Identification of Masquerade Attack and Prevention by using Data Driven Semi-Global Alignment Approach

Amol C.Devkate[1] Megha D.More[2] Sonali R.Misal[3] Vaibhav G.Pagare[4] Prof.Yogendra Y.Patil[5]
[1,2,3,4,5]S.B.P.C.O.E Indapur

Abstract— In the computer system masquerader is a outsider or a insider user who take the privileges or authentication to act as an legal user in system. Such user take identity of a legal user to performing some illegal or a fraud activity[3]. Till now various researches has been accomplish for detecting masqueraders. Semi-global alignment approach is one Best algorithm but it doesn't reach to that much of accuracy and performance which is required by high scale industries. To overcome such a drawback we discover a new technique called as Data Driven Semi-Global Alignment Approach(DDSGA). In this algorithm we tries to improve security of system by adopting scoring parameter for a each user. DDSGA also minimize the alignment overhead. The process of Detection and updating is done in a parallel manner. experimental results that show that DDSGA achieves a high hit ratio of 88.4 percent with a low false positive rate of 1.7 percent.

Keywords: Masquerade detection, sequence alignment, security, intrusion detection, attacks

I. INTRODUCTION

As we know masquerade is a attacker who authenticate as an a legal user and steal the user credentials and tries to do fraudulent activities. There are mainly two types of attacker Insider attackers and the outsider attacker.

Insider attacker or a masquerader is the legal user of a system that wrongly use its own privileges to performing unauthorized actions and accessing distinct accounts. While the outsider attackers are the person who takes privileges of a legal user for performing fraudulent activities such as changing the user authentications, installing software's with a malicious code, social engineering, spoofing, sniffing.

While the masquerader access the user data it creates a log which is a main source for detecting the masquerader. At first, masquerade detection generate a profile for each user by gathering information such as login time, logout time, location, session duration, CPU time, commands issued, user ID and user IP address. After that, it compares these profiles against logs and indicate as an attack any behavior that does not match the profile. In our Algorithm we tries to improve performance and accuracy of detection. The main idea of a DDSGA is to best alignment of the active session sequence to the recorded sequences of the same user[6].

There may be possibility that some areas of active session will mismatch with a user profile. Such a mismatch area marked them as anomalous and several anomalous are strong indication of masquerader user . To increase the hit ratio and reduce both false positive and false negative rates, we pairs each user with distinct gap insertion penalties according to the user behavior. In order to reduce both runtime overhead and live time of a masquerader inside the system, DDSGA implements both detection and updation in parallel thread.

II. EXISTING SYSTEM

The Technologies used before DDSGA techniques can be explained by the following table which contains various techniques with its advantages and disadvantages.

A. Support Vector Machine(SVM):

Masquerade is a illegal user which tries to take privilege of legal user. To Detect such a attacks and attackers various techniques . The first technique was a super vector machine. The primary idea about the compression technique is that new and old data of same user should compress at the equal 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[1].

B. Computer Intrusion: Detecting Masquerades:

In the Computer intrusion: Detecting masquerades Our analysis is only based on the first two fields, "Command name" and "User". In this technique use generate fixed number of unix commands for each user and analysis is made to detect a masquerade

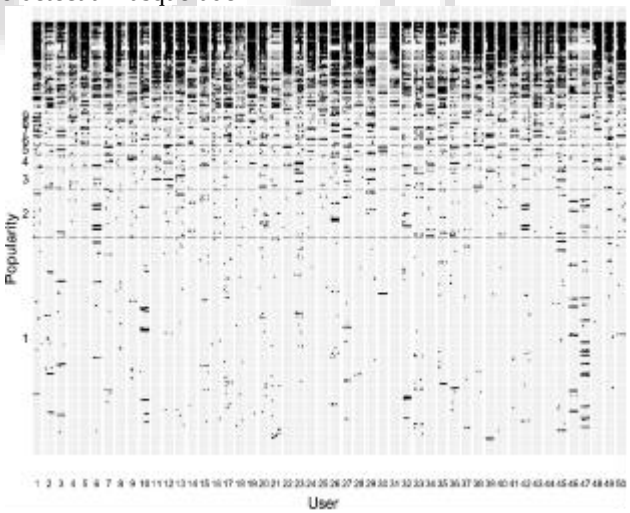

Fig. 1: Computer Intrusion: Detecting Masquerades

C. Sequence Alignment Algorithm:

pattern matching capability of sequence alignment algorithms to find masquerade attacks within sequences of information system audit data (e.g., command line entries). In bioinformatics, these algorithms are typically used to find areas of similarity between two sequences of biological data (e.g., DNA sequences)[2]. Sequence alignment feed several benefits beyond simple lexical matching by providing domain knowledge into the alignment process, such as likely

mutations in the sequences[6]. 'good' and 'bad' alignments based on this domain knowledge is provided by Customized scoring systems. Therefore, the alignments actually highlight areas of functional similarity between the aligned sequences based on the scoring system utilized[6].

Semi-global alignment algorithm is also known dynamic sequence alignment algorithm for detection of a masqueraders. Though, the algorithm proves better than any other pair-wise sequence alignment algorithms such as local and global alignment algorithms

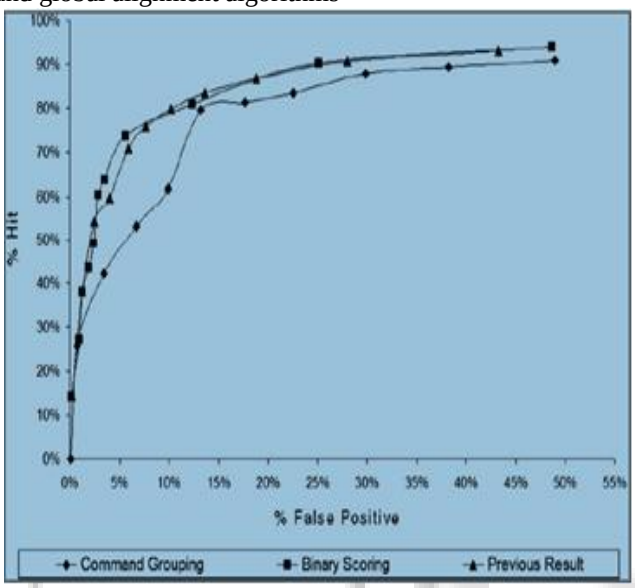

Fig. 2: Receiver operator characteristic curve for previous result, binary and command group scoring

D. Improved Semi-Global Alignment Approach:

To over the drawbacks of sequence alignment we defines a new techniques called as a Semi global alignment approach. To avoid same false positive, the signature is introduced a new behavior is encountered by exploiting the ability of SGA[8] .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

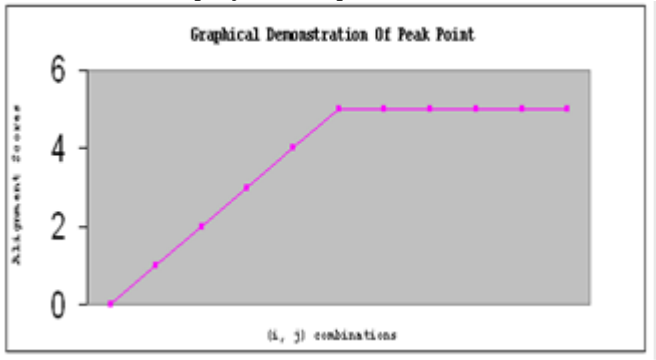

Fig. 3: Demonstration of peak point

III. PROPOSED SYSTEM

DDSGA work same as like a enhanced SGA with some modification in functionality which leads to performance improvement. The main strategy to detect a masquerader is to align users active session with previous of the same user and label misaligned area as a anomalous. Strong indications of masquerader is signaled when no of anomalous area is larger than the dynamic threshold of a user[9].

The overall task of detecting masquerader can be explained in three main phases A. Configuration B. Detection and C. Update.

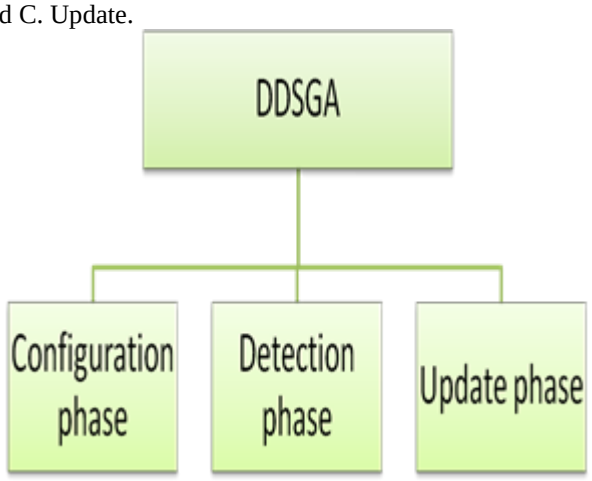

Fig. 4: Basic Phase of a DDSGA

A. Configuration Phase:

Configuration is the phase where some parameters are calculated for each user. This parameters calculation is necessary for a detection and update process. This parameters are as follows

– Mismatch score:

Alignment is process where user current user session is matched with previous user session. If the mismatched ie anomalous are found more than the detection update then there is strong indication of user is masquerader. We calculate such a mismatch score with a two methods free permutation system and restricted permutation one.

– Average optimal threshold:

DDSGA calculates different threshold value for each user. This value plays an vital role in detection and updation phase.

– Optimal gap penalties:

When inserting a gap into the test sequence and the signature sequence the optimal test gap penalty and the optimal signature gap penalty are paid. In the DDSGA we computer two distinct penalties for every user according to distinct user behavior.

For calculating above all parameters we need to generate a test and signature sequence of user signature. Which is done in the Initialization module

1) Initialization Module:

For generating test sequence we divide a user signature into nt non overlapped blocks of a length n. This test sequence is necessary for calculating the various user parameters. Signature sequence is generated by dividing user signature into group of overlapped sequence of length m=2n. The scoring alignment depends upon the match between the test and the signature subsequences,

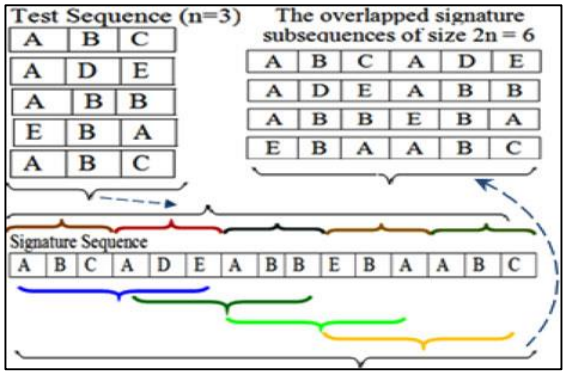

Fig. 5: The non-overlapped test sequences and the overlapped signature subsequences.

2) Users Lexicon Categorization Model:

In this model we generate lexicons for a each user. This module joint the user lexicon list and command grouping approach[6].

Users lexicon list	Functional group
Sh	Shell programming
Vi	Text processing
Kill	Process terminating
mail	Internet applications
search	Finding

Table 1: Lexicon List

3) Scoring Parameter Module:

Starting from the test and signature subsequences for each user, this module returns main three parameters: 1.optimal test gap penalty 2.optimal signature gap penalty 3.mismatch score. Here DDSGA maintain top_match_list. This list enables DDSGA to align the top match test sequence instead of all nt sequences.

Calculation of mismatch score can be done by using following formule MS=

$$\sum_{i=1}^{n}\sum_{k=1}^{nt} Min(Noccur_{Itself(pi)}, Noccur\_seq(pi))$$

- n is the length of the test sequence, nt is number of test sequences,
- Noccur_Itself (pi) is number of occurrence of pattern i in the current evaluated sequence,
- Noccur_Seq k(pi) is number of occurrence of pattern i in test sequence k

4) Average Threshold Model:

For each user this module compute a dyanamic average threshold used in detection phase and that may be updated in update phase. Alignment score is lower than the threshold in detection phase then the behavior is classified as a masquerade attack. While the Enhanced-SGA only considers snapshots of the user data, this module considers all the user data. The module applies Eq. to calculate the average alignment of test sequence i, avg_align_i, and the sub_average score for all previous alignment scores, score_align_i, of test sequence i. In the equation, max_score_align_i is the largest alignment score optened from the alignment of sequence i to all ns signature subsequences

Detection_update_threshold=∑avg_align/nt

B. Detection Phase:

We have run a complete alignment practices based upon the test and signature blocks of the data set to calculate the alignment parameters and the two scoring systems. Our focus on the effect of alignment parameters on false positive, false negative and hit ratio. The calculation of a hit ratio is done by following formula

Hit ratio=100-Totalfalsenegative

DDSGA calculates false positives, false negatives and hits are for each user, transformed into the corresponding rates that are then integrated and averaged over all 50 users.

TotalFalsePositive= ((∑fp/n)/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)

TotalFalseNegative=((∑fn/ni)/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 result of computation is improved by improving the hit ratio and minimizing false positive and false negative ratio. The increased hit ratio can be shown in below table which contains various approaches of detecting masquerade attack along with their result

Approach Name	Hit ratio percent	Hit ratio percent
DDSGA(Restricted Permutation)	83.3	3.4
DDSGA(Free Permutation)	80.5	3.8
SGA (Signature Updating)	68.6	1.9
SGA (Signature Updating þ Heuristic Aligning)	66.5	1.8
Na€ıve Bayes(With Updating)	61.5	1.3
Markov	75.8	7.7
Bayes 1-Step Markov	69.3	6.7

Table 2: A Comparison Between Our Two Scoring Systems And The Current Detection Approaches

AS from table 2 we can observe that the restricted permutation system results in higher hit ratio with corresponding low false positive rates. we have select the restricted permutation system as a appropriate scoring system for all the phases of DDSGA. The complexity of computation of SGA is quite large. As an example it requires approximate 500,000 operations to test one user session for masquerade attacks in the SEA data set, because the length of the signature sequence is 5,000 and that of the test sequence is 100. The resulting overhead is unacceptable in multiuser environments like cloud systems or in computationally limited devices like wireless systems. To minimize this overhead, we introduce two computational expansions that concern, respectively ,the Top-Matching Based Overlapping module and the parallelized detection module that are executed in each alignment session.

1) Top-Matching Based Overlapping Module:

The TMBO amend the Heuritic Aligning of the Enhanced-SGA. After splitting the signature sequence into a set of overlapped blocks of length L, it chooses the subsequence with the highest match to be used in the alignment process.

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

The valuation of the proposed TMBO approach mainly based upon two parameters: (a) Number of average alignments for the detection process, (b) The effect of the TMBO on false alarm rates and hit ratio. To evaluate the reduction of the workload due to TMBO, consider the Number of Asymptotic Computations(NAC) computed

$$NAC=Avg\_n\_align*Sig\_len*Test\_len$$

Where:

- Avg_n_align is the average number of alignments required for one detection session over all existing users.
- Sig_len is the length of the overlapped signature subsequence.
- Test_len is the length of the test sequence.

2) The Parallelized Detection Module:

Since TMBO partitions the user signature in a set of overlapped subsequences, 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 module the task of alignment score calculation, detection of a masquerader is done parallel by creating different thread for a each user and running them parallel. This reduces the live time of a masquerader in a system.

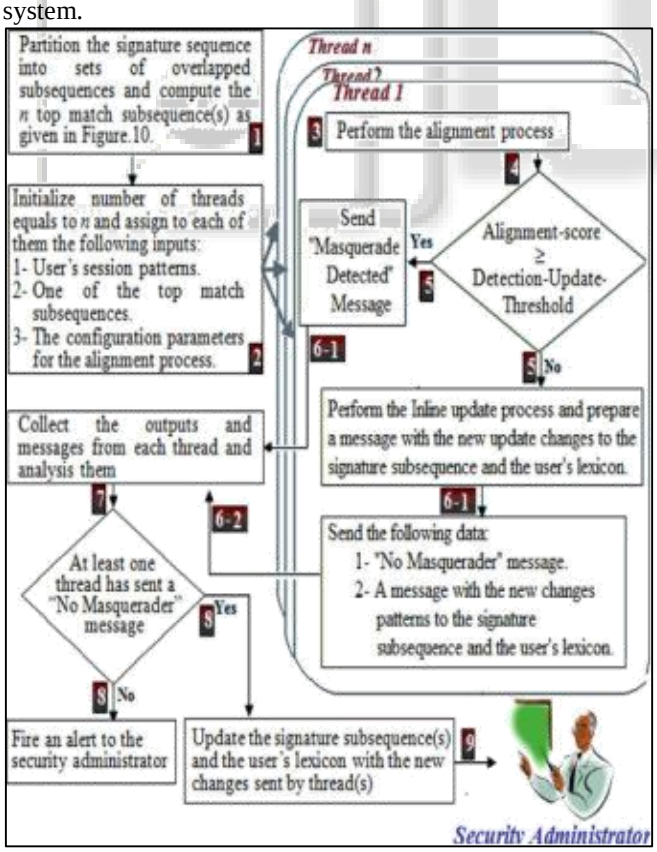

Fig. 6: The processes of the parallelized detection module.

C. Update phase:

The update of the user signature patterns is critical because any IDS should be automatically updated to the new legal behaviors of a user. This update is developed by two modules:

1) the inline update module
2) long term update one.

1) The Inline Update Module:

This module has two main tasks

1) Searching areas in user signature subsequences to be updated and accumulate with the new user behavior patterns[6].
2) Update the user lexicon by inserting new commands.

After each alignment every parallel thread must update both the user signature subsequences and user lexicons. This overall task done in the detection phase. There are different case in which we need to update the user signature some of which are case follows

Case1: The test sequence pattern matches the corresponding signature subsequence pattern..

In this case no update is required because the alignment correctly used the symbols in the correct subsequence to find optimal alignment.

Case 2. A gap is inserted into either or both sequences.

This case does not require an update because symbols that are aligned with gaps are not similar and should be neglected.

Case 3: There is at least a mismatch between the patterns in the two sequences.

In this case we consider all the mismatches within the current test sequence. Then, both the signature subsequence and the user lexicon are updated under two conditions. The first one states that we can insert into the user signatures only those patterns that are free of masquerading records.This happens anytime the overall_alignment_score for the current test sequence is larger than or equal to the detection_update_threshold. The second condition states that the current test pattern should have previously appeared inthe user lexicon or belongs to the same functional group ofthe corresponding signature pattern.

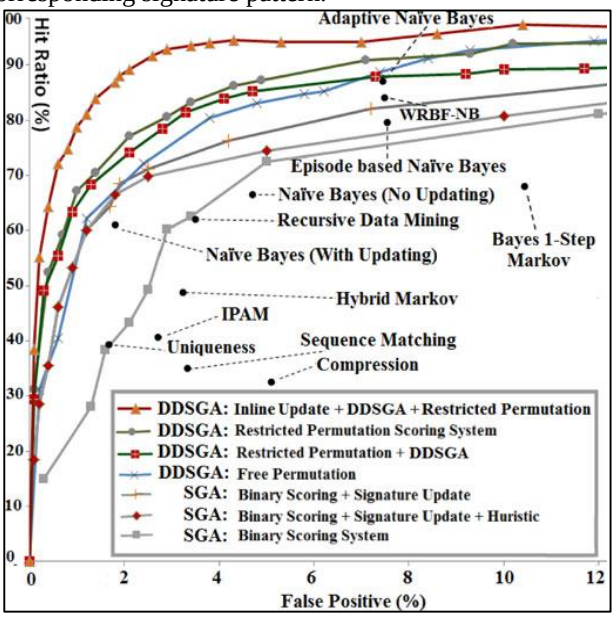

Fig. 7: Impact of Inline update on system accuracy.

2) Long Term Update Module:

In this module we reconfigures the system parameters through the outputs of the inline update module. Here some strategies used for a computation The periodic strategy used

for the reconfiguration step with a fixed frequency, i.e. 3 days or 1 week. To minimize the overhead, the idle time strategy runs the reconfiguration step anytime the system is idle[9]. This solution is accurate in highly overloaded systems that require an efficient use of the network and computational resources. The threshold strategy runs the reconfiguration step as soon as the number of test patterns inserted into the signature sequences reaches a threshold that is distinct for each user and frequently updated[9].

IV. CONCLUSION

Masquerade is one of the critical attack because attacker successfully controls the user privileges. SGA is based on sequence alignment and it is one of the most effective technique for detection of a masquerader[6]. But its not reach to that much of accuracy and effectiveness which was required by high scale industries[8]. To overcome this drawbacks we introduce a DDSGA algorithm which models more accurately the consistency of the behavior of distinct users by introducing distinct parameters. It reduces the live time of the masquerader in system[9] by creating detection and update threads in a parallel manner. Hence DDSGA is One of the best algorithm for a detection of a masquerader with high hit ration and low false positive rate.

REFERENCES

[1] M. Schonlau, W. DuMouchel, W. Ju, A. F. Karr, M. Theus, and Y.Vardi, "Computer intrusion: Detecting masquerades," Statist. Sci.vol. 16, no. 1, pp. 58–74, 2001.
[2] S. E. Coull, J. W. Branch, B. K. Szymanski, and E. A. Breimer, "Intrusion detection: A bioinformatics approach," in Proc. 19th Annu. Comput. Security Appl. Conf., Las Vegas, NV, USA, Dec.2003, pp. 24–33
[3] A. H. Phyo and S. M. Furnell. "A detection-oriented classification of insider it misuses," in Proc. 3rd Security Conf. 2004.
[4] S. K. Dash, K. S. Reddy, and A. K., Pujar"Episode based masquerade detection," i,in Proc. 1st Int. Conf. Inf. Syst. Security, 2005,
[5] A. Sharma and K. K. Paliwal, "Detecting masquerades using a combination of Na€ıve Bayes and weighted RBF approach," J. Comput.Virology, vol. 3, no. 3, pp, 237–245, 2007.
[6] Scott E. Coull, Boleslaw K. Szymanski, "Sequence alignment for masquerade detection" Computational Statistics and Data Analysis 52 (2008) 4116–4131
[7] S. Malek and S. Salvatore, "Detecting masqueraders: A comparison of one-class bag-of-words user behavior modelling".
[8] A. S. Sodiya, O. Folorunso, S. A. Onashoga, and P. O. Ogundeyi, "An improved semi-global alignment algorithm for masquerade detection," Int. J. Netw. Security, vo1. 12, no. 3, pp. 211–220, May 2011.
[9] Hisham A. Kholidy, Fabrizio Baiardi, and Salim Hariri DDSGA: data-driven semi-global alignment approach for detecting masquerade attack.

