CERIAS Tech Report 2002-70
Fraud Formalization and Detection
by B Bhargava, Y Zhong, Y Lu
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Fraud Formalization and Detection*

Bharat Bhargava, Yuhui Zhong, and Yunhua Lu

Center for Education and Research in Information Assurance and Security (CERIAS)
and Department of Computer Sciences
Purdue University, West Lafayette, IN 47907, USA
{bb,zhong,luy}@cs.purdue.edu

Abstract. A fraudster can be an impersonator or a swindler. An impersonator is an illegitimate user who steals resources from the victims by "taking over" their accounts. A swindler is a legitimate user who intentionally harms the system or other users by deception. Previous research efforts in fraud detection concentrate on identifying frauds caused by impersonators. Detecting frauds conducted by swindlers is a challenging issue. We propose an architecture to catch swindlers. It consists of four components: profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making component. Profile-based anomaly detector outputs fraud confidence indicating the possibility of fraud when there is a sharp deviation from usual patterns. State transition analysis provides state description to users when an activity results in entering a dangerous state leading to fraud. Deceiving intention predictor discovers malicious intentions. Three types of deceiving intentions, namely uncovered deceiving intention, trapping intention, and illusive intention, are defined. A deceiving intention prediction algorithm is developed. A user-configurable risk evaluation function is used for decision making. A fraud alarm is raised when the expected risk is greater than the fraud investigation cost.

1 Introduction

Fraudsters can be classified into two categories: impersonators and swindlers. An *impersonator* is an illegitimate user who steals resources from the victims by "taking over" their accounts. A *swindler*, on the other hand, is a legitimate user who intentionally harms the system or other users by deception. Taking *superimposition fraud* in telecommunication [7] as an example, impersonators impose their usage on the accounts of legitimate users by using cloned phones with Mobile Identification Numbers (MIN) and Equipment Serial Numbers (ESN) stolen from the victims. Swindlers obtain legitimate accounts and use the services without the intention to pay bills, which is called *subscription fraud*.

Impersonators can be forestalled by utilizing cryptographic technologies that provide strong protection to users' authentication information. The idea of separation of duty may be applied to reduce the impact of a swindler. The essence

* This research is supported by NSF grant IIS-0209059.

Y. Kambayashi, M. Mohania, W. Wöß (Eds.): DaWaK 2003, LNCS 2737, pp. 330–339, 2003.

is to restrict the power an entity (e.g., a transaction partner) can have to prevent him from abusing it. An empirical example of this idea is that laws are set, enforced and interpreted by different parties. Separation of duty can be implemented by using access control mechanisms such as role based access control mechanism, or lattice-based access control model [8]. Separation of duty policies and other mechanisms, like dual-log bookkeeping [8] reduce frauds but cannot eliminate them. For example, for online auctions, such as eBay, sellers and buyers have restricted knowledge about the other side. Although eBay, as a trusted third party, has authentication services to check the information provided by sellers and buyers (e.g. phone numbers), it is impossible to verify all of them due to the high quantities of online transactions. Fraud is a persistent issue under such an environment.

In this paper, we concentrate on swindler detection. Three approaches are considered: (a) detecting an entity's activities that deviate from normal patterns, which may imply the existence of a fraud; (b) constructing state transition graphs for existing fraud scenarios and detecting fraud attempts similar to the known ones; and (c) discovering an entity's intention based on his behavior. The first two approaches can also be used to detect frauds conducted by impersonators. The last one is applicable only for swindler detection.

The rest of this paper is organized as the follows. Section 2 introduces the related work. Definitions for fraud and deceiving intentions are presented in Section 3. An architecture for swindler detection is proposed in Section 4. It consists of a profile-based anomaly detector, a state transition analysis component, a deceiving intention predictor, and a decision-making component. The functionalities and design considerations for each component are discussed. An algorithm for predicting deceiving intentions is designed and studied via experiments. Section 5 concludes the paper.

2 Related Work

Fraud detection systems are widely used in telecommunications, online transactions, the insurance industry, computer and network security [1, 3, 6, 11]. The majority of research efforts addresses detecting impersonators (e.g. detecting superimposition fraud in telecommunications). Effective fraud detection uses both fraud rules and pattern analysis. Fawcett and Provost proposed an adaptive rule-based detection framework [4]. Rosset et al. pointed out that standard classification and rule generation were not appropriate for fraud detection [7]. The generation and selection of a rule set should combine both user-level and behavior-level attributes. Burge and Shawe-Taylor developed a neural network technique [2]. The probability distributions for current behavior profiles and behavior profile histories are compared using Hellinger distances. Larger distances indicate more suspicion of fraud.

Several criteria exist to evaluate the performance of fraud detection engines. ROC (Receiver Operating Characteristics) is a widely used one [10, 5]. Rosset et al. use accuracy and fraud coverage as criteria [7]. *Accuracy* is the number

of detected instances of fraud over the total number of classified frauds. *Fraud coverage* is the number of detected frauds over the total number of frauds. Stolfo et al. use a cost-based metric in commercial fraud detection systems [9]. If the loss resulting from a fraud is smaller than the investigation cost, this fraud is ignored. This metric is not suitable in circumstances where such a fraud happens frequently and causes a significant accumulative loss.

3 Formal Definitions

Frauds by swindlers occur in cooperations where each entity makes a commitment. A swindler is an entity that has no intention to keep his commitment.

Commitment is the integrity constraints, assumptions, and conditions an entity promises to satisfy in a process of cooperation. Commitment is described by using conjunction of expressions. An expression is (a) an equality with an attribute variable on the left hand side and a constant representing the expected value on the right hand side, or (b) a user-defined predicate that represents certain complex constraints, assumptions and conditions. A user-defined Boolean function is associated with the predicate to check whether the constraints, assumptions and conditions hold.

Outcome is the actual results of a cooperation. Each expression in a commitment has a corresponding one in the outcome. For an equality expression, the actual value of the attribute is on the right hand side. For a predicate expression, if the use-define function is true, the predicate itself is in the outcome. Otherwise, the negation of the predicate is included.

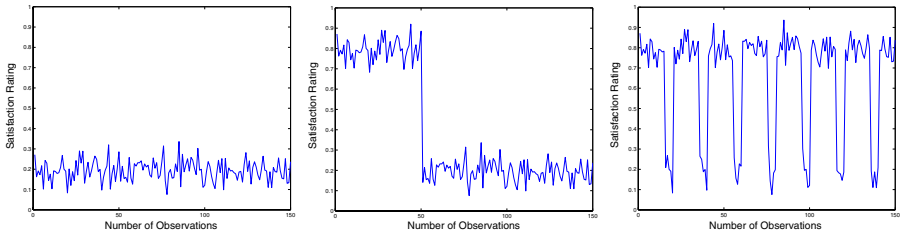
Example: A commitment of a seller for selling a vase is (Received_by = 04/01) $\wedge$ (Prize = \$1000) $\wedge$ (Quality = A) $\wedge$ ReturnIfAnyQualityProblem. This commitment says that the seller promises to send out one "A" quality vase at the price of \$1000. The vase should be received by April 1^{st}. If there is a quality problem, the buyer can return the vase. An possible outcome is (Received_by = 04/05) $\wedge$ (Prize = \$1000) $\wedge$ (Quality = B) $\wedge$ $\neg$ReturnIfAnyQualityProblem. This outcome shows that the vase of quality "B", was received on April 5^{th}. The return request was refused. We may conclude that the seller is a swindler.

Predicates or attribute variables play different roles in detecting a swindler. We define two properties, namely intention-testifying and intention-dependent.

Intention-testifying: A predicate P is intention-testifying if the presence of $\neg$P in an outcome leads to the conclusion that a partner is a swindler. An attribute variable V is intention-testifying if one can conclude that a partner is a swindler when V's expected value is more desirable than the actual value.

Intention-dependent: A predicate P is intention-dependent if it is possible that a partner is a swindler when $\neg$P appears in an outcome. An attribute variable V is intention-dependent if it is possible that a partner is a swindler when its expected value is more desirable than the actual value.

An intention-testifying variable or predicate is intention-dependent. The opposite direction is not necessarily true.



(a) Uncovered deceiving intention (b) Trapping intention (c) Illusive intention

Fig. 1. Deceiving intention

In the above example, ReturnIfAnyQualityProblem can be intention-testifying or intention-dependent. The decision is up to the user. *Prize* is intention-testifying since if the seller charges more money, we believe that he is a swindler. *Quality* and *received_by* are defined as intention-dependent variables considering that a seller may not have full control on them.

3.1 Deceiving Intentions

Since the intention-testifying property is usually too strong in real applications, variables and predicates are specified as intention-dependent. A conclusion that a partner is a swindler cannot be drawn with 100% certainty based on one intention-dependent variable or predicate in one outcome. Two approaches can be used to increase the confidence: (a) consider multiple variables or predicates in one outcome; and (b) consider one variable or predicate in multiple outcomes. The second approach is applied in this paper.

Assume a satisfaction rating ranging from 0 to 1 is given for the actual value of each intention-dependent variable in an outcome. The higher the rating is, the more satisfied the user is. The value of 0 means totally unacceptable and the value of 1 indicates that actual value is not worse than the expected value. For example, if the quality of received vase is B, the rating is 0.5. If the quality is C, the rating drops to 0.2. For each intention-dependent predicate P, the rating is 0 if ¬P appears. Otherwise, the rating is 1. A satisfaction rating is related to an entity's deceiving intention as well as some unpredictable factors. It is modelled by using random variables with normal distribution. The mean function $f_m(n)$ determines the mean value of the normal distribution at the the n^{th} rating.

Three types of deceiving intentions are identified.

Uncovered deceiving intention: The satisfaction ratings associated with a swindler having uncovered deceiving intention are stably low. The ratings vary in a small range over time. The mean function is defined as $f_m(n) = M$, where M is a constant. Figure 1a shows satisfaction ratings with $f_m(n)$=0.2. The fluctuation of ratings results from the unpredictable factors.

Trapping intention: The rating sequence can be divided into two phases: preparing and trapping. A swindler behaves well to achieve a trustworthy image before he conducts frauds. The mean function can be defined as:

$$f_m(n) = \begin{cases} m_{high}, & \text{n} \leq \text{n}_0; \\ m_{high}, & \text{otherwise.} \end{cases} \quad \textit{Where } n_0 \textit{ is the turning point.}$$

Figure 1b shows satisfaction ratings for a swindler with trapping intention. $\text{F}_m(\text{n})$ is 0.8 for the first 50 interactions and 0.2 afterwards.

Illusive intention: A smart swindler with illusive intention, instead of misbehaving continuously, attempts to cover the bad effects by intentionally doing something good after misbehaviors. He repeats the process of preparing and trapping. $\text{f}_m(\text{n})$ is a periodic function. For simplicity, we assume the period is N, the mean function is defined as:

$$f_m(n) = \begin{cases} m_{high}, & (\text{n mod N}) < \text{n}_0; \\ m_{high}, & \text{otherwise.} \end{cases}$$

Figure 1c shows satisfaction ratings with period of 20. In each period, $\text{f}_m(\text{n})$ is 0.8 for the first 15 interactions and 0.2 for the last five.

4 Architecture for Swindler Detection

Swindler detection consists of profile-based anomaly detector, state transition analysis, deceiving intention predictor, and decision-making. Profile-based anomaly detector monitors suspicious actions based upon the established patterns of an entity. It outputs *fraud confidence* indicating the possibility of a fraud. State transition analysis builds a state transition graph that provides *state description* to users when an activity results in entering a dangerous state leading

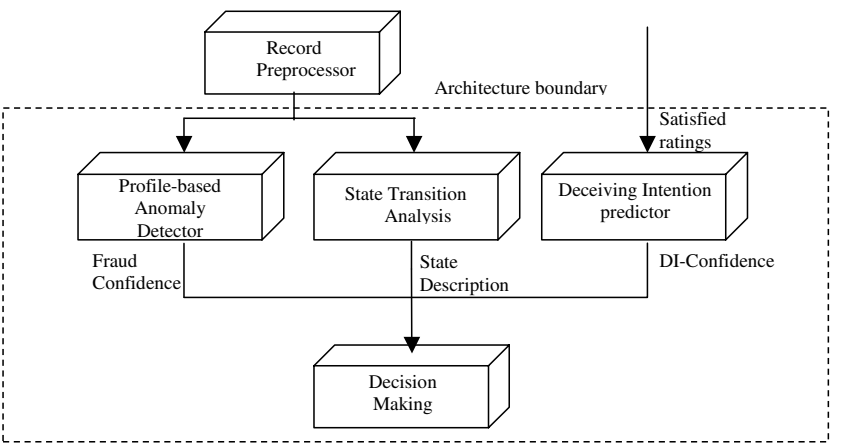


Fig. 2. Architecture for swindler detection

to a fraud. Deceiving intention predictor discovers deceiving intention based on satisfaction ratings. It outputs *DI-confidence* to characterize the belief that the target entity has a deceiving intention. DI-confidence is a real number ranging over [0,1]. The higher the value is, the greater the belief is.

Outputs of these components are feed into decision-making component that assists users to reach decisions based on predefined policies. Decision-making component passes warnings from state transition analysis to user and display the description of next potential state in a readable format. The expected risk is computed as follows.

f(fraud confidence, DI-confidence, estimated cost) = max(fraud confidence, DI-confidence) $\times$ *estimated cost*

Users can replace this function according to their specific requirements. A fraud alarm will arise when expected risk is greater than fraud-investigating cost. In the rest of this section, we concentrate on the other three components.

4.1 Profile-Based Anomaly Detector

As illustrated in fig. 3, profile-based anomaly detector consists of rule generation and weighting, user profiling, and online detection.

Rule generation and weighting: Data mining techniques such as association rule mining are applied to generate fraud rules. The generated rules are assigned weights according to their frequency of occurrence. Both entity-level and behavior-level attributes are used in mining fraud rules and weighting. Normally, a large volume of rules will be generated.

User profiling: Profile information characterizes both the entity-level information (e.g. financial status) and an entity's behavior patterns (e.g. interested products). There are two sets of profiling data, one for history profiles and the other for current profiles. Two steps, variable selection followed by data filtering, are used for user profiling. The first step chooses variables characterizing the normal behavior. Selected variables need to be comparable among different entities.

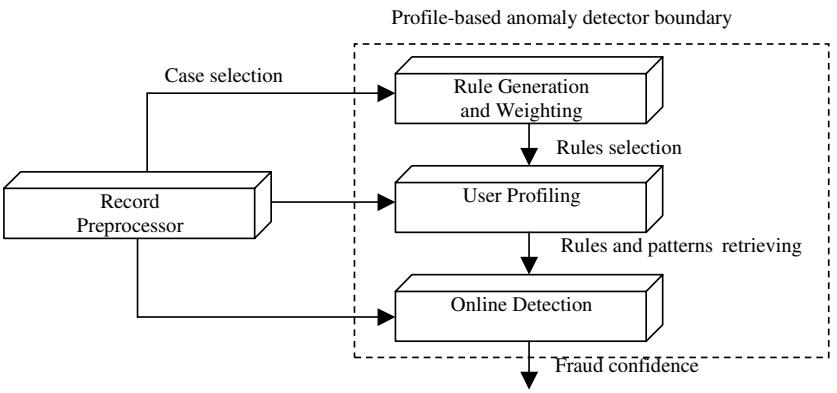


Fig. 3. Profile-based anomaly detector

Profile of the selected variable must show a pattern under normal conditions. These variables need to be sensitive to anomaly (i.e., at least one of these patterns is not matched in occurrence of anomaly). The objective of data filtering for history profiles is data homogenization (i.e. grouping similar entities). The current profile set will be dynamically updated according to behaviors. As behavior level data is large, decay is needed to reduce the data volume. This part also involves rule selection for a specific entity based on profiling results and rules. The rule selection triggers the measurements of normal behaviors regarding the rules. These statistics are stored in history profiles for online detection.

Online detection: The detection engine retrieves the related rules from the profiling component when an activity occurs. It may retrieve the entity's current behavior patterns and behavior pattern history as well. Analysis methods such as Hellinger distance can be used to calculate the deviation of current profile patterns to profile history patterns. These results are combined to determine fraud confidence.

4.2 State Transition Analysis

State transition analysis models fraud scenarios as series of states changing from an initial secure state to a final compromised state. The initial state is the start state prior to actions that lead to a fraud. The final state is the resulting state of completion of the fraud. There may be several intermediate states between them. The action, which causes one state to transit to another, is called the signature action. Signature actions are the minimum actions that lead to the final state. Without such actions, this fraud scenario will not be completed.

This model requires collecting fraud scenarios and identifying the initial states and the final states. The signature actions for that scenario are identified in backward direction. The fraud scenario is represented as a state transition graph by the states and signature actions.

A *danger* factor is associated with each state. It is defined by the distance from the current state to a final state. If one state leads to several final states, the minimum distance is used. For each activity, state transition analysis checks the potential next states. If the maximum value of the danger factors associated with the potential states exceeds a threshold, a warning is raised and detailed state description is sent to the decision-making component.

4.3 Deceiving Intention Predictor

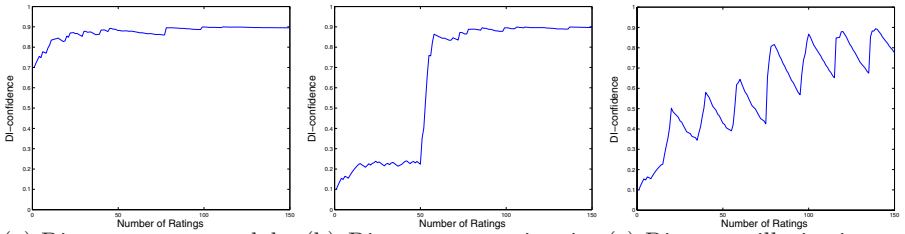
The kernel of the predictor is the deceiving intention prediction (DIP) algorithm. DIP views the belief of deceiving intention as the complementary of trust belief. The trust belief about an entity is evaluated based on the satisfaction sequence $<R_1, R_2, \ldots, R_n>$, R_n is the most recent one, which contributes to a portion of α to the trust belief. The rest portion comes from the previous trust belief that is determined recursively. For each entity, DIP maintains a pair of factors (i.e. *current construction factor* Wc and *current destruction factor* Wd). If integrating R_n will increase trust belief, $\alpha = Wc$. Otherwise, $\alpha = Wd$. Wc and

Wd satisfy the constraint $Wc < Wd$, which implies that more efforts are needed to gain the same amount of trust than to loose it [12]. Wc and Wd are modified when a foul event is triggered by the fact that the coming satisfaction rating is lower than a user-defined threshold. Upon a foul event, the target entity is put under supervision. His Wc is decreased and Wd is increased. If the entity does not conduct any foul event during the supervision period, the Wc and Wd are reset to the initial values. Otherwise, they are further decreased and increased respectively. Current supervision period of an entity increases each time when he conduct a foul event, so that he will be punished longer next time, which means an entity with worse history is treated harsher. The DI-confidence is computed as $1 - \mathit{current\ trust\ belief}$.

DIP algorithm accepts seven input parameters: initial construction factor Wc and destruction factor Wd; initial supervision period p; initial penalty ratios for construction factor, destruction factor and supervision $r1$, $r2$ and $r3$ such that $r1$, $r2 \in (0, 1)$ and $r3 > 1$; foul event threshold $fThreshold$. For each entity k, we maintain a profile P(k) consisting of five fields: current trust value $tValue$, current construction factor Wc, current destruction factor Wd, current supervision period $cPeriod$, rest of supervision period $sRest$.

```
DIP algorithm (Input parameters: Wd, Wc, r1, r2, r3, p,
               fThreshold; Output: DI-confidence)
    Initialize P(k) with input parameters
    while there are new rating R
        if R <= fThreshold then //put under supervision
            P(k).Wd = P(k).Wd + r1 * (1 - P(k).Wd)
            P(k).Wc = r2 * P(k).Wc
            P(k).sRest = P(k).sRest + P(k).cPeriod
            P(k).cPeriod = r3 * P(k).cPeriod
        end if
        if R <= P(k).tValue then    //update tValue
            W = P(k).Wd
        else
            W = P(k).Wc
        end if
        P(k).tValue = P(k).tValue * (1 - W) + R * P(k).W
        if P(k).sRest > 0 and R > fThreshold then
            P(k).sRest =  P(k).sRest - 1
            if P(k).sRest = 0 then  //restore Wc and Wd
                P(k).Wd = Wd and P(k).Wc = Wc
            end if
        end if
        return (1 - P(k).tValue)
    end while
```

Experimental Study DIP's capability of discovering deceiving intentions defined in section 3.1 is investigated through experiments. Initial construction fac-



(a) Discovery uncovered deceiving intention (b) Discovery trapping intention (c) Discovery illusive intention

Fig. 4. Experiments to discovery deceiving intentions

tor is 0.05. Initial destruction factor is 0.1. Penalty ratios for construction factor, destruction factor and supervision-period are 0.9, 0.1 and 2 respectively. The threshold for a foul event is 0.18. The results are shown in fig. 4. The x-axis of each figure is the number of ratings. The y-axis is the DI-confidence.

Swindler with uncovered deceiving intention: The satisfaction rating sequence of the generated swindler is shown in fig. 1a. The result is illustrated in fig. 4a. Since the possibility for the swindler to conduct foul events is high, he is under supervision at most of the time. The construction and destruction factors become close to 0 and 1 respectively because of the punishment for foul events. The trust values are close to the minimum rating of interactions that is 0.1 and DI-confidence is around 0.9.

Swindler with trapping intention: The satisfaction rating sequence of the generated swindler is shown in fig. 1b. As illustrated in fig. 4b, DIP responds to the sharp drop of $f_m(n)$ very quickly. After $f_m(n)$ changes from 0.8 to 0.2, it takes only 6 interactions for DI-confidence increasing from 0.2239 to 0.7592.

Swindler with illusive intention: The satisfaction rating sequence of the generated swindler is shown in fig. 1c. As illustrated in fig. 4c, when the mean function $f_m(n)$ changes from 0.8 to 0.2, DI-confidence increases. When $f_m(n)$ changes back from 0.2 to 0.8, DI-confidence decreases. DIP is able to catch this smart swindler in the sense that his DI-confidence eventually increases to about 0.9. The swindler's effort to cover a fraud with good behaviors has less and less effect with the number of frauds.

5 Conclusions

In this paper, we classify fraudsters as impersonators and swindlers and present a mechanism to detect swindlers. The concepts relevant to frauds conducted by swindlers are formally defined. Uncovered deceiving intention, trapping intention, and illusive intention are identified. We propose an approach for swindler detection, which integrates the ideas of anomaly detection, state transition analysis, and history-based intention prediction. An architecture that realizes this approach is presented. The experiment results show that the proposed deceiving

intention prediction (DIP) algorithm accurately detects the uncovered deceiving intention. Trapping intention is captured promptly in about 6 interactions after a swindler enters the trapping phase. The illusive intention of a swindler, who attempt to cover frauds with good behaviors, can also be caught by DIP.

References

[1] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–255, 2002. 331

[2] P. Burge and J. Shawe-Taylor. Detecting cellular fraud using adaptive prototypes. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management*, 1997. 331

[3] M. Cahill, F. Chen, D. Lambert, J. Pinheiro, and D. Sun. Detecting fraud in the real world. In *Handbook of Massive Datasets*, pages 911–930. Klewer Academic Publishers, 2002. 331

[4] T. Fawcett and F. Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1997. 331

[5] J. Hollmén and V. Tresp. Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In *Proceedings of Advances in Neural Information Processing Systems (NIPS'11)*, 1998. 331

[6] Bertis B. Little, Walter L. Johnston, Ashley C. Lovell, Roderick M. Rejesus, and Steve A. Steed. Collusion in the U. S. crop insurance program: applied data mining. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 594–598. ACM Press, 2002. 331

[7] Saharon Rosset, Uzi Murad, Einat Neumann, Yizhak Idan, and Gadi Pinkas. Discovery of fraud rules for telecommunicationsùchallenges and solutions. In *Proceedings of the fifth ACM SIGKDD*, pages 409–413. ACM Press, 1999. 330, 331

[8] Ravi Sandhu. Lattice-based access control models. *IEEE Computer*, 26(11):9–19, 1993. 331

[9] Salvatore J. Stolfo, Wenke Lee, Philip K. Chan, Wei Fan, and Eleazar Eskin. Data mining-based intrusion detectors: an overview of the columbia IDS project. *ACM SIGMOD Record*, 30(4):5–14, 2001. 332

[10] M. Taniguchi, J. Hollmén M. Haft, and V. Tresp. Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of the IEEE International Conference in Acoustics, Speech and Signal Processing*, 1998. 331

[11] David Wagner and Paolo Soto. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 255–264. ACM Press, 2002. 331

[12] Y. Zhong, Y. Lu, and B. Bhargava. Dynamic trust production based on interaction sequence. Technical Report CSD-TR 03-006, Department of Computer Sciences, Purdue University, 2003. 337