

# User Profiling in Window Title and Process Table

Chien-Chih Lin<sup>1</sup>, Eun Young Noh<sup>2</sup>, Youngping Yan<sup>3</sup>, and Edward Wegman<sup>4</sup>

As one of strategies of computer security, user profiling was conducted for the window title and process table data which was collected on an internet connected unclassified window NT network reflecting the currently ubiquitous window based computing environment. The statistical methodology in the NIDES (Next Generation Detection Expert System) was implemented on these data to profile the past behavior of users on a computer system and to learn the pattern of each user and to identify the unauthorized user or misuse of authorized user. The anomaly detection was based on the score value that represents how much the user's behavior is abnormal compared to the past patterns of behavior. As a method of measuring performance, a specified user was assumed as an authorized user and the other users were treated as unauthorized users. The score values of a specified user and other users gave quite different distributions and the rate of detecting other users from a specified user was about 60~70% in case of 10% false alarm rate.

**Key Words:** Intrusion Detection; User Profiling; Window Title; Process Table; Computer Security

## 1. Introduction

In the information age, information security is a crucial issue for the continuity of business operation. A well-designed information system should develop a security mechanism to protect its intellectual asset. Therefore, Computer Intrusion Detection System is needed to reduce the impacts of threats against the computer security and to prevent the misuse of computer system.

As one of the strategies of computer security, user profiling is a system that measures the user interactions with the computers and networks to extract some useful

---

<sup>1</sup> Chien-Chih Lin (clin3@gmu.edu) is the Ph.D. Student in the School of Computational Science, George Mason University.

<sup>2</sup> Eun Young Noh (enoh@gmu.edu) is the Ph.D. Student in the School of Computational Science, George Mason University.

<sup>3</sup> Youngping Yan (yyan1@gmu.edu) is the Ph.D. Student in the School of Information Technology and Engineering.

<sup>4</sup> Edward Wegman (ewegman@gmu.edu) is the Bernard J. Dunn Professor of Information Technology and Applied Statistics, the Chair of the Department of Applied and Engineering Statistics, and the Director of the Center for Computational Statistics in the School of Information Technology and Engineering, George Mason University.

information that represents the patterns of user behavior. Several possible measurements are designed for user profiling, such as intervals between keystrokes, mistyping, typing speed, text or command statistics, mouse events, computer usage statistics, window registry access statistics, email usage, window titles, and process tables. Because the users have their own patterns in their behaviors when they use the computer system, user profiling is often used to generate the alarms for the potential attacks or computer misuse by comparing the users' current behavior to their past patterns.

Many intrusion detection systems or anomaly detection techniques based on the user profiles are developed. Those systems use the keystroke timing, system log files, or command usage or other user profile data. Moreover, those systems apply the statistical methodology or machine learning approach to identify an unauthorized user from authorized user or to detect a misuse of computer system from authorized application.

In this project, data about window titles and process tables generated from window-based computing environment are analyzed. The statistical methodology in the NIDES (Next Generation Detection Expert System) was implemented on these data to profile the past behavior of users on a computer system, to learn the pattern of each user, and to identify the unauthorized users or misuse behavior of authorized users.

The source of data, preprocessing, and data structures are explained in chapter 2. The statistical methodologies implemented for the user profiling and anomaly detection are introduced in chapter 3. Chapter 4 shows the user profile for six users and the test for detecting the anomaly user.

## **2. Data**

The target data set was obtained from the Machine Learning and Inference Laboratory in the George Mason University. The data was collected on an internet-connected and unclassified window-based network environment. Each file is a session in which a stream of window title and related process table information appears from login to logout for a user. The original data set has 1292 sessions for 26 users. Each session has three versions according to the level of detail for the process information. In this project, the first version of data for six users (User1, 4, 7, 8, 19, 25) who have many sessions was

analyzed because some users have too small sessions to analyze. 1,024 sessions for these selected six users cover 78% of the total sessions.

Data was preprocessed in two steps. The first preprocessing that was conducted using the Unix shell script and AWK program from the Machine Learning and Inference Laboratory. A record for the information about window title and process table was generated whenever that information is measured in the original data. 31 useful features were extracted from the original window title and process tables. The second preprocessing was conducted in order to reduce some nuisance information. Indeed, some features were excluded because some of features are the logarithm of other features and some other features are not applicable. The second preprocessing reduced the amount of data tremendously. The summary for data structure and data preprocessing is represented in Table 1. The final data has 1014 sessions and 129,313 records for 6 users, and 21 features.

|                           | Users  | Sessions | Records | Features | Version       |
|---------------------------|--------|----------|---------|----------|---------------|
| Original data             | 26     | 1,292    | -       | -        | Three Version |
| Select Users and Versions | 6      | 1,014    | -       | -        | First Version |
| Preprocessing1            | 6      | 1,014    |         | 32       | First Version |
| Preprocessing2            | 6      | 1,014    | 129,313 | 21       | First Version |
|                           | User1  | 287      | 31,370  |          |               |
|                           | User4  | 134      | 26,170  |          |               |
|                           | User7  | 193      | 23,289  |          |               |
|                           | User8  | 167      | 17,597  |          |               |
|                           | User19 | 134      | 10,497  |          |               |
|                           | User25 | 99       | 20,390  |          |               |

Table 1: Summary of data structure and Preprocessing

### 3. Methodology

The NIDES (Next Generation Detection Expert System) has a statistical subsystem that profiles the past behavior of users on a computer system and learns the pattern of each user. The statistical subsystem of NIDES uses this monitored profile to identify the potential intrusion that can be unauthorized user or misuse of authorized user. These statistical methods were used for the detecting systems in which their targets are users or programs. In Both cases the system log files are analyzed as target data. Because the

statistical method of NIDES was implemented in this project to profile users' behavior and to identify the real users, its methodology would be introduced precisely in this chapter.

### 3.1 Score Value

A single statistic value that is called "score value" is generated for each audit records. The score value represents how much the user's behavior in recent past is abnormal. The meaning of the recent past is that the current statistics are calculated by exponential weights in which the weight reduces as the record is further from the current record.

The large score value means that the behavior of user is abnormal and the score value close to zero means that the behavior is normal. One or more critical values can be selected. For example, score value between 0 and  $n_1$  is "no concern", score values between  $n_1$  and  $n_2$  is "yellow alert", and score values between  $n_2$  and  $n_3$  is "red alert". The critical values are selected empirically considering the false alarm rate. Different critical values can be selected for different users when standard critical values yield too many false alarms for a particular user or when more precise monitors are needed for particular user.

Since score value is the measure that totally indicates the extent to which a user's behavior is abnormal, it is calculated from the measures of individual detecting variables that reflect the degree of abnormality of a particular type of recent behavior. If there are  $n$  detecting variables, score value is calculated as a kind of sum of squares of  $n$  individual measures.

$$IS = (S_1, S_2, \dots, S_n) C^{-1} (S_1, S_2, \dots, S_n)^t$$

where  $S_i$  ( $1 \leq i \leq n$ ) is the measure of individual components and  $C$  is the correlation matrix of  $(S_1, S_2, \dots, S_n)$  and  $(S_1, S_2, \dots, S_n)^t$ . Inverse of the correlation matrix is included to take the effect of correlations of components into account. If all the individual measures are independent the correlation matrix is identity matrix the score value can be simplified as sum of squares.

$$IS = S_1^2 + S_2^2 + \dots + S_n^2$$

Because the score value is a summary of all behaviors, it does not show the behaviors of individual component. However, when the score values are very large, the corresponding individual measures can be examined to see which components or which types of behavior have substantially contributed the large score value.

### **3.2 Individual Measure**

The individual measure,  $S$  that reflects the degree of the abnormality of a particular type of recent behavior is derived from a corresponding  $Q$  statistics. For example, if  $S$  is the measure of the degrees of abnormality of recent CPU time, then the corresponding  $Q$  is the actual CPU time used in recent past.

Before computing individual measure  $S$ , the users profile must be constructed for each component of each user. In the past profiling of users, some useful information that can characterize nicely the user behavior would be extracted from the original complicated data. In the NIDES, the probability distribution of each component is constructed by the relative frequencies of past  $Q$  values of the corresponding component during the profiled periods. The relative frequencies are computed in the pre-defined intervals, whose width can be determined differently for the users, and the last interval dose not have upper bounds. The user profiles are usually updated every night. The new data of that day are included in the profile system and the most past data are excluded. Moving the profile period reflects adaptively the change of user behavior.

Once the user past profiles are constructed, individual measure  $S$  can be computed by comparing the current value of  $Q$  to the relative frequency table, i.e. the past profile of the corresponding component. If the current  $Q$  value belongs to the interval with high relative frequency, the behavior of the corresponding component can be thought as normal and individual measure  $S$  has small values. Indeed, while the current  $Q$  value which falls into the interval with very low relative frequency indicates that the behavior of that component is abnormal and gives the large values to the individual measure  $S$ .

The computation of the individual measure S value from current Q statistics and its past profile is the key part of the anomaly detection using user profile. The precise algorithm is explained as following. The algorithm assumes that there are  $n$  intervals in the relative frequency distribution for a particular component of a particular user. Let the  $P_i, (i = 1, \dots, n)$  be the relative frequency for the  $i$ -th interval, and  $P_{(i)}$  be the order statistics of  $P_i$  from small to large relative frequencies. And let  $CUMP_i$  be the cumulative relative frequencies of  $P_{(i)}$  that increase as  $i$  increases and goes to 1 when  $i = n$ . For each  $CUMP_i$ , some value  $s_i$  that satisfies the equation  $\Pr \{ |N(0,1)| \geq s_i \} = CUMP_i$ , or  $s_i = \Phi^{-1}(1 - \frac{CUMP_i}{2})$ <sup>5</sup> is defined, where  $N(0,1)$  is the normally distributed variable with mean 0 and variance 1, and  $\Phi$  is the cumulative distribution of the  $N(0,1)$  variable. Therefore, the interval that has the smallest relative frequency has  $CUMP_1$ , and the corresponding  $s_1$  has the largest value and the interval with largest relative frequency has  $CUMP_n$ , and the value of  $s_n$  is 0.

For example, when there are 10 intervals and the relative frequencies  $P_i$  of the each interval is given, the corresponding values of  $P_{(i)}$ ,  $CUMP_i$ ,  $s_i$  are computed as in Table 2, and the histogram of relative frequencies and ordered relative frequencies are shown in Figure 1, and the relationship between  $CUMP_i$ , and  $s_i$  are shown in Figure 2

| I         | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10   |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| $P_i$     | 0.04  | 0.1   | 0.3   | 0.16  | 0.02  | 0.03  | 0.2   | 0.09  | 0.05  | 0.01 |
| $P_{(i)}$ | 0.01  | 0.02  | 0.03  | 0.04  | 0.05  | 0.09  | 0.1   | 0.16  | 0.2   | 0.3  |
| (i)       | 10    | 5     | 6     | 1     | 9     | 8     | 2     | 4     | 7     | 3    |
| $CUMP_i$  | 0.01  | 0.03  | 0.06  | 0.1   | 0.15  | 0.24  | 0.34  | 0.5   | 0.7   | 1    |
| $s_i$     | 2.567 | 2.170 | 1.881 | 1.645 | 1.440 | 1.175 | 0.954 | 0.675 | 0.386 | 0    |

Table 2: Frequency tables for the data with 10 intervals

<sup>5</sup>

$$\begin{aligned}
CUMP_i &= \Pr \{ |N(0,1)| \geq s_i \} = 1 - \Pr \{ |N(0,1)| \leq s_i \} \\
&\Rightarrow \Pr \{ |N(0,1)| \leq s_i \} = 1 - CUMP_i \\
&\Rightarrow \Phi(s_i) - \Phi(-s_i) = 2\Phi(s_i) - 1 = 1 - CUMP_i \\
&\Rightarrow \Phi(s_i) = 1 - \frac{CUMP_i}{2} \\
&\Rightarrow s_i = \Phi^{-1}(1 - \frac{CUMP_i}{2})
\end{aligned}$$

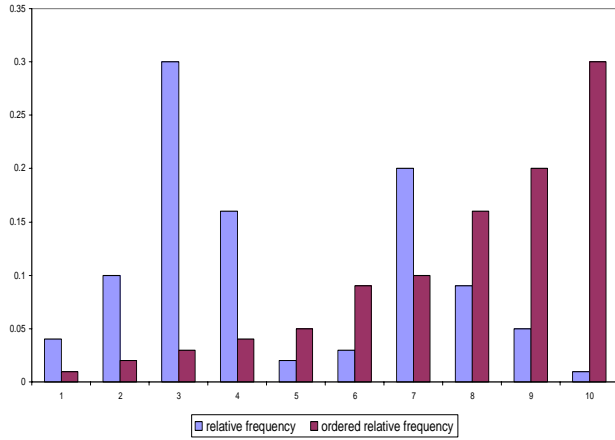


Figure 1: Relative Frequencies and ordered relative frequencies

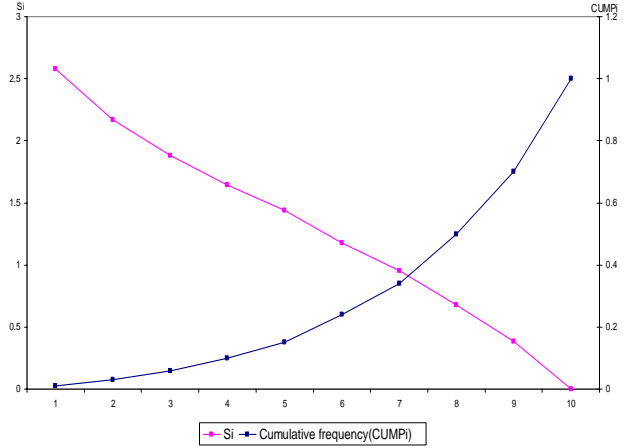


Figure 2: Relationship between  $CUMPI_i$  and  $s_i$

### 3.3 Individual Q Statistic

As we see, user past profile, individual measure  $S$ , and score value are based on the basic statistics  $Q$  for each component.  $Q$  statistic for each component is computed differently according to its aspects. The measures are divided into two main groups of ordinal, categorical. The categorical measures are more divided as linear or binary measures.

Each measure was defined as following by the paper [8]. “The ordinal measure is a count of some numerically quantifiable aspect of observed behavior. For example, the amount of CPU time used, the number of audit records produced is an ordinal measure. A categorical measure is a function of observed behavior over a finite set of categories. Its value is determined by the relative frequencies. A binary categorical measure does not count the number of times that each category of behavior occurs, only whether the category was invoked. This type of measure is sensitive in detecting infrequently used categories such as changing one’s password. Linear categorical measure has score function that counts the number of times each category of behavior occurs. For example, command usage is a linear categorical measure, where the categories span all the available command names for that system.”

$Q$  statistics in NIDES is the sum of audit record values in the recent past with exponential weights that give more weight to the record closer to the current value. The

concept of “half life” was applied to decide the decay rate in the weight. The formulas to compute Q statistics using exponential window for ordinary measure, binary categorical measure, and linear categorical measure are introduced in detail in paper [8]. In our project, however, the weighted sum of recent past record values was not applied because of some restrictions of the data and the computational burdens. First, the session data are gathered inconsecutively, for example user has often only few sessions in a month, while half-life of few hours or few hundred audit records are used to compute recent Q statistics in NIDES. Second, window title record that is used as an audit record has already preprocessed to have some summarized information of the process table. As a result, to simplify the problem and to apply the method adequately, the current record value itself was assigned into Q statistic for both of the ordinary and categorical measures.

## **4. Implementation**

### **4.1 Candidate Features for User Profile**

Among the 21 features in the finally chosen target data set, 18 features except the features for the identification such as User Id, Date, and Time are the candidate features for the user profile. 18 candidate features are listed in Table 3 with the descriptions. Five features have categorical measure and thirteen features have ordinary measure. Even though Q statistic is the record value itself for both measures, the division of categorical or ordinary measure is needed because they have different method in constructing the frequency tables for the past distribution. For the features with ordinary measure, the relative frequencies are computed in 10 equally spaced intervals. For each feature, the width of interval was decided considering the distribution of the corresponding feature and same interval was applied to all users. Meanwhile, the relative frequencies are computed in each possible category for the feature with categorical measure.

| Name | Description   | Measure  |
|------|---|----------|
| F1   | Host machine ID   | Category |
| F2   | Day of week   | Category |
| F3   | Time of day(hour)   | Category |
| F4   | Number of seconds from the start of the session   | Ordinary |
| F5   | Window process name or Process name   | Category |
| F6   | Window status   | Category |
| F11  | Window process ID   | Ordinary |
| F15  | Total elapsed time in window  | Ordinary |
| F17  | Ratio of Cpu time accrued by process within window to Total elapsed time in window or 0 | Ordinary |
| F18  | Delta time between window titles whenever NEW window is opened                          | Ordinary |
| F20  | Elapsed time since login whenever NEW window is opened                                  | Ordinary |
| F22  | Number of characters in protected words   | Ordinary |
| F23  | Number of characters in protected words / total number of characters                    | Ordinary |
| F24  | Total number of words in window title   | Ordinary |
| F25  | Number of protected words / Total number of words in window title                       | Ordinary |
| F26  | Number of process-level records in a single window unit                                 | Ordinary |
| F28  | Total number of windows opened  | Ordinary |
| F30  | Number of protected words in window title   | Ordinary |

Table 3: Candidate features for user profile

## 4.2 Period of User Profile

To build the users past profile for the history data, how long period of data will be included in the profile must be decided. Table 4 shows how the sessions of each user are distributed from November 2001 to January 2003 and which periods of data are used for users past profile and test. Data during eight months from November 2001 to June 2002 were used in making the past profile for the first period, and data of July 2002 were tested using this profile. The users past profile for the second period was built using data during eight month from December 2001 to July 2002 and data of August 2002 were tested based on the user profile for the second period. By doing so, users past profiles can be constructed from the data with same time periods. Furthermore, by moving the time

period used in users past profile the changing patterns of users behavior on computer usage can be reflected as time pass and new data are accumulated.

|        | 2001          |    | 2002 |     |    |    |     |     |      |    |    |    |    |    | 2003 |  |
|--------|---------------|----|------|-----|----|----|-----|-----|------|----|----|----|----|----|------|--|
|        | 11            | 12 | 1    | 2   | 3  | 4  | 5   | 6   | 7    | 8  | 9  | 10 | 11 | 12 | 1    |  |
| User1  |               | 17 | 17   | 24  | 21 | 22 | 23  | 37  | 23   |    |    | 8  | 49 | 29 | 17   |  |
| User4  | 3             | 12 | 18   | 12  |    | 17 | 20  | 8   | 20   | 12 | 11 |    | 1  |    |      |  |
| User7  | 4             | 13 | 21   | 17  |    | 15 | 17  | 20  | 16   | 7  | 17 | 17 | 19 | 10 |      |  |
| User8  | 1             | 10 | 28   | 11  |    | 13 | 32  | 11  | 11   | 7  | 19 | 18 | 6  |    |      |  |
| User19 | 4             | 18 | 20   | 16  |    | 12 | 15  | 20  | 19   | 10 |    |    |    |    |      |  |
| User25 |               |    | 3    | 21  | 19 | 18 | 21  | 14  | 3    |    |    |    |    |    |      |  |
| Total  | 12            | 70 | 107  | 101 | 40 | 97 | 128 | 110 | 92   | 36 | 47 | 43 | 75 | 39 | 17   |  |
|        | ← profile 1 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |
|        | ← profile 2 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |
|        | ← profile 3 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |
|        | ← profile 4 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |
|        | ← profile 5 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |
|        | ← profile 6 → |    |      |     |    |    |     |     | test |    |    |    |    |    |      |  |

Table 4: Number of sessions for each period and for each user Profile and test period

### 4.3 Selecting Features for User Profile

Figure 3 and Figure 4 shows the distributions of the features with categorical measure and with ordinary measure, respectively. If the distributions of a feature are quite different among the users, corresponding feature can be thought to have good properties to identify the user. While a feature that has very similar distributions among the users has not good properties to identify the user.

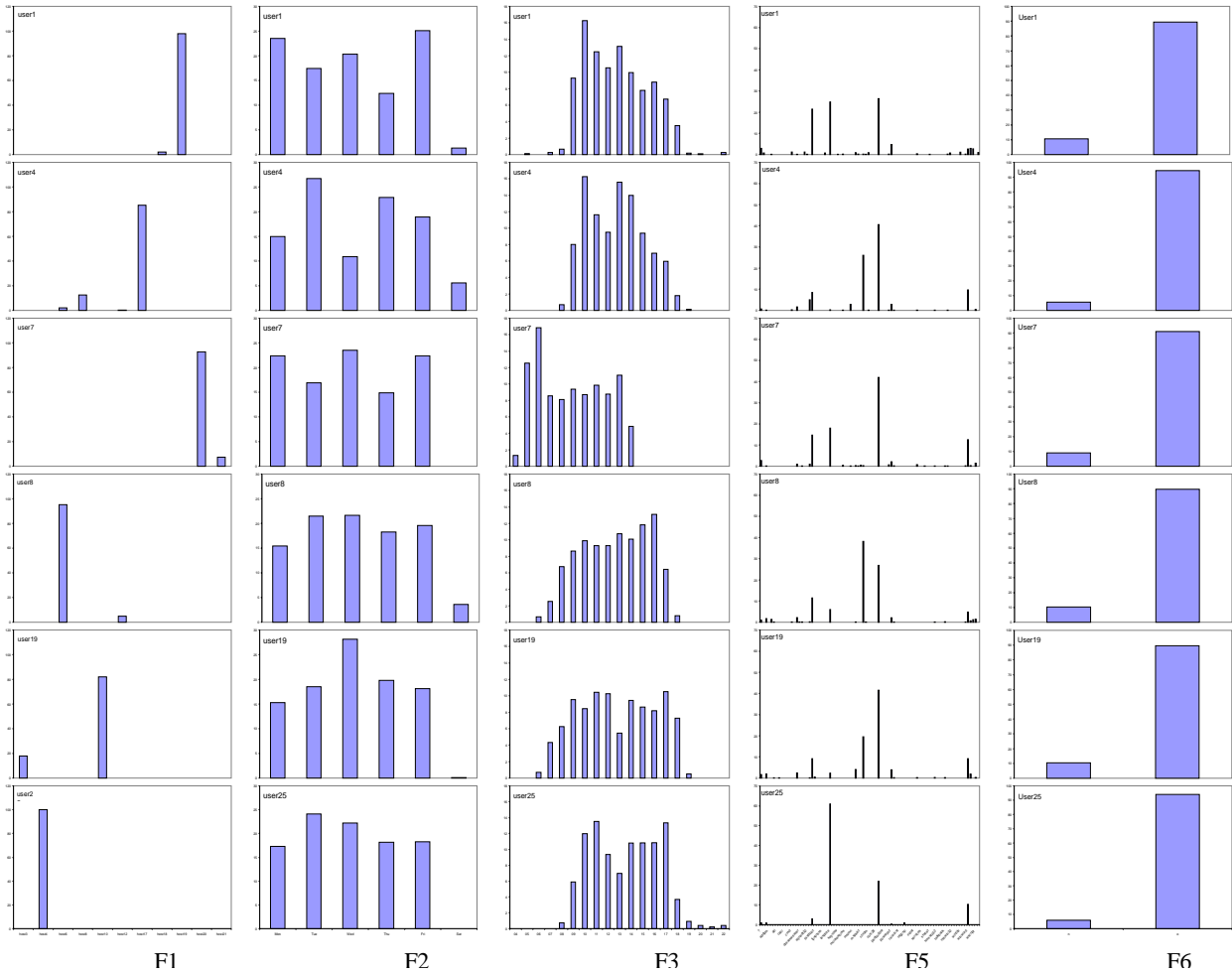


Figure 3: Histograms for the features with categorical measure

For the features with categorical measure, the frequencies of the possible categories are drawn as histogram in Figure 3. Histograms of F1 (host machine ID), F2 (day of week), F3 (time of day), and F4 (window process name) have different patterns among the users therefore these features can be selected as user profile and anomaly detecting features. However, F6 (window status), which has only two categories was not selected because it has almost same frequencies in two categories for all six users.

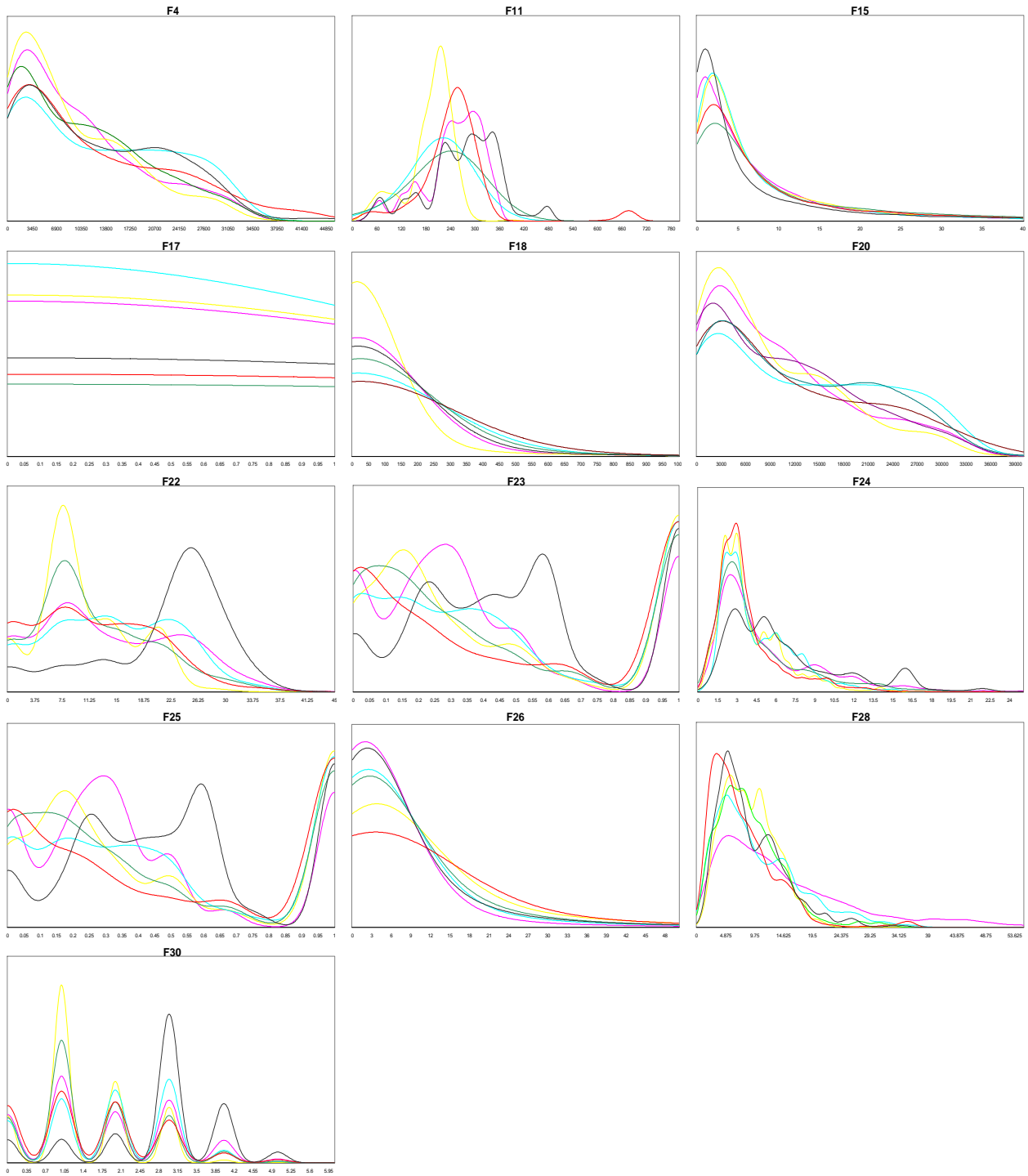


Figure 4: densities for the features with ordinary measure.

Figure 4 shows the probability densities of the features with ordinary measure that were estimated by the kernel density estimation. All the features show different patterns

of densities among the six users although the degrees of the differences vary. Even though the densities of F17 for each user are different, each density is not informative because they are uniformly distributed. Some features turned out to be highly correlated. F4 (number of seconds from the start of the session) and F20 (elapsed time since login whenever new window is opened) have almost same densities for each user, which means these two features are highly correlated. F23 (ratio of number of characters in protected words to total number of characters in window title) also has almost same densities with F25 (ratio of number of protected words to total number of words in window title). So, F20 and F23 were not included in the final feature sets.

Finally, 14 features (Categorical features: F1, F2, F3, F5; Ordinary features: F4, F11, F15, F18, F22, F23, F24, F26, F28, F30) were selected as feature sets. Based on these features, users past profile and the score values were computed by the methodology explained in chapter3 to detect unauthorized users.

#### **4.4 Test**

The performance of the anomaly detecting system using user profile is evaluated by how accurately the unauthorized users or the misuses of the authorized user are detected from authorized user or authorized usages under the allowed false alarm rate, where the false alarm is to detect authorized user as unauthorized user. The test of performance based on the real data is very hard to conduct because the unauthorized users are very rare in the host system and we don't know if there are unauthorized users in the test data.

To evaluate the performance of our system, all the records from six users in the test data were combined and assumed as records from specified user, for example User1. Each record is compared to the User1's past behavior that was learned from the past profile and produces the score value by the method explained in chapter 3. Since the score value is the measure of how much the user's behavior is abnormal compared to the past patterns of behavior, the records from real User1 are expected to have small score value because they will have the similar pattern to the past. While the records from five other users are expected to have large score values because they played a role of unauthorized user and will have different pattern from the past profile of User1. Nevertheless, since five other users are still authorized users and they may have common

patterns of behavior that is not expected in unauthorized users, this test method would give poorer performance than real situation.

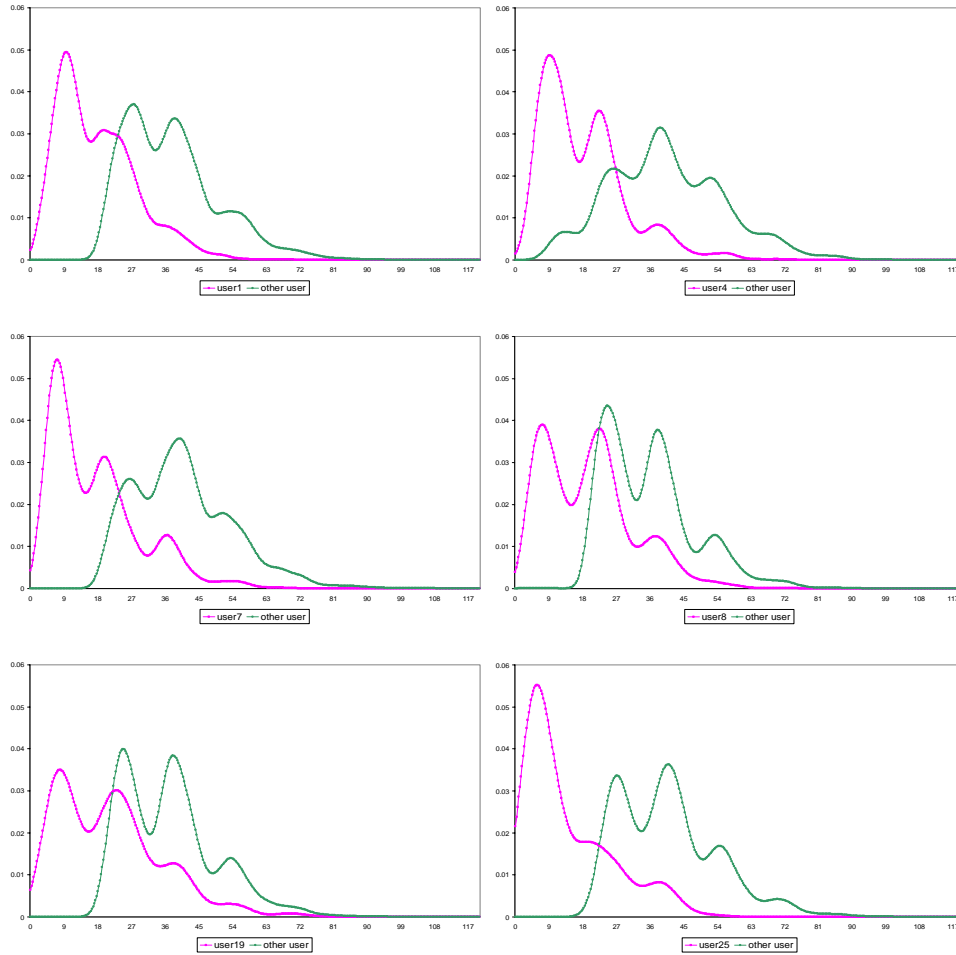


Figure 5: Densities of score values for each 6 user and other users groups

Figure 5 represents the estimated probability densities of score values for each user and other users group. For example, the first plot in the left top shows the situation where all users are assumed as User1 and the profile of User1 are used to compute score values. The distributions of score values for User1 and other users can be discriminated even though they are overlapped in some range of score values. Same procedures were conducted for other users. User1, User4, User7, User25 are quite well discriminated from other users group, while User8, and User19 show weak discriminations.

The false alarm rates for the real users were computed as the rate of records from a specified user with larger values than the given threshold to the total records from that specified user. The detecting rates for the unauthorized users were computed as the rate of the records from five other users with larger values than threshold to the total records from five other users. These rates are changed as the threshold values are changed. Table 5 shows the detecting rates for the three values of thresholds satisfying the given false alarm rate such as 1%, 5%, and 10%. The detecting performances for User1, User4, User7, and User25 are better than User8, and User19, as the results shown in Figure 5. The detecting rates under the false alarm rate of 1% are very low, and the detecting rates under 10% false alarm rate are about 60~70% for User1, User4, User7, User25. Again, this test is based on the authorized users. If data from unauthorized users are available, those data will give larger detecting rates.

| False Alarm Rate |                 | 1%    | 5%    | 10%   |
|------------------|-----------------|-------|-------|-------|
| User1            | Threshold Value | 47.4  | 38.2  | 31.9  |
|                  | Detecting Rate  | 19%   | 42.6% | 60.4% |
| User4            | Threshold Value | 54.8  | 40.2  | 34.3  |
|                  | Detecting Rate  | 16.7% | 46.3% | 64.8% |
| User7            | Threshold Value | 54.5  | 39.7  | 35.9  |
|                  | Detecting Rate  | 15.4% | 48.9% | 62.0% |
| User8            | Threshold Value | 53.3  | 41.3  | 37.3  |
|                  | Detecting Rate  | 10.3% | 26.3% | 41.1% |
| User19           | Threshold Value | 58.3  | 45.0  | 39.8  |
|                  | Detecting Rate  | 6.3%  | 22.1% | 36.2% |
| User25           | Threshold Value | 43.8  | 38.8  | 31.5  |
|                  | Detecting Rate  | 35.0% | 53.1% | 70.1% |

Table 5: Rates of detecting other users from real user under the false alarm rate of 1%, 5%, 10%

## 5. Conclusion

For the data related with window titles and process tables from unclassified window NT network, user profiling based on the frequency table from past data was conducted to learn the patterns of user behavior. Using these past profile information for six users, data from Jul 2002 to Dec 2002 were tested. When a specified user is assumed as an

authorized user and the other users are treated as unauthorized users, score values of a specified user and other users gave quite different distributions, especially when the specified users are User1, User4, User7, and User25. In the detecting system that detects a user as anomaly or unauthorized user when the score value is larger than given threshold, the rate of detecting other users from a specified user was about 60~70% in case of 10% false alarm rate. The performance is not so good in our test. One possible reason of low detecting rate is that the performance test was based on the other authorized users. The other possible reason is that user profiles were built based on so long periods (eight months) and updated every month because of the restrictions on the data. In the real situation, if the system detects unauthorized users rather than other authorized users, and user profiles are constructed based on shorter periods and updated everyday, this system is expected to give better performance.

One possible further works is suggested as following. Since exponential weighting method was not applied in our system, the score values have irregular component. Thus, to control this irregularity of the detecting performance, the future work for this project is to apply the exponential weighting method to this project and then, the detecting rates under the given false alarm rate are expected to be improved.

## **6. Acknowledgements**

The authors would like to thank Dr. Ryszard Michalski, Dr. Kenneth Kaufman and Mr. Jarek Pietrzykowski of the Machine Learning and Inference Laboratory for the use of the data set and preprocessing applications in the system.

## 7. References

- [1] Kaufman K., Cervone G. and Michalski R.S., "An Application of Symbolic Learning to Intrusion Detection: Preliminary Results From the LUS Methodology," Reports of the Machine Learning and Inference Laboratory, MLI 03-2, George Mason University, Fairfax, VA, June, 2003.
- [2] Lane, T. and Brodley, C.E., "Temporal Sequence Learning and Data Reduction for Anomaly Detection," ACM Transactions on Information and System Security 2, pp. 295-331, 1999.
- [3] Goldring, T., "Recent Experiences with User Profiling for Windows NT," Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, Baltimore, MD, 2002.
- [4] Sandeep Kumar and E. H. Spafford, "An Application of Pattern Matching in Intrusion Detection," CSD-TR-94-013, Coast TR 94-07, Department of Computer Sciences, Purdue University, 1994.
- [5] Jeremy Frank, "Artificial Intelligence and Intrusion Detection: Current and Future Directions," Division of Computer Science, University of California at Davis, 1994.
- [6] David J. Marchette, "Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint," New York: Springer, 2001.
- [7] Harold S. Javitz and Alfonso Valdes, "The SRI IDES Statistical Anomaly Detector". In Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy, May 1991
- [8] Teresa F. Lunt, "Detecting Intruders in Computer Systems", 1993 Conference on Auditing and Computer Technology
- [9] Debra Anderson, Teresa F. Lunt, Harold Javits, Ann Tamaru, and Alfonso Valdes, "Detecting Unusual Program Behavior Using the Statistical Component of the Next-generation Intrusion Detection Expert System (NIDES)" Computer Science Laboratory SRI-CSL-95-06, may 1995