A Comparison of Test Statistics for Computer Intrusion Detection Based on Principal Components Regression of Transition Probabilities

William DuMouchel
AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932
dumouchel@research.att.com

Matthias Schonlau
AT&T Labs-Research and
National Institute of Statistical Sciences
PO Box 14006
Research Triangle Park, NC 27709-4006

Abstract

One method of detecting an unauthorized user masquerading as a registered user is to compare in real time the sequence of commands given by each user to a profile of that user's past behavior. Our profiles are derived from each user's historical one-step transition probabilities of Unix commands. We compare various statistics for testing the null hypothesis that the observed command transition probabilities come from a profiled transition matrix, which is a smoothed version of historical transition counts. The primary statistical difficulty comes from the large sparse nature of the transition count matrix. Our example is based on the 100 most frequent commands. Hence we infer about a 100 by 100 matrix of transition probabilities, although most of these transitions will be unobserved in the test data.

Different test statistics are formed by varying the amount of smoothing of the transition probabilities in the training data, and by using different theoretical test criteria. To reduce the dimensionality of the test statistics, the alternative hypothesis is based on a principal component regression model. Using example data from a population of 45 (mostly research) users on a single computer, we compute error rates and ROC curves for the various test statistics when each user's test data is compared to their own and to other users' training data. We also discuss implementation issues such as storage and computational requirements.

1 Description of Statistical Methodology

Introduction. In computer intrusion detection one attempts to identify unauthorized accesses to computer accounts. There are two main approaches to intrusion detection: pattern recognition and anomaly detection. Pattern recognition is the attempt to recognize general "attack signatures" that stem from known attacks such as exploiting a software bug. The approach has the disadvantage that it cannot defend against previously unknown software bugs, or any unauthorized user with the knowledge of the account password. Anomaly detection, on the other hand, attempts to identify an unauthorized user by identifying unusual usage of the computer. Usually, for each user a historical profile is built and large deviations from the profile indicate a possible intruder. Therefore it is also referred to as the profile based approach. Intrusion detection systems like IDES (Lunt et al. 1992), NIDES and Emerald (Porras and Neumann 1997) use both approaches, presumably because neither one is uniformly superior to the other. In this paper we only consider the anomaly detection approach, which lends itself to a statistical treatment. Ryan et. al. (1998) suggested that each user on a computer system leaves a "print" that could be captured by training a neural network with historical data. When for new data from any user the neural network predicts that the data is more likely to stem from another user in the historical data, then an alarm for a possible intrusion is raised. Forrest et al. (1996) consider anomalies for unix processes (such as ftp, or root) rather than for users. In this paper we propose a test for anomaly detection based on hypothesis testing. Since users (including the root user) and system processes (such as ftp) both generate command - level data, we are able to test anomalies for unix processes and users simultaneously.

Command Transition Probabilities. Our method is based on comparing the sequence of each user's commands to a stored profile describing the probability distribution of that user's command sequences. Each command is one of a fixed number K of possible commands or command groupings. We represent each user's historical data in terms of a transition matrix of command

probabilities:

$$p_{jku} = P(\text{Next Com.} = k | \text{Prev. Com.} = j, \text{User} = u) \quad (1)$$

The commands are arbitrarily numbered as $j, k = 1, ..., K$ and we assume that historical data are available for $U+1$ users, arbitrarily numbered $u = 0, 1, ..., U$. We focus on User $u = 0$ as the user whose commands are being monitored and tested for unusual behavior at any given time, but the historical data from the other U users play a role in the test procedure for user 0.

1.1 Smoothing Historical Transition Probabilities

The probabilities (1) must be reliably estimated, with all $p_{jku} > 0$. Thus we smooth the observed proportions from the training period. Denote by N_{jku} the raw count of transitions from command j to command k for user u during the training period. This array will have many elements equal to 0 and many of its nonzero elements may be so small that they are statistically unreliable. For some small $\epsilon > 0$ (we use $\epsilon = 0.001$), let

$$N_{.ku} = \sum_j N_{jku}$$

$$N_{..u} = \sum_k N_{.ku}$$

$$q_{ku} = (N_{.ku} + N_{..u}\epsilon)/N_{..u}(1 + K\epsilon)$$

The q_{ku} are the marginal probabilities of command k for user u, modified using ϵ so that they are all positive. The transition probabilities p_{jku} are weighted averages of q_{ku} and the raw proportions $N_{jku}/N_{j.u}$, namely

$$p_{jku} = [(N_{jku} + M_{ju}q_{ku})/(N_{j.u} + M_{ju}) + \epsilon]/(1 + K\epsilon) \quad (2)$$

where $N_{j.u} = \sum_k N_{jku}$ and M_{ju} are weights chosen to be large if the raw transition frequencies are not significantly different from the marginal frequencies, but near 0 if the raw transition frequencies are very reliably different than the marginal ones. Specifically, let

$$M_{ju} = 1/\max\{.0001, [\sum_k (N_{jku} - N_{j.u}q_{ku})^2 / N_{j.u}q_{ku} - K + 1]/N_{j.u}(K-1)\} \quad (3)$$

The use of equations (2) and (3) is inspired by an empirical Bayes model in which it is assumed that each vector $(p_{j1u}, ..., p_{jKu})$ was generated from a Dirichlet prior distribution (O'Hagan 1994, Ch. 10) with mean vector $(q_{1u}, ..., q_{Ku})$ and probability density proportional to $\prod_k p_{jku}^{M_{ju}q_{ku}-1}$.

1.2 Hypothesis Testing Framework

Suppose that User 0 is logged on and has generated a sequence of $T+1$ commands $C_0, C_1, ..., C_T$. There is the possibility that these commands are being generated by someone other than User 0, and let the unknown true transition probabilities of this sequence be

$$\pi_{jk} = P(C_t = k | C_{t-1} = j) \quad t = 1, 2, ..., T \quad (4)$$

The corresponding transition counts for this user's test data are n_{jk}, where $\sum_{j,k} n_{jk} = T$. Let $n_{j.} = \sum_k n_{jk}$. By definition, $E[n_{jk}|n_{j.}] = n_{j.}\pi_{jk}$. We want to test the null hypothesis

$$H_0: \quad \pi_{jk} = p_{jk0} \quad j = 1, ..., K; \; k = 1, ..., K \quad (5)$$

Statistical hypothesis testing is a procedure in which a decision maker prespecifies a computable test statistic whose distribution is supposed to be completely known assuming that a null hypothesis H_0 is true. This allows the false alarm probability to be computable in advance for a decision rule that rejects H_0 whenever the test statistic exceeds a fixed value.

The standard test statistic for this situation is the log likelihood ratio statistic, namely

$$LR = 2\sum_j \sum_k n_{jk} \log(n_{jk}/n_{j.}p_{jk0}) \quad (6)$$

The range of j in the above summation is over the J values of j for which $n_{j.} > 0$. If T is very large and the assumptions (4 - 5) are true, then LR will have an approximate chi-squared distribution with $J(K-1)$ degrees of freedom, so that LR could be compared to the percentiles of this distribution to assess H_0. Unfortunately, T will rarely be large enough for this distributional assumption to be even approximately true. The usual rule of thumb is that every value of $n_{j.}p_{jk0} > 1$, and that most $n_{j.}p_{jk0} > 5$. This would imply that every $n_{j.} > 3K$, yet many rarely occurring commands would have many fewer occurrences than that in the test data. Another problem with using the test statistic (6) is that this is an omnibus test with power against all possible alternatives to (5), which is likely to waste statistical power testing against unlikely or nonsensical alternative sets of transition probabilities. Therefore we will construct an alternative hypothesis with fewer degrees of freedom, using the historical data from all users as a guide to which alternative transition probabilities are plausible.

Alternative Hypothesis. Suppose that the test data are being generated by the transition probabilities

$$H_1: \pi_{jk} = \pi_{jk}(\beta) = p_{jk0} + \sum_{u=1,U} (p_{jku} - p_{jk0})\beta_{ju} \quad (7)$$

where $\beta = (\beta_{11}, ..., \beta_{KU})$. Instead of considering all alternatives to H_0, (7) focuses on directions in the space of transition probabilities that the historical data confirm are occupied by one or more other users. The expression (7) becomes identical to H_0 if all KU elements of $\beta = 0$, so H_1 might also be stated as $\beta_{ku} \neq 0$ for some (k, u). However, (7) still has the disadvantage of too many degrees of freedom because many of the users may have similar historical probabilities so that the vectors $(p_{j1u} - p_{j10}, ..., p_{jKu} - p_{jK0})$ may be highly collinear for many values of u.

Principal Components Regression. The dimensionality of the alternative hypothesis is reduced by choosing linear combinations of user deviations from p_{jk0} that have maximum variance and are uncorrelated. First define the matrices X_j:

$$X_{jku} = (p_{jku} - p_{jk0})/\sqrt{p_{jk0}}$$

Then let Z_j be a $K \times U^*$ matrix consisting of the first $U^* < \min(U, K)$ principal components of X_j. Take uncentered principal components, so that $tr(Z_j' Z_j) \approx tr(X_j' X_j)$.

1.3 Test Procedure

Fisher's Score Statistic. After reducing the dimensionality in this way, the corresponding Fisher score statistics (Stuart and Ord 1991, Ch. 25) are defined as follows, where $v = 1, ..., U^*$:

$$\begin{aligned} Y_{vjk} &= Z_{jkv}/\sqrt{p_{jk0}} \qquad (8) \\ S_{jv} &= \sum_k n_{jk} Y_{vjk} \\ V_{jv} &= \sum_k p_{jk0} {Y_{vjk}}^2 \end{aligned}$$

The principal component scores S_{jv} have moments

$$E[S_{jv}] = 0 \qquad (9)$$

$$\mathrm{Var}(S_{jv}) = n_{j\cdot} V_{jv} \qquad (10)$$

$\mathrm{Cov}(S_{jv}, S_{j'v'}) = 0$ for all combinations $j \neq j'$ or $v \neq v'$. Note that if the $U^* K^2$ subscores (8) are stored, then only U^* additions are required to update the entire matrix S_{jv} after each observed transition $(C_{t-1} = j, C_t = k)$.

The test statistic

$$S^2 = \sum_{j,v} ({S_{jv}}^2 / n_{j\cdot} V_{jv}) \qquad (11)$$

has an approximate χ^2 distribution with $df = JU^*$ under H_0, or, using the Wilson-Hilferty (1931) transformation,

$$S^* = [(S^2/df)^{1/3} - 1 + 2/9df](9df/2)^{1/2}$$

a standard normal distribution. In our example, $U^* = 5$ principal components are used for each previous command j.

An Alternate Test Statistic. Fisher's score statistic treats all U^* principal component equally. Since one would expect principal components to decrease in importance as more and more are added, we consider an alternate test statistic that reflects that expectation:

$$R^2 = \sum_{j,v} ({S_{jv}}^2) / \sum_{j,v} (n_{j\cdot} V_{jv}) \qquad (12)$$

The statistic (12) has expectation 1 under H_0 but is not chi-squared distributed. Its distribution can be approximated by a gamma distribution having the same first two moments by computing its variance assuming that each S^*_{jv} has a normal distribution, and then using the Wilson-Hilferty approximation to approximate the resulting gamma distribution by a normal distribution. The result is to compare the quantity R^* to a standard normal distribution by a normal distribution.

$$\begin{aligned} R^* &= [R^{2/3} - 1 + r^2]/r \qquad (13) \\ r^2 &= 2 \sum_{j,v} (n_{j\cdot} V_{jv})^2 / (3 \sum_{j,v} n_{j\cdot} V_{jv})^2 \quad . \end{aligned}$$

This approach effectively downweights components with low variance and thus low information for estimating β. While the statistic S^* gives full weight to the first U^* principal components, the statistic R^* provides a more continuous and gradually decreasing weight to each component, so that a correct choice of U^*, the number of components used is less crucial for R^* than for S^*.

2 Data and Results

Testing methodology for intrusion detection is notoriously difficult because of the lack of data with actual intrusions. Even when such data do exist, tests usually leave the reader with the uncomfortable feeling that the procedure is particularly sensitive to the intrusions at hand and does less well in other situations. We therefore slightly change the task of detecting intrusions to discrimination among users: we compare test data to training data (profiles) for pairs of users and decide whether they stem from the same user or not. Ideally, an alarm should always be raised except when a user is tested against his or her own profile.

To establish user profiles, we use historical data from usage on our local unix machine. The data (user names and commands) are extracted from output of the unix `acct` auditing mechanism and consist of user names and

commands only (without their arguments). Some commands recorded by the system are implicitly generated and not explicitly typed by the user. For example, each execution of the .profile file or a make file generates commands contained in these files that are also recorded in the data stream. We use test data separated in time from training data according to the following cross-validation scheme involving four separate time periods.

2.1 Experimental Design

Data were collected from our local population during four separate time periods approximately a month apart. During each time period, the first 1000 command transitions by each user are included in the study. There were 45 users with that amount of data available from all four time periods. We form four separate replications of our study by considering in turn each of the four time periods as the test period and the other three time periods as historical training periods for collecting user profiles. Within each replication, we test each user's test data against each of the profiles in the historical data for a varying number of principal components ($U^* = 5, 10, 20$) and record both the test statistics S^* and R^*.

Within each set of 1000 test commands, we compute the test statistic for each of 10 sets of 100 commands, so the basic unit of study is observation of 100 commands from the user being validated. When the test statistic S^* (R^*) exceeds a threshold value, an alarm is raised. Depending on the chosen value of the threshold, different rates for false positives (false alarms) and false negatives (missing alarms) can be obtained. In the following we investigate the tradeoff between false negatives and false positives by choosing different thresholds.

2.2 Results

We first focuse just on the false alarm problem (comparing each test user to the same user's profile) and look at the test statistic R^* and S^* as a function of the number of commands N. Out of $3 \times 2 \times 4 = 24$ combinations of levels of number of principal components, test statistics and replications we choose two ($U^* = 10$, Test Period = 1 for both R^* and S^*) to display in Figures 1 and 2. The two horizontal lines in each of the Figures correspond to thresholds of $S^* = 10$ ($R^* = 10$) and of $S^* = 100$ ($R^* = 100$), respectively. The vast majority of the 45 users maintain $S^* < 10$ ($R^* < 10$).

Under the Null hypothesis, both R^* and S^* should have a standard normal distribution which would suggest a small threshold (e.g. 3). Our data show higher false alarm rates than this theory predicts, which we attribute to the fact that our model does not accommodate users' behavior changing over time, and some of our users changed significantly.

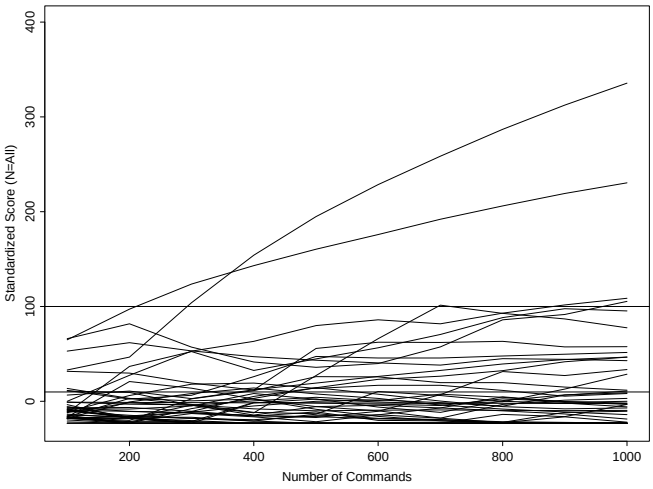

Figure 1: Time plot of S^* comparing each user's test data with their own training data profile. Training Periods = $2-4$. Test Period = 1. Number of Principal Components $U^* = 10$.

For the purpose of assessing the effect of different combinations, we compute the percentage of statistics greater than 10 for all possible 24 combinations of factor levels. The result is displayed in Figure 3. The percentages for S^* seem to be generally higher than for R^*.

Figures 4 and 5 show, for each of the 45×45 (test data, training data) pairs, the median value of S^* (R^*) for blocks of $N = 100$ commands within that pair. (Note that actually $25 + S^*$ ($25 + R^*$) is plotted to allow for the logarithmic scale on the vertical axis. A horizontal line is drawn at $S^* = 0$ ($R^* = 0$). Ideally, the median test score against a user's own historical profile (denoted by "+" in Figures 4 and 5) should be lower than all of the test scores run against other people's profiles (denoted by "."). Another way of saying this is that a user should only be able to break into his/her own account without causing an alarm to be raised. As we can see in Figures 4 and 5, the block medians discriminate very well between a user's own and other profiles. The distance between the pluses and dots seem to be somewhat smaller based on the S^* statistic in Figure 5.

We compute the average number of users scoring better than the true users (i.e. the average numbers of dots that are lower than pluses) to summarize the result for all 24 plots. The result is displayed in Figure 6. Note that even though the "+" is usually the lowest point, on average there are between about one and three dots

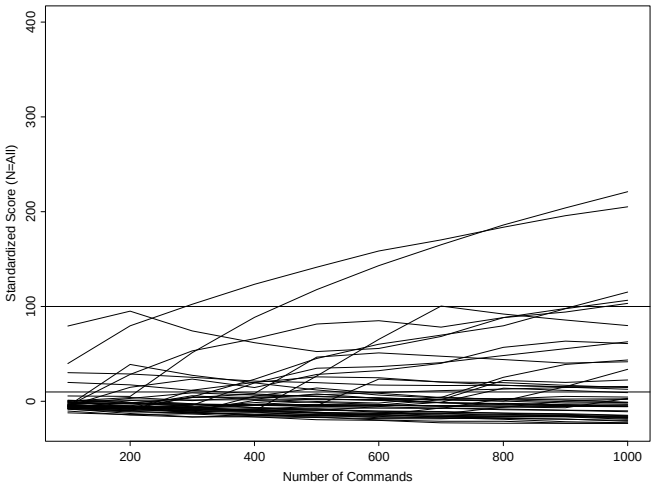

Figure 2: Time plot of R^* comparing each user's test data with their own training data profile. Training Periods $= 2-4$. Test Period $= 1$. Number of Principal Components $U^* = 10$.

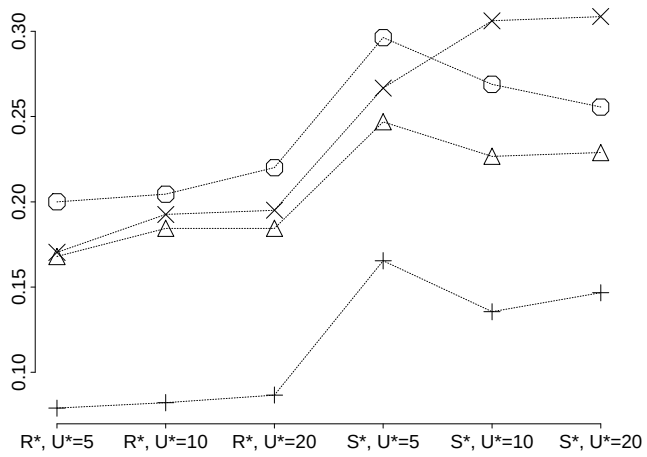

Figure 3: Profile Plot of the Percentage of Standardized Statistics Greater than 10. The Six Categories on the Horizontal Axis Represent the 2×3 Levels of the Two Factors (Number of Principal Components U^* and Test Statistic). Points With the Same Symbols Represent the Same Test Period/ Training Periods combination.

below the "+" due to a small number of cases in which the true user ("+") scores very high. Figure 6 indicates that for two of the time periods the S^* does better and for the other two about the same as the R^* statistic.

As the threshold where an alarm is raised changes, different rates of false alarms (false positives) and missing alarms (false negatives) can be obtained. Figure 7 and Figure 8 show the tradeoff between these false positive and false negative rates for all four time periods based on blocks of 100 commands, respectively. Each time period is used as testing data with the remaining three time periods serving as training data. The lower left corner represents the ideal scenario: no false alarms and no missing alarms. Note that both axes are presented on a logarithmic scale. From Figure 7 we can tell, for example, based on 100 transitions and a false positive rate of 5% we obtain a corresponding false negative rates between about 10% - 50%. These numbers are quite high and also indicate that there is a considerable variability from time period to time period.

The false alarm rate corresponding to a false positive rate of 5% is displayed in Figure 9 for possible all 24 combinations of factors. The S^* statistic again has lower false alarm rates for 2 or 3 of the time periods and about the same in the remaining time period.

It is surprising that the number of principal components U^* did not seem to have any noticable effect in this or the previous Figures. It implies that discrimination among users based on 5 principal components is as good

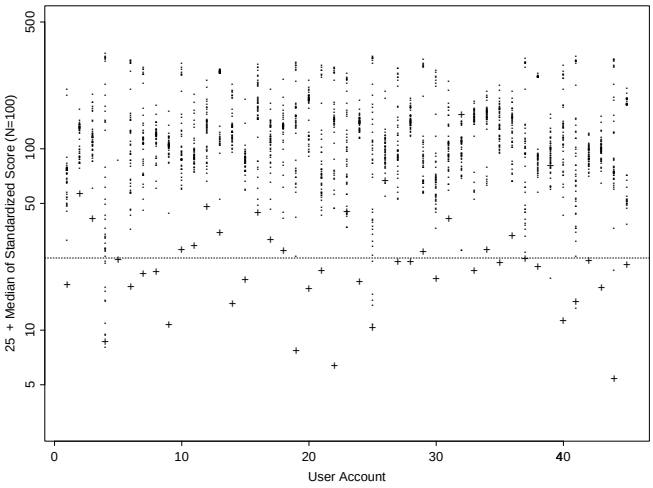

Figure 4: Medians of S^* Scores for Blocks of 100 Test Data Commands. Within Each Column Test Data For Different Users is Tested Against the Training Data of Only One User. "+" Denotes Comparison When Training and Test Data Stem from the Same User, "." When Test Data Stems from a Different User. Training Periods $= 1-3$. Test Period $= 4$. Number of Principal Components $U^* = 10$. A Horizontal Line is Drawn at $S^* = 0$.

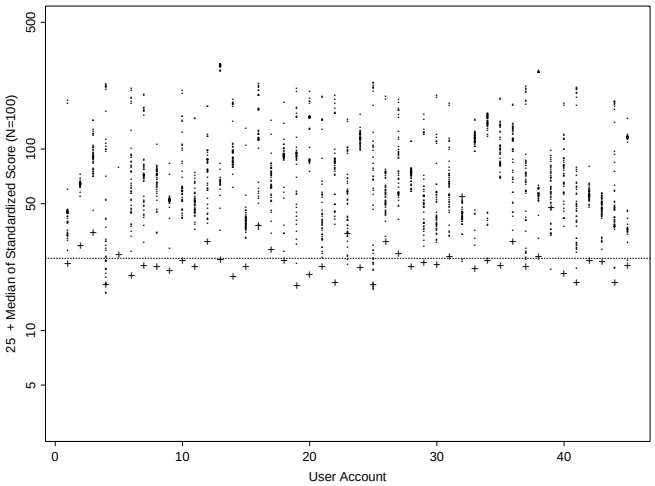

Figure 5: Medians of R^* Scores for Blocks of 100 Test Data Commands. Within Each Column Test Data For Different Users is Tested Against the Training Data of Only One User. "+" Denotes Comparison When Training and Test Data Stem from the Same User, "." When Test Data Stems from a Different User. Training Periods = $1-3$. Test Period = 4. Number of Principal Components $U^* = 10$. A Horizontal Line is Drawn at $R^* = 0$.

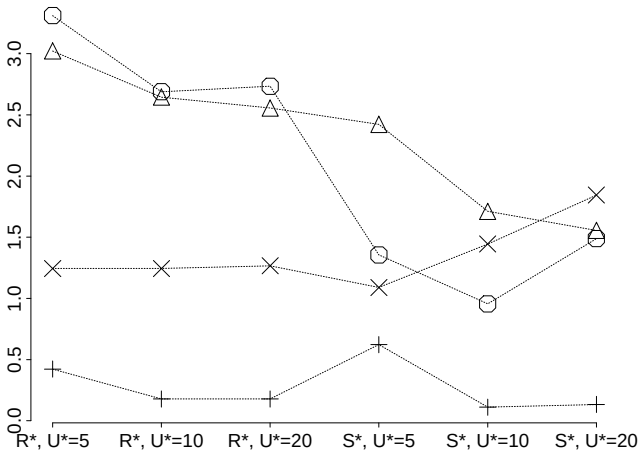

Figure 6: Profile Plot of Average Number of Users Scoring Better than the True User. The Six Categories on the Horizontal Axis Represent the 2×3 Levels of the Two Factors (Number of Principal Components U^* and Test Statistic). Points With the Same Symbols Represent the Same Test Period/ Training Periods combination.

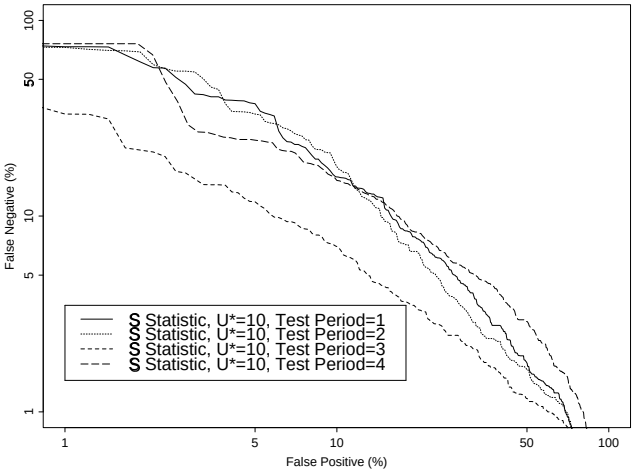

Figure 7: Tradeoff Between False Positive and False Negative Rates for S^*, Treating Every Block of 100 Commands as a Separate Experiment. Each of Four Periods is Used as Test Data With the Respective Other Three Periods Being Used as Training Data. Number of Principal Components $U^* = 10$.

as with 20 principal components. We prefer, of course, $U^* = 5$, because it requires less storage. Figures 6 and 9 indicate that the statistic S^* is to be preferred, whereas R^* is preferable based on Figure 3. The ROC curves are probably the most important ones and we therefore recommend the statistic S^*. We further note that much of the variation between time periods seems to be due to period 3. When period 3 is used as the test period, results seem to be consistently better compared to other periods.

2.3 Further Results

In an earlier experiment we have considered two additional factors.

For one of them two different definitions of transitions are considered. Currently, two subsequent commands recorded in the audit stream form a transition even though they may stem from different windows. Alternatively, a transition can be defined as any two subsequent commands that are generated from the *same* window. The underlying idea is that different windows may be used for different tasks (e.g. editing, compiling, debugging) and combining commands from different windows may add noise. Based on comparisons of ROC curves it turned out that the definition of transitions does not matter one way or the other.

We also used different amounts of smoothing in the

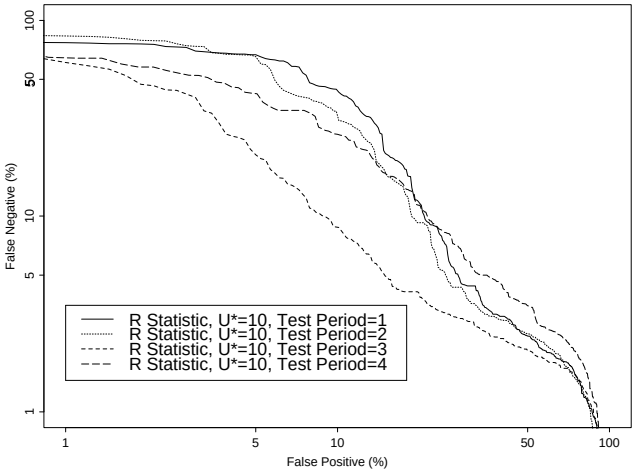

Figure 8: Tradeoff Between False Positive and False Negative Rates for R^*, Treating Every Block of 100 Commands as a Separate Experiment. Each of Four Periods is Used as Test Data With the Respective Other Three Periods Being Used as Training Data. Number of Principal Components $U^* = 10$.

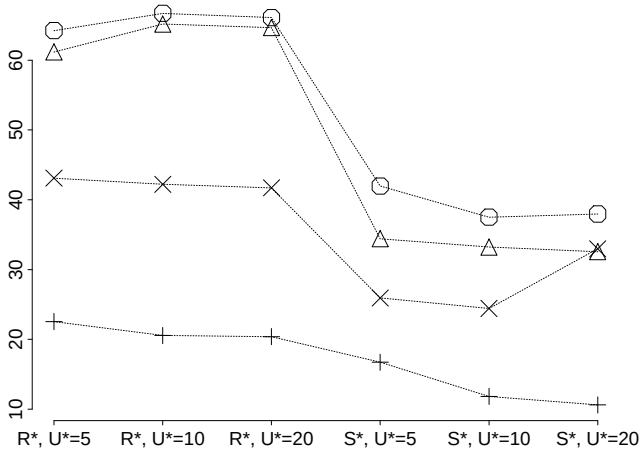

Figure 9: Profile Plot of False Negatives at 5% False Positives. The Six Categories on the Horizontal Axis Represent the 2×3 Levels of the Two Factors (Number of Principal Components U^* and Test Statistic). Points With the Same Symbols Represent the Same Test Period/ Training Periods combination.

historical data which translates to choosing different values for ϵ. We tried $\epsilon = 0.001$ (as in this paper) and $\epsilon = 0.0001$. The smaller ϵ, i.e. the less smoothing we use, the greater is the penalty (in the sense of a large contribution to the statistic) of a transition not seen in the historical data. It turned out that previously unseen transitions should not be penalized heavily. We interpret this to mean that the overall usage pattern of transitions discriminates better than a few previously unseen transitions. Note also that most of the 101×101 possible transitions were not seen in any one training data set.

3 Comparison with Static Statistical Tests

The methodology presented so far has conformed to the need for a statistic that can be updated economically in real time. Therefore, the training data was smoothed and then treated as known and fixed, and the test statistics were partitioned into orthogonal principal components that allowed each newly observed transition to contribute independently to the test statistics S_{jv}. In this Section, we compare the performance of the best performing statistic of the previous section, namely the score statistic S^* based on just 5 principal components per previous command, to standard test statistics for comparing frequency distributions. The raw frequencies from test and training data will be compared using the well known log likelihood ratio test statistics. All the data from each user will be used in a single static test statistic, so the comparison is always between a test sample of size 1000 and a training sample of size 3000. We consider two such tests, one comparing the marginal frequencies of the next command, ignoring the previous command, and one comparing the two sets of conditional frequencies of next command given previous command.

Suppose $n = \{n_{jk}\}$ is a matrix of transition frequencies during the test period, and $N = \{N_{jk}\}$ is a matrix of training period frequencies to be compared to n, where N and n may come from the same user (to test the false alarm rate) or from different users (to test discrimination ability). The fitted frequencies under the null hypothesis that n and N come from the same distributions of k given j are m_{jk} and M_{jk}, where

$$m_{jk} = n_{j.}(n_{jk} + N_{jk})/(n_{j.} + N_{j.})$$
$$M_{jk} = N_{j.}(n_{jk} + N_{jk})/(n_{j.} + N_{j.})$$

The log likelihood ratio test statistic for testing the null hypothesis of equal conditional probabilities is

$$LR_{Cond} = 2 \sum_{j,k} [n_{jk} log(n_{jk}/m_{jk}) + N_{jk} log(N_{jk}/M_{jk})]$$

The corresponding test statistic for the null hypothesis

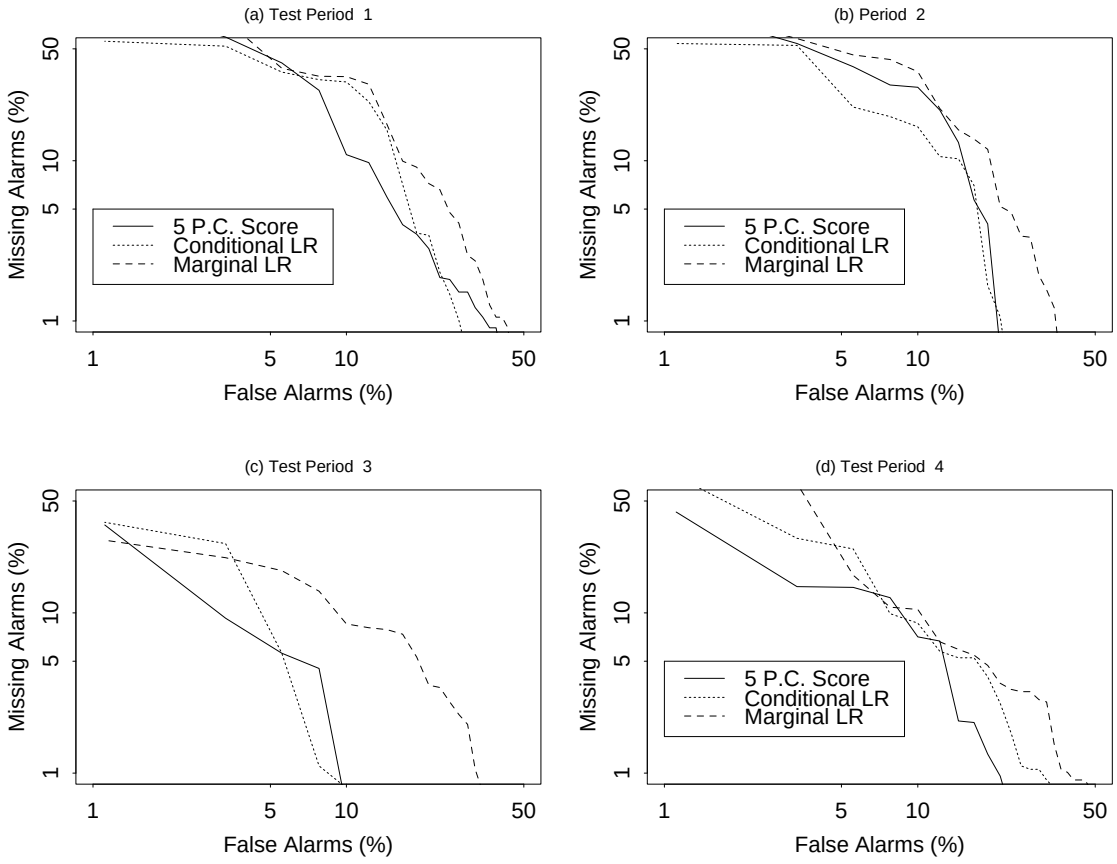

Figure 10: ROC Curves Based on Test Data for Four Different Time Periods. Curves for the Marginal Likelihood Ratio Test, Conditional Likelihood Ratio Test and for the Test Based on 5 Principal Components are Shown. Sample Sizes are 1000 Transitions in Test Periods, 3000 Transitions in Training Periods.

of equal marginal probabilities is

$$LR_{Marg} = 2\sum_k [n_{.k} log(n_{.k}/m_k) + N_{.k} log(N_{.k}/M_k)]$$

Where

$$m_k = n_{..}(n_{.k} + N_{.k})/(n_{..} + N_{..})$$
$$M_k = N_{..}(n_{.k} + N_{.k})/(n_{..} + N_{..})$$

The degrees of freedom associated with these statistics are

$$df_{Marg} = [\text{No. of } k \text{ such that } (n_{.k} + N_{.k}) > 0] - 1$$
$$df_{Cond} = \sum_j \{[\text{No. of } k \text{ such that } (n_{jk} + N_{jk}) > 0] - 1\} \quad (14)$$

In (14), only values of j for which both $n_{j.} > 0$ and $N_{j.} > 0$ are included in the summation. In order to compare test statistics with differing degrees of freedom, we transform to approximate normality using the Wilson-Hilferty transform, leading to the definitions

$$LR^*_{Cond} = \left[(LR_{Cond}/df_{Cond})^{1/3} - 1 + 2/9df_{Cond}\right](9df_{Cond}/2)^{1/2}$$
$$LR^*_{Marg} = \left[(LR_{Marg}/df_{Marg})^{1/3} - 1 + 2/9df_{Marg}\right](9df_{Marg}/2)^{1/2}$$

3.1 Results of Comparisons

Figure 10 shows the results comparing the three test statistics LR^*_{Marg}, LR^*_{Cond}, and the 5 principal component version of S^* on each of the 4 test periods of 1000 transitions per user, with the other three periods serving as training data with 3000 transitions. The four parts (a) - (d) of the Figure show the ROC curves for the three test statistics for each time period. Remember-

ing that curves shifted toward the lower left reveal more powerful test statistics, we see that in general comparing conditional distributions worked better than comparing marginal distributions, since the dotted curves are mostly below and to the left of the dashed curves in each of the 4 panels of the Figure.

The ROC curves for the five-p. c. score statistic S^* are drawn as solid curves in Figure 10. The statistic S^* does uniformly better than the other two statistics in test period 4, it falls between the other two in test period 2, and the conditional and score statistic curves cross for the data in test periods 1 and 3. In general, then, for this sample of users and usage, the whole-sample comparisons show the benefits of using the extra information in the conditional distributions as opposed to the marginal distributions. Our method of smoothing the training data in Section 1.1 involves shrinking the conditional distributions toward the marginal distributions, which attempts to gain the benefits of the extra information in the conditional distributions while mitigating the instabilities of the small conditional distribution cell counts.

Figure 10 indicates that the dynamic method involving smoothed historical data and principal components is at least as powerful as the more conventional likelihood ratio tests, but not much more powerful. It is somewhat disappointing that the principal components dimension reduction did not provide more power in addition to the benefits of easy updating. Further research is needed to understand this better.

4 Discussion

Ryan et. al. (1998) use a neural network approach and test classification errors based on 10 users. They have 11 successive days of data, 8 of which are used for training. One of the users only had little data. They report a false alarm rate of 7% and 4% missing alarms. Our test is more challenging in that we test with more users, and because there is a gap in time between historical and test data. Also, their decision criterion seems to assume that the intruder would be one of the other users in their training data. On the other hand, unlike them, we excluded users with very low account usage.

In order for an intrusion detection tool to be useful, the false alarm rate needs to be low – otherwise alarms tend to be ignored. To that extent a false alarm rate of 5% still seems high. Perhaps extending the length of the training period will make the profiles more robust to changes in user behavior. One possible way to improve the markov model is by consolidating series of cascading commands (generated, for example, by a makefile) into single (meta) commands, or by considering the next command conditional on more than one previous command. Note that our theoretical false alarm probabilities ignore the problem of multiple testing, in which repeated testing of the same null hypothesis as time goes on increases the chance of a false alarm rate. This problem, common to most control-chart like procedures, seems to be a less important cause of excessive false alarms than our failure to model how users tend to change their profiles over time.

A major strength of the approach presented is its speed. Only a few dozen operations are needed for updating the test statistic, and preliminary calculations indicate that it will be easily possible to implement this procedure in real time. The amount of storage required for the procedure is relatively large. Based on 5 principal components and 100 command categories, 50500 single precision numbers need to be stored for each user.

We need to perform further investigation of the optimal number of principal components to take. We will also investigate the effect of adding known intrusion signatures as profiles in the training data to make the principal components methodology more sensitive to such attacks. We also expect to be able to use this procedure to increase our understanding of local system usage.

Acknowledgements

Matthias Schonlau's work is funded in part by NSF grants DMS-9700867 and DMS-9208758. We are grateful for feedback from our network intrusion group with members from AT&T Labs Research, The National Institute of Statistical Sciences and Rutgers University. Their comments have led to a number of improvements of this paper.

References

Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff, T. (1996). In *Proceedings of the 1996 IEEE Symposium on Security and Privacy.* IEEE Computer Society Press, Los Alamitos, CA, pp. 120-128.

Lunt, T. Tamaru, A., Gilham, F., Jagannathan, R., Neumann , P., Javitz, H., Valdes, A., Garvey, T. (1992). "A Real-Time Intrusion Detection Expert System (IDES) - final technical report. " Computer Science Library, SRI International, Menlo Park, California.

O'Hagan, A. (1994). *Kendall's Advanced Theory of Statistics, Vol. 2B: Bayesian Inference* , New York: John Wiley.

Porras, P., and Neumann P. (1997). "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disurbances". In *Proceedings of the National Information Systems Security Conference.* (to appear).

Ryan, J. , Lin,M., and Miikkulainen, R. (1998). "Intrusion Detection with Neural Networks" . In Jordan, M. I., Kearns, M. J., and Solla, S. A. (editors), *Advances in Neural Information Processing Systems 10* (NIPS'97, Denver, CO). Cambridge, MA: MIT Press.

Stuart, A. and Ord, J.K. (1991). *Kendall's Advanced Theory of Stististics, Vol. 2: Classical Inference and Relationship*, New York, John Wiley.

Wilson EB, Hilferty MM (1931). "The distribution of chi-square", In *Proc. Nat. Acad. Sci.*, Wash. DC, 17, 684-688.