Modeling Malicious Network Packets with Generative Probabilistic Graphical Models

Ashe Magalhaes
ashemag@cs.stanford.edu

Gene Lewis
glewis17@cs.stanford.edu

***Abstract*— Cyber enterprise systems often are difficult to protect due to a large number of sub-components that must work in concert to remain resilient. In cyber enterprises where incoming traffic may approach a few megabits per second, an IDS and host system controlled by a Markov Decision Process may serve as an efficient resiliency solution. However, the structure of this model leverages very little information about the adversary. For example, attack signatures of well known attacks and the behavior of previous packets are not considered when the system decides if a network packet is malicious or normal.**

In this paper, we attempt a first step to augmenting such a resiliency system by learning about adversary behavior through modeling malicious packet data with a probabilistic graphical model. We examine the effects of weakening the Markov assumption on the behavior of an adversary, and investigate how well this Markov adversary model is borne out in real data for four different cyber attack types. Finally, we investigate how well our model captures intrinsic characteristics of malicious behavior by using log-likelihood scores of various attack models to train a discriminative classifier; we find that our classifier is able to attain anywhere from 93% to 98% classification accuracy, a strong indicator that our generative models have successfully captured the distribution of features and behaviors that comprise a malicious adversary vs. a benign one.

I. INTRODUCTION

Technological advances such as high-speed backbones, local area networks, and wireless technology have created a dynamic network of systems which have become mission critical for governments, companies, and institutions [1]. In recent years, a number of high profile attacks on cyber infrastructure have inspired a considerable amount of research into enhancing traditional protection mechanisms [2]. To reduce dependency on security experts, projects have used data mining and machine learning techniques to obtain the automatically learn and respond to common attack signatures [3]. This paper focuses on the problem of protection for a cyber enterprise, which consists of an elaborate web of applications, software, storage, and networking hardware. These systems have difficulty keeping up with incoming network traffic which excludes a few megabits per second.

A. Intrusion Detection Systems

An Intrusion detection system (IDS) is often deployed as a primary component for enabling security and resiliency within industrial control systems. Its objective is to detect ongoing intrusive activities in computer systems and networks. An IDS searches for evidence of malicious behavior by analyzing one or more event streams. Events may be represented by network packets, operating system calls, audit records produced by the operating system auditing facilities, or log messages produced by applications [4]. When an attack is detected, the IDS produces an alert that describes the type of attack. A false positive warning occurs when normal network behavior is labeled as an attack. A false negative warning occurs when malicious network behavior is not detected. Consequences of false positives include reduced system availability and a subsequent disregard of IDS warnings. Consequences of false negatives include reduced trust in the IDS and damages caused by the attacker [1].

B. Cyber enterprises as a Markov Decision Process

The problem of maintaining a secure and reliable enterprise can be addressed with a Markov Decision Process (MDP) model of a host and IDS system [5]. Resilient cyber infrastructure can be defined as the ability of a system to continue to function (though possibly in a degraded manner) in the face of behaviors that affect the proper operation of some of its components [6]. Modeling the system as a MDP allows for the incorporation of rigorous definitions for state awareness and operational resiliency in our modeling of real-time control systems.

The evaluation of such MDP systems requires examining the expected utility frontier for policies obtained from varying the model parameters. However, varying all model parameters over all possible values does not accurately reflect the behavior of the MDP system in handling real-world malicious network packets.

In order to augment the analysis of the MDP controller, we can leverage information about the adversary through generative probabilistic graphical models. This achieves the dual purposes of generating data to better categorize attack types as well as incorporating the generative models into the MDP system to boost resiliency.

C. Related Work

While controller-based autonomous systems has been implemented widely for applications in defense [7], sensor networks [8], and power management [9], to name a few, this paper builds on a novel approach to modeling an IDS and host as an MDP controller [5]. The connection between anomaly detection and probabilistic graphical modeling has been shown to provide robust operational solutions [10][11].

The contribution of this paper to the existing literature is to provide further exploration of the insight generative probabilistic graphical models yield into the behavior of adversaries in the context of cybersecurity. Our work diverges

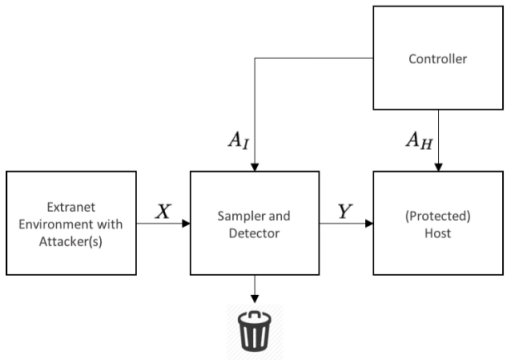

Fig. 1. Example of MDP Controller. Controller has actions A_I and A_H. Action A_I encodes the action space of the IDS (i.e. whether to pass or drop a packet). Action A_H encodes the action space of the host (i.e. whether to wait for a packet or reset) [5].

from those cited in its reliance on supervised learning as a means of modeling types of attacks.

II. MODELS

A. Packet Model

We collect a total of 40 different features for each packet, ranging from duration to protocol type to the number of files accessed. We model each continuous variable as a Gaussian distribution and model each discrete variable as a Multinomial distribution. Furthermore, we model the Bayesian Network of packet features with a Naive Bayes Model; given the observed state of the adversary, the distributions for each of the packet features are independent of each other. Our packet generation procedure creates a new packet by sampling each of the 40 packet features from their respective distributions, conditional on if the adversary choose to act normally or maliciously. To fit the parameters for each distribution, we calculate the maximum likelihood estimates for each parameter.

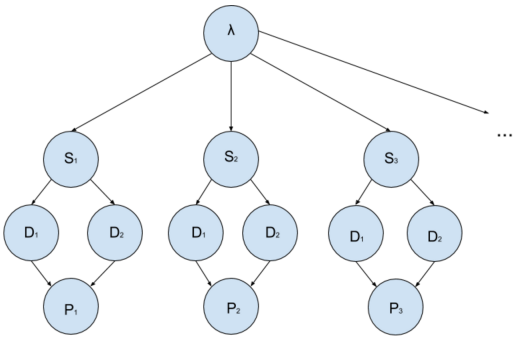

Fig. 2. Naive Bayes Network Packet Model

B. Adversary Naive Bayes Model

As a baseline approach, we model the adversary as a Naive Bayes Model; given the probability of an adversary acting maliciously, the observed states that of the adversary are independent. A packet is then generated from the corresponding distribution, as described above. To estimate the probability of an adversary acting maliciously, we calculate the maximum likelihood estimate.

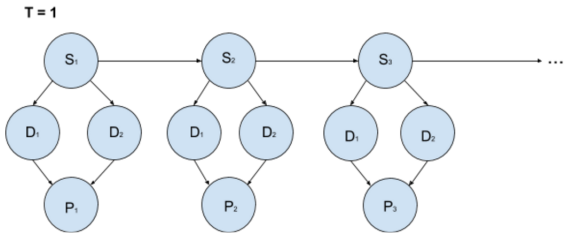

Fig. 3. Naive Bayes Adversary Model

C. Adversary Markov Model

As an improvement on our baseline approach, we break the Naive Bayes Assumption to capture more complex stochastic dynamics of adversary actions over time. In this model, each observed state of the adversary is dependent on past T observed states, where T is the number of previous time steps that influence the current state. For example, $T = 1$ gives the standard Markov Assumption that the current state is only dependent on the previous state. The transition probability for the adversary is given by a T-tensor, where each dimension has size 2 and each entry gives the probability of transitioning to the next state given the current and previous T states. To estimate the entries of the probability tensor we calculate the maximum likelihood estimates.

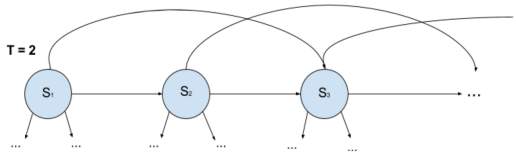

Fig. 4. Markov Adversary Model

III. EXPERIMENTS

A. Dataset

The maximum likelihood estimation of our model parameters relies on the NSL-KDD dataset [12]. Due to the confidential nature of network attacks, there are few publicly available data sets for network-based anomaly detection systems [5]. Since 1999, KDD CUP has been the most widely used dataset for the evaluation of anomaly detection methods [13]. Researchers Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani created a data set, NSL-KDD, to address the shortcomings of KDD. Specifically, the advantages of NSL-KDD over KDD include the following:

- It does not include redundant records in the train set so the classifiers will not be biased towards more frequent records
- The performance of the learners are not biased by the methods which have better detection rates on the frequent records
- The number of records in the train and test sets are reasonable, which allows for experiments on the complete set. Consequently, evaluation results by different research works will be comparable [13].

Consequently, NSL-KDD has been carefully constructed to be representative of existing real networks.

We group our attacks into four types [14].

1) Denial of Service Attacks (DoS)
   A DoS attack is a type of attack in which the adversary makes a computing resource too busy to serve legitimate networking requests. This denies users access to a machine. Examples of DoS attacks within the dataset include local area network denial (land), Neptune, ping of death (pod), smurf, and teardrop attacks.

2) Probe Attacks
   A probe attack is a type of attack in which the adversary scans a machine or a networking device in order to determine vulnerabilities that may later be exploited. This technique is commonly used in data mining. Examples of probe attacks within the dataset include portsweep, nmap, ipsweep, and satan.

3) Remote To User Attacks (R2L)
   A remote to user attack is a type of attack in which the adversary sends packets to a machine over the internet which s/he does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer. Examples of R2L attacks in the dataset include imap, spy, phf, multihop, and guessing the password.

4) User to Root Attacks (U2R)
   A user to root attack is a type of attack in which the adversary starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges. Examples of U2R attacks in the dataset include perl, buffer overflow, and rootkit. [15]

B. Training

Prior to training, each packet was converted to feature space and processed so that all discrete-valued features instead take on integers. We next implement each of our attack types (DoS, Probe, R2L, U2R) as a Markov Model and learn the model parameters via maximum likelihood estimation as described above. In order to analyze the dynamics of the past T adversarial states, we fit each of the above models

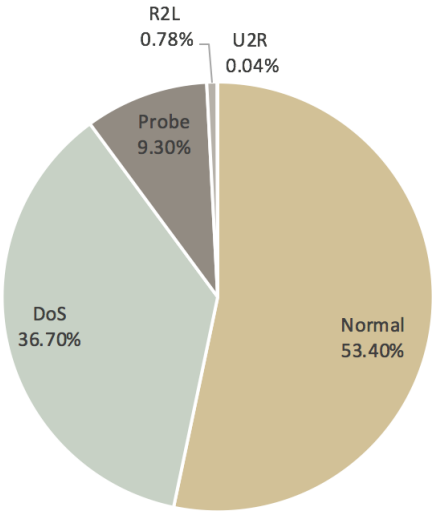

Fig. 5. Network Packet Types in NSL-KDD Dataset

for $1 \leq T \leq 10$. We then calculate the log likelihood of the data given our learned model and use this as an evaluation metric for the appropriateness of fit. Below we have plotted the log likelihood scores for each type of attack normalized against the maximum log likelihood achieved by that attack model; this allows us to examine the effect of T on each model relative to each other.

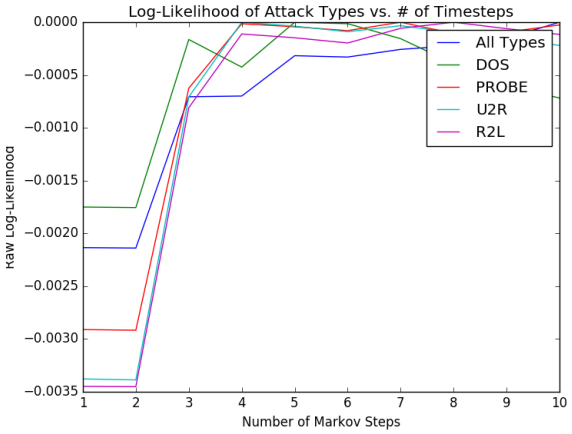

Fig. 6. Normalized Log Likelihood of Attack Types vs. Number of Timesteps

C. Testing

To examine the appropriateness of fit of the attack models from a different perspective, we fit a discriminative classifier using the log-likelihood responses of our attack models and then measure the classifier's ability to discriminate between malicious and non-malicious packets; the intuition behind this procedure is that an attack model should score a malicious data-packet of the corresponding attack type as more

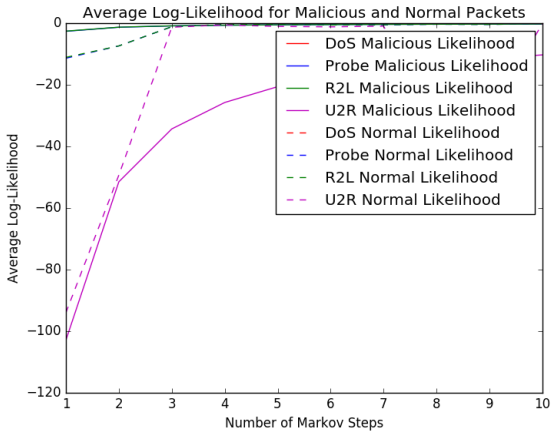

Fig. 7. Log Likelihood for Malicious and Normal Packets for different Attack Types vs. Number of Timesteps

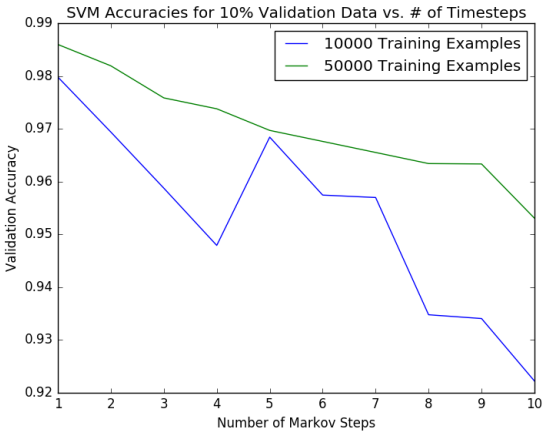

Fig. 9. SVM Validation Accuracy for Predicting Malicious Data

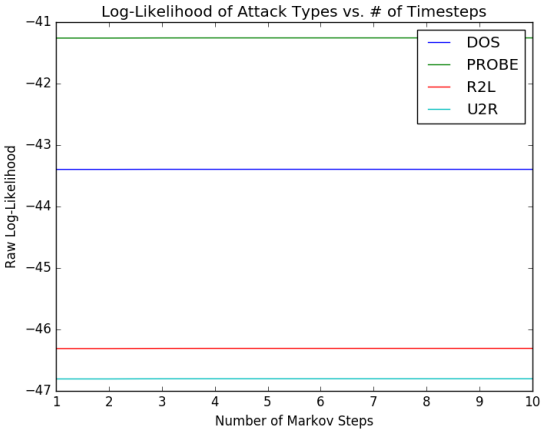

Fig. 8. Raw Log Likelihood of Attack Types vs. Number of Timesteps

likely than a normal data-packet or a malicious data-packet of a different attack type (e.g. a packet with the signature of a DoS attack should be scored highly by the DoS model and low by the Probe, U2R, and R2L models).

We first trained Markov models for each of the four attack types for varying degrees of Markov steps from $T = 1$ to $T = 10$. We then converted two sets of 10000 and 50000 data points into vectors $x_i \in \mathbb{R}^4$, where each feature in a given x_i is the log-likelihood score of the data point from one of the four different attack models. We then trained an SVM classifier with an radial basis function kernel [19] on these four dimensional vectors and made predictions on validation data of size 10% of the training data. Our validation accuracies are given in the figure below.

IV. ANALYSIS

We note that during training most models achieve a maximum log likelihood score with $T = 4$ or $T = 5$, with average likelihood score decreasing for larger T; this implies that most packet signatures are primarily influenced by the previous 4 or 5 packets, where modeling probabilities for longer timesteps leads to a model with too few independence assumptions that overfits the data. This result is intuitive, as we would expect certain time-dependent attack types such as Denial of Service to be better modeled and predicted by a model that takes into account extended time dynamics. We also note that the true log likelihood score of each attack model is proportional to its representation in the data set, and so our model for a given attack type is only as robust as this proportion.

During testing, we noted that our SVM classifier trained on log-likelihood scores performed surprisingly well, correctly classifying anywhere from 93% to 99% of 1000 - 5000 validation examples. We note a particularly surprising result, in which our classification accuracy decreases as the number of Markov steps our attack models were trained with increases. This result at first seems to contrast with our training results, in which our models attain a maximum log-likelihood of the data around $T = 5$; however, as Figure 7 shows, we find that the difference in average log-likelihood response of our models to malicious packets is at a maximum at $T = 1$ and decreases as T moves to 10. Stated differently, the log-likelihood responses of each model to malicious vs. normal data begins to converge as the number of previous timesteps considered increases. Thus, our likelihood scores contain the most discriminative power at $T = 1$, which coincides with our classification accuracy results.

V. CONCLUSION

In this work, we have constructed a generative probabilistic Markov model of four different attack types, fitting the parameters of the corresponding distributions from the data. We found that changing the time dependence of the Markov model had a strong influence of the appropriateness of fit of the model, in general reaching a maximum when considering the previous five timesteps. We also showed that the average log-likelihood response of our models between positive and negative examples generally converges, suggesting the discriminative power of the models drops as we consider more timesteps. Finally, we also showed that fitting

a discriminative classifier with the collective log-likelihood scores of our models generally achieves very good accuracy and supports the claim that our models do indeed capture much of the intrinsic structure differentiating an adversarial model from a malicious one, along with the behavior of such an adversary in choosing states. Future includes investigating how we can further leverage these models by incorporating them into an MDP resiliency system or intrusion detection system as discussed in the introduction.

ACKNOWLEDGMENTS

The authors thank Professor Mykel Kochenderfer, Vineet Mehta, and Paul Rowe for their consistent support and inspiration. The authors also thank Dr. Arash Habibi Lashkari from the ISCX Research Center, UNB, Canada for sharing the NSL-KDD dataset.

REFERENCES

[1] Valeur, Fredrik, et al. "A Comprehensive approach to intrusion detection alert correlation."Dependable and Secure Computing, IEEE Transactions on1.3 (2004): 146-169.

[2] Melin, Alexander M., et al. "A mathematical framework for the analysis of cyber-resilient control systems."Resilient Control Systems (ISRCS), 2013 6th International Symposium on. IEEE, 2013.

[3] Yu, Zhenwei, Jeffrey JP Tsai, and Thomas Weigert. "An adaptive automatically tuning intrusion detection system."ACM Transactions on Autonomous and Adaptive Systems (TAAS)3.3 (2008): 10.

[4] Gowadia, Vaibhav, Csilla Farkas, and Marco Valtorta. "Paid: A probabilistic agent-based intrusion detection system."Computers & Security24.7 (2005): 529-545.

[5] Vineet Mehta, Paul Rowe, Mykel Kochenderfer, Cost Optimal Cyber Resilience Analysis For an Intrusion Detection System, unpublished.

[6] Choudhury, Sutanay, et al. "Action Recommendation for Cyber Resilience."Proceedings of the 2015 Workshop on Automated Decision Making for Active Cyber Defense. ACM, 2015.

[7] Armstrong, Derek, et al. "A controller-based autonomic defense system." DARPA Information Survivability Conference and Exposition, 2003. Proceedings. Vol. 2. IEEE, 2003.

[8] Panangadan, Anand, Syed Muhammad Ali, and Ashit Talukder. "Markov decision processes for control of a sensor network-based health monitoring system." Proceedings of the National Conference on Artificial Intelligence. Vol. 20. No. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[9] Qiu, Qinru, and Massoud Pedram. "Dynamic power management based on continuous-time Markov decision processes." Proceedings of the 36th annual ACM/IEEE Design Automation Conference. ACM, 1999.

[10] Ferragut, Erik M., et al. "Automatic construction of anomaly detectors from graphical models." Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on. IEEE, 2011.

[11] McCusker, Owen, et al. "A combined discriminative and generative behavior model for cyber physical system defense." Resilient Control Systems (ISRCS), 2013 6th International Symposium on. IEEE, 2013.

[12] NSL-KDD Dataset: http://www.unb.ca/research/iscx

[13] Tavallaee, Mahbod, et al. "A Detailed Analysis of the KDD CUP 99 Data Set."Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009. 2009.

[14] KDD Dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[15] Paliwal, Swati, and Ravindra Gupta. "Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm." International Journal of Computer Applications 60.19 (2012).

[16] F. Aleserhani et al. Evaluating Intrusion Detection Systems in High Speed Networks. In: Fifth International Conference on Information Assurance and Security. 2009.

[17]] Dacier M. Design of an intrusion-tolerant intrusion detection system. Tech. Rep. D10. IBM Zurich Research Laboratory; 2002.

[18] Kochenderfer, Mykel J., Christopher Amato, and Hayley J. Davison Reynolds.Decision making under uncertainty: theory and application. MIT press, 2015.

[19] Park, Jooyoung, and Irwin W. Sandberg. "Universal approximation using radial-basis-function networks." Neural computation 3.2 (1991): 246-257.