ISSN 2395-1621

Autonomic IDS and DDoS Attack In Website

Pratik Bobade [#1], Aditya Borge [#2], Chetan Ingulkar [#3], Akshay Jadhav [#4]

1chetaningulkar121@gmail.com

2jadhavakashyme@gmail.com

[#1234]Department of Computer Engineering ,
Navsahyadri Education Society's Group of Institutions,
University of Pune, India

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 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 as per the growth of e-Business there are number of web site increasing in market hacker attacks on this website by sql injection and xss attacks so we are detecting this attacks by useing automatic IDS.

Keywords— SQL injection, XSS Attacks ,Autonomic Intrusion Detection System, DDoS Attacks.

ARTICLE INFO

Article History

Received : 18th April, 2015

Received in revised form :

20th April 2015

Accepted : 24th April, 2015

Published online :

22th may2015

I. 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 . In this attacks happen 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.

Our solution is general, because the mechanisms and algorithms are only based on the clients and server. 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 remains open), the situation that adversary is able to launch a directed attack to the application's ports after eaves dropping is limited. 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 connection between communication pc is bounded and its average overhead is observed to follow an exponential style of decay.

As growing of E-business the number of web site are increasing per day. This web site contain some database or some important information in database (Example: customer id, customer account, etc)

So hacker try to access this database by using Sql injection or XSS attacks.

In Sql injection attacker invokes the application passing as an input a sql statements which the application executes. using this attacker can access the database as an unauthorized user and damage the data stored in the database.

In XSS attacker uses social engineering to convince a victim to click on a URL that contain malicious HTML, JavaScript code user browser then display HTML and execute JavaScript which is part of malicious URL. This attacks can affect the Browser Cookies and another sensitive part of data or database.

II. RELATED WORK

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

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.

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.

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.

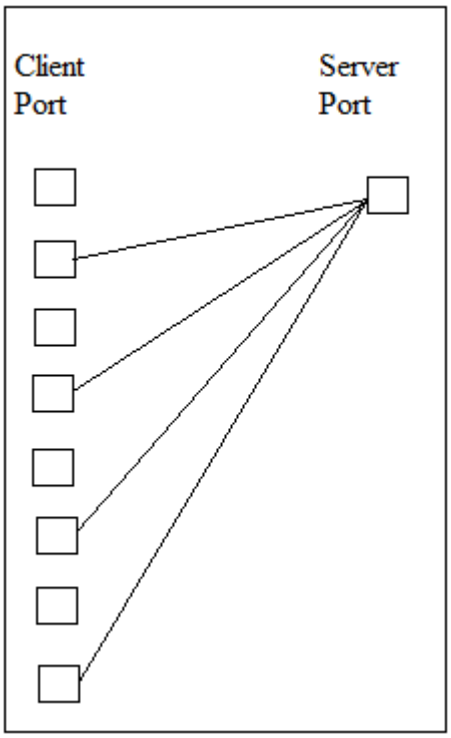


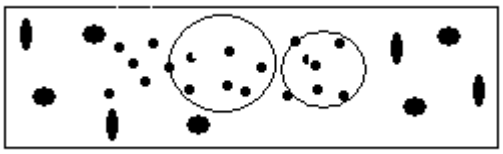
Fig1:-Port-hopping method

Automatic IDS

- Normal data/cluster data
- Anomaly/reservoir
- suspicious/reservoir

Fig2:-Initial stage

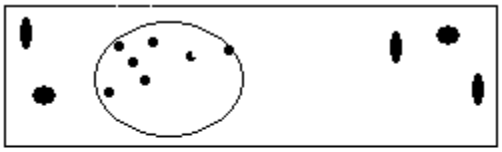
Whenever the data is came or transfer between two network the data is blind in one or more packet during this sometime hacker or any another person add the anomaly means virus between it and the packet is transfer to us. As the figure show the initial condition of packet where the data is transfer as 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.



- Normal data/cluster data
- Anomaly/reservoir
- suspicious/reservoir

Fig3:-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.



- Normal data/cluster data
- Anomaly/reservoir
- suspicious/reservoir

Fig4:-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 autonomic 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.

III. METHODOLOGY

1. Building initial detection models with Affinity Propagation :

Initial detection models

Let $\mathcal{E}=\{e_1 \ldots .e_N\}$ 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.

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

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

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.

$$S\left(ei, ec(i)\right) = -d(ei, ej)^2 \quad \text{if } i \neq j$$
$$= -S \quad (s^* >= 0)$$

otherwise.

$-S$ = 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:

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

e_j ranges over all items associated to exemplar ei.

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 ($\mu_i > \mathcal{E}$) 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 $\mathcal{E} < d(e_t,e_i) < \lambda$ then item is **suspicious.**

Otherwise e_t is **normal** and assigned to i^{th} cluster.

If **Item** is **normal** Model is updated

Update exemplar…

$e_i = e_i$ Exemplar remains same.

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

$t_i=t$ 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 Nreservoir ,

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

{(ei,ni)}∪ { (e'j,1)}

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

Adaptation process for model rebuilding as

S(ei,ei)=$-s*\times\Sigma d(e,ei)i$

S(ei,ei)= $-ni\ d(ei,ej)2$

S(ei,e′j)= $-ni\ d(ei,ej')2$

S(ej′,ei)= $-\ d(ei,ej')2$

S(ej′,ej′)= $-s*$

IV. EXPERIMENTAL RESULT OF HTTP

As we seeing the growth of web site per day so thus is an important issues in computer network security. Because there are number of attacks on web site increasing per day. Different from many other traditional intrusion detection method using separated data set in our experiments we detect web attacks with real HTTP traffic streams.

In the experiments we collected two large HTTP traffic stream we find the common attacks which happen repeat are XSS attacks, Sql injection , DDoS attacks. As per studying this attacks we find that this are mostly done by URL.

EXAMPLE:

1. http://www.aweb.com/msgpage.php?msgID=<script>MSG.location+'http://www.aweb.com
2. http://www.aweb.com/MsgPage.php?msgID=/content/base/build/explore/none.php?.....etc:passwd
3. http://www.aweb.com/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

So we checks this attacks with the traditional intrusion detection like k-NN, PCA, SVM, Adaptive SKL In this we seen that some attacks happen continuously and some rare .The continuously attacks are 239 in a second which mean that rate of attack is high. The difference between traditional intrusion detection and automatic intrusion can be seen in following fig.

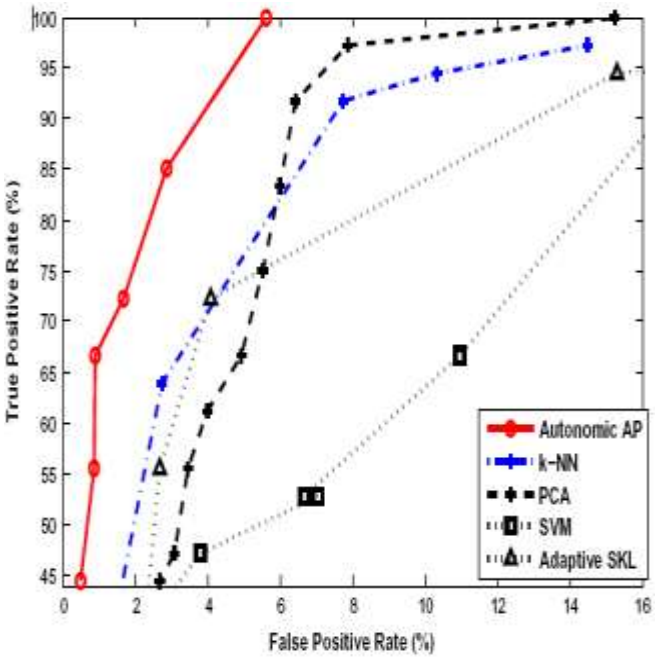


Fig. 5. ROC curves for rare ttack detection with AP, PCA, SVM, SKL and k-NN.

Table 1

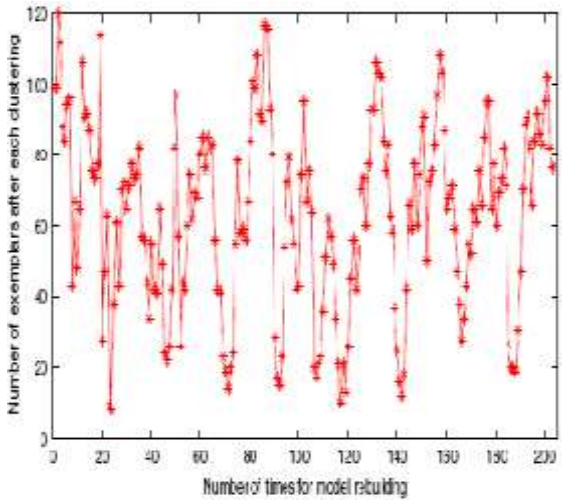
Parameters used in the experiments.

$N_{reservoir}$	R	Δ	Δ	Nsize	Λ
300	60%	2000	10,000	4	1.2

V. RESULT ANALYSIS

As we have graph of system and also the table 1 we can say that the automatic model is more effective than k-NN, PCA. While our model detects 100% of attacks whith a false positive rate of 5.62% ,k-NN detects 97.22% attacks with a higher false positive rate of 14.48%.The automatic model is also better than PCA, SVM, Adaptive SKL in terms of detection accuracy.

Although adaptive SKL works online and adaptively, it can learn abnormal behaviours if an attack has been falsely identified as normal. The adaptive SKL thus leads to a biased model that results in higher FPRs. Our autonomic model, however, consists of exemplars (clusters) that can be considered as micro-models. The detection can be effective even few clusters are biased. Moreover, we define the third status of incoming data items as suspicious. The suspicious data is not directly used for updating the models. Our model thus is more effective for online anomaly detection.



.
Fig.6. The number of exemplars (clusters) after each clustering.

Table 2

Autonomic AP

Ɛ	TRP(%)	FRP(%)
0.150	55.56	0.86
0.129	66.7	1.31

k-NN

Ɛ	TRP(%)	FRP(%)
0.200	38.89	1.32
0.175	55.56	2.15

Fig. 6 shows the number of exemplars after each clustering. The number of exemplars is always changing over time. This indicates that the behaviour of the data is evolving and static methods hence may not be effective.

VI. FUTURE WORK

Growth of the web site will increase day by day. So the problem of computer network security will be solve by this application. It will be helpful for the web site to keep their data secure from the sql injection, XSS attack, DDos attack. Also keep unaffected website from malicious and mainly it will prevent form the DDoS attacks by which helpful for our system that can be run properly with any interrupt occur.

VII. CONCLUSION

In this project we are providing security to authorised person who enters the certain websites. Also database for maintain the record with security of server from DDOS attacks.

REFRENCES

[1] ”Autonomic intrusion detection :Adaptively detecting anomalies over unlabeled audit data strems in computer networks & Security” Thomas guyt.

[2]] “Mitigating Distributed Denial of Servie Attacks in Multiparty Applications in the Presence of Clock Drifts,”
By Z. Fu, M. Papatriantafilou, and P. Tsigs, Proc. IEEE Int Symp. Reliable Distributed Systems .

[3] ”Effective approach towards the intrusion Detection Systems using data mining technique”
by G.V.Nadiammai, M.Hemalatha.

[4] D. Dean, M. Franklin, Trans. Information Security, vol. 6, no. 2, pp. 125-137.

[5] Stephanie Forrest, Steven A. Hofmeyr, IEEE S&P, 1996, pp. 125

[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] D.G. Andersen, “Mayday: Distributed Filtering for Internet Servies,” USENIX Symp. Internet Technologies and Systems, p. 3, 2003.

[8] Irina Rish, Mark Brodie, S Alina Beygelzimer,Genady Grabarnik, Adaptive diagnosis in distributed systems, IEEE Trans. Neural Networks 16 (5)1088–1109.

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

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

