KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection

Maheshkumar Sabhnani
EECS Dept, University of Toledo
Toledo, Ohio 43606 USA

Gursel Serpen
EECS Dept, University of Toledo
Toledo, Ohio 43606 USA

Abstract

Automated rule induction procedures like machine learning and statistical techniques result in rules that lack generalization and maintainability. Developing rules manually through incorporation of attack signatures results in meaningful but weak rules as it is difficult to define thresholds. This paper utilizes a hybrid procedure for developing rules by combining signature analysis with automated techniques to improve readability, comprehensibility, and maintainability of rules. Through the proposed rule-formulation technique, heuristic rules were developed for two remote-to-local (R2L) attacks using the KDD intrusion detection features and dataset. Empirical results show that high detection rates with low false alarms are observed for the warezmaster and warezclient attacks in the KDD data set. The utilized technique also highlighted a mislabeling problem in the KDD dataset for the two R2L attacks considered.

Keywords: Intrusion detection, Remote-to-local attack, Misuse detection, KDD dataset, Heuristic rule, Detection probability, False alarm rate

1. Introduction

Heuristic rules have been commonly used to detect remote-to-local (R2L) attacks [1] [2] [3]. R2L attack category offers the most diverse set of attacks in terms of attack execution, implementation, and dynamics: R2L attacks differ vastly in terms of signatures and the host against which they are executed. The diverse knowledge required to detect R2L attacks inspired many expert systems in literature including P-BEST [4], EMERALD [5], and RIPPER [6].

Rules are typically developed using features present in a given dataset. Automated rule-generation techniques including the C4.5 decision tree algorithm, which leverage machine learning and statistics, can build the optimal set of rules on given data. Though these rules are the best fit for the given dataset, they are difficult to comprehend and manage if the original dataset is slightly revised in terms of new records or new features are added. Consequently, it is problematical, at the least, to be able to predict their performance in an unknown environment as these rules are always data specific and not signature specific. Rules can also be developed using signature analysis of attacks: often such rules will be highly meaningful leading to elevated comprehensibility. Furthermore, rules formulated through signature analysis offer better generalization capability by the virtue of the process of generation. Another significant advantage that can easily be associated with signature-based rules is the increased level of maintainability and applicability. Relevant features can be extracted by studying the attack dynamics. The major challenge occurs when thresholds are defined for these rules, which is empirical in nature. Not defining thresholds properly results in weak rules and hence low probability of detection for the targeted attacks.

The advantage of using heuristic rules (data or signature based) is that they can detect attacks in real-time as they typically require very little processing time. Once it becomes feasible to define precise and comprehensive heuristic rules for a specific application, the misuse detection performance is guaranteed to be high. In the case of R2L attack detection, the expected performance criteria incorporate high detection and low false alarm rates. There are a number of limitations of heuristic rules as well. It is not a trivial task to develop precise rules, at least for certain applications, because human expert knowledge might not readily lend itself to an effective formulation in the form of an IF-THEN rule template. It is also a challenging task to write rules that are robust enough to work in unknown environments where there is significant noise as the rules might not offer adequate generalization capabilities.

The recent literature presents a set of significant efforts to detect R2L attacks using heuristic rules in networked computing environments: specifically the Defense Advanced Research Project Agency (DARPA) and Knowledge Discovery in Databases (KDD) datasets were utilized for most of these studies. In 1998, DARPA funded an "Intrusion Detection Evaluation Program (IDEP)" in Lincoln Laboratory at the Massachusetts Institute of Technology [7]. DARPA intrusion detection data was recreated on the simulated military network environment along with the different attacks embedded in it. The victim machines subjected to these attacks ran

Linux, SunOS^{TM}, and Solaris^{TM} operating systems. Three kinds of data was collected: transmission control protocol (TCP) packets using the "tcpdump" utility, basic security module (BSM) audit records using the Sun Solaris BSM utility, and system file dumps. Stolfo and Lee [3], one of the participants in DARPA 1998 program, used TCP packets to build the KDD dataset, which consisted of records based on individual TCP sessions. Each record had 41 features and the method used to derive these features is discussed in [8]. Data-mining techniques were utilized to generate features using the TCP packets for different connections. The KDD dataset is accessible through the UCI KDD archive [9].

Agarwal and Joshi [1] proposed a two-stage general-to-specific framework for learning a rule-based model (PNrule) to learn classifier models on a data set that has widely different class distributions in the training data set. The PNrule technique was evaluated on the KDD testing data set, which contained many new R2L attacks not present in the KDD training dataset. The proposed model was able to detect only 10.7% attacks in the R2L attack category although insignificant amount of false alarms were generated. The obvious disadvantage of this algorithm is that the rules are automatically generated leading to data set dependency with negligible generalization capability.

Levin [2] used Kernel Miner tool on the KDD data set. Kernel Miner is data-mining tool for classification of data and prediction of new cases using automatically generated decision trees. Using this tool and the KDD data set, Levin created a set of locally optimal decision trees (called the decision forest) from which optimal subset of trees (called the sub-forest) was selected for predicting new cases. Levin used only 10% of the KDD training data set randomly sampled from the entire training data set. Multi-class detection approach was used to detect different attack categories in the KDD data set. The final trees gave very high detection rates for all classes including the R2L in the entire training data set. The proposed classifier achieved only 7.32% detection and 2.5% false alarm rates for the R2L attacks in the KDD testing data set.

Yeung and Chow [10] proposed a novelty detection approach using non-parametric density estimation based on Parzen-window estimators with Gaussian kernels to build an intrusion detection system using normal data only. This novelty detection approach was employed to detect attack categories in the KDD data set. 30,000 randomly sampled normal records from the KDD training data set were used as training data to estimate the density of the model. Another 30,000 randomly sampled normal records (also from the KDD training data set) formed the threshold determination set, which had no overlap with the training data set. It is important to note that this model detects whether a record is intrusive or not. For the R2L attack category, 31.17% of R2L records in the KDD testing dataset were detected as intrusive patterns: authors did not report any information on false alarm rates. Even so this technique also failed to identify R2L attacks with a high detection rate.

Lee and Stolfo [11] used data mining techniques to collect KDD features from DARPA 1998 dataset. RIPPER rules were created using these data mining techniques to detect R2L attacks. The proposed model could detect only 20% of R2L attacks with a false alarm rate of 0.01 for the DARPA 1998 testing dataset.

The results of DARPA 1998 IDS evaluation program [12] indicate that the best model proposed was EMERALD that could detect only 35% of R2L attacks in the testing dataset. KDD dataset was used in the UCI KDD 1999 competition. The aim of the competition was to develop intrusion detection system models to address attack categories Probe, DoS, U2R, and R2L. Results of the KDD Cup 1999 competition [13] indicate that the winner was able to correctly classify only 7.82% of R2L records present in the KDD testing dataset.

Literature survey shows that the intrusion detection models proposed for R2L attacks failed to demonstrate desirable performance with high detection and low false alarm rates using KDD dataset. Heuristic rules seem to be popular to detect R2L attacks possibly due to the nature of these attacks. The intrusion detection models performed well on the KDD training dataset but failed to detect R2L attacks in the KDD testing dataset. This indicates that the attack signatures present in KDD training and testing datasets might not be correlated. This lack of correlation could occur if there are many new attacks in the testing dataset that have signatures different than those present in the training dataset. Hence to build a successful R2L detection model using the KDD data, both training and testing datasets will need to be analyzed. Further analysis of failure of various models in the literature indicates that R2L attacks significantly vary in terms of signatures and hence models that try to detect all R2L attacks using the same algorithm are highly likely to fail. This observation leads to the finding that each R2L attack should be considered and addressed with a detection algorithm individually.

This paper shall propose heuristic rules that will be created by analyzing the signatures of selected attacks in the R2L category. The rule formulation technique will use signature analysis to define a set of features and use automated techniques to assign various threshold values needed in the formulation of heuristic rules. A comprehensive and detailed understanding of R2L attack dynamics, mechanisms, and signatures is likely to help develop heuristic rules that will not only potentially offer high probability of detection but also a low false alarm rate. This paper shall analyze selected R2L attacks by combining both KDD training and testing datasets and propose heuristic rules to improve detection rate and reduce false alarms. The advantage of using signature-based heuristic rules is that they are not affected by noise in the training dataset. Hence well-formed, signature-based heuristic rules are expected to detect the attacks with low false and missed alarm rates if the KDD feature set can potentially provide adequate observables of attack dynamics.

The KDD dataset consists of various R2L attacks that project noticeably different and diverse attack signatures: warezmaster and warezclient involve uploading and downloading data from ftp servers, respectively. These two attacks represent typical attacks executed against any ftp server. This paper studies warezmaster and warezclient attacks as representative instances from the R2L category: attack specifications in terms of signatures, services used, and how the victims are involved for these two attacks are analyzed in detail. These two attacks will be used to elaborate on the proposed rule generation technique.

Section 2 presents proposed rules for the two R2L attacks considered. It discusses the signatures of the same two R2L attacks, typical features that must be observed, and relevant features present in the KDD dataset that can help with the detection of these attacks. The same section will also discuss how features and thresholds were set for different rules to detect these attacks. Section 3 discusses the performance of proposed rules on the KDD dataset. Finally conclusions are discussed in Section 4.

2. Formulation of heuristic rules for R2L attacks

In this section, signatures of various R2L attacks will be analyzed. The aim will be to extract relevant features from signatures that must be selected to conclusively observe the attack in a networked environment. These rules will directly map the attack signatures. Since these rules will be tested on the KDD testing data set, an attempt shall be made to formulate rules in terms of the KDD features. Various thresholds (in premises of rules) will be set as they are observed in the KDD datasets or from the rules derived through the C4.5 decision tree algorithm on the KDD datasets. The proposed rules shall be discussed with respect to both attack signatures and the feature sets to facilitate understanding of the mapping between the two.

Thresholds for different rules created were obtained as follows. First the KDD training and testing data sets were merged. For a given attack under consideration in this paper, each record in the merged KDD data set was labeled as either belonging to or not belonging to that attack. C4.5 decision tree was then applied on this relabeled data set to obtain the rule set for the same attack. These rule sets were consulted whenever needed to define thresholds for features. This technique can easily be extended to other R2L attacks as well.

2.1. Warezmaster Attack

Warezmaster exploits a system bug associated with a file transfer protocol (FTP) server. Normally, guest users are never allowed write permissions on an FTP server. Hence they can never upload files on the server. Most public domain FTP servers have guest accounts for downloading data. Anyone can login to an FTP server using guest accounts. This attack takes place when an FTP server has, by mistake, given write permissions to users on the system. Hence any user can login and upload files. During the execution of the attack, the attacker logs on the server using the guest account. The attacker then creates a hidden directory and uploads "warez" (copies of illegal software) onto the server. Other users can then later download these files. One simple and obvious way to prevent this attack is to assign correct permissions to the users on the FTP server.

Since this attack requires uploading files during an FTP connection, the relevant features that can be observed are an FTP session in progress, files being uploaded, and hidden directories being created. All required features were present in the KDD data set except the one monitoring the upload of data. This can be indirectly observed by analyzing the amount of data that is exchanged between the source and destination in a given amount of time. If a huge amount of data is sent from source as compared to that from destination then it can be assumed that data is being uploaded. A KDD feature that records hot indicators can be used to monitor if any hidden directories are created during the FTP session. Consequently, two separate and independent rules are proposed to detect this attack. Features used in Rules 2.1a and 2.1b are the result of signature analysis. The thresholds are set based on the observations from the C4.5 generated rules. Rules 2.1a and 2.1b are formulated as follows:

> *"If during an FTP session, large amount is data is sent from source as compared to destination then warezmaster attack can be concluded."*

(duration > 265) ∧
(protocol = tcp) ∧
(service = ftp ∨ ftp_data) ∧
(source_bytes > 265616) ∧
(destination_bytes = 0) ⇒ Warezmaster Attack

… **Rule 2.1a**

> *"If a guest has logged in through an FTP connection, and hidden directories are created then warezmaster attack can be concluded."*

(protocol = tcp) $\wedge$
(service = ftp $\vee$ ftp_data) $\wedge$
(hot > 0) $\wedge$
(hot <= 2) $\wedge$
(is_guest_login = 1) $\Rightarrow$ Warezmaster Attack

… **Rule 2.1b**

In formal terms, a super rule that leverages the above two rules can be stated as follows:

(Rule 2.1a $\vee$ Rule 2.1b) $\Rightarrow$ Warezmaster Attack

An FTP connection can be observed by verifying that the protocol is TCP and the service is FTP or FTP_DATA. Rule 2.1a suggests that the FTP connection has been active for an extended period of time (*duration* > 265 seconds) and a large amount of data has been transferred from the source machine (*source_bytes* > 265616 bytes) with no data received from destination (victim) machine (*destination_bytes* = 0 bytes). This indicates that the user is uploading data to the FTP server. Thresholds for *duration* and *source_bytes* have been selected from the following rules generated using the C4.5 algorithm on the merged KDD training and testing data sets:

(duration > 265) $\wedge$
(destination_bytes <= 688) $\wedge$
(is_guest_login = 1) $\Rightarrow$ Warezmaster Attack

… **Rule C2.1a**

(source_bytes > 265616) $\wedge$
(source_bytes <= 283618) $\Rightarrow$ Warezmaster Attack

… **Rule C2.1b**

Rule 2.1b suggests that hidden directories (*hot* = 1 or 2) are created when a guest has logged in (*guest_login* = 1) the victim machine. This also indicates that guest account has write permissions on the machine. Though activity monitored by *hot* indicators does not necessarily mean that the guest user is creating hidden directories, it definitely indicates improper activity not compatible with the privileges of a guest user. Threshold for *hot* indicator was set by empirically observing the KDD records, where most records having more than two *hot* indicators were labeled as not a warezmaster attack.

2.2. Warezclient Attack

Warezclient attack can be launched by any legal user during an FTP connection after warezmaster attack has been executed. During warezclient attack, users download the illegal "warez" software that was posted earlier through a successful warezmaster attack. Since this process requires downloading files from the FTP server, attack dynamics project a perfectly legal process. The only feature that can be observed to detect this attack is downloading files from hidden directories or directories that are not normally accessible to guest users on the FTP server. This will require keeping track of all legal directories and checking whether the files being downloaded during FTP sessions belong to legal directories or not. The KDD content feature '*hot'* can be utilized to detect whether such suspicious activity took place. In brief, if many *hot* indicators are being observed in a small duration of time during the FTP session, it can be concluded that warezclient attack is being executed on the victim machine. Note that since the user is downloading files from the server, he/she needs to be logged in as a normal or a guest/anonymous user. The rule to detect warezclient attack in compliance with the KDD data set features can be stated as follows:

> *"If a user, during an FTP session, triggers notably many hot indicators to be set in a small duration of time then the user maybe downloading illegally posted software from the server."*

In terms of the KDD features the rule can be defined as:

(duration < 5) $\wedge$
(protocol = tcp) $\wedge$
(service = ftp $\vee$ ftp_data) $\wedge$
(logged_in = 1 $\vee$ is_guest_login = 1)
(hot > 25) $\Rightarrow$ Warezclient Attack

… **Rule 2.2**

Rule 2.2 suggests that if within five seconds (duration < 5 seconds) of an FTP connection/session, there are many *hot* indicators (hot > 25) being set by a logged user then it is highly likely that warezclient attack is being executed.

Threshold value of *hot* can be inferred by comparing the Rules C2.2a and C2.2b, shown below and created by the C4.5 algorithm on the merged KDD training and testing data sets.

Rule C2.2a suggests that if number of *hot* indicators set is less than or equal to 25 then the KDD record does not represent warezclient. Rule C2.2b suggests that if the number of *hot* indicators set is more than 25 then warezclient attack can be concluded. Threshold value for *duration* was set empirically by executing multiple rules having different threshold values of *duration* on the KDD data set. Threshold value of 5 for the duration gave minimum number of false alarms and maximum detection rate in the KDD training data set. Results obtained using Rule 2.2 is discussed in Section 3.2.

(duration <= 4685) ∧
(hot > 0) ∧
(hot <= 25) ⇒ not Warezclient Attack

… **Rule C2.2a**

(destination_bytes <=3299) ∧
(hot > 25) ⇒ Warezclient Attack

… **Rule C2.2b**

3. Performance evaluation of proposed rules on the KDD dataset

Section 2 defined three rules for the two attacks considered in the R2L attack category. These rules used basic and content features from the KDD dataset and next will be tested on the KDD training and testing data sets to observe their performance with respect to detection, false alarm, and missed alarm rates. This section presents the test results of these rules on the merged KDD training and testing data sets. If the rules are well-formed, then the detection rate is expected to be high, while concurrently achieving low false alarm and missed alarm rates.

3.1. Warezmaster Attack

Two rules were proposed for the warezmaster attack in Section 2.1. Table 1 indicates the performance of these rules on the KDD training and testing data sets combined, which had a total of 1622 warezmaster attack records. Table 1 indicates that two warezmaster rules (2.1a & 2.2b) were able to detect more than 65% attack records with very low false alarm rates (0.005%). This suggests that these rules adequately but not necessary precisely map the signatures of the warezmaster attack. The missed alarms (557 records) show that some attack records were not detected. Figure 1 presents examples of some of these missed records as they exist in the KDD data set. These missed alarms were generated because records similar to the ones in Figure 1 indicated that there was no source data transferred to victim machine (*source_bytes* = 0) during an FTP session while these records were labeled as warezmaster attack record in the KDD data sets. Additionally some other records indicated that there was no user logged on the FTP session, but the record was still labeled as warezmaster. In both of these situations, warezmaster attack is not possible suggesting that these records were possibly incorrectly labeled in the KDD training and testing data sets.

Table 1. Performance of proposed rules for warezmaster attack on the KDD dataset

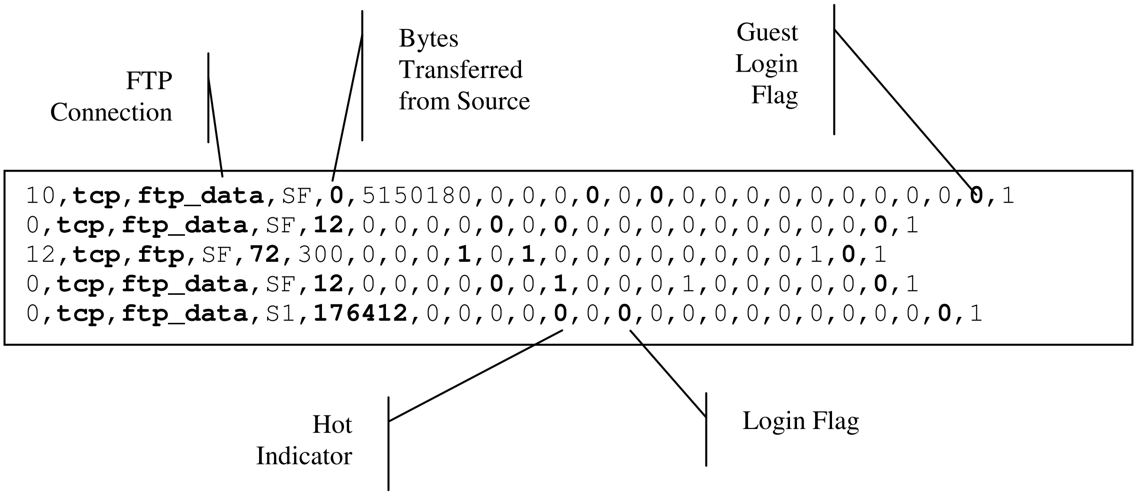
	Number of Records	Percentage of Records
Positive Detection	1065	65.6597
Negative Detection	1384328	99.9950
False Alarms	70	0.0051
Missed Alarms	557	34.3403

3.2. Warezclient Attack

Rule 2.2 was tested on both KDD training and testing datasets and the performance is shown in Table 2. There were a total of 893 warezclient attack records present in the KDD training and testing data sets combined. Note that the process of downloading files, as typically happens during a warezclient attack, is not an illegal process and hence many missed alarms are expected to occur. Table 2 shows the rule achieved ideal negative detection with 0 false alarms. The positive detection was more than 30%.

Table 2. Performance of proposed rule for warezclient attack on the KDD dataset

	Number of Records	Percentage of Records
Positive Detection	270	30.2352
Negative Detection	1385127	100.0000
False Alarms	0	0.0000
Missed Alarms	623	69.7648



```
10,tcp,ftp_data,SF,0,5150180,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
0,tcp,ftp_data,SF,12,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
12,tcp,ftp,SF,72,300,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1,0,1
0,tcp,ftp_data,SF,12,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,1
0,tcp,ftp_data,S1,176412,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
```

Figure 1. Missed alarm examples for warezmaster attack in the KDD dataset

The reason for a large number of missed alarms is because the dynamics of this attack is very similar to the normal behavior of the FTP download process. Some missed alarms observed are shown in Figure 2. These examples indicate that *destination_bytes* was equal to zero, hence there was no information sent back by the victim i.e. FTP server. This means no file downloads occurred. Hot indicator values generated during the FTP session were small enough to conclude the attack. This is because the user accesses the hidden directories and hence a large value for hot indicators is expected to occur. These observations suggest that there might be mislabeling problems in the KDD dataset for warezclient records.

KDD training and testing datasets had a total of 1,386,020 unique records. 2,515 of these records represented warezmaster and warezclient attacks. The utilized technique could detect 1,335 out of 2,515 as attack records. Hence a combined detection of 53.08% was achieved. Also, only 70 false alarms were generated from the remaining 1,383,435 normal records achieving a combined false alarm rate of 0.005% for the two considered R2L attacks.

4. Conclusions

This paper utilized a technique for creating heuristic rules by combining both signature analysis and automated methods. The technique assisted in improving readability, comprehensibility, and maintainability of heuristic rules. Heuristic rules were proposed for two R2L attacks – warezmaster and warezclient. Probability of detection and false alarm rates are computed on the KDD training and testing datasets combined for each considered attack. The overall performance, combined for the three attacks is reasonably good: an average 53.08% detection rate is achieved with only 0.005% false alarm rate for the two specific attacks in the R2L category.

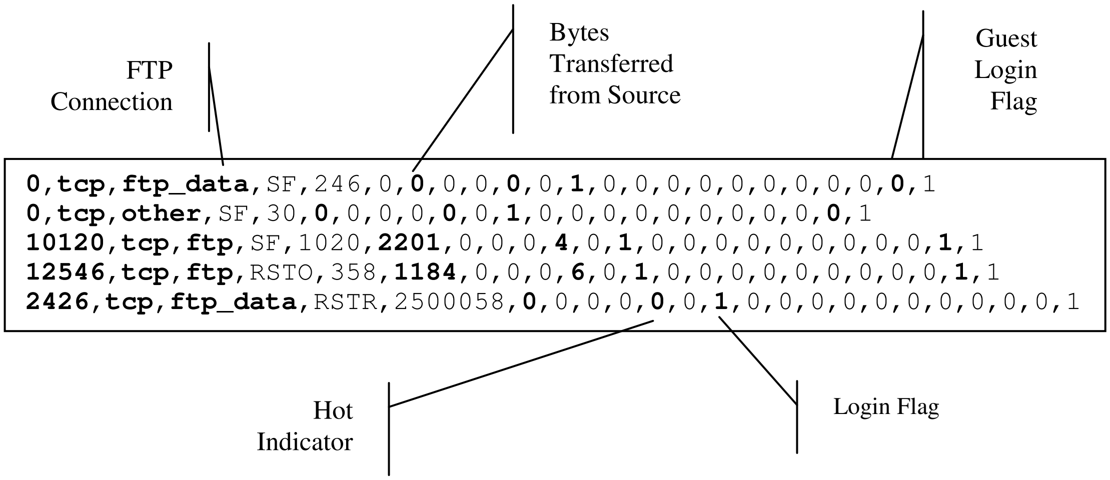
Since the rule thresholds were set using knowledge of both training and testing data sets, the performance of proposed rules on new records in KDD testing data set cannot be interpreted in a precise fashion. On the other hand, signature analysis performed on two R2L attacks is likely to facilitate a predictably good level of detection and false alarm rates for other attacks in the KDD data sets and other type of data derived on real host and network traffic.

Data mislabeling problems in KDD dataset were highlighted for records showing missed alarms. If the record labels are consistent with the attack signatures, then the proposed heuristic rules are poised to perform much better. Nevertheless, the heuristic rule set model performs appreciably well with relatively high probability of detection and low false alarm rates.

5. References

[1] R. Agarwal, and M. V. Joshi, "PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection)", *Technical Report TR 00-015*, Department of Computer Science, University of Minnesota, 2000.

[2] I. Levin, "KDD-99 Classifier Learning Contest LLSoft's Results Overview", *SIGKDD Explorations, ACM SIGKDD*, January 2000, Vol. 1 (2), pp. 67-75.



```
0,tcp,ftp_data,SF,246,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1
0,tcp,other,SF,30,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1
10120,tcp,ftp,SF,1020,2201,0,0,0,4,0,1,0,0,0,0,0,0,0,0,0,1,1
12546,tcp,ftp,RSTO,358,1184,0,0,0,6,0,1,0,0,0,0,0,0,0,0,0,1,1
2426,tcp,ftp_data,RSTR,2500058,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1
```

Figure 2. Missed alarm examples for warezclient attack in the KDD dataset

[3] W. Lee, S. J. Stolfo, K. W. Mok, “A Data Mining Framework for Building Intrusion Detection Models”, IEEE Symposium on Security and Privacy, Oakland, California, 1999, pp. 120-132.

[4] U. Lindqvist, P. Porras, “Detecting Computer and Network Misuse through the Production-based Expert System Toolset (P-{BEST})”, *IEEE Symposium on Security and Privacy*, 1999, pp. 146-161.

[5] P. A. Porras, and P. G. Neumann, “EMERALD: Event monitoring enabling responses to anomalous live disturbances”, *In Proceedings of the 20th National Information Systems Security Conference*, Baltimore, Maryland, 1997, pp. 353-365.

[6] W. W. Cohen, “Fast effective rule induction”, *Proceedings of the 12th International Conference on Machine Learning (ML-95)*, Lake Tahoe, CA: Morgan Kaufmann, 1995, pp. 115-123.

[7] DARPA data set, 1998. http://www.ll.mit.edu/IST/-ideval/data/1998/1998_data_index.html, cited April 2003.

[8] W. Lee, S. J. Stolfo, and K. W. Mok, “Mining in a Data-Flow Environment: Experience in Network Intrusion Detection”, *In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999, pp. 114-124.

[9] KDD data set, 1999; http://kdd.ics.uci.edu/databases/-kddcup99/kddcup99.html, cited April 2003.

[10] D. Y. Yeung, and C. Chow, “Parzen-window Network Intrusion Detectors”, *In Proceedings of the Sixteenth International Conference on Pattern Recognition*, Quebec City, Canada, August 2002, Vol. 4, pp. 385-388.

[11] W. Lee, and S. Stolfo, “A Framework for Constructing Features and Models for Intrusion Detection Systems”, *ACM Transactions on Information and System Security*, November 2000, Vol. 3 (4), pp. 227-261.

[12] R. P. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, “The 1999 DARPA Off-Line Intrusion Detection Evaluation”, *Computer Networks*, 2000, Vol. 34 (4), pp. 579-595.

[13] C. Elkan, “Results of the KDD’99 Classifier Learning”, *SIGKDD Explorations, ACM SIGKDD*, January 2000, Vol 1. (2), pp. 63-64.