An Improved Semi-Global Alignment Algorithm for Masquerade Detection

Adesina Simon Sodiya, Olusegun Folorunso, Saidat Adebukola Onashoga, and Omoniyi Paul Ogunderu
(Corresponding author: Adesina Simon Sodiya)

Department of Computer Science, University of Agriculture, P. M. B. 2240, Abeokuta
(Email: sinaronke@yahoo.co.uk, {folorunsolusegun, bookyy2k, omoniyiogunderu}@yahoo.com)

(Received Dec. 14, 2009; revised and accepted Apr. 7, 2010)

Abstract

Masquerading is a security attack in which an intruder assumes the identity of a legitimate user. Semi-global alignment algorithm has been the best of known dynamic sequence alignment algorithm for detecting masqueraders. Though, the algorithm proves better than any other pairwise sequence alignment algorithms such as local and global alignment algorithms, however, the problem of false positive and false negative have not been reduced to the barest minimum. Many previous works on masquerade detection using sequence alignment have difficulty at choosing the scoring system on which the algorithms base their optimal scores on. Hence, they resolved to assuming (or picking) a set of scores which they referred to as a unique scoring function for their experiment. In this work, an improved semi-global alignment called Cross-semiglobal algorithm, is designed to improve the efficiency of masquerade detection. In the previous pair-wise algorithms, a fix value is always assumed as the gaps score. In Cross-semiglobal algorithm, the scoring function on which the algorithms based their scores is constructed from legitimate users' sequence of commands. This principle was implemented using platform independent C/C++ framework. The designed was tested using a systematically generated ASCII coded sequence audit data from Windows and UNIX operating systems as simulations for standard non-intrusive and intrusion data. The result shows a reduction in false positive rate from 7.7% using semi-global alignment to 5.4% using cross-semiglobal. The detection efficiency was also improved by 7.7%.

Keywords: Cross-semiglobal algorithm, gaps scores, masquerading, sequence alignment, semi-global algorithm

1 Introduction

Intrusions on computer infrastructures are now growing problems [29]. In the field of computer security, one of the most damaging attacks or intrusion is masquerading, in which an attacker assumes the identity of a legitimate user in a computer system. Masquerade attacks typically occur when an intruder obtains a legitimate user's password or when a user leaves their workstation unattended without any sort of locking mechanism in place. It is difficult to detect this type of security breach at its initiation because the attacker appears to be a normal user with valid authority and privileges. This difficulty underlines the importance of equipping computer systems with the ability to distinguish masquerading attacker actions from legitimate user activities [6].

Forecasting the unknown and detecting the known threats and targeted attacks are the most concern of network security especially in large scale environment [1]. The information security industry has been very active in recent years. In order to counterwork security threats to computer systems and networks, many technologies have been developed and applied in security operations such as Intrusion Detection System (IDS), firewalls, routers. All those security application devices, whether aimed at prevention or detection of attacks, usually generate huge volumes of security audit data [37]. The traditional form of IDS and prevention systems are either signature-based or anomaly-based. Both require updates to maintain their signature database or they must have a period of time to develop a behavioral baseline to identify accurately "suspicious" or anomalous activities [1, 16].

The detection of a masquerader relies on a user signature, a sequence of commands collected from a legitimate user. This signature is compared to the current user's session. The underlying assumption is that the user signature captures detectable patterns in a user's sequence of commands. A sequence of commands produced by the legitimate user should match well with patterns in the user's signature, whereas a sequence of commands entered by a masquerader should match poorly with the user's signature. Designing algorithms to distinguish legitimate users and masqueraders based on user signatures has been extensively studied [7, 21].

In the past, sequence alignment algorithms such as global, local and semi-global alignments have been used for detecting masquerading. Out of these algorithms, the most efficient is semi-global alignment. The problem with use of semi-global alignment lies in determination of the best scoring system used by the algorithm. Other works on semi-global alignment, had resolved to a fixed scoring system and this scheme is repeated for all individual. However, in this work, instead of fixing a scoring system, we designed an algorithm, called Cross-semiglobal algorithm to compute the best scoring scheme for semi-global alignment. These scores are not expected to be the same for all persons since human patterns of behaviors are not the same. This method of variant scoring system provides an improvement to semi-global alignment algorithm for efficient masquerading detection. This will go along way reducing the number of false negative and false positive alarms.

The rest of this paper is organized as follows: Section 2 presents related works on masquerade and intrusion detection using sequence alignment method. The improved semi-global alignment called "Cross-semiglobal algorithm" is presented in Section 3. The implementation procedure and evaluation of Cross-semiglobal algorithm are well enumerated in Section 4. Section 5 presents future works and conclusion.

2 Literature Review

Since intrusion detection field started with the work of [2], many techniques have been used to design Intrusion Detection Systems (IDS). Some of the early techniques used in designing IDS are statistics [18, 34], Neural network [9, 14, 21], Data mining [12, 21, 24] and Expert System IDIOT [20], ASAX [23], DIDS [27], IDES [10], and NIDES [3]. Most of the early expert system-based IDSs are for misuse detection. Some other early works are [11, 13, 15]. More recently is the application of mobile agent (MA) to intrusion detection because of the problems with centralized systems. Some of the MA-based IDSs are MA-IDS [30], IDA [4], Micael [8], Sparta [19] and MSAIDS [29].

However, since masquerading is a significant part of computer intrusions, masquerade detection is now gaining the attention of security researchers. A technique for detecting masquerade attacks through the use of a number of statistical methods was presented in the work of [25]. They used the uniqueness of a command in a sequence of command line entries as an anomaly metric. If a particular command was rarely used previously, its score would be proportionally lower than a command that was used more often. The underlying idea was that legitimate sequences of command line data should be consistent with the commands found in the user's signature and any deviation would indicate possible masquerade attacks. Of course, this approach has several shortcomings such as ignoring sequencing information by assuming command independence, ignoring command functionality, and ignoring variations in human behavior by unduly punishing any change from past command line entries. Lane and Brodley presented a string matching approach by attempting to lexically match subsequences of the users' signature with subsequences of the monitored session and used the number of commands that were matched to create a similarity metric [21]. The method proposed in [21], like [25], ignores the underlying functionality of the commands in the sequences, relying instead on finding exact lexical matches.

For early detection of network flooding and intrusion and non-intrusive packet monitoring, Stolfo et al. proposed an anomaly based detection mechanism that analyses the traffic flowing out of the network [31]. In their work, each packet is monitored and compared with the stored patterns to discover the anomaly. In contrast to this, misuse-based or pattern detection approaches store the signatures of the known attacks in a database. Then the current traffic is compared with the database entries to find the patterns matching. The obvious drawback of misuse detection approaches is that they can only detect known attack patterns and are not for detecting new attacks that do not match with stored patterns. Their implementation adopted an anomaly-based approach that uses information entropy to detect DoS attacks using the information present in the packets and the source IP address as parameters to detect anomaly. The objective of the proposed system is to prevent the suspicious packets from flooding the victim.

There have also been several attempts at applying more advanced machine learning techniques to the problem of masquerade detection. Lee et al. provided a very good result by using a two-class Naive Bayes classifier to detect masqueraders [22]. The most important contribution made in their work is the use of updating mechanisms that dynamically update the classifier probabilities as monitored sequences are classified. Thus, this approach adapts to changes in user behavior. However, despite the improved performance of the classifier, sequencing information and the functional semantics of the commands are ignored. Valdes and Skinner applied one-class Naive Bayes and Support Vector Machine classifiers, and find that their results were comparable to those of the two-class classifiers [35]. This approach, however, suffers from the same weaknesses as the [22] approach by ignoring sequence and functionality information. Unfortunately, [35] did not provide specific false positive and true detection scores, thereby making direct comparison to their techniques impossible. Subramanian and Angamuthu also used a one-class Support Vector Machine and implemented a novel recursive data mining strategy to perform dimensionality reduction [32]. Unlike the Support Vector Machine of [32, 35] did provide some considerations for sequencing information in their dimensionality reduction technique; however, functionality is ignored as is the possibility of variation in user behavior.

In the work of Bhukya and Suresh, Hidden Markov Model (HMM) was used to compute and formulate the effectiveness of masquerade detection and also to present a highly effective approach to masquerade detection [5]. They made use of the Schonlau Dataset, abbreviated to

SEA data. The SEA dataset has been one of the notable data source for testing the effectiveness of intrusion detection systems or techniques. Their approach seems effective than some of the earlier approaches that implores hidden markov model analysis.

Several other bioinformatics tools have been applied to computer security problems. Wang et al. were among the first to consider the use of bioinformatics techniques beyond biological data when they applied the TEIRESIAS pattern discovery algorithm to sequences of system call data [37]. This algorithm finds recurring patterns of maximal length sequences and uses these recurring patterns to build a database of valid system call sequences. More recently, Szymanski and Zhang used the concept of motifs or conserved areas of recurring behaviors, to discover anomalies within sequences of audit data [33]. Wepsi et al. used Hidden Markov Models, which are typically used to align many biological sequences at once, to detect the presence of various application protocols within encrypted tunnels [38]. Another related bioiformatic work is that of [1]. The paper presented an intrusion detection and prediction system using cooperative co-evolutionary immune system for distributed data networks. This was an intelligent technique based on genetic algorithm and co-evolutionary immune system where the detectors can discriminate the existing incidents and predicting the new incidents in a distributed environment [1]. They prepared a prototype of CoCo-IDP in a Jini platform running grid computing in distributed systems. Evaluation results show that, the CoCo-IDP can adaptively converge for the best answer and can detect or predict the incidents in a selected boundary. Moreover, the system generates the flexible detectors with diversity in a variable threshold. In comparison with pure Immune System (IS), the obtained results show that the proposed system has simpler rules, more powerful detection and prediction capabilities with high accuracy metric.

This work is mainly concerned with improving on existing sequence alignment techniques for masquerade detection. Previous works, using sequence alignment approach, have fixed arbitrary scores as the algorithm scoring parameters. Two previous works in which fixed parameters were used for scoring system, were presented in [6, 7]. The scoring scheme in [6] used $(-2, -3)$ combination to represent penalties for presence of gaps in the test block and the signature block respectively. The Binary Scoring and Command Grouping scoring systems newly proposed in [7] also did not put variant human behaviors into considerations. These scoring systems contradict the fundamental norm that "an individual behavior is distinct, so all persons pattern of behavior cannot be scored using the same scoring function". In this work, however, apart from the Match and Mismatch scores being kept at 1 and 0 respectively, (this score is a standard for intrusion detection algorithms using dynamic programming), the gaps scores are computed by Cross-semiglobal algorithm in order to take into consideration changes in users' behaviors. This algorithm, for detection of the best gaps scores for a particular training of an intrusion detection system on a normal user sequence of commands, coupled with the semi-global alignment in particular, yields a considerable improvement in masquerade detection than ordinary semi-global alignment.

3 Methodology

Sequence alignment algorithms require six parameters to execute the pair-wise alignment. The parameters are the User Block, the Test Block, the Match Score, the Mismatch Score, the User Gap Score and the Intrusion Gap Score. Out of these parameters, the two gaps scores will be determined by the Cross-semiglobal algorithm. These gaps scores coupled with the Match and Mismatch scores are then used for future alignments of test block on normal user block. These four scores are normally referred as to as the Scoring System.

3.1 The Scoring System

The Match Score is a positive score for reward of perfect match of commands in the test sequence and the user signature. The Mismatch Score is a penalty for any mismatch in the command sequence of the test block and the user signature. The intrusion gap and user gap scores are penalties for presence of gaps within the test block and the signature block respectively.

This system of combination of the four parameters is used by alignment algorithm to compute the optimal score. The optimal score is a value function from the alignment of the test block and the signature sequence. This score is then compared with the threshold score to determine if the test block is an intrusion block or not.

In this work, the Match and Mismatch Scores, by convention, are taken to be 1 and 0 respectively. Nonetheless, the Intrusion Gap and Signature Gap are left to Cross-semiglobal alignment to determine from the training of the intrusion detection system on the normal user threshold sequence of commands.

3.2 Legitimate User and Masquerader Behaviour Patterns

An intrusion detection system based on pattern matching of user behaviors has to be trained with the sequence of audit generated from the user behavioral pattern of approach towards the information system. This process of training the masquerade detection system (or IDS) is described as the Learning Phase of the system. Once the system has learnt the behavior pattern of the legitimate user and provided a score (often referred to as threshold score) for his behavior based on the audit data, subsequent activities of the user on the system will then be audited, scored and the optimal score will be compared with the user threshold score. The underlying assumption is that, the optimal score of the normal user activities will be within the neighborhood of the threshold score. While the score of a masquerader will show a significant deviation from the normal user threshold score.

3.3 Semi-Global Alignment Algorithm

In this context, emphasis is made wholly on the works presented in Coull et al. on Semi-global alignment algorithm [6, 7]. In their works, the signature sequence, which represents the user's typical command behavior, is referred to as the UserSig. The monitored command sequence, which may contain a possible subsequence of masquerader commands, is referred to as the IntrBlck (tested block).

Algorithm 1 Semi-global alignment algorithm

```
Begin
Align(userSig of length $m$, intrBlck of length $n$, matchScore, misMatchScore, gapSigScore, gapTestScore)
for $i = 0$ to $m$ do
  for $j = 0$ to $n$ do
    if $i = 0$ or $j = 0$ then
      $D[i][j] = 0$
    else
      if $i = m$ or $j = n$ then
        $top = D[i][j-1]$
        $left = D[i-1][j]$
      else
        $top = max(0, D[i][j-1] + gapSigScore)$
        $left = max(0, D[i-1][j] + gapTestScore)$
      end if
      if $Signature[i-1] = Test[j-1]$ then
        $diagonal = D[i-1][j-1] + match$
      else
        $diagonal = D[i-1][j-1] + mismatch$
      end if
      $D[i][j] = max(top, left, diagonal)$
    end if
  end for
end for
return $D[m][n]$
End
```

The alignment algorithm, Algorithm 1 uses dynamic programming to discover the optimal alignment among all possible alignments. It begins by initializing an $(m+1)$ by $(n+1)$ matrix, called D. Starting at position $(0, 0)$ (i.e., the upper left corner) in the matrix, we iterate through each position whose value is determined through a choice of three transitions to that position:

- **Diagonal Step:** Indicates an alignment between the $(i-1)$ symbol in Signature with the $(j-1)$ symbol in Test. The alignment score added to the value of the matrix position at $(i-1, j-1)$ measures the level of alignment of the symbols defined in the scoring system, denoted as diagonal.
- **Vertical Step:** Indicates the insertion of a gap into Signature, and alignment of the gap with the $(j-1)$ symbol in Test. The gap penalty is added to the value of the matrix position at $(i, j-1)$, denoted as top. The gap penalty for this transition is dependent on the scoring system used.
- **Horizontal Step:** Indicates the insertion of a gap into Test, and alignment of the gap with the $(i-1)$ symbol in Signature. The gap penalty is added to the value of the matrix position at $(i-1, j)$, denoted as left. As with the vertical step, the gap penalty for this transition is dependent on the scoring system used.

The maximum value of these three possible transitions is used as the value for the current matrix position and indicates the actual alignment made. Thus, given the dynamic programming principle, each position, (i, j), in the matrix represents the score of the optimal alignment of all symbols up to location $(i-1)$ in Signature and $(j-1)$ in Test. By induction, the score given in position (m, n) represents the score of the optimal alignment of the two sequences given the scoring system, and by tracing the transitions made in deriving that score we can recreate the alignment of the two sequences. The resultant score at the (m, n) position of the matrix represents a metric for the similarity of the two strings according to the scoring system used. We use this score as an indicator for masquerade attacks.

3.4 Cross-Semiglobal Algorithm

The Cross-Semiglobal algorithm is presented in this work to improve on the method for determining optimal score for different user behaviors. It plays an important role in the determination of the penalties for the inclusion of gaps in the commands blocks. The Cross-Semiglobal algorithm works on the principle of selecting the highest of all scores and determines the gap scores that produce the highest score. The highest score then represents the optimal score for that alignment.

During the training of the masquerade detection system, Cross-Semiglobal algorithm is applied to analyze and determine the suitable scoring system for a particular user. The algorithm for Cross-Semiglobal alignment is stated as follows:

The Cross-semiglobal algorithm first makes several calls to the ordinary semi-global alignment while varing the values of i and j to compute the score for each combination of (i, j). It then proceeds futher to compute the (i, j) combination that produces the maximum score, the first occurrence of such (i, j) combination is returned as the *PeakPoint*. This then makes the value of i and j as penalties for inclusion of gaps in the user block and the test block respectively.

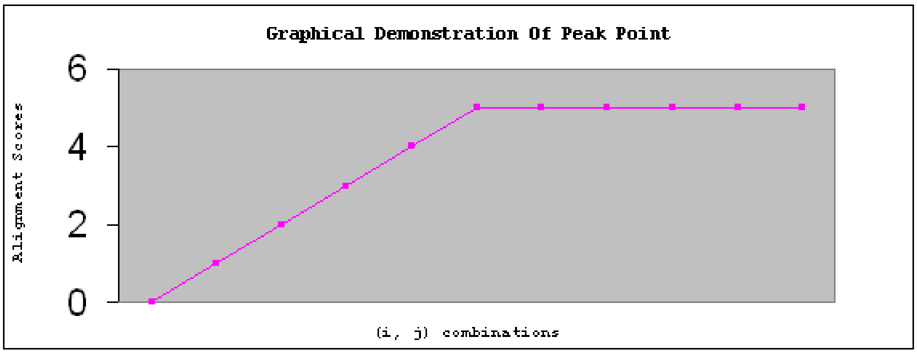


Figure 1: Demonstration of peak point

Algorithm 2 Algorithm Cross-Semiglobal

```
Begin
// the list of method parameters are given below:
userSig: normal user signature.
testSig: monitored command sequence for intrusion.
matchScore: the score for matches during the alignment.
misMatchScore: the score for mis-matches during the alignment.
$i$: the score for the userGap.
$j$: the score for the intrusionGap.
OptimalValue$(i, j)$: two dimensional array for storing alignment scores corresponding to $(i, j)$ combinations.

// the algorithm logic is described below:
for $i = -n$ to -1 step 1 do
  for $j = -n$ to -1 step 1 do
    $OptimalValue(i, j)$=$SemiglobalAlignment(userSig, testSig, matchScore, misMatchScore, i, j)$
  end for
end for
// determining the peak point$(i, j)$
for $i = -n$ to -1 step 1 do
  for $j = -n$ to -1 step 1 do
    if $(max(OptimalValue(i, j)) = OptimalScore)$ then
      $PeakPoint = (i, j)$
    end if
    return $PeakPoint$
  end for
end for
End
```

3.5 The Presence of Gaps in Blocks

The pairwise alignment algorithm, or dynamic alignment algorithms, splits the sequences of commands into all possible permutations. In these arrangements, gaps are included to cater for dissimilarity of human pattern of behaviors amongst the sequences merged to compute the threshold sequence. Hence, the penalty for the presence of anomaly in patterns is severely high.

3.6 Indeterminate Gap Scores

For effective masquerade detection and reduced False Positive and False Negative rates, we cannot make the gap scores constant. Naturally, human behaviors are different. This is why Cross-semiglobal algorithm is implored to compute the gaps score suitable for that particular user behavior. The algorithm will not return the same gaps scores for all persons; there will be dissimilarities in the scores which is commensurate with the distinct behaviors of users.

3.7 The Optimal Gaps Scores

Since Cross-semiglobal algorithm has $i = -n$ to -1 and $j = -n$ to -1 where $i = user\ gap$ and $j = intrusion\ gap$, sets of combination of (i, j) will always return the same value corresponding to the optimal score of the alignment. However, as more and more the value of (i, j) increases after the first occurrence of the optimal score, no such combination of (i, j) will give a value greater than the optimal score even if $n \rightarrow -\infty$.

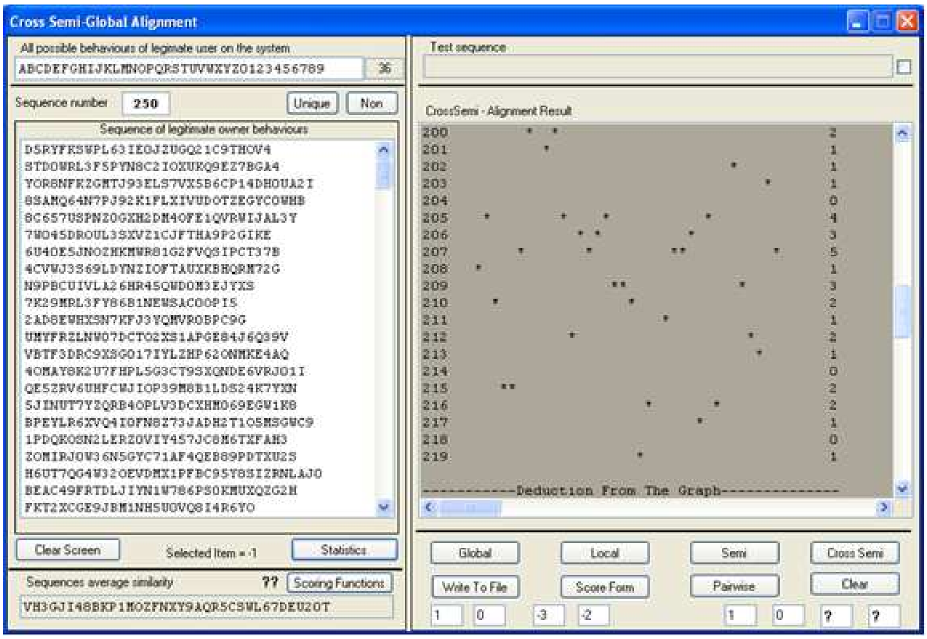
Hence, the Cross-Semiglobal algorithm is designed to return the (i, j) combination, which satisfies the condition of the optimal score that took place. The returned (i, j) is termed the peak point.

3.8 Graphical Demonstration of Peak Point

Figure 1 depicts how the peak point is attained by the Cross-Semiglobal algorithm in determining the best gaps scores.

4 Implementation Procedure and Evaluation

The implementation of the new algorithm was demonstrated using C/C++ Application Programming Interface due to its support for Object Oriented design, portability and user interface flexibility.



All possible behaviours of leginate user on the system

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 36

Sequence number 250

Sequence of leginate owner behaviours

```
D5RYFKSWPL63IEGJ2UGQ21C9THOV4
8TDOWRL3F5PYN8C2IOZUKQ9EZ7BGA4
YOR8NFKZGMTJ93ELS7VX5B6CP14DHOUA2I
8SAMQ64N7PJ92K1FLXIVUDOTZEGYCOWHB
8C657USPNZ0GXH2DM4OFE1QVRWIJAL3Y
7WO45DROUL3SXVZ1CJFTHA9P2GIKE
6U4OE5JNOZHKMWR81G2FVQSIPCT37B
4CVWJ3S69LDYNZIOFTAUXKBHQRM72G
N9PBCUIVLA26HR45QWDOM3EJYXS
7K29MRL3FY86B1NEWSACOOPI5
2AD8EWHXSN7KFJ3YQMVROBPC9G
UMYFRZLNW07DCTO2XS1APGE84J6Q39V
VBTF3DRC9XSG017IYLZHP62ONMKE4AQ
4OMAY8K2U7FHPL5G3CT9SXQNDE6VRJO1I
QE5ZRV6UHFCWJIOP39M8B1LDS24K7YXN
5JINUT7YZQRB4OPLV3DCXHMO69EGW1K8
BPEYLR6XVQ4IOFN8Z73JADH2T1O5MSGWC9
1PDQKOSN2LERZ0VIY457JC8M6TXFAH3
ZOMIRJOW36N5GYC71AF4QEB89PDTXU2S
H6UT7QG4W32OEVDMX1PFBC95Y8SIZRNLAJ0
BEAC49FRTDLJIYN1W786PS0KMUXQZG2H
FKT2XCGE9JBM1NHSUOVQ8I4R6YO
```

Selected Item = -1

Sequences average similarity ??

VH3GJI48BKP1MOZFNXY9AQR5CSWL67DEU20T

Test sequence □

CrossSemi - Alignment Result

```
200        *  *                                      2
201          *                                       1
202                                    *             1
203                                         *        1
204                                                  0
205  *          *     *             *                4
206               *  *            *                  3
207       *        *           **            *       5
208 *                                                1
209                    **                *           3
210   *                  *                           2
211                            *                     1
212              *                        *          2
213                                        *         1
214                                                  0
215    **                                            2
216                         *        *               2
217                                *                 1
218                                                  0
219                       *                          1

------------Deduction From The Graph--------------
```

1 0 -3 -2 1 0 ? ?

Figure 2: Cross-semiglobal alignment evaluation interface

4.1 Data Source

We collected ASCII coded sequence data generated from Windows and UNIX operating systems as our normal and test block sequences respectively. Different platforms audit logs were chosen because of the disparity in the event logging principles. The user repeated similar pattern of activities on the two operating systems. The implemented system has the capability to audit user activities on a system and convert the audited data to an ASCII coded sequence data. By imitating the normal user behavior pattern, masquerader sequence data were simulated. The implemented system can generate over 99999 ASCII coded sequence data from the systems audit log.

4.2 Gui Design and Partition

The GUI is divided into two compartments. The first compartment is for the demonstration of the training of the intrusion detection system and the other is for the evaluation of intrusion data and statistics of result.

Figure 2 shows a description of the interface.

4.3 Traning the System

The masquerade detection system was first trained with the normal user sequences of behavior to generate a threshold sequence, threshold score and scoring system for the user. The sequence of behaviors is a subset of possible behaviors of the user on the Windows operating system. Secondly, UNIX behavior sequence was also used to train the system for comparison. The two generated threshold sequences were compared and evaluated.

4.4 Threshold Sequence and Score

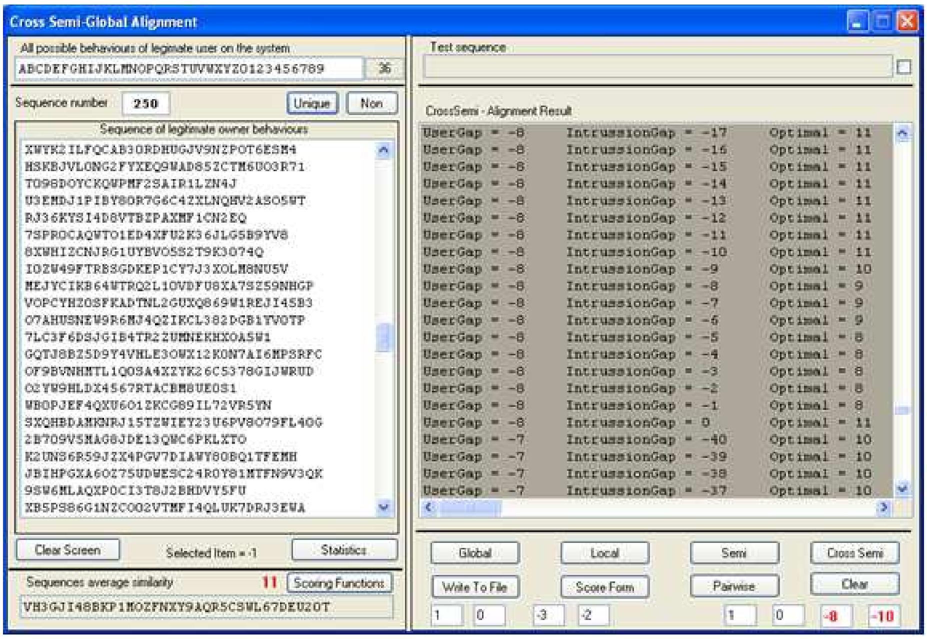
The threshold sequence serves as the master sequence generated which contains all the similarity between the normal user sequences supplied for the training of the masquerade detection system. All other sequences of activities audited on the system against the user session will be compared (sequentially aligned) with the threshold sequence. The scores of this comparison are computed using the scoring system of the user whose session was used. The disparity of the alignment optimal score with the threshold score will indicated the occurrence of instruction (masquerading) or not.

Figure 2 shows sequences of normal user behaviors supplied as training data for the intrusion detection. On the left side is the user sequences used to train the system. The Statistics button generates the threshold sequence for that training. On the right is a scatter curve depicting the graphical behavior of the characters forming the threshold sequence. The threshold sequence generated was: VH3GJI48BKP1MOZFNXY9AQR5CSWL67DEU20T.

4.5 Cross-Semiglobal Scoring Functions

The Scoring Function button generates the best scoring functions for the training. This best scoring function is the peak score already described. For the training described in Figure 3, the best scoring function is (-8,-10) and the optimal score of alignment of the threshold with the possible sequence is 11.

In Figure 3, the value labelled red represents the optimal score and the peak scores. The result of the Cross-Semiglobal computation is displayed on the right side of the application interface.



All possible behaviours of leginate user on the system: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 36

Sequence number: 250

Sequence of legitimate owner behaviours

```
XWYK2ILFQCAB3ORDHUGJV9NZPOT6ESM4
HSKBJVLONG2FYXEQ9WAD85ZCTM6UO3R71
TO98DOYCKQWPMF2SAIR1LZN4J
U3EMDJ1PIBY8OR7G6C4ZXLNQHV2ASO5WT
RJ36KYSI4D8VTBZPAXMF1CN2EQ
7SPROCAQWTO1ED4XFU2K36JLG5B9YV8
8XWHIZCNJRG1UYBVO5S2T9K3O74Q
IOZW49FTRBSGDKEP1CY7J3XOLM8NU5V
MEJYCIKB64WTRQ2L1OVDFU8XA7SZ59NHGP
VOPCYHZOSFKADTNL2GUXQ869W1REJI45B3
O7AHUSNEW9R6MJ4QZIKCL382DGB1YVOTP
7LC3F6DSJGIB4TR2ZUMNEKHXOASW1
GQTJ8BZ5D9Y4VHLE3OWX12KON7AI6MPSRFC
OF9BVNHMTL1QOSA4XZYK26C5378GIJWRUD
O2YW9HLDX4567RTACBM8UEOS1
WBOPJEF4QXU6O1ZKCG89IL72VR5YN
SXQHBDAMKNRJ15TZWIEY23U6PV8O79FL4OG
2B7O9V5MAG8JDE13QWC6PKLXTO
K2UNS6R59JZX4PGV7DIAWY8OBQ1TFEMH
JBIHPGXA6OZ75UDWESC24ROY81MTFN9V3QK
9SW6MLAQXPOCI3T8J2BHDVY5FU
XB5PS86G1NZCOO2VTMFI4QLUK7DRJ3EWA
```

Selected Item = -1

Sequences average similarity 11

VH3GJI48BKP1MOZFNXY9AQR5CSWL67DEU20T

Test sequence □

CrossSemi - Alignment Result

```
UserGap = -8     IntrussionGap = -17     Optimal = 11
UserGap = -8     IntrussionGap = -16     Optimal = 11
UserGap = -8     IntrussionGap = -15     Optimal = 11
UserGap = -8     IntrussionGap = -14     Optimal = 11
UserGap = -8     IntrussionGap = -13     Optimal = 11
UserGap = -8     IntrussionGap = -12     Optimal = 11
UserGap = -8     IntrussionGap = -11     Optimal = 11
UserGap = -8     IntrussionGap = -10     Optimal = 11
UserGap = -8     IntrussionGap = -9      Optimal = 10
UserGap = -8     IntrussionGap = -8      Optimal = 9
UserGap = -8     IntrussionGap = -7      Optimal = 9
UserGap = -8     IntrussionGap = -6      Optimal = 9
UserGap = -8     IntrussionGap = -5      Optimal = 8
UserGap = -8     IntrussionGap = -4      Optimal = 8
UserGap = -8     IntrussionGap = -3      Optimal = 8
UserGap = -8     IntrussionGap = -2      Optimal = 8
UserGap = -8     IntrussionGap = -1      Optimal = 8
UserGap = -8     IntrussionGap = 0       Optimal = 11
UserGap = -7     IntrussionGap = -40     Optimal = 10
UserGap = -7     IntrussionGap = -39     Optimal = 10
UserGap = -7     IntrussionGap = -38     Optimal = 10
UserGap = -7     IntrussionGap = -37     Optimal = 10
```

1 0 -3 -2 1 0 -8 -10

Figure 3: Cross-semiglobal Intrusion Gaps computation

4.6 Sample Test

A typical test of ordinary Semi-global alignment using $(1, 0, -3, -2)$ combination according to [6] and $(1, 0, -8, -10)$ combination according to this work was carried out on a user pattern of behaviors on Windows while an equivalent pattern was generated on UNIX. On the windows operating system, VH3GJI48BKP1MOZFNXY9AQR5CSWL67DEU20T was computed as the threshold sequence. UNIX audit was used as the test blocks on which the degree of disparity is to be computed. Note that, approximately, same pattern of behavior were performed on the two platforms. But based on the mode of operation of these platforms, there is a degree of disparity. However, we expect this degree of dissimilarity not to throw an intrusion alarm if the alignment algorithms are efficient enough.

A gaps scores of $(-8, -10)$ were generated by the cross-semiglobal algorithm from the training of the system. Then pair-wise alignments were carried out using semi-global alignment with the ordinary score and cross-semiglobal algorithm score combinations. The differences were then computed for further computation.

The optimal scores of the alignments were:

- ORDINARY SEMI-GLOBAL ALIGNMENT: 4
- CROSS-SEMIGLOBAL ALIGNMENT: 8

The test results show that the Cross-semiglobal algorithm performed better than the ordinary Semi-global algorithm in the determining the best alignment score. The Cross-semiglobal uses the peak function as against the fixed scoring scheme presented in [6].

4.7 Experiment Metrics, Test and Parameters

This test focuses on the effects of changing the various parameters of the alignment algorithm on the false positive and false negative rates as it was done in [6, 7]. One of the benefits of this particular approach is the sheer number of tunable parameters.

To best facilitate comparison with other masquerade detection algorithms, False Positive Rate, False Negative Rate, and Hit Rate metrics were used to determine how well the Cross-semiglobal alignment algorithm performed. A False Positive is a non-intrusion block that the algorithm labelled as containing an intrusion. A False Negative is an intrusion block that the algorithm has labelled as non-intrusion. Finally, a Hit is an intrusion block that the algorithm has properly labelled as containing an intrusion. False Positives, False Negatives and Hits are computed for each user, transformed into corresponding rates, which are then summed and averaged over all 25 users.

The italicized statements below summarize the metric calculations used by the algorithm.

Metric 1: False Positive metric:

$$FalsePositive_{overall} \quad = \quad ([\sum_{i=1}^{n}(f_i/n_i)]/u) * 100,$$

where f denotes number of false positives, n denotes number of non-intrusion command sequence blocks, and u denotes number of users (25 in this case).

Table 1: Comparison of Semi-global and Cross-semiglobal alignments algorithms

Technique	**%Hit Rate**	**%False Positive**
(algorithm)	**(score)**	**(score)**
Semi-global	75.8	7.7
Cross-semiglobal	82.1	5.4

Metric 2: False Negative metric:

$$FalseNegative_{overall} \quad = \quad ([\sum_{i=1}^{n}(f_{ni}/n_i)]/u) * 100,$$

where f_n denotes number of false negatives, n denotes number of intrusion command sequence blocks, and c denotes number of users who have at least one intrusion block.

Metric 3: Hit Rate metric:

$$HitRate_{overall} = 100 - FalseNegative_{overall}.$$

It is suggested that other AI techniques such as Artificial Neural Network, Fuzzy Logic etc. can be used to improve the computation of the scoring system.

4.8 Comparison to Ordinary Semi-Global Alignment

This metric result presents the performance of an ordinary Semi-global alignment to that using Cross-semiglobal algorithm. For simplicity, Cross-semiglobal alignment is used to depict Semiglobal alignment using Cross-semiglobal algorithm.

The result above is interpreted to mean that Cross-semiglobal technique of scoring system is 8.3% more efficient than ordinary Semi-global alignment using a fixed scoring scheme as proposed in [6, 7].

5 Conclusion and Future Works

The motive that prompted this work is to find a better algorithm in place of the Semi-global alignment algorithm as emphasized in this work and especially in the works of [6, 7]. However, Cross-Semiglobal alignment algorithm was developed as an add-in algorithm to improve the effectiveness of Semi-global alignment.

It should be noted that Cross-Semiglobal algorithm does not directly align sequences but compute a unique scoring scheme peculiar to the training of the intrusion detection system on the defined user. These scores are then used for semi-global alignments which in-turn decrease the False Positive Rate and improve the Detection Rate.

References

[1] M. R. Ahmadi, "An intrusion prediction technique based on co-evolutionary immune system for network security (CoCo-IDP)," *International Journal of Network Security*, vol. 9, no. 3, pp. 290-300, Nov. 2009.

[2] J. P. Anderson, *Computer Security Threat Monitoring and Surveillance*, Technical Report Contract 79F26400, James P. Anderson Co., Box 42, Fort Washington, PA, 19034, USA, Feb. 26, revised Apr. 15 1980.

[3] D. Anderson, T. Frivold, Tamaru, and A. Valdes, *NIDES: Software Users Manual: Beta-update Release*, SRI International, Dec. 1994. (www.sdl.sri.com/papers/7sri)

[4] M. Asaka, S. Okazawa, A. Taguchi, and S. Goto, " A method for tracing intruders by use of mobile agents," *INET '99*, pp. 1-12, June 1999.

[5] W. N. Bhukya and K. G. Suresh, "A study of effectiveness in masquerade detection," *EIGAR 1999 Best Paper Proceedings*, pp. 34-50, 1999.

[6] S. Coull, J. Branch, B. Szymanski, and E. Breimier, "Intrusion detection: A bioinformatics approach,“ *Proceedings of the 19th Annual Computer Security Applications Conference*, pp. 24-33, 2003.

[7] S. Coull and B. Szymanski, "Sequence Alignment for Masquerade Detection," *Computational Statistics and Data Analysis*, vol. 52, no. 8., pp. 4116-4131, Apr. 2008.

[8] J. D. De Querioz, Da Costa Carmo, L. F. R., and L. Pirmez, "Micael: An autonomous mobile agent system to protect new generation networked applications," *2nd annual workshop on Recent Advances in Intrusion Detection*, vol. 10, no. 1, pp. 20-23 Sep. 1999.

[9] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," *Proceedings of the IEEE symposium on Research in Computer Security and Privacy*, pp. 240-250, Oakland, CA,1992.

[10] D. E. Denning and P. G. Neumann, *Requirement and Model for IDES - A Real-time Intrusion Detection Expert System*, Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA.

[11] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222-232, Feb. 1987.

[12] T. Fawcett and F. Provost, "Combining data mining and machine learning for effective user profiling," *Proceedings of the Knowledge Discovery and Data Missing*, pp. 7-13, 1997.

[13] S. Forrest, S. A. Hofmeyr, A. Somayagi, and T. A. Longstaff, "A sense of self for unix processes," *Proceedings of IEEE symposium on Research in Security and Privacy*, pp. 120-128, IEEE Computer Society Press, 1996.

[14] A. K. Ghosh, J. Wanken, and F. Charron, *Detecting Anomalous and Unknown Intrusion*

Against Programs, Reliable Software Technologies. (www.rstlotp.com)

[15] S. Hofmeyr, S. Forrest, and A. Somayaji., "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, pp. 151-180, 1998.

[16] K. Hwang, M. Cai, and Y. Chen, "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 1, pp. 41-55, 2007.

[17] K. Ilgun, "USTAT: A real-time intrusion detection system for UNIX," *Proceedings of IEEE Symposium on Research in Security and Privacy*, pp. 16-28, 1993.

[18] H. S. Javitz, and A. Valdes, *The NIDES Statistical Component: Description and Justification*, Technical report, SRI computer science Laboratory, Menlo Park, CA, March 1994. (http://www.sdl.sri.com/nides/index5.html)

[19] C. Kruegel, and T. Toth, *Applying Mobile Agent Technology to Intrusion Detection*, Technical Report Number TUV-1841-2002-31, Technical University of Vienna.

[20] S. Kumar, and E. H. Spafford, "A pattern matching model for misuse intrusion detection," *Proceedings of the 17th National Computer Security Conference*, pp. 11-21, 1994.

[21] 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, pp. 43-49, 1997.

[22] 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*, pp. 89-100, June 2001. (http://www.cs.columbia.edu/ids.2001)

[23] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 219-228, 2002.

[24] A. Mounji, *Languages and Tools for Rule-based Distributed Intrusion Detection*, PhD thesis, Faculte Universitaire Notre de la Paix de Namur, 1997.

[25] S. Noel, D. Wijesekera, and C. Youman, "Modern intrusion detection, data mining, and degrees of attack guilt," *Applications of Data Mining in Computer Security*, pp. 2-25, Kluwer, 2002.

[26] M. Schonlau, W. DuMouchel, W. H. Ju, A. F. Karr, M. Theus, and Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical Science*, vol. 16, no. 1, pp. 58-74, 2001.

[27] M. Schonlau, and M. Theus, "Detecting masquerades in intrusion detection based on unpopular commands," *Information Processing Letters*, vol. 76, no. 1, pp. 33-38, 2000.

[28] S. R. Snapp, S. E. Smaha, D. M. Teal, and T. Grance, "The DIDS (Distributed Intrusion Detection System) prototype," *Proceedings of the summer USENIX Conference*, pp. 227-233. 1992.

[29] A. S. Sodiya and H.O.D. Longe, "Critical analysis of anomaly-based intrusion detection techniques," *The Journal of Computer Science and it's Applications*, vol. 10, no 1, pp. 64-74, 2004.

[30] A. S. Sodiya, "Multi-level and secured agent-based intrusion detection system," *Journal of Computing and Information Technology*, vol. 14, no. 3, pp. 217-223, 2006.

[31] 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*, pp. 74-81, Newport Beach, CA, USA, Aug. 1997.

[32] M. Subramanian and T. Angamuthu,"An autonomous framework for early detection of spoofed flooding attacks," *International Journal of Network Security*, vol. 10, no. 1, pp. 39-50, Jan. 2010.

[33] B. K. Szymanski and Y. Zhang, "Recursive data mining dormasquerade detection and author identification," *Proceedings of the 5th Annual IEEE System, Man, and Cybernetics Information Assurance Workshop*, pp. 424-431, 2004.

[34] G. Tandon, P. Chan, and D. Mitram, "MORPHEUS: Motif oriented representations to purge hostile events from unlabelled sequences," *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, pp. 16-25, 2004.

[35] A. Valdes and K. Skinner, "Probabilistic alert correlation," *Recent Advances in Intrusion Detection (RAID 2001)*, LNCS 2212, pp. 54-68, Springer-Verlag, Davis, California, Oct. 2001.

[36] K. Wang and S. J. Stolfo, "One-class training for masquerade detection," *Proceedings of the 3rd IEEE International Conference on Data Mining Workshop on Data Mining for Security Applications*, 2003.

[37] L. Wang, A. Ghorbani, and Y. Li, "Automatic multi-step attack pattern discovering," *International Journal of Network Security*, vol. 10, no. 2, pp. 142-152, Mar. 2010.

[38] A. Wespi, M. Dacier, and H. Debar, "An intrustion-detection system based on the teiresias pattern-discovery algorithm," *EICAR 1999 Best Paper Proceedings*, pp. 1-15, 1999.

[39] C. V. Wright, F.Monrose, and G. M. Masson, "On inferring application protocol behaviors in encrypted network traffic," *Journal of Machine Learning Research*, vol.7, pp. 2745-2769, 2006.

Sodiya Adesina had M.Sc. in Computer Science from University of Lagos, Nigeria in 1999 and Ph.D in Computer Science from University of Agriculture, Abeokuta, Nigeria in 2004. He is presently a Senior Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta, Nigeria. His areas of research interests are computer security and software engineering. He has published about 30 publications in both national and international journals. He has also attended a number of conferences.

Folorunso Olusegun is presently a Senior Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta, Nigeria. His areas of research interests are Information Systems, Visualization and Security. He has published in national and international journals and conferences.

Onashoga Saidat Adebukola is presently a Lecturer in the Department of Computer Science, University of Agriculture, Abeokuta, Nigeria. Her areas of research interests include computer security and data mining. She has published in both national and international journals. She just completed her Ph.D. research work on intrusion detection system.

Ogunderu Omoniyi Paul received the B.Sc in Computer Science from University of Agriculture, Abeokuta, Nigeria in 2010. Currently he is the IT Project Manager, Software Developer and Researcher in an IT Project Management Firm, Scriptwares Solutions. His research interests are network security with a focus on Authentication and Intrusion Detection Systems and Software Engineering.