ABSTRACT
Masqueraders, despite widespread use of security products such as firewalls and intrusion detection systems, are serious threats to organizations. Although anomaly detection techniques have been considered as an effective approach to complement existing security solutions, they are not widely used in practice due to poor accuracy and relatively high degree of false alarms. In this paper, we performed an empirical study investigating the effectiveness of SVM and sequence-based kernel methods. Sequence-based kernel methods showed slightly better performance than generic RBF kernel with same frequency of false alarms. In addition, the composition of two kernel methods showed that frequency of false alarms could be further reduced.

Keywords
Masquerade Detection, Support Vector Machine (SVM), User Commands Profile, Anomaly Detection

1. INTRODUCTION
A masquerader is someone who impersonates other users. One must assume that sophisticated masqueraders possess insider’s knowledge on system topologies, potential system vulnerabilities, and how various security products have been installed to defined intrusions. Worse yet, authentication and access control mechanisms are highly unlikely to be effective because masqueraders already own sufficient information to pass the tests. Yet, from organizations processing sensitive intelligence information (e.g., CIA and FBI) to companies developing state-of-the-art technologies, threat by inside attackers is real and serious. There are numerous incidents of security policy violation by insiders[14]. According to the annual CSI/FBI computer security survey, insider abuse makes up 59% of all the reported attacks in 2004[3]. In addition, Gartner estimates that more than 70% of unauthorized access to information systems is committed by employees, as are more than 95% of intrusions that result in significant financial losses. Because insiders often know what security mechanisms are installed and where their respective weaknesses lie, violation of security policies and abuse of computer resources by inside users as a form of masquerade is a serious security threat.

In this research, we improve the performance of masquerade detection system with high detection rate by applying sequence-based information.

2. PREVIOUS WORKS
Schonlau et al.[11] built a data set based on UNIX user truncated commands and evaluated six different approaches to masquerade detection. Maxion et al.[9] showed better performance than those formed in Schonlau’s experiment by applying the Naive Bayes classification algorithm using the same data set. Maxion improved the accuracy while maintaining false alarm rate of about 1%. Unfortunately this level of detection is insufficient for a commercial product. Kim et al.[6] also showed good performance using SVM with RBF kernel by applying a common command scheme and a sliding windows method. They provided a reasonable hit ratio of 80.1%, but were hampered by a 9.7% false alarm rate, which must be decreased for a practical masquerade detection system. Wang et al.[13] used one-class SVM for only self-user training. Szymanski et al.[12] applied a recursive data mining algorithm. These two works used frequency-based information of user command types as a feature of user profiles. However, their results are no better than those produced by Kim’s experiment[6].

3. SEQUENCE-BASED KERNEL METHODS
SVM has been successfully used in many applications in-
Most masquerade detections based on SVM[6, 13] use the RBF kernel due to its easy application and relatively good performance. Although UNIX command sequence information is a good feature for a user behavior model, the RBF kernel uses frequencies of features without sequence information. For example, although <a.out, cd, cpp> and <cpp, ed, a.out> could have different behaviors, their word count features for the RBF kernel have the same input vector: <n(a.out), n(cd), n(cpp)> = <1, 1, 1>.

To reduce the number of false alarms in masquerade detection systems, we use sequence information as an effective feature in order to develop an accurate user behavior model. One of the sequence-based methods is a string kernel proposed by Lodhii[8]. The string kernel compares two strings according to the noncontiguous sequences of substrings they contain. The string kernel has two parameters: n which starts for the number of noncontiguous sequence and λ ∈ (0, 1) which is a decay factor used to support noncontiguous substrings. Equation 1 represents a kernel function

\[ K(x, y) = \langle \phi(x), \phi(y) \rangle \]

that is an inner product between mapped feature vectors in a feature space.

Another sequence-based kernel method is the K-gram kernel which is based on simple text representation technique of n-grams. A K-gram kernel transforms strings into high dimensional feature vectors where each feature corresponds to a contiguous substring. Equation 2 shows a K-gram kernel function which has a parameter n, the number of contiguous sequences. We decide to restrict the minimum number of exact sequence as a one not zero to avoid very sparse vector; that is, we add a one to each element of feature vector showed in equation 2. String and K-gram kernels calculate the similarity of two strings which can be different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.

In this experiment, we used Schonlau’s data set to compare with the result of previous experiments[11, 9, 6, 7]. To build user profile as a model learning, two-class approach was used: one is a self and 49 users are non-self for each of his/her profile. Once user profile was developed, SVM classifier examined test data and determined if they were to be considered normal or abnormal.

Table 2 shows that experimental results achieved a 89.61% detection rate with a 14.19% false alarm rate with a K-gram kernel and 97.40%, 23.77% with a string kernel respectively. Although most pattern recognition problems suffer from the excessive cost of data preprocessing, the masquerade detection system shown in figure 1 does not need any preprocessing and can simply use user command sequences of different lengths. String and K-gram kernels help to use the sequence-based features internally calculated in SVM. A general masquerade detection system can use a series of UNIX command sessions of different lengths from login to logout, but we use a sequence of 100 commands as a block in order to provide for a fair comparison with other previous experiments.
The string kernel shows the same detection rate regardless of different λ parameters. In addition, string and K-gram kernels show bad performance when K parameter has a score of 4 or more.

5. HYBRID OF DIFFERENT KERNELS

To achieve a lower false alarm rate, we did an experiment on the compositions of two kernel methods. Since string and K-gram methods have high detection rates of anomalous behaviors, they are used for masquerade detection. Similarly, because RBF method has relatively fewer false alarms, it is used for normal user validation. One example of a simple hybrid method conducts parallel masquerade detection; we decided on abnormal behavior only when both masquerade detectors flagged abnormal alarms, and on normal behavior when either of two masquerade detectors gave normal. To check the similarity between the kernel methods, commonly used Pearson’s correlation analysis was used. The result showed that K-gram and RBF kernels have the lowest correlation coefficient (0.59) when distinguishing between a masquerader and a normal user. Therefore the two kernels will complement each other best so that we focused on a hybrid of between sequence and frequency[6] information of each users command as a feature for user behavior model.

Figure 3 shows slightly better performance than the simple kernel method. We did experiments on the SVM with various kernel methods and hybrid methods, however the false alarms are still not sustainable, because high false alarms need more cost to investigate them in depth. Furthermore the ultimate goal of a masquerade detection system is to get a low number of false alarms and without losing any intrusions.

6. CONCLUSIONS

In this paper, we demonstrated the effectiveness of sequence-based kernel method for SVM to detect masquerading attacks. In the first experiment in which simple sequence-based methods were used, K-gram and string kernel methods were shown to be more effective than generic RBF kernel method. The composition method of K-gram and RBF kernel was shown to improve the accuracy and minimize the rate of false positives slightly.

7. REFERENCES