Appeared in AAAI '08

Proactive Intrusion Detection

Benjamin Liebald, Dan Roth, Neelay Shah and **Vivek Srikumar**
Department of Computer Science
University of Illinois at Urbana-Champaign.
benjaminliebald@gmail.com, danr@uiuc.edu, neelay.n.shah@gmail.com, vsrikum2@uiuc.edu

Abstract

Machine learning systems are deployed in many adversarial conditions like intrusion detection, where a classifier has to decide whether a sequence of actions come from a legitimate user or not. However, the attacker, being an adversarial agent, could reverse engineer the classifier and successfully masquerade as a legitimate user. In this paper, we propose the notion of a *Proactive Intrusion Detection System (IDS)* that can counter such attacks by incorporating feedback into the process. A proactive IDS influences the user's actions and observes them in different situations to decide whether the user is an intruder. We present a formal analysis of proactive intrusion detection and extend the adversarial relationship between the IDS and the attacker to present a game theoretic analysis. Finally, we present experimental results on real and synthetic data that confirm the predictions of the analysis.

Introduction

Machine learning systems that use classifiers are deployed to detect malicious activities such as spam, online identity theft and intrusion. The perpetrators of these activities have significant incentive to reverse engineer the classifiers. (Dalvi et al. 2004) studies the problem of learning classifiers in adversarial conditions, whereas (Lowd and Meek 2005) considers the inverse setting and presents efficient algorithms for an attacker to reverse engineer a classifier.

We propose a protocol to address the problem of an adversarial agent that can reverse engineer classifiers, and study it in the domain of intrusion detection. An *Intrusion Detection System* (IDS) performs the vital task of identifying security breaches. Detection of intruders is a machine learning problem, where the task is to recognize illegal users based on patterns of activity. Conventionally, anomaly detection systems aim to solve this problem by learning a model of the legitimate users' normal behavior. Users whose actions deviate from the learned behavior are classified as intruders. These ideas were applied to detect masqueraders in (Schonlau et al. 2001), which compared six methods for masquerader detection. Further improvements were suggested by (Maxion and Townsend 2002), (Yung 2003) and others.

The threat of attackers using reverse engineered classifiers is compounded if they are insiders, who have both the motivation and opportunity to "learn" to masquerade as legitimate users. A recent survey (cf. (Gordon et al. 2006)) shows that insider attacks are indeed a significant threat to corporate intellectual property.

We propose the notion of *Proactive Intrusion Detection* to reliably identify attackers who mislead intrusion detection systems. Consider the following scenario – Alice is an engineer and Bob an accountant in a company. Bob wants to steal engineering plans and manages to get Alice's password. If Bob accesses the system using Alice's password, then the anomaly detection system will detect this as an intrusion because Bob's actions will be different from Alice's. However, if Bob observes Alice's behavior and learns to imitate her, then an IDS that looks for anomalous behavior could be misled. Yet, it is reasonable to assume that even if Bob manages to masquerade as Alice, he may not necessarily be able to handle *uncommon situations* the same way as Alice does. For example, in the event of a database error, Bob's actions need not be similar to the ones of Alice, who, being an engineer, may address the problem differently and perhaps more consistently. This motivates the idea of an IDS that proactively presents uncommon situations to the users and observes their reaction. This allows the IDS to observe users' behavior in other situations than those they "prepared" for and keep the system secure.

The underlying idea behind proactive intrusion detection is to expose users of the system to multiple contexts, in a way that poses no problem to legitimate users but creates difficulties for attackers who learned to use the system in a specific context and revert to their own behavior in other contexts. The different situations that the IDS presents to the user, called *modes* of the IDS, depend on the system that is being protected. One of the modes will be the normal mode of operation. The ability to change modes allows the IDS to influence the user's actions and observe their reactions.

We show that it is possible for an masquerading intruder to circumvent a traditional machine-learning based IDS in a behavioral biometrics domain. However, a proactive IDS can detect the same attacker. We present a formal model for intruder detection which shows that that a proactive IDS can identify intrusions better than a traditional system. In addition, we formalize the adversarial relationship between the attacker and the IDS as a game. This leads to algorithms for a proactive IDS which meet the dual goals of maximizing security with limited disruption to legitimate users. Finally, we present results of experimental verification of the performance these algorithms.

Problem Definition

Consider a interactive system in which user's actions are driven by their eventual goal *and* the feedback from the computer. At each step, a user draws an element from a (possibly changing) distribution over a set of actions. Let $Alice$ be a user of the system and Bob an attacker. We focus on intruders who pretend to be legitimate users. In order to do so, not only does the intruder Bob obtain the login credentials of $Alice$, but also learns an internal model of $Alice$'s actions and uses this to generate actions. This definition of an attacker is smarter than the ones in (Schonlau et al. 2001) and other similar work, where the intruders only know the login credentials, but do not transform their input to match the user's input patterns.

The task of an IDS is to identify whether a window of actions of size $\mathcal{N}$ was performed by the legitimate user or not. Let x_i denote the i^{th} action in the window and x_1^i the sequence $(x_1, x_2, \cdots, x_i)$. Then, the IDS is a classifier which identifies whether the set of actions $x_1^{\mathcal{N}}$ was performed by $Alice$ or not. We limit our discussion to the following probabilistic classifier – If the current user is expected to be $Alice$, then set of actions $x_1^{\mathcal{N}}$ is labeled as an intrusion if, for a user-specific threshold θ, $P(x_1^{\mathcal{N}}|Alice) < \theta$. Many anomaly detection systems in the literature (for example, (Ju and Vardi 2001; DuMouchel 1999)) models the user's actions as a Markov model. With this assumption, a window of actions is labeled as an intrusion if the following holds –

$$\prod_{i=1}^{\mathcal{N}} P(x_i|x_1^{i-1}, Alice) < \theta \qquad (1)$$

The intrusion detection system is completely defined by training a classifier for each legitimate user. While we use a probabilistic definition of the classifier, most results of this paper hold for any classifier that can generate a score for each input. Almost all classifiers can be used this way (refer (Niculescu-Mizil and Caruana 2005)) and this model of an IDS covers the systems defined by existing masquerade detection systems (cf. (Schonlau et al. 2001; Maxion and Townsend 2002; Yung 2003) and others). A common assumption in the literature is that each legitimate user provides uncontaminated data for training.

Proactive Intrusion Detection

$Alice$ draws actions from some probability distribution at every time step. Bob, the $Alice$-masquerader has learned an approximation of that distribution. It is reasonable to assume that, while Bob can masquerade as $Alice$ in *normal* situations, he will not have enough data to learn $Alice$'s behavior in all situations. This assumption is especially true if $Alice$ is an "expert" in her daily usage of the system and knows how to handle uncommon situations. When these uncommon situations arise, it is unlikely that Bob will behave similar to $Alice$, simply because he did not have enough data to learn her behavior. An IDS should take advantage of such distinctive responses of users in specific situations.

This notion forms the basis of **Proactive Intrusion Detection**. In this architecture, the anomaly detector observes a user's interaction with a computer in many situations and learns context specific classifiers for each one of the situations. As in traditional systems, we assume that there is a training phase when uncontaminated data is available for all the situations. After training is complete and the IDS is deployed, it occasionally requires the user to perform actions in these different situations and observes the user's behavior in this context. If the user's behavior is different in any of these situations, then he is declared an intruder.

The different situations in which the IDS observes the user are called *modes* of the IDS. The actual semantics of the modes will depend on the specific system. The key idea is that a user's behavior must be consistent in multiple situations – one of which reflects normal usage of the system. In all other modes, the IDS *simulates* a situation that does not normally occur and observes the user's reactions to it. It must be noted that the IDS need not actually change the internal state of the system and only changes the user's perception of the system. This takes advantage of the fact that the system is an interactive one.

There are two advantages of using a proactive intrusion detection system. First, since the user's behavior is more distinctive in specific situations, their actions will span a smaller space. This makes the task of classification easier. Second, during the design of the IDS, if modes are chosen so that the reaction to them is at an instinctive level, rather than at a conscious level, then learning to impersonate a legitimate user across different modes becomes very difficult, even if the attacker does observe them.

Modes: Example

Consider a transaction processing system in which users access a database. If this system needs to be protected with a proactive IDS, then the different modes could be the following – (i) Normal usage, (ii) Simulate "Deadlock Found", and (iii) Simulate "Integrity constraint violation".

When deployed, the IDS does not affect the system's normal operation. However, it does affect the way the user interacts with the system – the output seen by the user depends on the current mode of the IDS. The most common mode would be the first mode, where the IDS does not alter the output. Occasionally, the IDS will change the mode to one of the other modes based on a mode selection policy that will be discussed later and the system simulates the appropriate error messages. The IDS observes the user's reaction and performs its classification task. If the masquerader has seen only the normal behavior of the users, then the responses of the masquerader to other modes will be different and the masquerade will be detected.

Modeling a Proactive IDS

Our notation can be extended to accommodate the proactive IDS. Let the set of modes of the IDS be denoted by $\mathcal{M} = \{M_1, M_2, \cdots, M_{|\mathcal{M}|}\}$. The mode M_1 will be designated as the normal mode, where the IDS will not transform the output seen by the users. In all other modes, the IDS will transform the output. At this point, we do not specify the explicit transformation that each mode performs.

We extend the definition of the classifier defined in (1) to include the modes. As with the user's actions, let the i^{th} mode within a block be m_i and m_1^i be the sequence $(m_1, m_2, \cdots, m_i)$. The classifier has the same form as defined earlier; it has additional input to make its decision. The classifier is redefined to report an intrusion if –

$$\prod_{i=1}^{\mathcal{N}} P(x_i, m_i | x_1^{i-1}, m_1^{i-1}, Alice) < \theta \tag{2}$$

Since all modes other than M_1 are disruptive, we may wish to quantify and limit the disruption. Let $d(M_i)$ represent a non-negative real number representing the disruption caused the IDS being in mode M_i. These values could be ascertained by surveying the users of the system.

Let the IDS use a policy D to pick its mode. The *mode selection policy* is a probability distribution over the modes. A proactive IDS is completely defined by two independent aspects – the classifier $\mathcal{C}$ and the mode selection policy D.

Bounds on Attacker Performance

We wish to place a bound on the number of iterations N before which the attacker has to be identified. Here, one iteration is defined as a window of actions after which the IDS makes a classification decision. We want to limit the probability that the IDS does not identify an intrusion to δ. For the discussion in this section, we assume that the IDS changes mode at the start of a window. This is a simplification of the general proactive IDS described earlier. The mode is chosen using the mode switching policy D. At the end of each iteration, the IDS classifies the window as legitimate or not. Let the generalization error of the classifier be ϵ.

We assume that the attacker has observed the user's interactions with the system for some time and built a model of the legitimate user's usage. Let γ_m represent the probability that the user's and the attacker's actions in mode m come from different distributions. Let the normal mode M_1 be defined as that mode for which the attacker's performance is the best, i.e., $\gamma_{M_1} \leq \gamma_m$ for every mode m.

Theorem 1. *If a proactive IDS with modes $\mathcal{M}$ has a generalization error ϵ, then with a probability greater than $1-\delta$, an attacker who is γ_m-similar to a legitimate user in mode m will be identified in N iterations, where N is bounded by the following inequality –*

$$N \leq \frac{\log(1/\delta)}{\log\left(\frac{1-\epsilon}{\epsilon}\right)\sum_{m\in\mathcal{M}}\gamma_m D(m) + \log(1-\epsilon)} \overset{Def}{=} N^B \tag{3}$$

Proof Sketch. In mode m, the intruder will not be identified correctly in two situations – (i) The attacker perfectly mimics the real user and hence, the IDS does not identify an intrusion, or (ii) The attacker is imperfect and the IDS makes an error. This means that the probability that the masquerader is not identified in mode m is $(1-\gamma_m)(1-\epsilon) + \gamma_m\epsilon$. The probability that the masquerader is not identified for all modes is the term in the parenthesis in (4). Since the IDS makes its decision after each iteration independently, bounding the probability of error by δ, we get (4).

$$\left(\sum_{m\in\mathcal{M}} D(m)\left[(1-\gamma_m)(1-\epsilon) + \gamma_m\epsilon\right]\right)^N < \delta \tag{4}$$

Applying the arithmetic-geometric inequality twice and rearranging it, we get the required result. □

The inequality (3) says that an IDS with an error ϵ should observe a masquerader who is γ_m-similar to the real user for at least N iterations to guarantee a failure probability less than δ. Using (3), we can show that a proactive IDS can identify masqueraders better than a traditional IDS. A **passive IDS**, that is, a traditional anomaly detector, operates only in the normal mode. In the context of the proactive IDS, it can be seen that the passive IDS is a proactive IDS in which the mode selection policy is fixed to $D(M_1) = 1$ and $D(m) = 0$ for all other m.

Corollary 1. *Consider a proactive and a passive IDS with identical classifier errors ϵ. Assume a γ_m-similar attacker, identified by both the IDSs with confidence $1-\delta$. Then, the following hold – (i) $N^B_{Proactive} \leq N^B_{Passive}$, where $N^B_{Proactive}$ and $N^B_{Passive}$ are defined as in (3), and (ii) There exist mode selection policies for which the inequality in (i) is strict.*

Proof Sketch. For the passive IDS, (3) gives us

$$N_{Passive} \leq \frac{\log(1/\delta)}{\log\left(\frac{1-\epsilon}{\epsilon}\right)\gamma_{M_1} + log(1-\epsilon)} \overset{Def}{=} N^B_{Passive} \tag{5}$$

The attacker's performance is best for mode M_1. That is, γ_{M_1} is the least among all γ_m and hence also less than any weighted average of the γ_m's. That is, for any $D(m)$, $\gamma_{M_1} \leq \sum_{m\in\mathcal{M}} D(m)\gamma_m$. In particular, there exist D where the inequality is strict. (For example, if the weight for $D(0)$ is set to zero, then the inequality can be made strict.)

Since $\epsilon < \frac{1}{2}$, comparing (3) and (5) shows that the upper bound on $N_{Proactive}$ is less than the upper bound on $N_{Passive}$. This, with the observation that there exist mode selection policies where the improvement is achievable, gives the required result. □

Mode Selection

The earlier discussion assumed that we know the accuracy of the attacker's model in each mode. In this section, we examine different policies for the selection of modes without knowledge of the attacker's model. For the analysis in this section, we remove the restriction that the IDS changes its mode only at the beginning of each window.

Passive and Random IDS

A **Passive IDS** is one that always operates in the normal mode. This view of a passive IDS shows that the set of proactive intrusion detection systems is a strict superset of the set of passive intrusion detection systems. Another simple mode selection policy is to pick the next mode randomly from a uniform distribution. In other words, a **Random Proactive IDS** is defined by the following mode selection policy – for every mode m, $D(m) = \frac{1}{|\mathcal{M}|}$.

Game Theoretic Proactive IDS

While the random proactive systems do improve performance , we would like to have better control over the modes. Here, we analyze the IDS based on the worst-case assumption that the IDS and the user have an adversarial relationship. The starting point of this analysis is the inequality (2). The implicit goal of the intruder Bob is to make the classifier of the IDS believe that he is $Alice$. In our current setting, his goal will be to make the left hand side of (2) (called the *score* of the classifier) more than the threshold θ.

The multiplicative form of the score can be interpreted as follows: each term in the product makes an incremental contribution of $P(x_i, m_i | x_1^{i-1}, m_1^{i-1}, Alice)$ to the final score. At the i^{th} step, the goal of Bob is to pick an action x_i such that its incremental contribution is maximized. The assumption of an adversarial relationship between the IDS and the user will imply that the proactive IDS should pick mode m_i such that the mode's incremental contribution to the final score is minimized for any action of Bob's. This gives us the following **Decision Theoretic** policy to pick modes –

$$m_i = arg_M \left(\max_X \min_M P(x_i, m_i | x_1^{i-1}, m_1^{i-1}, Alice) \right) \tag{6}$$

The adversarial notion can be analyzed as a zero sum game between the IDS and the user. In the decision theoretic analysis presented above, the IDS picks a single mode whose incremental contribution to the final score is maximized for all possible actions of the user. Instead of picking a single mode, we can derive a probability distribution over the modes such that the *expected incremental contribution* to the final score is maximized for all possible user actions. If $\mathbf{p} = (p_1, p_2, \cdots, p_{|\mathcal{M}|})$ is a probability distribution over the modes where p_i is the probability that the IDS will pick mode M_i, then we wish the following to hold –

$$\mathbf{p} = \max_X \min_p E_p \left[P(x_i, m_i | x_1^{i-1}, m_1^{i-1}, Alice) \right] \tag{7}$$

where $E_\mathbf{p}$ denotes the expectation over the distribution p. Once we have the optimal $\mathbf{p}$, the next mode is picked by drawing from the distribution. This gives us the **Mixed Strategy IDS** . This representation allows us to use a principled approach for solving the min-max problem. Linear programming is a commonly used technique to solve zero-sum games (see, for example, (Vaserstein and Byrne 2002)). Computing the optimal set of probabilities $\mathbf{p}$ can be done by converting the game to a linear program and solving it.

It must be noted that in this analysis, there is a departure from traditional game theoretic analysis for mixed strategies. Traditionally, it is assumed that both the players make their move simultaneously. Clearly this is not the case here. Yet, it is shown in the experiments that the mixed strategy is among the best performing policies. We believe that this is because, in reality, the user is not really perfectly adversarial. The goal of the user, in addition to thwarting the IDS, is also to perform some activity on the system.

Constrained Proactive IDS

We may want to limit the total disruption caused by the proactive IDS per block to $\mathcal{D}_{Max}$. Extending the random proactive IDS, a **Constrained Random Proactive IDS** picks modes randomly while ensuring that the total disruption for the whole block is not more than $\mathcal{D}_{Max}$. A similar extension to the Mixed Strategy IDS limits the expected disruption to $\mathcal{D}_{Max}$. Adding this constraint will not make the original linear program infeasible because $d(M_1) = 0$ and hence, always picking M_1 will be a feasible solution. Constrained games and their relation to linear programming are discussed in (Charnes 1953). The IDS that uses this constrained game to pick its mode will be referred to as the **Constrained Mixed Strategy Proactive IDS.**

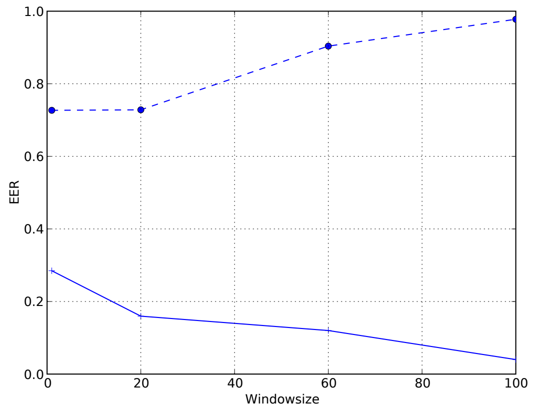
Experiments

We present results of two experiments. First, using real data from users, we demonstrate a successful masquerade attack against a traditional IDS and show the capacity of a proactive IDS to identify it. In the second, we compare the different mode selection policies in terms of intruder detection accuracy and disruption to normal usage.

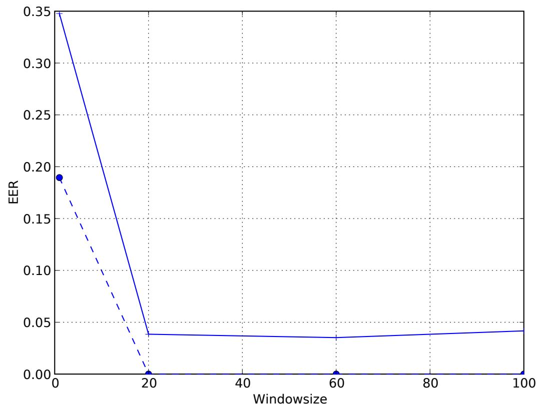
Behavioral Biometrics

Behavioral biometric systems authenticate users based on their usage of input devices. Recent work ((Gamboa and Fred 2003; Pusara and Brodley 2004; Gunetti and Picardi 2005)) has shown the success of systems based on keystroke and mouse movement dynamics. As a baseline, we implemented the system described in (Gamboa and Fred 2003), which distinguishes users based on their mouse movement.

We collected mouse movement data (pointer position and button state) from ten users interacting with three Linux applications and segmented it into *strokes*, which represent the data collected between successive mouse clicks. For each stroke, we used statistics like the mean and standard deviation of the horizontal, vertical and angular velocities as features. (For a complete description of features, see (Gamboa and Fred 2003)). We implemented the passive and random proactive IDS for the users, using 90% of the data for training and the rest for testing. The IDSs were trained using Support Vector Machines with a Gaussian kernel tuned according to the procedure described in (Hsu, Chang, and Lin 2006). During evaluation, we added the raw scores of the classifier over a window of strokes and compared the sum to a threshold to decide if there is an intrusion. We tuned the classifier such that it achieved equal error rate, i.e., the rate of false positives and false negatives were made equal.



(a) **Example of a successful attack**: Error of the passive system for increasing window lengths under masquerade attack using 25% of training data. While the system is able to detect untransformed strokes of other users (solid line) its error is high when detecting strokes transformed by the attacker (dashed line).



(b) **Detection of a smart attacker by a proactive IDS**: Error of the proactive detection system under masquerade attack using 100% of training data. Unlike the passive system, this IDS is able to correctly identify strokes both before (solid line) and after (dashed line) they are transformed by the attacker.

Figure 1: Biometrics comparison of the passive and random proactive IDS. In both cases, the graphs show the variation of the equal error rate (EER) (that is, the error of the system that is tuned such that the number of false positives and false negatives are equal.)

We implemented a masquerade attack in this setting. The goal of the attacker was to move the mouse pointer and execute a click. Our attacker had access to a fraction of the data used to train the system and knew the learning algorithm and features of the IDS. We implemented a masquerader script that used a genetic algorithm to generate a path between two points that would be accepted by the IDS. The fitness function of the genetic algorithm was the classifier A mutation consisted of changing individual points within a stroke, while a crossover split two strokes into two parts, and interchanged them to get two new solutions. After each generation, the ten highest scoring solutions were retained.

To implement the random proactive IDS, we chose the following six modes – (i) Normal usage, (ii) Dropping short mouse sequences, (iii) Introducing delay, (iv) Introducing random jitter, (v) Slowing down the pointer, and (vi) Speeding up the pointer. We trained a classifier for each user for every mode using data obtained from six users. In this case, the attacker learns the "average" behavior of each user.

We show the results of our evaluation in Fig. 1(a) and 1(b), which show the performance of the passive and proactive IDS respectively. We compare the performance of the systems for simple masquerade attacks (i.e., attackers who do not transform the strokes of the target user) and smart masqueraders (i.e., attackers who learn to transform the strokes of the target user). We can see that while the passive IDS can identify attackers who do not masquerade, masqueraders are not identified for any window lengths. On the other hand, the proactive IDS can identify both masquerading and non-masquerading attackers. The fact that we can identify attackers in the biometrics experiments with a proactive IDS validates the assumption that users behave consistently even in unexpected situations. For attackers who do not transform their input, even at a window size of 20, the proactive IDS achieves the best performance of the passive IDS.

Comparison of Mode Selection Policies

We compared the different mode selection policies with respect to their ability to spot intruders and the disruption they cause to normal activity. The most effective IDS is one that can identify intruders with minimum disruption. For this set of experiments, we used synthetically generated data. Our experimental setup is similar to that of (Schonlau et al. 2001), which showed that the best model for users is a Markov model. We modeled twenty users' actions with random Markov chains. We defined six modes, without explicitly defining their semantics because the algorithms do not depend on the semantics of the mode. As earlier, we designated mode M_1 as the normal mode with zero disruption and assigned unit disruptions for five other modes. For each user, we defined as session as 50 non-overlapping blocks, with 100 actions in each block.

Masqueraders were inserted into a user's session similar to (Schonlau et al. 2001): If the previous block was not a masquerade, then the current block is not a masquerade with probability p_u. Otherwise, the current block is a masquerade with probability p_m. In our experiments, we set the values of both p_m and p_u to 0.8. When exceptional situations are encountered, a masquerader reverts to his own behavior. This is captured by having the masquerader learn the behavior of the user in normal mode using the IDS's training data and using random Markov chains for all other modes.

We used a Markov model as the IDS's internal model of the users. For each user, during training, the IDS computes

IDS Type	**Average Error**	**Effectiveness $\times 10^{-4}$**
Passive	0.427	–
Random Proactive	0.019	2.360
Decision Theoretic	0.049	2.256
Mixed	0.027	2.334
Constrained Mixed	0.107	3.614
Constrained Random	0.303	2.788

Table 1: **Evaluation of different mode selection policies.** All the proactive policies have a lower error than the passive IDS. Random selection of modes is the best in terms of average error followed by the mixed strategy IDS. However, when we constrain the disruption to 2500 we see that the constrained mixed strategy IDS outperforms the constrained random IDS. Effectiveness, which captures the need for high accuracy *and* low disruption, shows that the most effective IDS is the constrained mixed strategy IDS.

the probability of an action given the mode and the previous action. The classification was done per-block by computing the probability of the actions in the block given the modes. We tuned the threshold to get equal error rate in the passive IDS. For each IDS, we measured the error and the total disruption for the entire user session for each IDS. For each proactive IDS, we defined a metric called *Effectiveness* that captures the need for high accuracy and low disruption. The effectiveness of a proactive IDS is defined as the ratio of the accuracy of the IDS to the total disruption.

The errors in masquerade detection the effectiveness for each mode selection policy are shown in Table 1. All the proactive IDSs perform better than the passive IDS in terms of average error and the best performing proactive systems are the random proactive IDS and the mixed strategy proactive IDS. However, in terms of the effectiveness, we see that the constrained mixed strategy IDS performs best. This gives us a parameter to trade off disruption and security.

Conclusions

We introduced the novel idea of proactive intrusion detection to detect smart attackers. The key intuition is that while average behavior could be masqueraded, exceptional situations force people to revert to their own behavior and this idea can be used to detect intrusions. The IDS, instead of just observing the user's actions, participates in influencing them with its own set of actions called modes, thus broadening its capability. These ideas are presented without special emphasis on any specific domain or specific classifiers and can be used with different classifiers in diverse domains. We analyzed the interaction of the user and the IDS as a game and developed algorithms for proactive intrusion detection systems. We experimentally and analytically showed that proactive intrusion detection does detect masqueraders and we presented an application in the domain of behavioral biometrics. From a broader perspective, our paradigm of employing a proactive agent can be applied to any interactive environment where there is an incentive for an adversarial agent to reverse engineer a machine learning based system.

Acknowledgements

This work was supported by a research grant from Boeing.

References

Charnes, A. 1953. Constrained games and linear programming. *Proceedings of the National Academy of Sciences of the United States of America* 39(7):639–641.

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 99–108. ACM Press.

DuMouchel, W. 1999. Computer intrusion detection based on bayes factors for comparing command transition probabilities. Technical Report TR91, National Institute of Statistical Sciences (NISS).

Gamboa, H., and Fred, A. 2003. An identity authentication system based on human computer interaction behaviour. In *Pattern Recognition in Information Systems*, 46–55. ICEIS Press.

Gordon, L. A.; Loeb, M. P.; Lucyshyn, W.; and Richardson, R. 2006. CSI/FBI computer crime and security survey. Computer Security Institute Publications.

Gunetti, D., and Picardi, C. 2005. Keystroke analysis of free text. *ACM Transactions on Information and System Security* 8(3):312–347.

Hsu, C.; Chang, C.; and Lin, C. 2006. A practical guide to support vector classification. Technical report, National Taiwan University.

Ju, W.-H., and Vardi, Y. 2001. A hybrid high-order Markov chain model for computer intrusion detection. *Journal of Computational and Graphical Statistics* 10(2):277–295.

Lowd, D., and Meek, C. 2005. Adversarial learning. In *Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 641–647. ACM Press.

Maxion, R., and Townsend, T. 2002. Masquerade detection using truncated command lines. In *Proceeedings of the International Conference on Dependable Systems and Networks*, 219–228. IEEE Computer Society Press.

Niculescu-Mizil, A., and Caruana, R. 2005. Predicting good probabilities with supervised learning. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, 625–632. ACM Press.

Pusara, M., and Brodley, C. 2004. User re-authentication via mouse movements. In *ACM workshop on Visualization and Data Mining for Computer Security*, 1–8. ACM Press.

Schonlau, M.; DuMouchel, W.; Ju, W.-H.; Karr, A. F.; Theus, M.; and Vardi, Y. 2001. Computer intrusion: Detecting masquerades. *Statistical Science* 16(1):58–74.

Vaserstein, L., and Byrne, C. 2002. *Introduction to Linear Programming*. Pearson Education, Inc.

Yung, K. H. 2003. Using feedback to improve masquerade detection. In *Applied Cryptography and Network Security*, 48–62. Springer.