International Journal of Advanced Engineering & Innovative Technology (IJAEIT)
Volume 1, Issue 4, October -2014
ISRA IMPACT FACTOR: 0.42

Automatic IDS and Varying HOPERAA Algorithm for DDoS Attack In Network

Pratik Bobade, Aditya Borge, Chetan Ingulkar, Akshay Jadhav

ABSTRACT — As the technology in network growing these days is increasing the users of internet rapidly. In network there is most harmful attacks such as DDoS and virus attacks. The growth rate of these attacks is increasing per day. The DDoS attacks is nothing but if acknowledgments loss during communication or exchange acknowledgment with each other the another client make acknowledgment and start communicate with each other because of port to be open for longer time another client will get request this include DDoS attacks so for avoiding these attacks the HOPERAA Algorithm is used also some time attacks is done through the packet some packet contain anomaly. This anomaly is nothing but virus which crash down our system so for that purpose we are using Automatic IDS. In which we are using framework that holds potential for self-managing, self-labelling, self-updating and self-adapting.

Index Terms— Automatic Intrusion Detection System, DDoS Attacks.

1 INTRODUCTION

Today the growth of the internet is faster that if draw a graph it go in upward direction. This growth is increasing as the need of internet is increasing. The need means increase of user if we calculate the rate of internet user growing per day it will also in upper direction. As the growth rate is increase the security area is also increasing per day because hackers are trying to access private data or want to crash the system down which helps them to ruff some important data. DDoS attack is most of famous in today life. In this attacks normally when the client send acknowledgments to the server that time the server resend the server side acknowledgments to client during this time if the acknowledgments is lost then the port remain open for longer time waiting for client or server transferring data. That time when port is open another client makes the request and access the data which is known as DDoS attack. [1-4]
Our solution is general, because the mechanisms and algorithms are only based on the clients and server(s). It can be a complementary mechanism to the ones against bandwidth attacks. By adjusting the hopping period (i.e. roughly the time that communication ports remain open), the situation that the adversary is able to launch a directed attack to the application's ports after eaves dropping is limited. [5]

Potential message loss due to the hopping period deviation caused by the clock-rate drifts can be controlled by adjusting a parameter in the HOPERAA algorithm. The message overhead for setting connections between communication parties is bounded and its average overhead is observed to follow an exponential style of decay.
Now these days there is also another attack which known as virus or anomaly attacks. This is done by normally packet transfer. the packet contain the data which is going to transfer in this packet there is some anomaly or virus which are trying to enter our system and crash down our import file. this attack is also playing big role is today's life. There are lot of company which are providing solution for it but they have some limitation they need to update their system as the new virus or new anomaly attack is occurs.[6]
Our solution for this attack is the automatic IDS system in which it will detect this attack without self updating. Also it will manage self-managing self-labelling, and self-adapting. It will work on attribute value it will check the attribute value of data in the packet. And bind the data which are having same or upper or down attribute value.

- *Pratik Bobade, Aditya Borge, Chetan Ingulkar, Akshay Jadhav is currently working as Research Scholar at Department of Computer Engineering ,Navsahyadri Education Society's Group of Institutions, University of Pune, Naigaon, Pune 412213 India*

2 Related Work

As the network is growing there also area if security is growing fast. So everyone trying their best solution for this security now let's see how the HOPERAA and Automatic IDS is best for it.

3 HOPERAA

The communication between two network or client and server is done by normally sending request and then receiving the acknowledgement form both side then transferring the data which is requested but the DDoS attacker send the request to server and start access the data from server. The DDoS attacker sends the blind request to server if server loss connection with client then it starts sending the data to a attacker. It considers the attacker as the client and sends the data. This shown in figure how actual it works. [7]

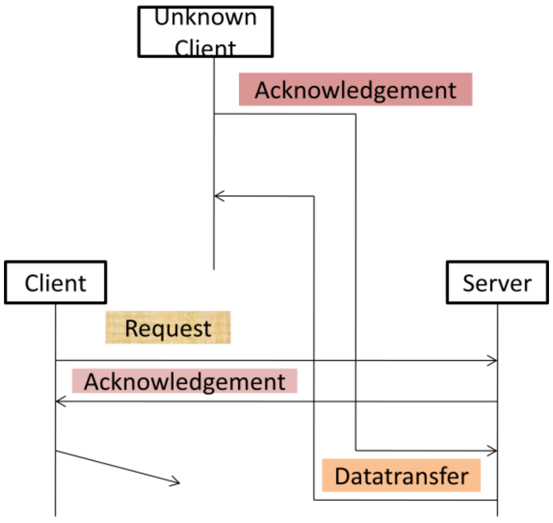

Fig. 1-DDoS attack occurrence

For avoiding this kind of attack we are using port-hopping method. In which after sending the request of client server send the port-hopping in that server send the port number in even or in odd sequence means during the transferring the data to the client the port number is given and data is is transfer to that particular port number. Example if first data is send at port number one than the second data is send two port number 3 after that port 5,7,.....The sequence could be even or odd dependent on sever. Also it calculate the time interval between the transferring the data. Client and server bough will calculate it and set their path they will calculate the speed of transferring the data at which data is transferring at low, medium, high and set their time. It will helpful because bough will know the at which rate data is coming if any another request is come it will identify.[8] HOPERAA(hopping period Alignment and adjustment) use the method is also known as the clock drift method in which time is calculate the HOPERAA is the best way to prevent the DDoS attacks. Also the figure represent the port-hopping method in which number of port are decide and data is transferred to that port number.

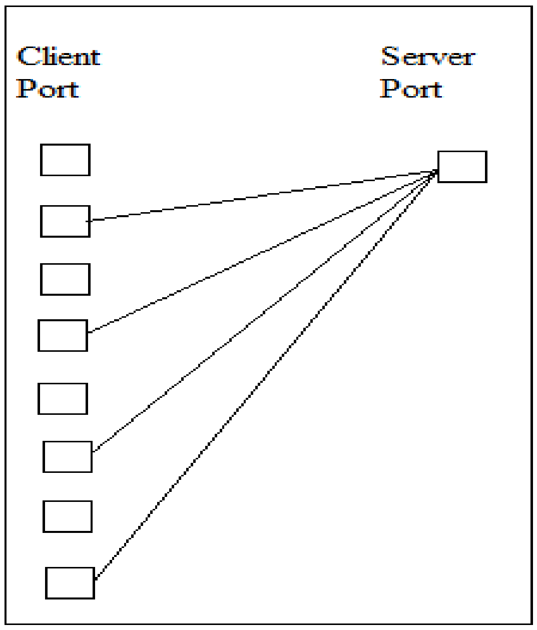

Fig.2:-Port-hopping method

2) Automatic IDS

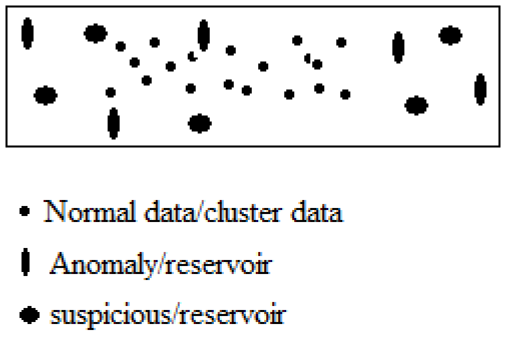

Fig 3:-Initial stage

Whenever the data is came or transfer between two network the data is blind in one or more packet during this sometime hacker or any another person add the anomaly means virus between it and the packet is transfer to us. As the figure show the initial condition of packet where the data is transfer as show it the anomaly contain different value then the normal data but we cannot say that this is anomaly because may some time the data contain different value or may any another possibility so we use the automatic let us see how it work[.9-10]

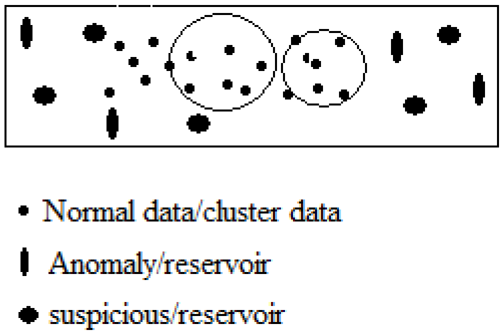

Fig. 4 :-Data binding

As we have say that in automatic self-labelling, self-managing is done. So as we see in figure in automatic the self-labelling is done on basic of attribute value first it calculate first attribute value and mark it as centre and checks for similar or upper, lower matching value. After matching the value it bind that data and reaming data is put as the suspicious or reservoir data.

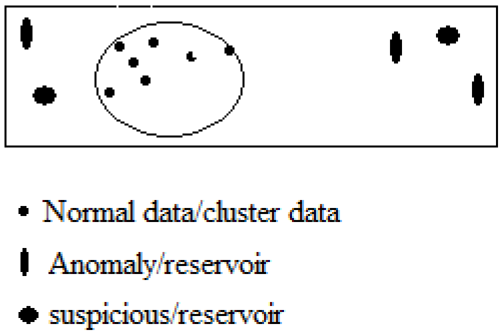

Fig 5:-Rebinding data which is reservoir

The data which is mark is again self-adapted and labeling is done again for the renaming data. This is doing same process but now it did not keep the reservoir data it declare the remaining data as the anomaly if it is suspicious then also it is declare it anomaly.

As it is automatic updating system it check the anomaly attacks and save it attribute value for next time means when the next packet is coming that time we are going to check first anomaly by checking packets all attribute value and our last attribute value which we declare as anomaly if it matches then anomaly is detected and move and start our process again.

A) Methodology:

1. Building initial detection models with Affinity Propagation :

Affinity Propagation is employed to build initial detection models

Let $\mathcal{E}=\{e_1.....e_N\}$ be a set of data items, and let $d(e_i, e_j)$ denote the distance (e.g., an Euclidean distance) between items e_i and e_j.

$d(e_i, e_j) = ||e_i - e_j||$

$$E(c) = \sum_{i=i}^{N} S(e_i, e_{c(i)})$$

E(c)= Fitness Function to cluster the data items
c(i)= index of the exemplar representing the item e_i in a cluster.
$S(e_i, e_{c(i)})$ = Represents the similarity Matrix.

$$\begin{aligned} S(ei, ec(i)) &= -d(ei,ej)^2 && \text{if } i\neq j \\ &= -S^* && (s^* >= 0) \quad \text{otherwise.} \end{aligned}$$

$-S^*$ = Represents a preference that e_i itself be chosen as an exemplar.

e_i = The exemplar of the cluster i
n_i = The number of items associated to exemplar i (i.e., the number of items in the cluster i)
μ_i = The mean distance between exemplar ei and all its associated items ti The last timestamp
when an item was assigned to ei (i.e., the timestamp when a cluster has lastly been updated)
N_{size} =Threshold for identification of suspicious items: minimum number of items for forming a normal cluster.
$\mathcal{E}$ = Threshold for identification of suspicious items: maximum distance between a normal item and its nearest exemplar or maximum mean distance between a normal exemplar and all its
Associated items.
λ = Threshold for immediate anomaly identification
$N_{reservoir}$ = Parameter for rebuilding criterion: the number of suspicious items in the reservoir
r = Parameter for rebuilding criterion: the percentage of suspicious items since last clustering
δ = Parameter for rebuilding criterion: time window length
Δ = Parameter for forgetting mechanism: time window length in which no item is newly assigned to an exemplar.

The detection model is a set of clusters after initial clustering is finished. Each cluster is represented by a 4-tuple (e_i,n_i, μ_i,t_i)
μ_i is calculated as:

$$\mu_i = \frac{1}{ni}\sum_{j=1}^{ni} d(ei, ej)$$

e_j ranges over all items associated to exemplar

2. Identifying anomalies as well as suspicious items and updating the models

Identify the suspicious items by looking at the size and the sparseness of each cluster.

If (ni < Nsize) then cluster is very small, and all items are marked as suspicious.

If ($\mu_i > \varepsilon$) then the cluster is very sparse , and all items are marked as suspicious.

For each new incoming item e_t at time t, its nearest exemplar e_i is found.

if $d(e_t,e_i) \geq \lambda$ (i.e. distance of new incoming item from its nearest exemplar is greater than predefined threshold) then e_t marked as anomalous.

If $\varepsilon < d(e_t,e_i) < \lambda$ then item is suspicious.

Otherwise e_t is normal and assigned to ith cluster.

If Item is normal Model is updated

Update exemplar…

$e_i = e_i$ Exemplar remains same.

$$ni = ni \times \frac{\Delta}{\Delta+(t-ti)} + \frac{1}{ni+1} \quad \text{(Number of Items increased.)}$$

$$\mu i = \left(\frac{\Delta}{\Delta+(t-ti)}\right) + \frac{\mu i \times ni + d(e,ei)}{ni+1} \quad \text{(Update the mean distance.)}$$

t_i=t update the time when the model is last updated.

$\frac{\Delta}{\Delta+(t-ti)}$ is forgetting factor.

if an exemplar ei has never been assigned with a single item in a time window Δ the exemplar is simply reset as a common item:

$e_i=e_i$, $n_i=1$, $\mu_{i=0.}$

Note : Δ or (t – ti) is not a real time interval. It denotes the number of items flowing in during the period between the two timestamps.

3. Rebuilding the model and identifying attacks

The detection model should be rebuilt if a behavioural change is detected.
The model is rebuilt if any of the following three criteria is met: the number of incoming suspicious items exceeds a pre-defined threshold $N_{reservoir}$,
The time window length exceeds threshold δ after the latest clustering,
The percentage of suspicious items since the latest clustering exceeds threshold r.

Upon a rebuilding criterion is triggered, the model will be rebuilt with the current detection model and all the suspicious items in the reservoir

$\{(e_i,n_i)\} \cup \{(e'_j,1)\}$

(e_i,n_i) is an exemplar of the current detection model and ej is a suspicious item in the reservoir.

The adaptation process for model rebuilding is defined as

$S(ei, ei) = -s * \times \sum_i d(e, ei)$ $\quad S(ei, ei) = -ni\, d(ei, ej)^2$

$S(ei, e'_j) = -ni\, d(ei, e'_j)^2$ $\quad S(e'_j, ei) = -\, d(ei, e'_j)^2$

$S(e'_j, e'_j) = -s *$

4 FUTURE WORK

As seeing the growth of network there also growing of hackers so to keep our private data safe we need more security. That is going to provide us in this system which will help us to keep our data safe from hacker. Also keep our data safe form the virus without updating our system it do it automatically and reduce human effort of self-updating or managing.

5 REFERENCES:-

[1] Z. Fu, M. Papatriantafilou, and P. Tsigas, "Mitigating Distributed Denial of Service Attacks in Multiparty Applications in the Presence of Clock Drifts," Proc. IEEE Int'l Symp. Reliable Distributed Systems (SRDS), May/June. 2012.

[2] CERT Advisory CA-1997-28 IP Denial-of-Service Attacks, http://www.cert.org/advisories/ca-1997-28.html, 2010.

[3] D. Dean, M. Franklin, and A. Stubblefield, “An Algebraic Approach to IP Traceback,” ACM Trans. Information and System Security, vol. 5, no. 2, pp. 119-137, 2002.

[4] D.G. Andersen, “Mayday: Distributed Filtering for Internet Services,” Proc. Fourth Conf. USENIX Symp. Internet Technologies and Systems (USITS ’03), p. 3, 2003.

[5] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, Thomas A. Longstaff, A sense of self for unix processes, in: IEEE S&P, 1996, pp. 120–128.

[6] Wei Wang, Xiaohong Guan, Xiangliang Zhang, Liwei Yang, Profiling program behaviour for anomaly intrusion detection based on the transition and frequency property of computer audit data, Computer. Secur. 25 (7) (2006)539–550.

[7] KDD-Data, Kdd cup 1999 Data, 1999. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (retrieved February 2014).

[8] Irina Rish, Mark Brodie, Sheng Ma, Natalia Odintsova, Alina Beygelzimer,Genady Grabarnik, Karina Hernandez, Adaptive diagnosis in distributed systems, IEEE Trans. Neural Networks 16 (5) (2005) 1088–1109.

[9] Snort. Snort, 2014. <http://www.snort.org/> (retrieved February 2014).

[10] Shobha Venkataraman, David Brumley, Subhabrata Sen, Oliver Spatscheck,Automatically inferring the evolution of malicious activity on the internet, in:NDSS, 2013