection program' threshold. The number of alert generated by random fluctuation in normal traffic is small. With a DDoS detection program threshold >5, the false positive rate drops to less than 1%. However in a highly distributed attack only after sufficient number of attack flows were merged, the deviation is detected by the routers.

Front End Web Interface

A front end web interface has been built on the LAMP (Linux Apache MySQL Python) Stack [20]. The web interface is simple and display different images depicting the path through which attack had taken place. When DDoS attack takes place it is shown specifically for each router and each destination as a tree as shown in fig. 4.

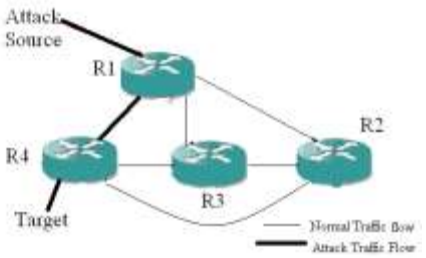

Fig. 4 Attack Tree

Conclusions

The proposed work strikes the idea of distributed version of detection methodology. Implementing multiple threshold levels makes the system to be more sensitive to anomalous traffic flows. Since traffic statistics is on per flow basis, the clear indication of attack traffic flows can be made. Also, the project portrays how the distributed methodology of detection provides a secure environment to the entire network domain involved rather than any particular network domain.

This work has some extensible parts which may make the system much robust. The extension that can be done is to the ranking methodology. Presently the system requires the network admin to set the initial rank that has to be aware of the traffic fluctuations in the router for recent times. This part can be automated so that the overhead on network admin reduces. For this, a particular parameter can be considered (e.g., CPU usage) and fluctuations in the traffic pattern can be correlated to the parameter considered and a conclusion of initial rank can be arrived. This may be termed as a training period for the system, when the system involves monitoring the parameter and traffic pattern to conclude the ranking scheme. The detection scheme algorithm involves a communication module. This module has been carefully analyzed and charted out for minimizing the redundancy of generating alerts, so that the traffic due alert generation is low. But still, there may be some redundant alert packet that may be generated. This can be further reduced.

Acknowledgment

We wish to acknowledge the Smart and Secure Environment Lab for providing the Environment for the attack generation.

References

[1].W.Stallings., "SNMP and SNMPv2: the infrastructure for network management", *in the IEEE Transaction on Communications,* Vol 36, 2002.

[2]. S. Bellovin, J. Schiller, and C. Kaufman, "Security Mechanism for the Internet", *RFC 3631*, Internet Eng. Task Force, 2003.

[3]. H. Wang, D. Zhang, and K. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," in the *IEEE Trans. on Dependable and Secure Computing,* Vol. 1, Oct.-Dec., 2004.

[4]. J. Sommers and P. Barford, "Self-Configuring Network Traffic Generation", *in Proc. of ACM Internet Measurement Conference, Taormina, Sicily, Italy*, Oct. 25-27, 2004.

[5]. S. Chen and Q. Song, "Perimeter-Based Defense against High Bandwidth DDoS Attacks," in the *IEEE Trans. on Parallel and Distributed Systems,* Vol. 16, No. 6, June 2005.

[6]. Yu Chen, Yu-Kwong Kwok, and Kai Hwang "MAFIC:Adaptive Packet Dropping for Cutting Malicious Flow to Push Back DDoS Attacks", *in Proc.of the 25th IEEE International Conference on Distributed Computing Systems* (ICDCSW'05), 2005

[7]. J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, and P.Reiher, "Benchmarks for DDoS Defense Evaluation", in the *Proc. Of the Milcom 2006,Oct 2006*

[8]. K.M.Elleithy., D.Blagovic., W.Cheng., and P.Sideleau., "Denial of Service Attack Techniques: Analysis, Implementation and Comparison", Systemics, Cybernetics and Informatics Vol. 3,No 1, 2006.

[9]. S.Lin., T.Chiueh.," A Survey on Solutions to Distributed Denial of Service Attacks", Stony Brook University, 2006.

[10]. S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, "DDoS resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection", in the IEEE INFOCOM 2006, Barcelona, April 23-29, 2006.

[11]. S.Meenakshi.,S.K.Srivatsa.,"Comprehensive Mitigation Mechanism Against DDoS Attack-A Comparitive Study ",Asian Jornal of Information Technology5(7) page 691-695,2006.

[12]. Yu Chen, "Collaborative Detection of DDoS Attacks over Multiple Network Domains", *in the IEEE Transaction on Parallel and Distributed Systems, 2007*.

[13]. B.H.Song., J.Heo.,*and* C.S.Hong, "Collaborative Defense Mechanism Using Statistical Detection Method against DDoS Attacks", IEICE TRANS. COMMUN., VOL.E90–B, NO.10 Oct 2007.

[14]. M.Sachdeva., G.Singh., K.Kumar., and K. Singh., "A Comprehensive Survey of Distributed Defense Techniques against DDoS Attacks", International Journal of Computer

Science and Network Security IJCSNS, VOL.9 No.12, Dec 2009.
[15]. A.Singh., D.Juneja., "Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks", International Journal of Engineering Science and Technology Vol. 2(8), page 3405-3411,2010.
[16]. P. Jain., J.Jain., and Z. Gupta.," Mitigation of Denial of Service(DoS) Attack", International Journal of Computational Engineering & Management IJCEM ,Vol 11 Jan 2011.
[17]. www.neye.unsupported.info
[18]. www.dynagen.org
[19]. Net-SNMP Tool, http://www.net-snmp.org.

Table I Parameters in the alerts during attack traffic

Source- Destination Pair	Packet Count	Mean Packet Count	Rank	Mean Rank
10.7.3.1, 10.4.3.1	16	16	4	4
10.4.3.2, 10.4.3.1	2	9	3	4
10.4.3.4 10.4.3.1	1	9	3	3
10.7.3.2 10.4.3.1	16	13	4	4

Table II Working of statistical analyzer program module

Inputs : Received Alerts from the database server.
Calculate the mean packet arrival rate and the other parameters using change point detection technique[3] [1] If (Current Packet Count) > ((Mean Rank of the Flow/2) * Mean packet count of the flow till the interval(i-1)) then the flow is classified as Suspicious [2] If Current Packet Count > (Mean Rank of the Flow * Mean packet count of the flow till the interval (i-1)) , then the flow is classified as critical . [3] If Current Packet Count > (Mean Rank of the Flow *> Mean Rank of the Flow)* Mean packet count of the flow till the interval (i-1)) , then the flow is classified as DoS. [4] If the above three conditions are not met then the flow is termed as normal.

Table III Components of the Alert Class

Router List	Traffic Type	Destination IP	Source Id
R1,R3	DoS	10.7.6.2	R3

Table IV Working of DDOS detection program module

Inputs : Received Alerts from the database server.

1. Deserialise the arriving packet. Let it be the new information.
2. If new info. Id != 'local_stats_analyzer' then
Check whether for the same 'dest ip' its local_stats_analyzer has raised any DoS alert
If 'yes' then
Recursively Check whether previous routers have raised any critical or DoS flow.
Store the new information in database server
Send this information to neighbouring routers
Else
Merge old information and new information
Store it in database server
 	
4. Update traffic type, list of routers in the database server.
5. Check whether 'Destination ip' belongs to its internal network
If 'yes' then
If traffic type='DoS'
Declare Dos Attack has established
Else If considerable no. of routers is there in router list
Declare DDoS has been established
Send message to Network Admin