IJSRM
INTERNATIONAL JOURNAL OF SCIENCE AND RESEARCH METHODOLOGY
An Official Publication of Human Journals

Human Journals
Review Article
June 2017 Vol.:6, Issue:4

Anomaly Detection of Masquerade Attacks Using Optical Semi-Global Alignment (OPSGA)

IJSRM
INTERNATIONAL JOURNAL OF SCIENCE AND RESEARCH METHODOLOGY
An Official Publication of Human Journals

Rohit R. Urade[*1], N. Z. Tarapore[2]

[1]*Research scholar, Dept. of Computer Engineering*
Vishwakarma Institute of Technology, Pune, India

[2]*Assistant Professor*
Dept. of Computer Engineering
Vishwakarma Institute of Technology, Pune, India

Submission: 5 June 2017
Accepted: 10 June 2017
Published: 25 June 2017

Keywords: Attacks, Masquerade detection, Optical semi-global alignment algorithm, Intrusion detection.

ABSTRACT

A masquerade attack is an attack which shows himself as a genuine user to use the privileges and user services within the system. The semi-global alignment (SGA) method is most effective to detect this type of attacks but when it comes to multiuser systems in large scale, the accuracy and performance have not reached that level until now. To enhance SGA method in performance and efficiency, we are proposing optical semi-global alignment approach (OPSGA). OPSGA has new transition matrix for faster alignment between two sequences and also better scoring system than early one. OPSGA also bear small changes in the user sequence, these behavioral changes can be updated in the user signature sequence according to current behavior. This approach optimizes the alignment overhead which also reflects in less run time overhead of the system. This approach was implemented using net framework. The results were tested by using SEA, Purdue and standard masquerade datasets. After explaining the OPSGA phases, we show the experimental results in which OPSGA acquires the hit ratio of 95.4 percent and low false positive rate of 2.7 percent. Hence, OPSGA results in increase of hit ratio by about 7 percent.

1. INTRODUCTION

A masquerader is an attacker which validates as genuine user of the system by sneaking into to its login ID or by breaching into the verification services. The system can be attacked by someone within it or from outside the system. Legal system user is an insider masquerader that misuses his/her rights to get entry in different accounts and achieve illegal actions. Outsider targets to exploit all the rights of a genuine user, for example, if a genuine user is logged in into the system and leaves the workstation open without any protection, at that point masquerade attacker can be co-worker. There exist many other implementations for masquerade attack. For instance, finding user password by duplicating or ex-filtration, packet sniffing, software with malicious code, hidden programs running within the system or by eavesdropping. These attacks leave some footprints in the system log, after inspection, it can be linked with some users. Many attacks which don't leave imprints whenever entered into system, but masquerade attack detection can be observed by the user behavior. At start, system begins to create profile for each user by gathering all the information, for instance, employee ID, session time, port, location, commands and other user activities for masquerade detection. Identification of a masquerade attacker in the system can be determined by the user pattern, an order of instructions which are gathered together from the genuine user. The current user sequence pattern will be evaluated with the stored user signature sequence pattern whenever the user profile is logged in. If commands that are produced by genuine user, it matches properly with the signature then the user is not an attacker or else it will be detected as masquerade attacker. For finding masquerade attack the most useful algorithm is semi global alignment (SGA). Existing approaches like Enhanced-SGA, HSGAA for detecting attacks have not attained the level of precision and performance. This paper introduces Optical-Semi Global Alignment Algorithm (OPSGA), which enriches the alignment scoring system and transformation of matrix for getting better alignment helps to increases both computational performance and accuracy of the system. Once the system finds out there are a mismatch alignment areas in user sequences we can tag them as misbehave users, it can be masquerade attack if there is strong indication of many mismatch areas within the user sequence. OPSGA bear small changes in the user sequence that are happened by small behavioral changes in user while accessing commands in the system. OPSGA system will help to increase hit ratio, decrease the runtime overhead of system and also reduce both false negative and false positive rates.

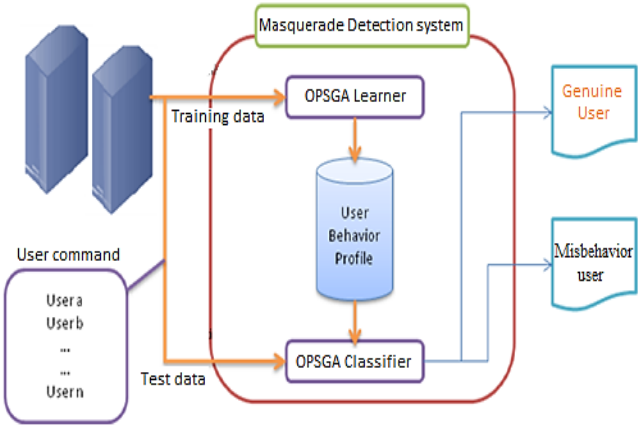


Fig 1: OPSGA System Architecture Overview

2. RELATED WORK

Meanwhile, masquerading is performing a significant part of computer intrusions, masquerade detection is now gaining awareness for security researches. We describe some masquerade detection approaches. A practical method for identifying masquerade in uncontrollable state over the use number of statistical methods was presented in their work [11]. They used singularly of command in succession of command line entries as an unusual metric function. The inherent though was that decriminalize series of command line data must be reliable with the commands originated in the user signature and slight variation or difference would signify feasible masquerade uncontrollable condition or attack. This approach has several limitations such as refuse to acknowledge sequence of information by assuming command or uncontrollable condition of independency, refuse to acknowledge the command serving for capable purpose and refusing fluctuation in human behavior by overly penalize slightly variation from past command line entries. An idea of matching two identical by making an effort to lexically match sequel of the user signature with sequel of the observed session and matched to generate a similarity metric by using the number of commands which was presented by Lane and Brodley [12]. The method declines the characteristic, however, the command in sequence and trust instead on determining precise lexical matches.

Stolfo *et al.* proposed detection mechanism that analyses the traffic flowing out of the network is an anomaly based. Each package is watched and evaluated with the stored pattern to notice the difference [13]. Distinguishing to this, mishandling or pattern detection methods

collect the signature of the recognized attacks in a database. Then the database entries to attain the pattern matching equated with the current session. But this method can only detect recognized attack patterns and are not for noticing new attacks that do not match with collected patterns is only the drawback of misuse detection approaches.

Lee *et al.* used two-class Naïve Bayes Classifier to notice masquerade attack that obtained a very good result [14]. The use of renewing methods that constantly update the classifier possibilities as observed sequences that are categorized in their work. Hence, this method adapts to change in user behavior. Even though commands are ignored, despite the better functioning of the classifier, sequencing data and the useful semantics.

Valdes and Skinner used two-class Naïve Bayes Classifier which was comparable with the results of applied one class Naïve Bayes and support vector machine classifiers [15]. But this procedure undergoes from the similar weak point as in which neglect to detect sequence and functionality information [14]. This method [15] however they did not show particular false positive and accurate detection scores and that's why making direct comparison to their techniques is improper. To analyze efficiency of masquerade detection and better method towards masquerade recognition is used in Hidden Markov Model (HMM) by Bhukya and Suresh [16]. Ordinary semi-global alignment was performing much better than then HMM model, it also has too less enhancements in masquerade detection.

3. SEMI GLOBAL ALIGNMENT (SGA)

This section provides description about SGA and some ways to enrich it. Current approaches are not precise and effective as much as Semi-global alignment. False alarm rates, low false positive and high hit ratio can be seen in semi-global alignment. It can be accepted in diverse background with different operating systems since different sample data is collected. For instance, command line entries, user mouse movements, instructions to system to perform task, registry of each event, keeping track of windows title opened by user, changes in files and folder names. Collecting data from network access what user has visited and what time. SGA tries to align large sequence from the given sequence to give the best possible output as in global alignments, as per maintaining the true form of local alignments. The SGA tries to align the specific area with maximum similarity in the sequences. SGA can pass over both prefixes and suffixes in the sequence, if unnecessary in inspecting similarity. An example of SGA with important parameters for an alignment specifically: Signature gap penalty,

Mismatch score, test gap penalty, Match score and detection threshold is shown in figure 2 [1].

To get the best arrangement of sequences, SGA uses dynamic programming. For this reason, it sets an m+1 by n+1 matrix score (M), and then the value of each cell of matrix is decided by one of three conditions, see figure 3 [1].

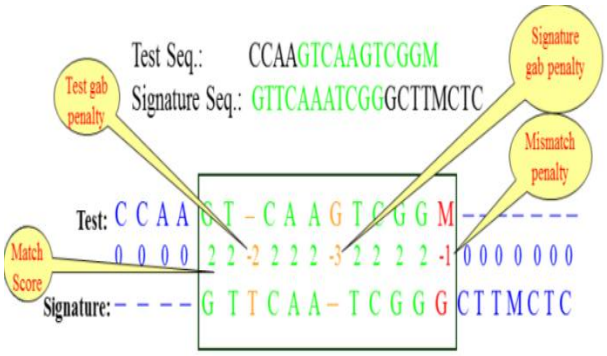


Fig 2: Example of Semi-global alignment algorithm

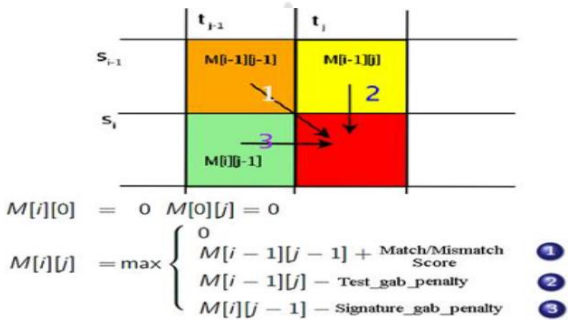


Fig 3: Transition matrix cells are filled by using 3 moves

We go over each position and value is calculated by three conditions according to that position.

1] Diagonal Move: Signifies a position between the (i-1) character in user signature sequence with the (j-1) character in user test sequence. Scoring system will first look up for any similarity in both sequences and then alignment score will be added to matrix position at (i-1, j-1).

2] Vertical Move: Denotes the position of gap with (j-1) character in the test sequence and insertion of gap into signature sequence. Scoring system will calculate the gap penalty for this move. Then at matrix location (i, j-1) the value of gap penalty will be inserted.

3] Horizontal Move: Denotes the position of a gap with (i-1) character in signature sequence and insertion of gap into test sequence. The value of gap penalty for this move will be calculated by scoring system similar to vertical move. Then at matrix location (i-1, j), the value of gap penalty will be inserted.

4. OPTICAL SEMI GLOBAL ALIGNMENT (OPSGA)

OPSGA is used for detecting masquerade attacks based on Enhanced-SGA [3]. It lineups the current user session which has sequence of activities and commands with the previous sequence of the same user and if the alignment is not matching then the region is denoted as anomalous. A masquerade attack is indicated depending upon the user threshold values, if the proportion of anomalous region is greater than a normal then masquerade attack is signaled. OPSGA can allow small changes in the user command sequence and then it is processed by three phases, configuration phase, detection phase and update phase. The position of each user command order is designed in the configuration phase, and these values are handed over to the detection and update phase. In detection phase, genuine user signature sequence is tested with the current user profile sequence which is logged in into the system. There are two methods that are Top-Matching Based Overlapping and Parallelized detection module, these are used to achieve better results in detection phase. Update phase is used to update user signatures in the database with new patterns in their sequence. Figure 4 shows three phases and its sub-modules.

In all pairwise algorithms are used to calculate the sequence alignment between two sequences. The two sequence which will be used by algorithm can be of unequal length. These algorithms try to align the sequence in best possible manner. The most effective are semi-global alignment rather than global and local alignment algorithm. To get possible alignment we need to calculate transition matrix for both sequences using dynamic programming. But when these sequences are aligned in best possible manner the alignment is always found in diagonal area of transition matrix. This diagonal alignment pattern is found in all the pairwise alignment algorithms as shown in figure 5. So why not to focus on diagonal area of matrix only to align those two sequences. It will also help to do less calculations in transition matrix.

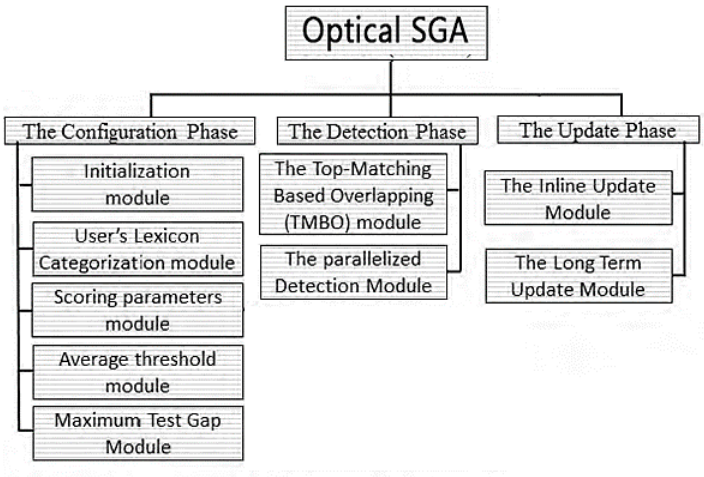


Fig 4: OPSGA Phases and Modules

In our proposed optical semi-global alignment algorithm (OPSGA), we need to calculate only values of diagonal area rather than calculating full transition matrix values. From this transition matrix, we get sequence alignment with the help of trace backing. Let's consider an example, in which sequence 1 and sequence 2 are of length 12. In dynamic programming, if we leave 1st row and 1st column of matrix, remaining each cell value of matrix is decided by three conditions and from that highest value is selected. As per early alignment algorithm, we need to calculate 144 cell values, in which 1st row and 1st column cell values are calculated easily. But for rest 121 cells values the algorithm has to perform 3 operations to get 1 value for each cell. To fill full transition matrix, 23 values (1st row & 1st column) + 121 × 3 = 23 + 363 = 386 total operations has to be performed as shown in figure 6.

If we consider calculating diagonal elements only then it will require only 102 operations for 34 cells to fill diagonal cell values. Number of operations will increase if we increase number of cells in diagonal area, but still, the number of operations will be less than half. That's why we don't need to calculate full transition matrix for generating score value for each cell, instead of that we calculate only values which come under diagonal pattern as shown in figure 7.

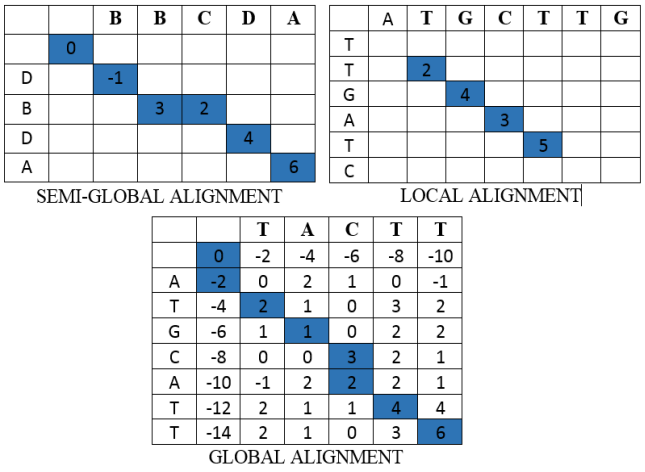


Fig 5: Different types of pairwise alignment in diagonal pattern

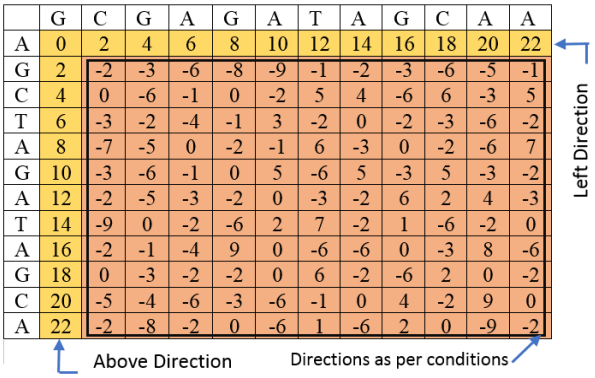


Fig 6: Transition matrix with 144 cells values

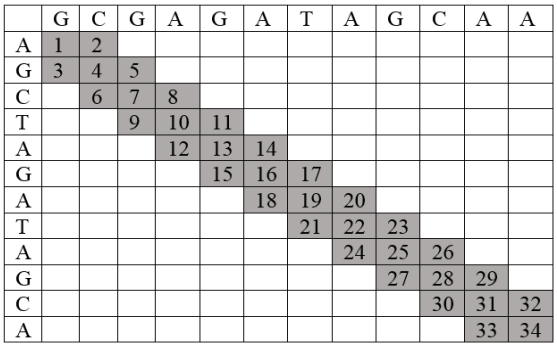
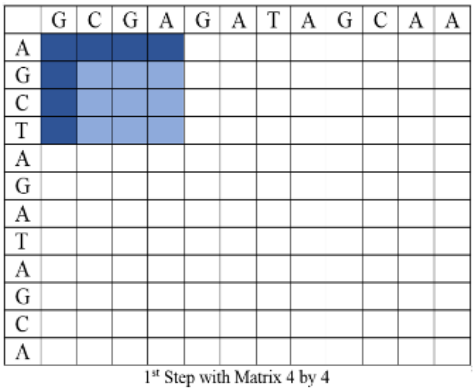


Fig 7: Sequence alignment values in diagonal pattern

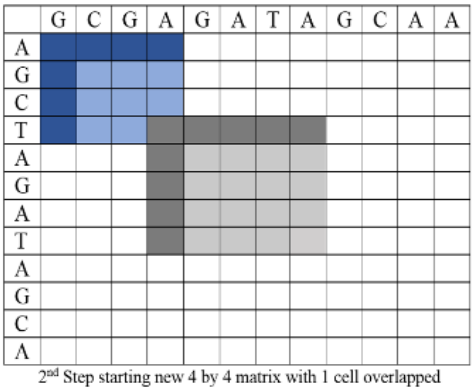
In early example of two sequences of length 12, we will try to align them with the help of optical semi global alignment (OPSGA). 1st our objective is to divide the sequences into equal parts. If our sequence length is of 12 we can divide it into 3 parts of each length 4. But we know sometimes the sequences come to align are of unequal length, in that case, we can

put “GAP” at the end or start or at both sides of the sequence to make them equal as much as possible. It will help to divide the sequences equally.

We can then start calculation in fixed size of 4 by 4 matrix. We will consider first 4 sequence characters of both string then apply global sequence alignment algorithm, row and column will be calculated by adding gap value rest 3 by 3 matrix cell values are computed by selecting maximum value by three conditions described in section 3. Once we calculated 1st 4 by 4 matrix we can start calculating 2nd 4 by 4 matrix. 2nd matrix will start from the last element of 1st matrix value. We are overlapping 1 cell between 1st matrix and 2nd matrix it will help in backtracking as we are splitting sequence in parts. In backtracking directions are very important while trace backing the sequence from last point to start point, therefore 1st row and 1st column of formed matrix will have directions left and above respectively whereas overlapped cells will have direction diagonal and remaining cell directions will be calculated as per from the cell value is coming it can be from above, left, or diagonal direction. Likewise, we can go on till the full sequence is processed in diagonal pattern with the help of this method. We have used global alignment algorithm as it can form matrix with the help of single value and gap score as shown in figure 8. For instance, selecting two sequences of length 12 if we try to find out sequence alignment through any pairwise alignment method we will have to calculate values for 144 cells. Whereas if we go through optical semi global alignment method we have to calculate values for only 64 cells. We can save computation time as we are skipping calculation of 144 - 64 = 80 cells, with this, runtime overhead of the system will be affected. Thus, it will help to improve the system efficiency by using less time in finding sequence alignment, shown in figure 9.



1st Step with Matrix 4 by 4



2nd Step starting new 4 by 4 matrix with 1 cell overlapped

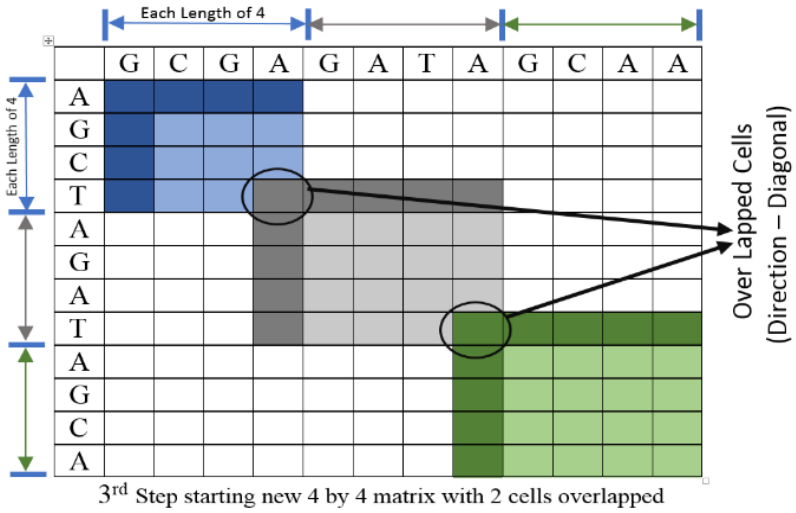


Fig 8: OPSGA 3 steps of 4 by 4 transition matrix

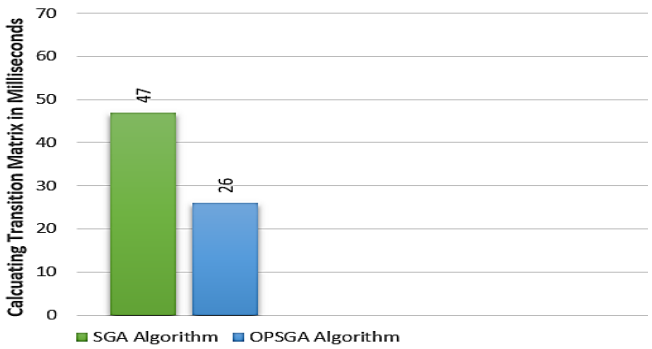


Fig 9: Formation of transition matrix by SGA & OPSGA

We have done more additions to optical semi global alignment, as compared to early sequence alignment algorithms, gap score was taken as fixed value for computations for all the individual user. If the gap score is kept constant, we will not be able to reduce false positive and false negative rates for better detection of masquerade attacks. Obviously, Behavior varies as user or human changes. Because of this Optical-semi global algorithm is made to calculate the appropriate gap score for specific user behavior pattern. Similar gap scores for all users will not be returned by the algorithm, whereas scores which have dissimilar values will correspond to distinct behaviors of users. Therefore, changing the gap value as per number of match or mismatch within the sequence. For example, if match value is greater than 10 then we will assign gap value as -2 or if mismatch value is greater than 10 and match value both then gap can be assigned as +2 as per this gap value can be changed as per given conditions as shown in figure 10.

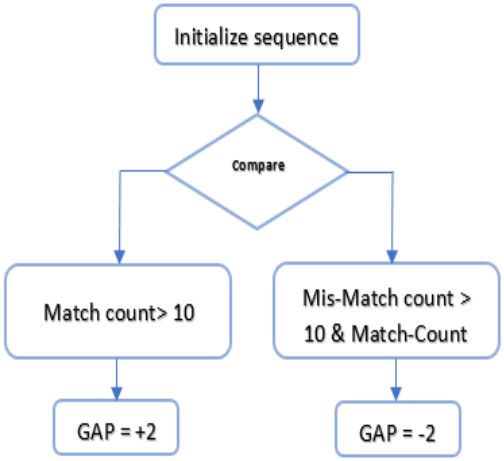


Fig 10: Change of GAP value

Instead of fixing the early scoring system, optical semi global algorithm the scoring function is calculated by sequence of commands collected by genuine user of the system through which the algorithm gives score value and as human behavior pattern are not same for all the person the score value of each individual will be different. The optical-Semi global algorithm runs on the principle by choosing the highest value from all calculated scores and it also decides the gap value that has generated the highest scores. For the alignment of sequence highest score signifies as optimal score shown in figure 11. Optical Semi global algorithm is useful to examine and find out the appropriate scoring system for specific users. This also will reflect chances of decreasing false negative and false positive alarms.

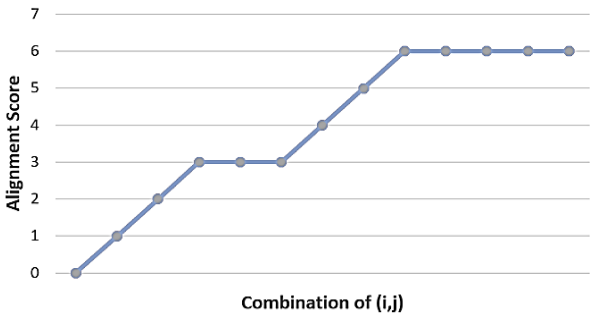


Fig 11: Illustration of highest point

4.1 THE CONFIGURATION PHASE

This phase processes parameters for each user within the system and these values are used by the subsequent sections within it.

4.1.1 OPSGA initialization module

In Configuration phase, Separate block of test sequence and signature sequence are provided in this module. The user signature sequence (nt) which is provided will be divided into many non-overlapped blocks of length n and that will be used as test sequence further for particular user. The sequence which is created by all the possible combinations of user sequence will be used by each part in the configuration phase. As per match or mismatch among the test sequence and signature sequence it calculates suitable scoring alignment.

4.1.2 User Lexicon grouping module

This section, for each user lexicon profile, is created, i.e. as per their functionality the lexical patterns are categorized and these are used to bear changes in the user behavior pattern. Pattern of user behavior consists of UNIX commands which we get in Purdue and SEA dataset. The command categorizing and user lexicon list are joined together in this section using the approach introduced in [3].

4.1.3 Scoring parameter module

Each user profile has test sequence and signature subsequence from starting. Mismatch score, optimal signature gap penalty and optimal test gap penalty these three parameters are returned by this module. OPSGA allows to record and aligns the test sequences which has top match score only rather than checking all the “nt” sequences. The top match list is created by selecting the highest match score for all “nt” sequences. Match score MS is calculated for each test sequence using feasible gap penalty and the overlapped signature subsequence “ns” [1].

4.1.4 Average threshold module

For every user behavior, an average threshold value is calculated which is used in detection phase and also it can be further updated in detection phase. If the alignment score of user behavior has greater value than the threshold value in detection phase then user is not a masquerade attacker, or else user is attacker if value is less than threshold. This value is calculated by taking all the previous alignment scores of test sequence [1].

4.1.5 Maximum factor of test gap

This section relates to find the maximum number of gaps which can be added into the user's test sequence alignment. To calculate the maximum factor of test gap, we can divide the maximum test gap module, according to the average threshold module that will add gaps into the test sequences of user. Further, these values will be used by detection phase.

Table 1. Comparing other current detection approaches

Method Name	Hit Ratio %	False positive %	Maxion-T Cost
OPSGA (Change in GAP value)	93.4	2.9	30.0
DDSGA (Restricted Permutation) [1]	83.3	3.4	37.1
SGA (signature updating) [4]	68.6	1.9	42.8
SGA (signature updating) + Heuristic [4]	66.5	1.8	44.3
Naïve Bayes (with update) [11]	61.5	1.3	46.3
SGA (Binary scoring) [4]	60.3	2.9	57.1
Adaptive Naïve Bayes [17]	87.8	7.7	58.4
Naïve Bayes (No updating) [11]	66.2	4.6	61.4
Episode based Naïve Bayes [19]	77.6	7.7	68.6
Hybrid Markvo [18]	49.3	3.2	69.9
SGA (traditional scoring) [3]	75.8	7.7	70.4
Sequence Matching [12]	36.8	3.7	85.4

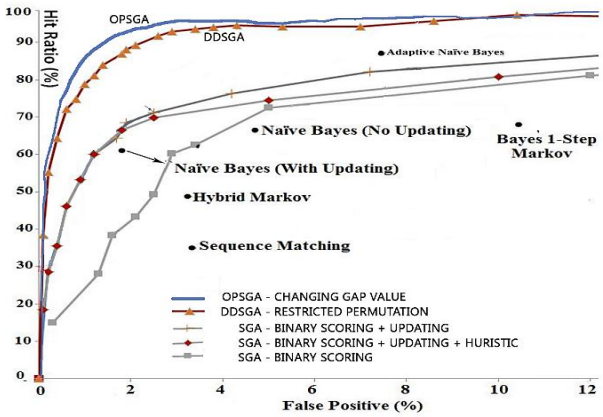


Fig 12: Effect of TMBO approach on system performance

4.2 THE DETECTION PHASE

Detection phase has two modules namely "Top Match based overlapping Module" and "Parallelized Detection Module". Our main aim is to check effect on the alignment factors of hit ratio, false negative and false positive.

4.2.1 Top Matching Based Overlapping (TMBO)

In this module, the current user sequence pattern is overlapped with the user signature sequence pattern. The TMBO selects the subsequence pattern with the highest match value with the help of user test sequence and signature sequence. The TMBO model is processed in three stages. In first stage, TMBO calculates the length which is used for overlapping sequence. In second stage, The TMBO shows the overlapped sequence matching among the user test sequence and signature sequence. And in third stage, TMBO selects the sequence which has highest match score value.

Table 2. TMBO Approach in Masquerade Detection

Method Name	Avg-n-align	NAC(1 User)	NAC (50 user)
OPSGA (L = 144)	4.7	4.7*144*100 = 67,680	67,680*50 = 3384000
DDSGA (L = 145.73) [1]	5.13	5.13 * 145.73 *	74759.49 * 50 = 3737974.5

		100 = 74759.49	
SGA - Heuristic Aligning (L = 200) [4]	4.5	4.5 * 200 * 100 = 90000	90000 * 50 = 4500000
SGA – Traditional (L = 200) [4]	49	49 * 200 * 100 = 980000	980000 * 50 = 49000000

Table 3. Masquerade detection approach against OPSGA

Method Name	Hit Ratio %	False positive %	Maxion-T Cost
OPSGA (Without updating)	90.4	3.4	30.0
OPSGA (Without Updating + TMBO)	88.6	3.2	30.6
DDSGA (Without updating) [1]	83.3	3.4	37.1
DDSGA (Without updating + TMBO) [1]	81.5	3.3	38.3
SGA (signature updating) [4]	68.6	1.9	42.8
SGA (signature updating) + Heuristic [4]	66.5	1.8	44.3
Naïve Bayes (With updating) [11]	61.5	1.3	46.3
SGA (Binary scoring + No updating) [4]	60.3	2.9	57.1

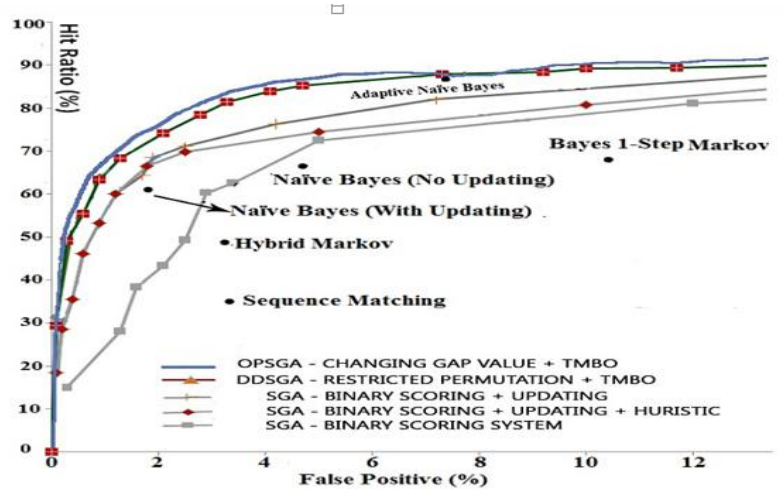


Fig 13: Effect of TMBO approach on system performance

4.2.2 Parallelized detection module

In parallelized detection module, we are performing the threads in parallel same time. If score value is equivalent to detection threshold rate then message is displayed as "No masquerade attack". Else if the value is less than detection threshold then that specific thread throws a message to systems as user is "Masquerade attacker". Figure 14 shows detection time of masquerade dataset [6] which has 50 user's files and figure 15 shows detection time of SEA dataset [1]. The experiments were taken on Windows 7 SP1 operating system, which was having Intel Core i5-2450M 3rd generation processor with speed of 2.5 GHz with 4 cores and 12 GB of RAM which is machine 1. Whereas machine 2 is running Intel Core i3-3110M with 2.40 GHz with 4 cores and 6 GB of RAM.

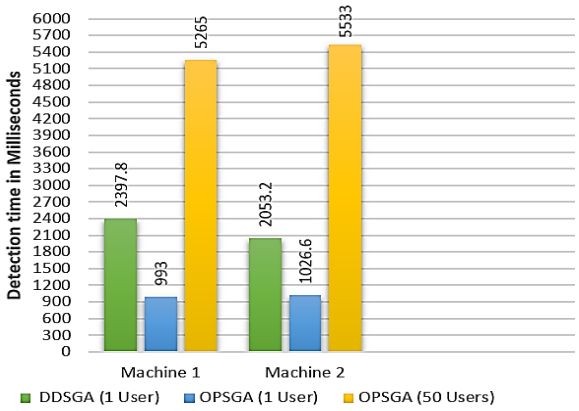


Fig 14: Detection time in Machine A & Machine B without threading (Using Masquerade Dataset)

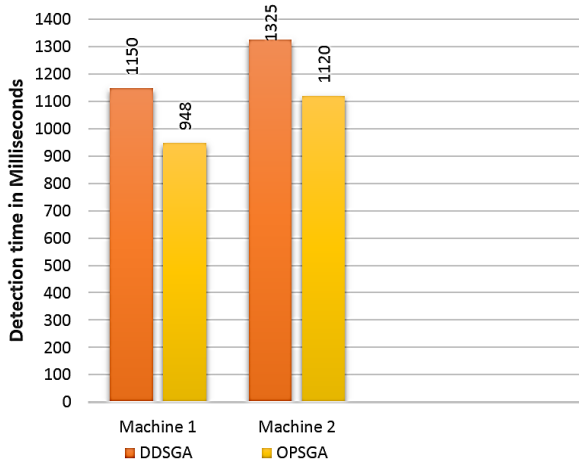


Fig 15: Detection time in Machine A & Machine B without threading (Using SEA Dataset)

4.3 THE UPDATE PHASE

The update phase is important as it updates the user signature pattern, intrusion detection system (IDS) helps to update the new behavior pattern of genuine user automatically. The update phase can be applied by using two modules: the inline update and the long-term update module.

4.3.1 The inline update module

Inline update module has two significant tasks:

a] Searching regions in user signature sequence to be updated and extended the sequence with the new user behavior pattern.

b] When adding new commands the user lexicon list should be updated.

There are three situations possible in Trace backing algorithm:

1] The pattern of user test sequence should match equivalent signature pattern of genuine user.

2] In test sequence and signature sequences a gap is inserted or it is inserted into any one of the sequences.

3] The two sequences can have at least mismatch values between the patterns.

In situation 1, as the alignment has properly matched with the sequences with the best optimal alignment score has therefore update is not necessary. Even in situation 2, update is not necessary as the symbols of sequences are properly arranged with the gaps are not similar and it must be ignored. In situation 3, test sequences of current instances will collect all the mismatches within it. Then two conditions are used to update the signature sequence and user lexicon. According to first condition, we can insert those patterns which don't come under masquerading records into the user signature. The alignment score value for current results should be larger or else equivalent to the value of detection threshold. As per another condition, the present pattern must occur in the user lexicon or it must fit into similar functional group of the signature pattern. These two situations indicate that it will update new pattern into user lexicon if they don't belong to lexicon. The user signature is increased by inserting resulting subsequence in the signature pattern without modifying the original pattern shown in figure 16 and performance affecting system accuracy after updating shown in figure 17.

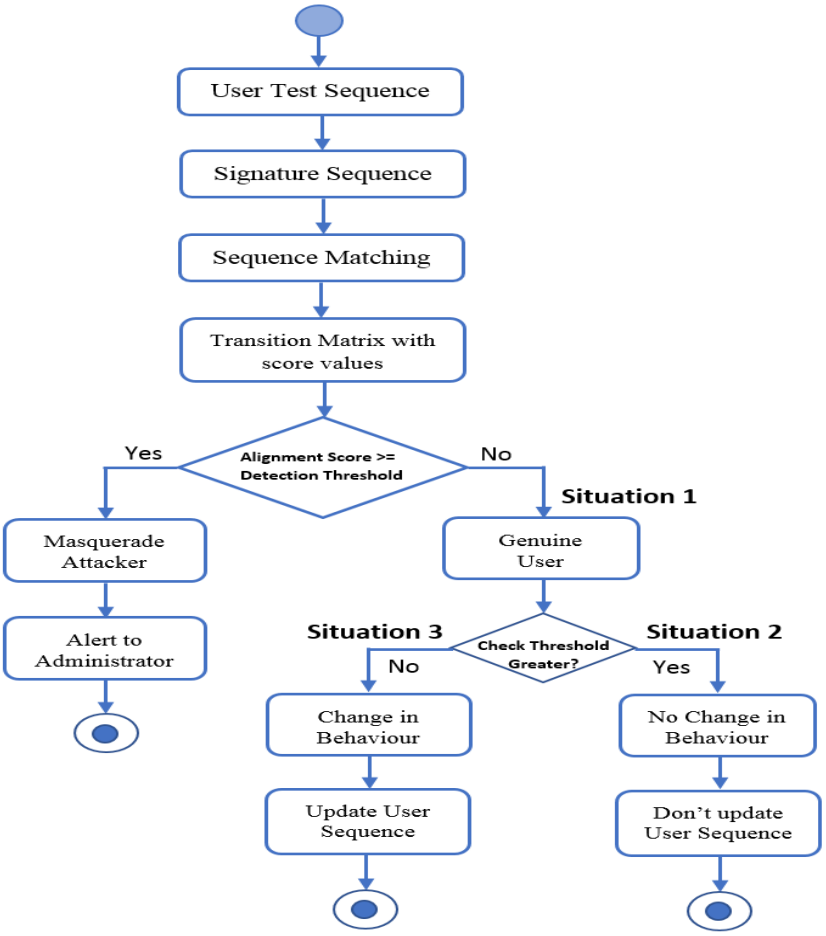


Fig 16: Update user sequence as per situations

Table 4. Results of masquerade detection methods

Method Name	Hit Ratio %	False positive %	Maxion-T Cost
OPSGA (Inline update + TMBO + Signature update)	95.4	2.7	23.8
DDSGA (Inline update + TMBO)	88.4	1.7	37.1
DDSGA (Without updating)	83.3	3.4	38.3
DDSGA (Without updating + TMBO)	81.5	3.3	42.3
SGA (signature updating)	68.6	1.9	42.8
SGA (signature updating) + Heuristic	66.5	1.8	44.3
Naïve Bayes (With updating)	61.5	1.3	46.3
SGA (Binary scoring + No updating)	60.3	2.9	57.1

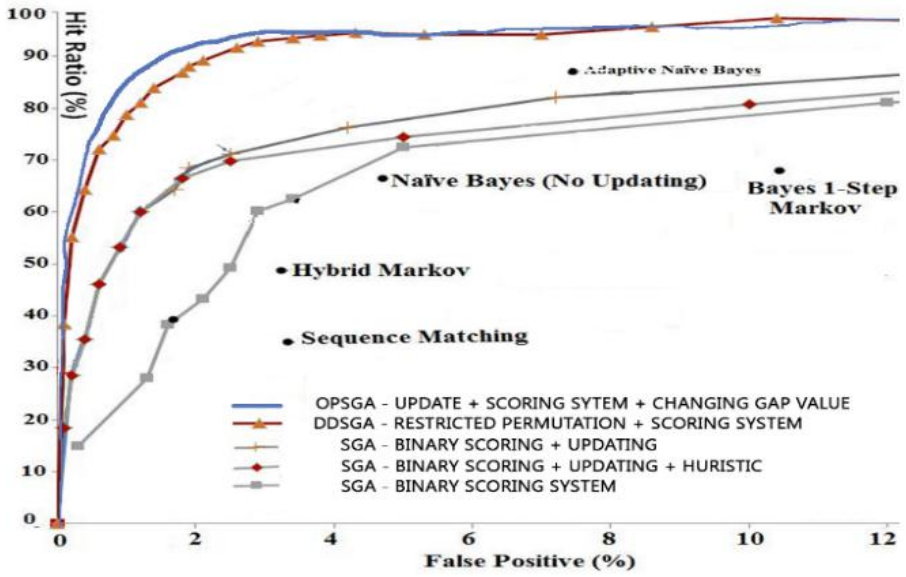


Fig 17: Affecting system accuracy after updating

4.3.2 Long term update

In this module, the system is reconfigured such that it will update in timely manner. To run this module there are three plans: Cyclic, inactive time, and threshold. The monitored system selects the proper plan according to specific necessities.

The cyclic plan proceeds with fixed rate like 4 days, 1 week or 1 month to update the system. To decrease workload of the system, the inactive time plan proceeds when the system is inactive to update the system. This solution is used when systems are highly loaded, they use more CPU and network bandwidth. The threshold plan proceeds when it reaches a threshold value while adding the test patterns into the signature sequence. When the signature sequence is modified at that time only it runs the module, therefore, this method is very effective as shown in figure 18.

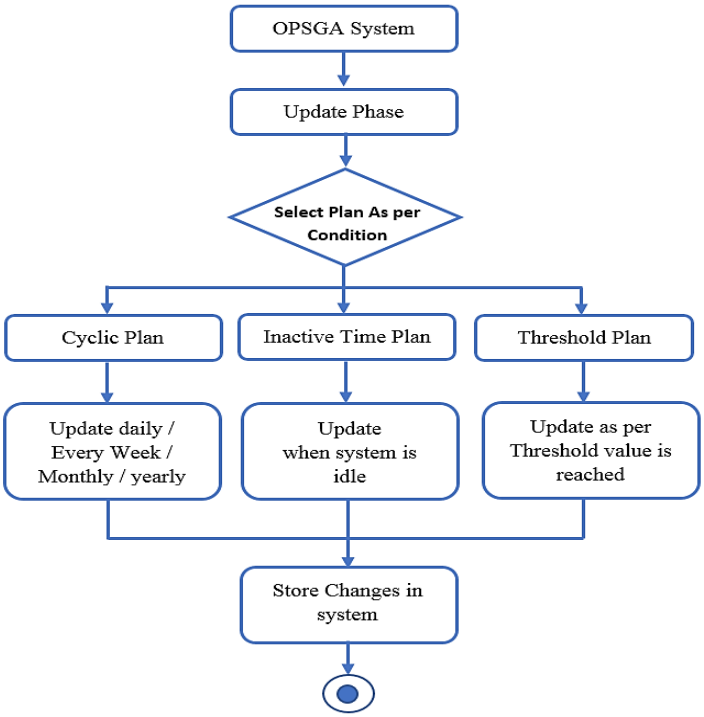


Fig 18: Selecting update plan as per condition's

5. IMPLEMENTATION OF SYSTEM

SNo	User's FullName
1	dhanu dhanu
2	abcd abcd
3	pqr pqr
4	Rohit u
5	ravi u

SNo	Id	Signature Sequence	Test sequence	Login Session	
1		AAAFDEBGDBAECF	AAFDEGBF	08 Mar 2017 10:13PM	Compare
2		AAAFDEBGDBAECF	EFABGCFD	08 Mar 2017 10:14PM	Compare
3		AAAFDEBGDBAECF	BBBCAEFDGAD	08 Mar 2017 10:15PM	Compare
4		AAAFDEBGDBAECF	AEFDBCGBCEFD	08 Mar 2017 10:19PM	Compare
5		AAAFDEBGDBAECF	AEBBFDEGBBA	08 Mar 2017 10:20PM	Compare

Fig 19: User test & signature sequence as per user login

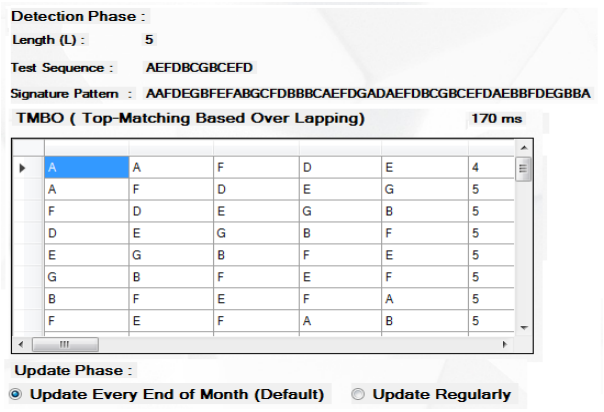


Fig 20: Detection (TMBO) & Update Phase

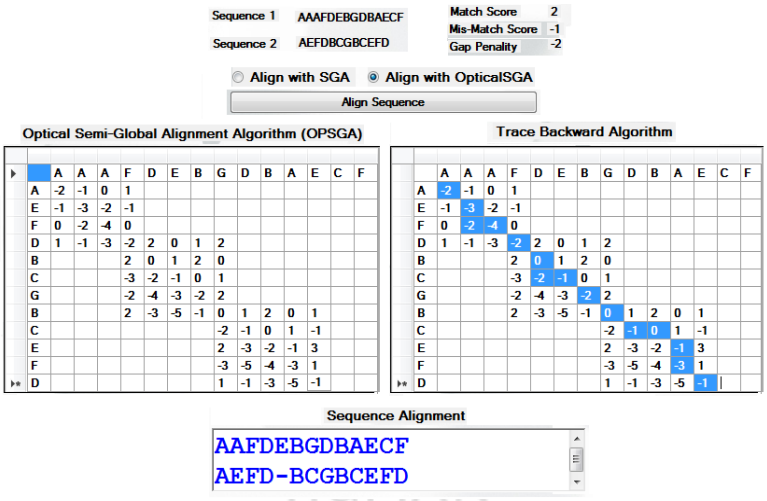


Fig 21: Sequence Alignment using SGA & Optical-SGA

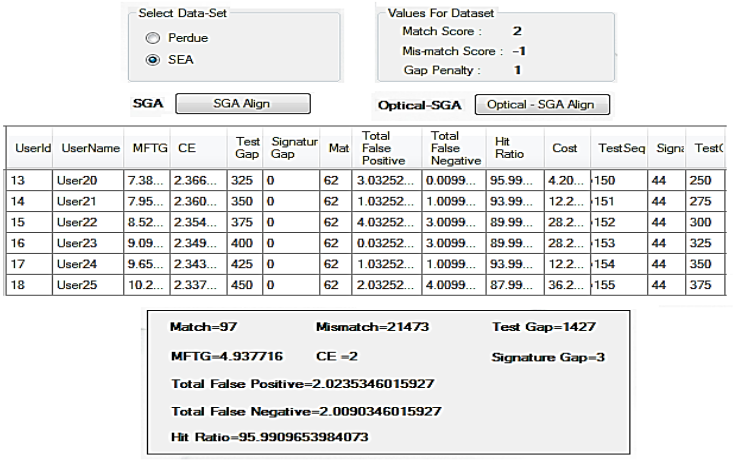


Fig 22: Using dataset in SGA & Optical-SGA phases

6. CONCLUSION

Masquerade attacker can enter into system by logging into it successfully and then it can control the system as per his needs, that's why masquerading come under most dangerous attacks. The semi-global alignment comes under dynamic programming that is based on sequence alignment. This technique is used for detection which separate sequences of audit data. Whereas SGA and Enhanced-SGA have not reached the level of accuracy and its functioning, thus the motive of our work is to find better algorithm in place of Semi-global alignment algorithm as highlighted in this work and particularly in the work of [3,4]. This is the main purpose to create Optical Semi-Global Alignment Approach (OPSGA). As per the protection point of view, OPSGA model shows more precision towards the behavior of different users by initiating different factors. Moreover, the main system presents two scoring method that bears modifications even if changes are minor within the command functionality by classifying user commands and positioning them in proper way within the same group without decreasing the alignment score. The scoring system can bear variations in commands as well as alterations in the user behavior as per time changes. All these aspects help to decrease false alarm rates and false positive rates and better hit ratio for detection of attacker which results in better performance of OPSGA system than DDSGA [1].

REFERENCES

[1] Hisham a. Kholidy, Fabrizio Baiardi, and Salim Hariri. "DDSGA: a data-driven semi-global alignment approach for detecting masquerade attacks," IEEE transactions on dependable and secure computing, vol. 12, no. 2, April 2015.

[2] A. H. Phyo and S. M. Furnell. "A detection-oriented classification of insider it misuses," In Proceedings of the Third Security Conference, 2004.

[3] Coull, S. E., Branch, J. W., Szymanski, B. K., and Breimer, E. A. "Intrusion Detection: A Bioinformatics Approach," 19th Annual Computer Security Applications Conference, Las Vegas, NV, December 2003, pp 24-33.

[4] Coull, S. E., Szymanski, B. K. "Sequence alignment for masquerade detection," Journal of Computational Statistics & Data Analysis, April 2008, pp 52.

[5] Hisham A. Kholidy, Fabrizio Baiardi. "CIDS: A Framework for Intrusion Detection in Cloud Systems", 9th Int. Conf. on Information Technology: New Generations ITNG, Las Vegas, Nevada, USA, April 2012, pp 16-18.

[6] Schonlau, M., DuMouchel, W., Ju, W., Karr, A. F., Theus, M., and Vardi, Y. "Computer Intrusion: Detecting Masquerades," Statistical Science 16(1), 2001, pp 58-74.

[7] Hisham. A. Kholidy, Fabrizio Baiardi. "CIDD: A cloud intrusion detection dataset for cloud computing and masquerade attacks," 9th International Conference on Information Technology: New Generations ITNG, Las Vegas, Nevada, USA, April 2012, pp 16-18.

[8] R.A.Maxion, T.N. Townsend. "Masquerade Detection Using Truncated Command Lines", International Conference on Dependable Systems and Networks, Washington, D.C., June 2002.

[9] R.Posadas, J.C. Mex-Perera, R. Monroy, J.A. Nolazco-Flores. "Hybrid method for detecting masqueraders using session folding and hidden markov models", In Gelbukh, Garcia, eds: MICAI. LNCS 4293, Springer, 2006, pp 622-631.
[10] Dumouchel, W. "Computer intrusion detection based on Bayes Factors for comparing command transition probabilities", Technical Report 91, National Institute of Statistical Sciences, 1999.
[11] S. Noel, D. Wijesekera, and C. Youman, "Modern intrusion detection, data mining, and degrees of attack guilt," Applications of Data Mining in Computer Security, Kluwer, 2002, pp. 2-25.
[12] T. Lane, and C. E. Brodley, "Sequence matching and learning in anomaly detection for computer security," Proceedings of the AAAI-97 Workshop: AI approaches to fraud detection and risk management, vol. 49, 1997, pp. 43-49.
[13] S. J. Stolfo, A. L. Prodomidis, S. Tselepis, W. Lee, D.Fan, and P. K. Chan, "JAM: Java agents for meta-learning over distributed databases," Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, Aug. 1997, pp. 74-81.
[14] W. Lee, S. J. Stolfo, P. K. Chan, E. E. Wofan, M. Miller, S. Hershkop, and J. Zhang, "Real time data mining-based intrusion detection," Proceedings of DISCEX II, June 2001, pp. 89-100.
[15] A. Valdes and K. Skinner, "Probabilistic alert correlation," Recent Advances in Intrusion Detection (RAID 2001), LNCS 2212, Springer Verlag, Davis, California, Oct. 2001, pp. 54-68.
[16] W. N. Bhukya and K. G. Suresh, "A study of effectiveness in masquerade detection," EIGAR Best Paper Proceedings, 1999, pp. 34-50.
[17] Subrat Kumar Dash, Krupa Sagar Reddy, Arun K. Pujari, "Adaptive Naive Bayes method for masquerade detection," Security and Communication Networks 4(4), 2011, pp. 410-417.
[18] Ju, W. and Vardi, Y., "A hybrid high-order Markov chain model for computer intrusion detection," Technical Report 92, National Institute Statist. Science, 1999.
[19] Dash, S.K., Reddy, K.S., Pujari, A.K., "Episode based masquerade detection," Lecture Notes in Computer Science, Springer, Berlin, 2005, vol. 3803, pp. 251–262.