A Survey report on DDOS attacking tools, detection and prevention mechanisms

PRACHI SHEVATE

Master of Computer Applications, Fergusson College, Pune, India

Abstract—*Denial of service is a technique to deny access to a resource by overloading it, such as packet flooding in the network context. Denial of service tools have existed for a while, whereas distributed variants are relatively recent. The distributed nature adds the "many to one" relationship. In this paper, we will be surveying various tools, detection mechanisms and prevention methods for DDoS attacks.*

Keywords—*DOS, DDOS, Tools, Detection, Prevention.*

I. INTRODUCTION

The spread of Internet has prompted a blast in different system related exercises like banking, E-trade, Defense Networks, Radar Systems, Social Engineering, Medical and relatively every field that can be thought of. With the expansion in these administrations, there is additionally an ascent of attacks on these administrations exhibit on the system. A Distributed Denial of Service (DDoS) attack is an endeavor to make an online administration inaccessible by overpowering it with activity from numerous sources. They focus on a wide assortment of vital assets, from banks to news sites, and present a noteworthy test to ensuring individuals can distribute and get to imperative data. They likewise influence Bitcoin trades. The most well-known attack on trade sites and their stages is a DDoS attack. These attacks are a vital part of online life - as banking systems, web based shopping stages and different administrations suppliers are usual targets of DDoS attacks.

Framework and system security is a key component for all these assortment of uses. Encryption, Authentication components, Intrusion Detection Systems, Security Management can be utilized to build the security of the system of PCs. In this survey paper, we will be discussing and comparing two tools used for DDos attack, its detection and prevention methods.

II. DDOS ATTACKING TOOLS

One of the real reason that make the DDoS attacks across the board and simple in the Internet is the accessibility of attacking devices and the capability of these apparatuses to create attacking movement. There are a wide range of DDoS attack apparatuses on the Internet that enable aggressors to execute attacks on the objective framework. Similarly, as the system security and hacking world is ceaselessly advancing, so too are the DDoS attack devices used to complete dispersed disavowal of administration (DDoS) attacks. For instance, DDoS instruments, for example, Trinoo and Stacheldraht were broadly utilized when the new century rolled over, yet these DDoS apparatuses ran just on the Linux and Solaris working frameworks. Particular DDoS attack instruments have since advanced to focus on different stages, rendering DDoS attacks more hazardous for targets and substantially simpler for programmers to do.

A portion of the more current DDoS apparatuses, for example, Low Orbit Ion Cannon (LOIC) were initially created as system push testing instruments however were later adjusted and utilized for malignant purposes. Different DDoS attack instruments, for example, Slowloris were created by "dark cap" programmers whose point is to guide thoughtfulness regarding a specific programming shortcoming. By discharging such DDoS instruments openly, dark cap programmers drive programming designers to fix helpless programming with a specific end goal to maintain a strategic distance from substantial scale attacks. The absolute most normal instruments are examined underneath:

A. Trinoo

It is otherwise called Trin00. College of Minnesota was the first to fall prey for DDoS assault caused by Trin00 instrument in August 1999. The reports specify that it was a two-day assault which included flooding servers with UDP bundles beginning from a great many machines. The assailant reacted just by bringing new daemon machines into the assault. It was first found as a twofold daemon on various bargained Solaris 2.x frameworks. Any objective framework can be utilized to dispatch this assault utilizing UDP flooding. As we will think about in the later part that Trin00 conveys ace/slave design and assailant controls various Trinoo ace machines. TCP and UDP conventions are dependable to perform correspondence amongst assailant and ace and amongst ace and slave. Both ace and slaves are secret key shielded to keep them from being assumed control by another aggressor.

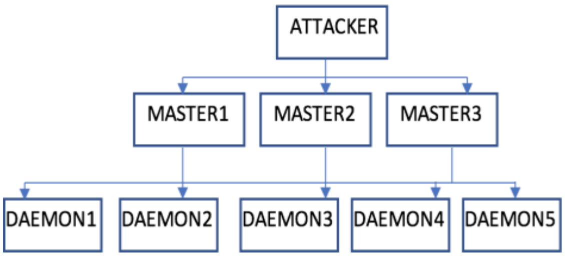
In this area, we depict how a DDoS assault is finished utilizing the Trinoo instrument.

1. Attacker associates with ace by means of telnet to TCP port and enters a secret word. Assailant sends various summons to the ace. For instance, quit summon permits to log off from the ace. mdos summon is utilized to dispatch different DDoS assaults
2. Master passes order line contentions to daemons through UDP port. Note: Commands are watchword secured. Scarcely any summons are: aaa secret word IP, dle - shutdown the daemon.
3. Daemons react to aces on UDP
4. Master needs to monitor daemons by checking in the event that they are alive or not.

Installing a Trin00 network

1. A record is stolen and set up. It comprises of all pre-assembled daemon and ace projects and rundown of hosts.
2. Target is recognized from among every one of the frameworks.
3. A rundown of defenseless frameworks is then used to make a content that plays out the endeavor, sets up an order shell running under the root account that tunes in on a TCP port and associates with this port to affirm the achievement of the adventure.
4. A subset with the coveted design is then decided for a trinoo organize. Pre-ordered parallels of the trinoo daemon are made and put away on a stolen account some place on the Internet.
5. A content is then run which takes this rundown of "claimed" frameworks and delivers yet another content to computerize the establishment procedure, running every establishment out of sight for greatest multitasking.
6. There is likewise an office to introduce rootkit which shrouds the nearness of projects and records. This is more essential on the ace framework, since these frameworks are vital to the trinoo arrange.

Mechanism of Trin00 works as follows:

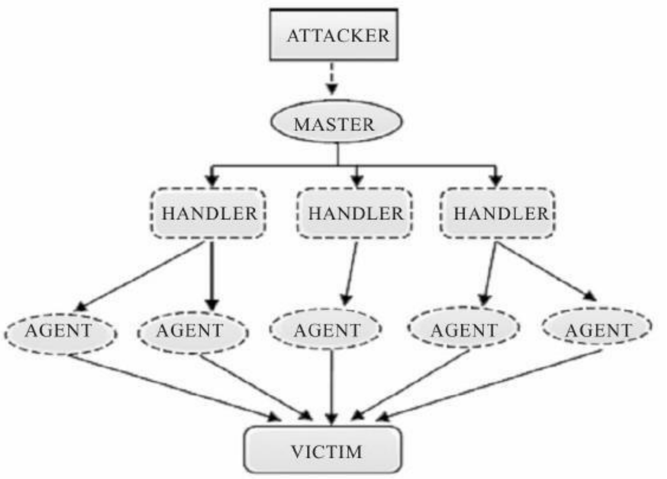


System of Trin00 fills in as takes after:

The attacker(s) control at least one "ace" servers, each of which can control numerous "daemons" (referred to in the code as "Bcast", or "communicate" has.) The daemons are altogether taught to organize a parcel based attack against at least one casualty frameworks. All that is then required is the capacity to set up a TCP association with the ace hosts utilizing "telnet" and the secret word to the ace server to have the capacity to wage gigantic, facilitated, disavowal of administration attacks.

B. Shaft

Shaft is likewise from the same DDOS attacking family as that of Trinoo. It was at first recouped in the year 1999 as paired code in a source frame for the operator. Shaft's unmistakable highlights are the capacity to switch handler servers and handler ports on the fly which makes its recognition by Intrusion discovery frameworks exceptionally troublesome. For the most part, a ticket instrument is utilized to connect exchanges and the



Specific enthusiasm for bundle insights. The pole organize comprises of client(s), handler(s), agent(s) and the objective or the casualty. It is comprised of at least one handler programs ("shaftmaster") and a vast arrangement of specialists ("shaftnode"). The aggressor utilizes a telnet program (“customer”) to interface with and speak with the handlers. A “Pole” system would resemble this:

"Shaft" is designed according to Trinoo, in that correspondence amongst handlers and operators is accomplished utilizing the questionable IP convention UDP. Remote control is by means of a basic telnet association with the handler. "Shaft" utilizes "tickets" for monitoring its individual specialists. The two passwords and ticket numbers need to coordinate for the specialist to execute the demand. A straightforward letter-moving is being used.

To finish up, "Shaft" is another DDoS variation with autonomous starting points. The code that was recuperated had all the earmarks of being still being developed. There are a few other key highlights that demonstrate developmental patterns as the class creates. This implies the recognition of this DDos establishment will turn out to be significantly more troublesome as the instrument progresses. The nearness of such operators can in any case be all the more promptly dictated by examination of movement peculiarities with an imperative on time and asset for the site manager and security groups chipping away at identifying the interruption.

C. Comparison table of Trinoo and Shaft

Tool Name:	TRINOO	SHAFT
Year Discovered:	2000	2000
Target Impact:	Bandwidth	Bandwidth
Scope:	DDOS	DOS, DDOS
Type of Attack:	UDP,TCP,HTTP	UDP,ICMP,TCP
OS Supported:		Linux, Unix
Makes Botnet:	Yes	Yes
Implementation lang:	C	Not known
Architecture Model:	Agent	Agent

III. DDOS Detection Mechanisms

DDoS attacks are dealt with as a clog control issue, but since most such blockage is caused by malevolent hosts not obeying conventional end-to-end blockage control, the issue must be taken care of by the switches. Usefulness is added to each switch to distinguish and specially drop parcels that likely have a place with an attack. Upstream switches are additionally advised to drop such parcels all together that the switch's assets be utilized to course authentic activity. There has been different promising countermeasure to DoS attacks in the current years. In this segment, we talk about the location component of DDoS attacks.

A. SNORT

Grunt is an open source interruption identification and avoidance framework which is prepared to do constant movement investigation and parcel logging. Grunt is a standout amongst the most well-known NIDS. Grunt is Open Source, which implies that the first program source code is accessible to anybody at no charge, and this has enabled many individuals to add to and break down the projects development. Grunt utilizes the most widely recognized open-source permit known as the GNU General Public License. Grunt is coherently separated into various parts. These segments cooperate to identify specific attacks and to create yield in a required arrangement from the identification framework. Grunt's design comprises of four fundamental segments: sniffer, preprocessor, identification motor, yield.

Packet Sniffer

A packet sniffer is a gadget (either equipment or programming) used to take advantage of systems. Packet sniffers have different utilizations: Network investigation and investigating, Performance examination and benchmarking, Eavesdropping for clear-content passwords and other intriguing goodies of information.

Preprocessor

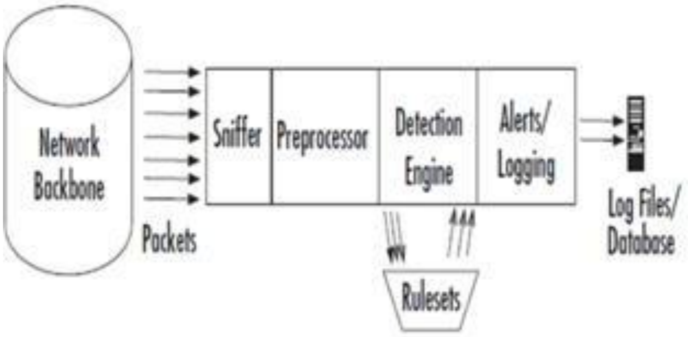
Grunt underpins numerous sorts of preprocessors and their orderly modules, covering many regularly utilized conventions and additionally bigger view convention issues, for example, IP discontinuity taking care of, port filtering and stream control

Detection Engine

The detection engine is the meat of the mark based IDS in Snort. The location motor takes the information that originates from the preprocessor and its modules, and that information is checked through an arrangement of standards. On the off chance that the principles coordinate the information in the parcel, they are sent to the ready processor. The mark based IDS work is refined by utilizing different rulesets. The rulesets are assembled by classification and are refreshed routinely.

Alerting/Logging Component

After the Snort data goes through the detection engine, it needs to go out somewhere. If the data matches a rule in the detection engine, an alert is triggered. Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump- style files or some other form.



B. TIME SERIES ANALYSIS

Some methods detect the sudden changes in traffic by converting the data into a time series and analyzing the time series. They analyze the time series in two different ways:

- Finding the distribution of data in a sampling period and if it is above certain threshold terming this as anomalous.
- Finding the monotonous change in some parameter with time and if the change is substantial terming the data as anomalous

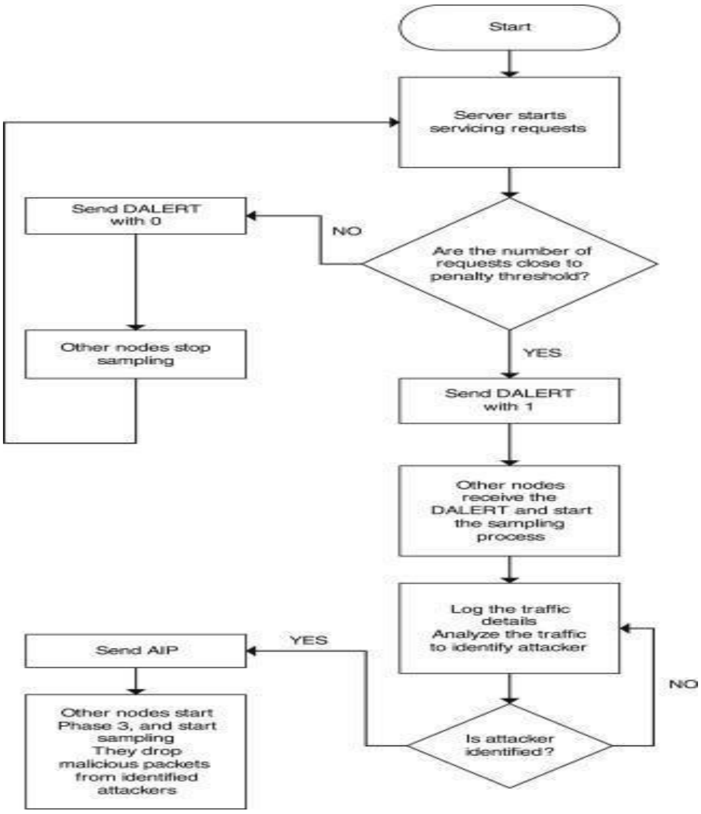
The rationale behind detection methods that look for sudden changes is an assumption that the proportion between certain parameters remains roughly uniform as long as traffic is normal. The parameters considered are the number of requests made and the number of responses received. During a DoS attack, the usual proportion between these parameters breaks and this is detected using Change Point detection method which detects whether the given time series is statistically homogeneous or not.

Distribution of IP addresses in the network traffic has been considered as an important parameter that gets effected by DoS attacks. Chi-Square statistic and covariance are also used to detect statistical heterogeneity in the time series. Wavelet analysis is able to capture complex temporal correlation across multiple time scales. It used energy distribution based on wavelet analysis to detect DoS attacks. When traffic behaviors are affected by DoS attack, energy distribution variance changes markedly. This change in distribution is used to detect DoS attack.

IV. DDOS Prevention Mechanisms

A. *DLSR Protocol*

DDoS preventing optimized link state routing protocol. The principal objective of the DLSR protocol is to prevent a DDoS attack from disrupting the services provided by the server. DLSR is a modification of the existing OLSR protocol. The functioning of the DLSR protocol comprises the following three phases:


Start
Server starts servicing requests
Are the number of requests close to penalty threshold?
NO
Send DALERT with 0
Other nodes stop sampling
YES
Send DALERT with 1
Other nodes receive the DALERT and start the sampling process
Log the traffic details
Analyze the traffic to identify attacker
Is attacker identified?
NO
YES
Send AIP
Other nodes start Phase 3, and start sampling
They drop malicious packets from identified attackers

Phase I: Detecting a DDoS attack

A threshold is setup to analyze the number of service requests coming in. If the number of service requests is above the service threshold then there is a possibility that DoS may ensue. So long as the number of service requests is below the service capacity of the server, a DoS will not occur.

If the server finds that the service threshold is exceeded, the server sends an alert message (DALERT) to all the nodes in the network. Upon receiving this alert, the nodes go into the attack identification mode, while the server continues to monitor and service all requests.

In this phase, all other nodes simply function as OLSR nodes. No special actions are performed.

Note: The start of the attack identification phase does not confirm an attack but merely indicates the possibility of one.

Phase II: Attack identification phase

Once the DALERT message is received, the nodes are aware that a server in the network is on the verge of DoS. From the DALERT message, the node is able to identify the server's IP address. At such an instant, all nodes are only aware of the possibility of a DoS, but have no information of the attacker(s). In order to gain information about the attacker, the nodes begin to sample the incoming data and make a note of the hosts that are requesting services from the server. All nodes are alerted about being attacked by multiple hosts. The attacker's information is sent using an Attacker Information Packet (AIP). The AIP contains the IP addresses of all hosts that have been found trying to execute a DoS attack. And they enter into the defense phase.

Phase III: DDoS defense phase

All the nodes are informed about the attack on Server. The DDoS defense phase starts on a node when it receives an AIP. In this phase, the nodes continue to sample incoming traffic. Packets from unidentified nodes are discarded. If any are found, then the node sends this information using the AIP. The discarding of packets from identified attackers helps in reducing the number of service requests (that reach the server) and, thus, reducing the load on the server. This, in turn, reduces the possibility of a DoS attack. A fake IP address can also be used by the attacker. One can argue that the dropping of packets would result in the genuine user being denied access. But our end goal is to keep the server secure and therefore even if the user is legitimate user, it is sacrificed the server is kept functioning.

It is important to note that the sampling of packets at a node will not guarantee that all the packets from the attacking host will be identified and discarded. Only a certain fraction of the total DoS attack packets will be caught. As presented in an IEEE paper, in order to increase the probability of detection and removal of malicious DoS attack packets, the proposed routing algorithm will route the packets such that they traverse a greater number of nodes before reaching the server. This helps to reduce the number of malicious service requests that reach the server. As each packet traverses through more nodes, the probability of detecting the malicious packets increases. There needs to be a limit by which the length of the path can be increased.

B. Probabilistic approach

As the name suggests, this method is used to find out the number of packets being malicious among massive number of packets. In wireless networks, mobile zombie devices can be used to send out flooding traffic which can consume all spectrum resources or at least significantly reduce the capacity of communication channels available to normal traffic.

REFERENCES

[1] Shaft: http://home.adelphi.edu/~spock/shaft_analysis.txt

[2] Snort: http://www.aboutdebian.com/snort.htm

[3] Time Series Analysis: https://www.slideshare.net/cisoplatform7/threat-detection- using-analytics-amp-machine-learning

[4] Probabilistic Approach: http://ieeexplore.ieee.org/document/4809180/metrics

[5] S. Jin and D. S. Yeung. A covariance analysis model for ddos attack detection. In IEEE Communications Society, 2004