© 2017 IJSRST | Volume 3 | Issue 4 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Detection of Masquerade Attack by Data Driven Semi-Global Alignment Approach

Snehal G.Sarade, Gorakh R. Bankar, Yogeshwari B. Narsale

Department of Computer Engineering, T.A.E. Kondhwa, Maharashtra, India

ABSTRACT

Masquerade attackers behave like a authorized user to utilize user requirements. The semi-global alignment algorithm (SGA) is one of the most optimize and unique techniques to find out these attack but it has not extend the correctness and executions required by large scope, multiuser systems. To increase all the accuracy and the execution of this algorithm, we recommend the Data-Driven Semi-Global Alignment, DDSGA approach. For security purpose, DDSGA improve the scoring systems by altering various alignment arguments for each user. like wise, it allow small replacement in user command series by assinging small suitable different in the low-level showing of the command to ability to perform a task . It seems to make appropriate changes in the client using technique by updating the pattern of the a user as per to its current using technique. To fix the runtime located, DDSGA to make as little the alignment context and parallelizes the search out and to update. After showing the DDSGA phases, we show the experimental outputs. This output is to represent that DDSGA get the high hit ratio of 88.4% with low wrong positive rate. It improves the hit ratio of advanced SGA and minimizes Maxion-Townsend cost. So, DDSGA results in improving all the hit ratio and false positive rates with a capable calculation context.

Keywords : Masquerade attack, sequence alignment,mismatch alignment, security, intrusion attack.

I. INTRODUCTION

The legal user which uses the services and identity of the user can be easily found by Masquerader. For this purpose,first SGA (Semi Global Alignment) algorithm can be applied. SGA is most efficient and optimal algorithm, but the drawback is that it cannot handle atmost accuracy for the multiuser systems. For that purpose,the DDSGA (Data Driven Semi Global Alignment) algorithm can be introduced. It can easily find out the attacks. It increase the effectiveness and efficiency more than the SGA algorithm.For each user by altering distinct alignment, DDSGA improves the scoring system.It reduces the overhead of alignment and finds parallel and renew it. After recounting the DDSGA phases, we can got experimental outputs and this output represents the DDSGA can find the high hit ratio of 88.4% with low false positive ratio. DDSGA increase false positive rate and minimizes Marion Townsend cost.

II. RELATED WORK

We understand some detection techniques for masquerade detection.The distinctive nature considers that command that have not been found in the training data to detect a masquerader. Again, the chance that a masquerader has issued a command is controversially related to the number of the users that use such command. While performance of uniqueness is relatively poor. One-step markove is based upon one-step transitions from a command to the next. This method false alarm rate is not satisfactory. Sconlau et al. toggled between a Markove model and the and simple not dependent. This approach accomplishes the good performance among the regard methods.

The main idea about the compression approach is that new and old data of same user should compress at the same ratio. Masquerading user will compress data in different ratio. For binary data classification Support Vector Machine(SVM) indicate set of machine learning algorithm SVM can gives a large set of pattern but it result in high false alarm and low detection rate[1]. Maxion and Townsend .applied a Naïve Bayes classifier widely used in text classification task and also classify user command data sequences into masquerader. An episodes is introduce which is based on Naïve Bayes technique.

According to Naïve Bays algorithm these episodes are Masquerade or normal. Which is used to the number of command in Masquerade block. This technique improves the hit ratio but there is high false positive rates. So this algorithm do not update the user profile. In Naïve Bayes algorithm information on the probabilities of commands one user over the other users[5]. The WRBF similarity measure based on the frequency of commands f, the weight associated with the frequency vector.

WRBF-NB similarity also increases the overall overhead by computing Naïve Bayes and WBRF and also integrate their results. It neglect the low level presentation of user commands. In Naïve Bayes algorithm both the command of legitimate user and those of an attacker may be different from the train signature. Due to the attacker one persists longer, the deviation of legitimate the user is momentary.

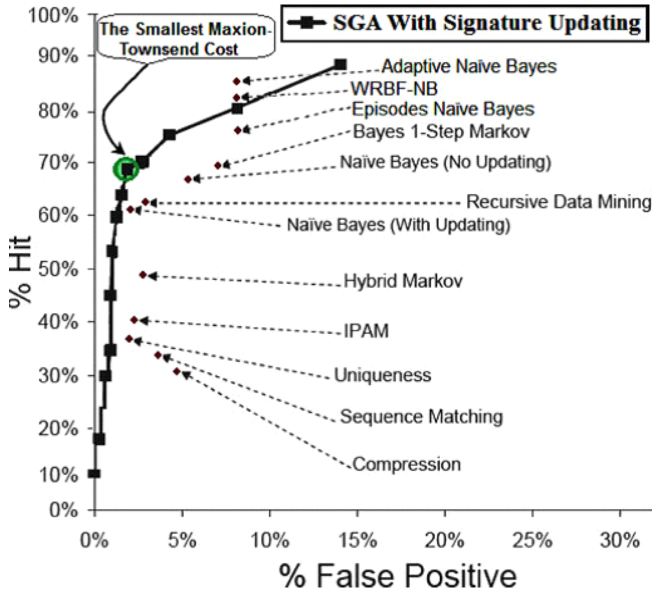


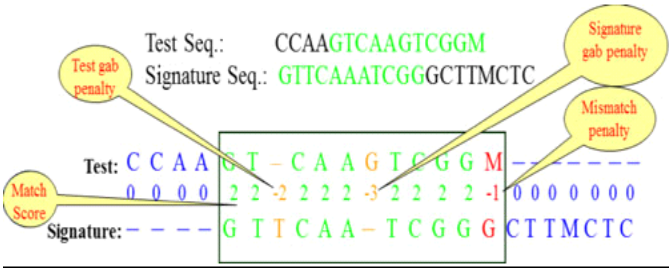
Figure 1. ROC curves for detection techniques that use SEA Dataset.

Malek and Salvator used for the user os commands as bag-of-words without timing information. They used for the one-class support vector machine . The sequence alignment algorithm used to find area of similarity. Behavior of the normal user should be created by collecting sequence of audit data[7].

SGA is more accurate and efficient. It has low false positive and missing alarm rate and high hit ratio. SGA exploits dynamic programming. It initializes an m+1 by n+1 score matrix, M and then shows value of each position of M. In Below diagram there are three stages

- Diagonal Transition
- Vertical Transition
- Horizontal Transition

These three transitions are used to fill each cell in the transition matrix.



Test Seq.: CCAAGTCAAGTCGGM
Signature Seq.: GTTCAAATCGGGCTTMCTC
Test gab penalty
Signature gab penalty
Mismatch penalty
Match Score
Test: C C A A G T – C A A G T C G G M - - - - - - -
0 0 0 0 2 2 -2 2 2 2 2 -3 2 2 2 2 -1 0 0 0 0 0 0 0
Signature: - - - - G T T C A A - T C G G G C T T M C T C

Figure 2. SGA AND THE ENHANCED SGA

The Enhance SGA

TO avoid same false positive, the signature is introducing a new behavior is encountered by exploiting the ability of SGA .The signature update scheme is augments the current signature sequence and the user lexicon. The modification heuristic aligning has been tested on the SEA data set for to simplify the comparison[8].

III. PROPOSED SYSTEM

To defeat the weakness of the existing system we projected a new system which has algorithm called DDSGA. It entirely based on the Enhanced-SGA. The main approach is to align the user active session sequence to preceding one of the same user & labels the misalign areas as irregular. DDSGA tries to leave little mutation in user command. DDSGA workflow can be shown as follows-

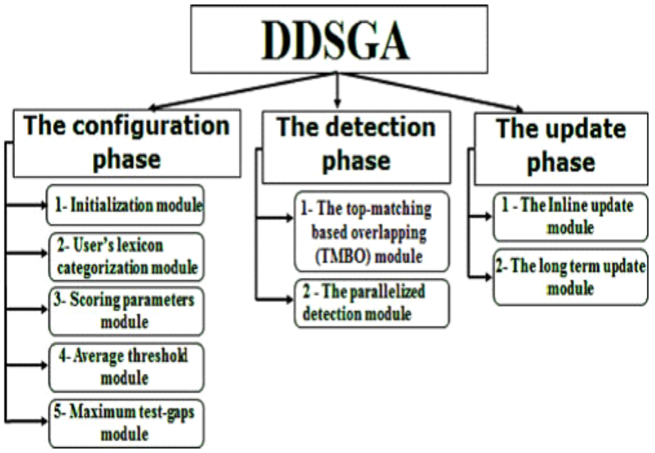

DDSGA
The configuration phase
1- Initialization module
2- User's lexicon categorization module
3- Scoring parameters module
4- Average threshold module
5- Maximum test-gaps module
The detection phase
1- The top-matching based overlapping (TMBO) module
2 - The parallelized detection module
The update phase
1 - The Inline update module
2- The long term update module

Figure 3. Phases and modules.

Above fig represents three main phases of DDSGA & first is for configuration of user & other two phases are based on alignment parameters. The phases of DDSGA can be explained as follows-

- **The Configuration Phase-**

There are some parameters which should be estimate for each user in this phase. The comprehensive explanation of each parameter can be given in following way.

- Mismatch Score

DDSGA calculates the mismatch score through two systems i.e 1) restricted permutation scoring system &2) free permutation one.

- Optimal gap penalties

There are some situations which requires to insert a gap into the test sequence & signature of user which called as optimal gap penalties. In the Enhanced–SGA all the users contribute the same fixed penalties. DDSGA calculates two different penalties for each user as per distinct manners.

- Average optimal threshold

DDSGA detects distinct threshold value for each user as per change in behavior. It's compulsory in both detection and update phase.

- Maximum factor of test gaps(mftg)

This constraint is associated to biggest number of gaps added into the user test sequence to the length of that sequence. The detection phase employ this constraint to estimate the utmost length of overlapped signature sequences.

- **Initialization Module**

Among all the test & signature sequences we require to detect separate set for the configuration phase of each user. So,in this module we divide the user signature into nt non-overlapped blocks each of length n & use this sequence as test sequence. This generated sequences represents all possible arrangement of user signature sequences & all the modules in the design. This sequences are dissimilar from those used in detection phase.

In contrast to non-overlapping features of test sequence. We require to create an overlapped signature subsequence for that, we split the user signature sequence into a set of overlapped groups of length m=2n. In that manner, the last n symbols of a block as well emerge as the first n of next one. ns are the number of signature subsequence's which is equal to nt-1 groups to consider all probable nearby pairs of the signature sequences of size n. The overall process elaborated above can be exposed in straightforward design as shown as follows.

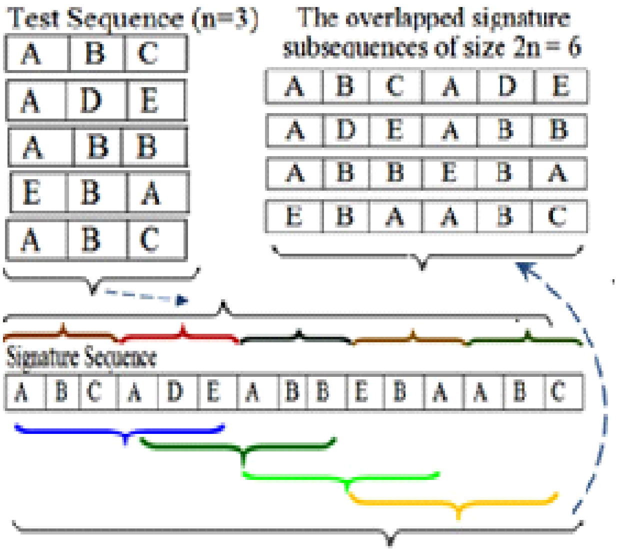


Figure 4. The non-overlapped test sequences and the overlapped signature subsequences.

- **User's lexicon Categorization Module**

In this module,for each user we require to build a lexicon as per their functionality. Lexicons are such commands to achieve particular work. Assume an example of command grep , we can be associated with locate because both belongs to “searching”.

- **Scoring Parameter Module-**

It is required to determine the score, it returns three parameters: optimal test penalty, optimal signature gap penalty, & mismatch score.

At initial, the module adds into the record top-match-list, we choose highest match scores for all the nt sequences. After that top-match-list sequences are aligned to the ns overlapped subsequence's by using any possible gap penalty. The range of test gap consequence range from 1 to n, while the signature gap penalty range from 1 to n. The mismatch score is 0 & match score is +2

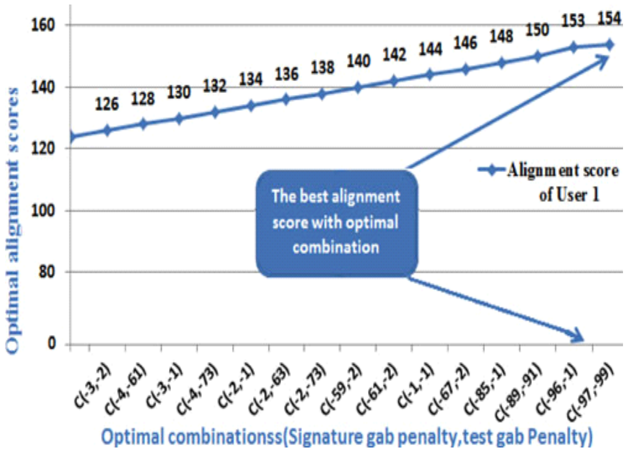


Figure 5. The best alignment score that corresponds to the optimal combination of gap penalties for user 1 in SEA Dataset.

- **Average Threshold Module-**

This is unique type of module in which calculation of average threshold for each user to be used in the detection phase & that may be inform in update phase. If alignment score is lower than the threshold in this phase, then the behavior is classified as masquerade attack. This module uses the same test & signature subsequence's of the initialization module & also it can alter with test sequence of any length as in convenient deployment user test session can be of whichever length.

Avg-align-I = (

- **Maximum Test Gap Module-**

As we identify the Enhanced-SGA Heuristic Aligning splits the signature subsequence's into 2n overlapped subsequence's because if subsequence's of length n are aligned, the maximum number of gaps that can be added into the test sequences is n for all users. The maximum test gap for each user modifies as per level of correspondence between the subsequence's in the user signature & to the length of test sequence. Even if the length of test sequence is long sufficient, the no of gaps is maximum half of sequence of length. After separating of signature sequence into 2n overlapped subsequence's the maximum test gap module can divide it as follow.

L=n+[max {]

- **DETECTION PHASE :**

We have execute a whole alignment test on the origin of the test and signature blocks of the SEA data set to estimate the alignment parameters and the two scoring systems. To derive a isolation with other methods, we choose to use the ROC curve and the Maxion-Townsend cost function. Our main focus is on the property of the alignment parameters on the false positive and false negative rates and on the hit ratio.

Total False Positive= (() nu)*100

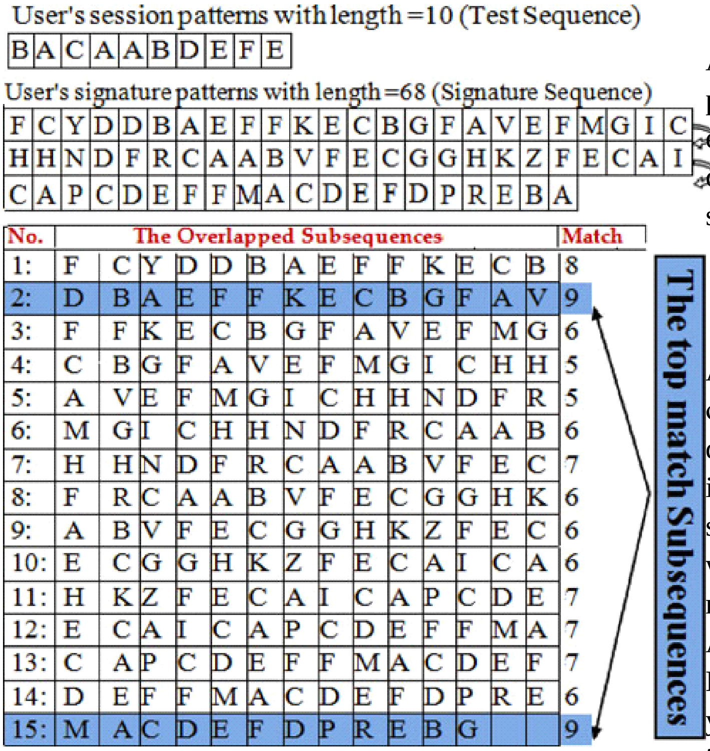
Where:
- fp = No. of false positive alarms,
- n = No. of non-intrusion command sequence blocks,
- nu = No. of users (50 in our case)

Total False Negative= (()/nui)*100

Where:
- fn = No. of false negatives,
- ni = No. of intrusion command sequence blocks,
- nui = No. of users who have at least one intrusion block

- The Top-Matching Based Overlapping Module

To align the session patterns to a set of overlapped subsequence’s of the user signatures in these module,confidential permutation scoring system, Maximum Factor of Test Gaps (mftg) & scoring parameter of each user are used. After dividing the signature sequence into a set of overlapped blocks of length L, it selects the subsequence with the highest match to be used in the alignment process. We have verified that on average, the number of alignments is rather smaller because of the distinction between the overlapped signature subsequence’s.


User's session patterns with length =10 (Test Sequence)

B	A	C	A	A	B	D	E	F	E

User's signature patterns with length =68 (Signature Sequence)

F	C	Y	D	D	B	A	E	F	F	K	E	C	B	G	F	A	V	E	F	M	G	I	C
H	H	N	D	F	R	C	A	A	B	V	F	E	C	G	G	H	K	Z	F	E	C	A	I
C	A	P	C	D	E	F	F	M	A	C	D	E	F	D	P	R	E	B	A				

No.	The Overlapped Subsequences													Match
1:	F	C	Y	D	D	B	A	E	F	F	K	E	C	B	8
2:	D	B	A	E	F	F	K	E	C	B	G	F	A	V	9
3:	F	F	K	E	C	B	G	F	A	V	E	F	M	G	6
4:	C	B	G	F	A	V	E	F	M	G	I	C	H	H	5
5:	A	V	E	F	M	G	I	C	H	H	N	D	F	R	5
6:	M	G	I	C	H	H	N	D	F	R	C	A	A	B	6
7:	H	H	N	D	F	R	C	A	A	B	V	F	E	C	7
8:	F	R	C	A	A	B	V	F	E	C	G	G	H	K	6
9:	A	B	V	F	E	C	G	G	H	K	Z	F	E	C	6
10:	E	C	G	G	H	K	Z	F	E	C	A	I	C	A	6
11:	H	K	Z	F	E	C	A	I	C	A	P	C	D	E	7
12:	E	C	A	I	C	A	P	C	D	E	F	F	M	A	7
13:	C	A	P	C	D	E	F	F	M	A	C	D	E	F	7
14:	D	E	F	F	M	A	C	D	E	F	D	P	R	E	6
15:	M	A	C	D	E	F	D	P	R	E	B	G			9

Figure 5. Overlapped Signature Subsequences of Size 14.

The functioning of the planned TMBO method primarily depends on two parameters: (a) Number of average alignments for the detection process, (b) The effect of the TMBO on false alarm rates and hit ratio. The main task of TMBO calculate the below length of the overlapped subsequence’s as per following equ[9].

$$L= (n+ [mftg+n])$$

In the current phase the current overlapping runs with length L rather than 2n.

The secondary step estimates the match as per each subsequence as exposed in the front of each subsequence’s. The third step choose the top match subsequence's, As in the fig subsequence’s 2 and 15 , as the best signature subsequence’s to be aligned adjacent to the test session patterns of the user. To compute the minimization of workload because of TMBO, consider the Number of Asymptotic Computations (NAC) computed.

$$NAC=Avg\_n\_align*sig\_len*telt\_len$$

As described in the update phase, if at least one of the previous eight alignments has a score larger than or equal to the detection_update_threshold, then a process of inline update should be executed for the signature subsequence and the user lexicon.

- The Parallelized Detection Module

As TMBO divides the user signature in a set of overlapped subsequence’s, we can parallelize the detection algorithm because it can align the commands in the user test session to each top match signature subsequence separately. In this phase we tries to find out whether or not the masquerade is detected. For that we require to perform a simple operation. We just whether Alignment score is less than the Detection_Update_Threshold[9]. If result of this test is yes then thread raises a “Masquerader Detected ” alert & if not then perform the signature update by inline update process. This overall process can be shown as follows.

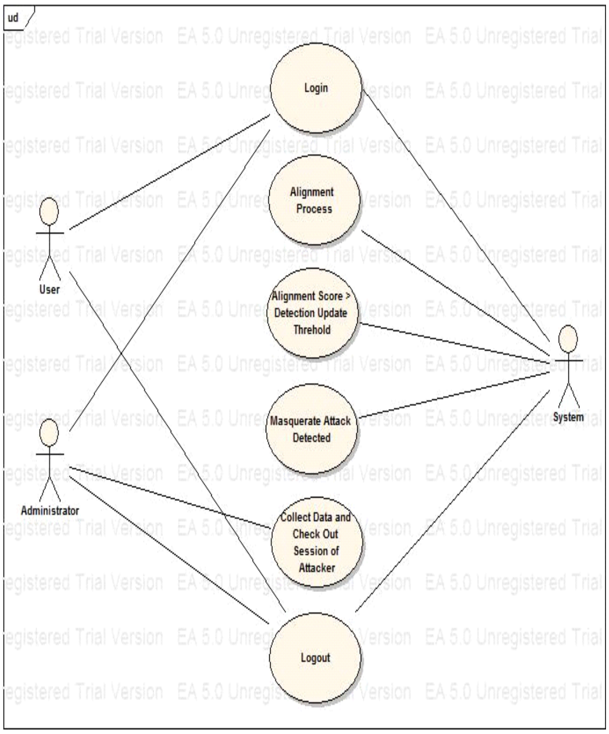

ud
Login
Alignment Process
Alignment Score > Detection Update Threhold
Masquerate Attack Detected
Collect Data and Check Out Session of Attacker
Logout
User
Administrator
System

Figure 6. The processes of the parallelized detection module.

- **The Update Phase** :

Its optional phase when user is not masquerade. Update of user signature is not an easy task because any IDS should be involuntarily update to the new authorized activities of user. The update is taken place by two modules: One is the inline update module and other is long term update .

- The Inline Update Module

This module has two primary functions:

- Scanning areas in user signature subsequence's to be updated and gather with new activities pattern.
- Update the lexicon of user with adding new commands.

Three cases are possible in the TBA that are as follows-

- The test sequence pattern checks the corresponding signature subsequence pattern,
- A gap is added into one or both sequences
- There is at least a variance between the patterns in the two sequences.This can be shown by fig below

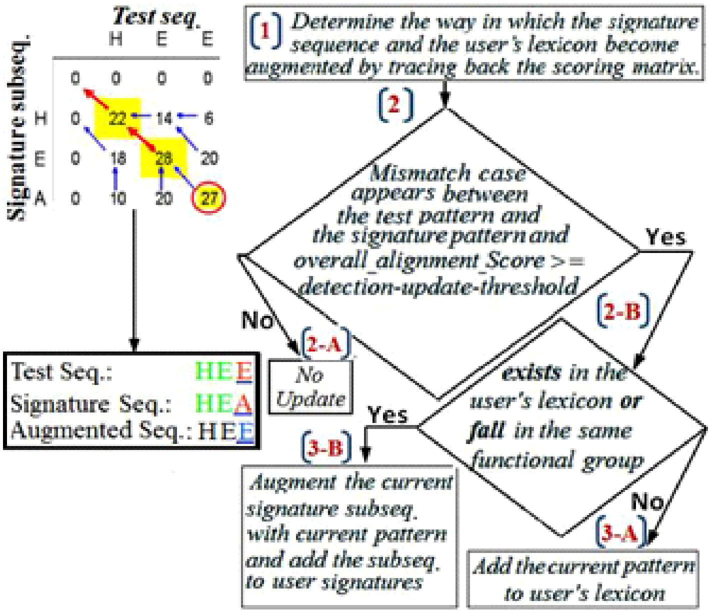

Test seq.
H E E
Signature subseq.
H
E
A
0 0 0 0
0 22 14 6
0 18 28 20
0 10 20 27
Test Seq.: HEE
Signature Seq.: HEA
Augmented Seq.: HEE
[1] Determine the way in which the signature sequence and the user's lexicon become augmented by tracing back the scoring matrix.
[2]
Mismatch case appears between the test pattern and the signature pattern and overall_alignment Score >= detection-update-threshold
No
Yes
[2-A]
No Update
[2-B]
exists in the user's lexicon or fall in the same functional group
Yes
[3-B]
Augment the current signature subseq. with current pattern and add the subseq. to user signatures
No
[3-A]
Add the current pattern to user's lexicon

Figure 7. The inline update steps.

- The Long Term Update Module

In that module we reset system configuration the system parameters through the results of the inline update module. There are three main approaches to run the module: Periodic, idle time, threshold. The periodic approach finishes the reconfiguration step with a Static frequency, i.e. 3 days or 1 week. To minimize the complexities, the idle time approach runs the reconfiguration step anytime when the system is idle. This solution is suitable in highly overloaded systems that need a sophisticated use of the network and computational resources. The threshold approach finishes the reconfiguration step as early as the number of test patterns entered into the signature sequences arrives a threshold that is separate for each user and commonly updated

IV. CONCLUSION

Masquerading is the serious attack which is done for its own purpose.So attacker steals the identity of authorize user and can easily handle the system.SGA is a model that uses sequence alignment term which used to find out the different sequential data, but it has extremely short wrong positive rate and missing alarm rates. low precision even its new edition or achieved the correct exactness and also not given the performance for practical operation. So to overcome from SGA problem there is DDSGA model.and this model is security perception and with more accuracy .It keeps the consistency by giving different parameters to different users and then it offers two level scoring system that bear means ignore change in the low level commands functionality of user command and aligning commands in the same class but without minimizing the alignment score . The scoring systems also permit all to carry out of its commands and changes in the user actions extra time. All features robustly corrupt false positive and absent alarm rates and improves the detection hit ratio. In the SGA data set, the ability of DDSGA is always better as compare to one of SGA. Top-Matching Based Over-lapping technique minimizes the computational context of alignment by minimizing the pattern sequence into a smaller set of overlapped subsequences. Additionally, the detection and the inform processes can be parallel with no failure of accuracy.

V. REFERENCES

[1]. T., S. E. Coulla and B. K. Szymanski, “Sequence alignment for masquerade detection,” J. Comput. Statist. Data Anal., vol. 52, no. 8, pp. 4116–4131, Apr. 2008.

[2]. T. Lane and C. E. Brodley, “An application of machine learning to anomaly detection,” in Proc. 20th Nat. Inf. Syst. Security Conf., 1997, pp. 366–380.

[3]. B. Christopher, “A tutorial on support vector machines for pattern recognition,” Data Mining Knowl. Discovery, vol. 2, no. 2, pp. 121–167, 1998.

[4]. Hisham. A. Kholidy and Fabrizio Baiardi, “CIDD: A cloud intrusion detection data set for cloud computing and masquerade attacks,” in Proc. 9th Int. Conf. Inf. Technol.: New Generations, Las Vegas, NV, USA, Apr. 2012, pp. 16–18.

[5]. S. Malek and S. Salvatore, “Detecting masqueraders: A comparison of one-class bag-of-words user behavior modeling techniques,” in Proc. 2nd Int. Workshop Managing Insider SecurityThreats, Morioka, Iwate, Japan. Jun. 2010, pp. 3–13.

[6]. B. Szymanski and Y. Zhang, “Recursive data mining for masquerade detection and author identification,” in Proc. IEEE 5th Syst., Man .Cybern. Inf. Assurance Workshop, West Point, NY, USA, Jun.2004, pp. 424–431.

[7]. Subrat Kumar Dash, K. S. Reddy, and K. A. Pujari, “Adaptive Naive Bayes method for masquerade detection”, Security Commun.Netw., vol. 4, no. 4, pp. 410–417, 2011.

[8]. A. Sharma and K. K. Paliwal, “Detecting masquerades using combination of Na€ıve Bayes and weighted RBF approach,” J. Comput.Virology, vol. 3, no. 3, pp, 237–245, 2007.