HMMs for Optimal Detection of Cybernet Attacks

Justin Grana
David Wolpert
Joshua Neil
Dongping Xie
Tanmoy Bhattacharya

SFI WORKING PAPER: 2014-06-022

SANTA FE INSTITUTE

HMMs for Optimal Detection of Cybernet Attacks

Justin Grana
Economics Department
American University
Washington, DC
jg3705a@student.american.edu

David Wolpert
Santa Fe Institute
1399 Hyde Park Rd.
Santa Fe, NM 87501
davidwolpert.weebly.com

Joshua Neil
Los Alamos National Laboratory
PO Box 1663 MS B264
Los Alamos, NM
jneil@lanl.gov

Dongping Xie
Economics Department
American University
Washington, DC
xdp668@gmail.com

Tanmoy Bhattacharya
Santa Fe Institute
1399 Hyde Park Rd.
Santa Fe, NM 87501
tanmoy@santafe.edu

Russell Bent
Los Alamos National Laboratory
PO Box 1663 MS B264
Los Alamos, NM
rbent@lanl.gov

June 20, 2014

Abstract

The rapid detection of attackers within firewalls of computer networks is of paramount importance. Anomaly detectors address this problem by quantifying deviations from baseline statistical models of normal network behavior. However anomaly detectors have many false positives, severely limiting their practical utility. To circumvent this problem we need to evaluate both the likelihood of observed network behavior given that no attacker is present (as in anomaly detectors) and the likelihood given that an attacker *is* present. Any realistic stochastic model for behavior of a compromised network must work in continuous time, with many latent variables. Here we develop such a stochastic model of a compromised network's behavior, and show how to use Monte Carlo methods to integrate over its latent variables. This allows us to evaluate the likelihood of observed behavior in a compromised network. We then present computer experiments showing that a likelihood ratio detector that combines our attacker model with a model of normal network behavior has far better ROC curves than an anomaly detector that only uses the model of normal network behavior.

1 Introduction

Many existing systems for detecting intrusions into cybernetworks monitor data streams only at the perimeter of the network. There is no examination of the behavior of communicating computers within the network that might reveal the penetration of the firewall once it has occurred. In addition, perimeter-focused tools are typically based upon matching previously known attack signatures to current data, and so are unable to detect attacks that avoid the current database of signatures [3, 15]. The extremely fast development time of new attack vectors, resulting in "zero-hour exploits", makes such matching increasingly problematic. For these two reasons, the rapid detection of attackers *within* network perimeters, without reliance on signature matching, is of paramount importance.

Machine learning provides some of the most promising approaches to such detection within the network. An example is anomaly detectors, which quantify deviations from baseline statistical models of normal network behavior when the network has not been penetrated [14, 9]. However in practice, many reported anomalies end up being false, reflecting behavior that is unusual but benign, severely limiting the usefulness of anomaly detectors. The underlying problem is that anomaly detectors do not exploit any model for the alternative to normal network behavior, i.e. for the behavior of the network behavior once it has been penetrated. Since our goal is to distinguish benign behavior from behavior indicative of an attack, we should be able to achieve far better performance if we could evaluate the likelihood of a given set of data under models for both types of behavior.

The challenge is how to model behavior of a network that has been penetrated without pre-supposing attacker methods, since these methods evolve so rapidly. To see how this might be done, consider the movement of an attacker through a network. Often in order to traverse the network the attacker will steal administrator credentials [12], using techniques such as pass-the-hash [10]. However *no matter what attack method they use*, typically they will conduct reconnaissance to guide their movement, perhaps to insert malware, or perhaps to collect increasingly valuable data for later exfiltration.

This means that there is a definite sequence in the movement of the attacker across the net, from computers with low value (for any of the goals of inserting malware, extracting data, or stealing credentials) to computers with higher value. This will be true *no matter what precise methods the attacker uses* to achieve that movement. Moreover, it will leave a trace of increasing network traffic going from low value computers to progressively higher value ones. Accordingly, this trace of the attacker's movement within the net — an inherently global property of the data traffic — can be used as the basis of a model of network behavior once it has been penetrated.

Since this trace has a definite time-ordering, we must model with a Markov process, not an IID process. However evaluating the likelihood of a given dataset of observed network traffic under such a statistical model of an attack is challenging. One of the main issues is that traffic occurs so rapidly that an accurate model must treat time as a continuous variable. Another is that traffic monitoring equipment does not detect the most important variables governing the traffic, e.g., the infection states of the computers in the network at a given time. The challenge then is to evaluate the likelihood of observed traffic under two hidden Markov process models, one for the case of no attacker on the network, and one for the case where there is an attacker, but their locations in the network at any given moment are unknown.

In this paper we show how one can do this, using Monte Carlo techniques to approximate the relevant integrals. We then present computer experiments on toy scenarios that show that a likelihood ratio detector which combines our attacker model with a model of normal network behavior has far better ROC curves than an anomaly detector that only uses the model of normal network behavior.

In Section 2, a brief discussion of past approaches to model-based anomaly detection in computer networks is given, using the Generalized Likelihood Ratio Test (GLRT) to produce anomaly scores. We follow that in Section 3 with a detailed exposition of our Markov process model for traffic between pairs of computers, both for a network not under attack and for one that is. Section 4 discusses the Monte Carlo integral estimates required to evaluate the associated likelihoods for any given dataset of traffic patterns. Section 5 then presents experimental results showing remarkable improvement in detection performance when attack models are taken into account. Finally, we discuss future directions and conclude in Section 6.

2 Background

Model-based anomaly detection proceeds by first estimating the parameters of a null model for expected behavior. We denote these historical estimates as $\hat{\theta}$. Next, given a data set $\mathbf{X}$ under question, the likelihood of the parameters given that data can be evaluated: $\mathcal{L}(\hat{\theta} \mid \mathbf{X})$. One can test whether a more likely alternative parameterization is present given $\mathbf{X}$, by calculating the GLRT:

$$\lambda = \frac{\mathcal{L}(\hat{\theta} \mid \mathbf{X})}{\sup_{\theta \in \mathbf{\Theta}} \mathcal{L}(\theta \mid \mathbf{X})}$$

where $\mathbf{\Theta}$ is an alternative parameter space.

Typically, we choose what data $\mathbf{X}$ to collect to facilitate statistical discovery of security breaches. The associated likelihood model may involve a graph connecting computers (nodes) with edges representing time-series of traffic. Since attacks typically cover multiple nodes and edges, subgraphs can be used to group data from multiple nodes and edges into $\mathbf{X}$ for increased detection power. Graph based methods include [1, 5, 6, 14, 17]. However, in no work identified is the stochastic behavior of attackers as they traverse the network collecting reward discussed.

To include this behavior, we will introduce an attack model. If an attacker is present and behaving according to an alternative parameterization, $\theta_{\mathbf{A}}$, then the uniformly most powerful test for rejecting the null hypothesis that {no attacker is present} is the test where $\theta_{\mathbf{A}}$ is used in the denominator:

$$\tilde{\lambda} = \frac{\mathcal{L}(\hat{\theta} \mid \mathbf{X})}{\mathcal{L}(\theta_{\mathbf{A}} \mid \mathbf{X})}.$$

We will use this fact to design optimal attack detectors.

A good survey of general anomaly detection is provided by [2]. Specifically for cyber security applications, machine learning and statistical approaches have shown promise in detecting malicious behavior in computer networks. The first robust statistical approach to this was done by Lee and Stolfo [13], and a good survey of the modern literature is given in [7]. The underlying problem of using partial observations to estimate the parameters of a system undergoing stochastic dynamics is also well studied, see [11] for a review.

3 Model

We model a cybernet as a directed graph, potentially with cycles, where each node represents either a computer or a human, either inside the firewall or outside it. Each node has an associated state. Examples of human nodes are users, system administrators, and hackers, whose states can represent their knowledge, their strategies, etc. Each directed edge represents a potential communication directly connecting one node (human or computer) to another node (human or computer). These edges have associated states, which represent communication messages. So the cybernet evolves according to a Markov process across all possible joint states of every node and every edge. (See Supplementary Material for a review of Markov processes over discrete state spaces.)

In this initial project, we only consider computer nodes, treating the human using a particular computer as part of that computer. We also only consider those computers that are inside the firewall. Each node can be in one of two states, “normal” or “infected”. Similarly, each edge can be in one of two states, “no message”, or “message in transit”. When a node is in a normal state, it sends benign messages along any of its directed edges according to an underlying Poisson process with a pre-specified rate. When a node is infected, it still sends benign messages at the same rate as when it is not infected, but now it superimposes malicious messages. These are generated according to another Poisson process, with a much lower rate.

For simplicity we assume that if an edge from an infected node to a non-infected node gains a new malicious message at time t, then with probability 1.0 the second node becomes infected and the new malicious message disappears immediately, leaving a trace on our net-monitoring equipment that that message traveled down that edge at t. (Formally, we model by this by having the Markov rate constants for message absorptions all be much larger than the rate constants for message emissions.) No node can become infected spontaneously, and no node can become uninfected.

3.1 Definitions

Let $G = (V, E)$ be the directed graph of a cybernet where $V = \{v_1, v_2 ... v_N\}$ is the set of nodes. Use 1 to represent the normal state of a given node and 0 to represent the infected state. Let $\sigma \in \mathbb{B}^N$ denote the state of all nodes in the network and σ_{v_i} denote the state of node v_i. The Markov process governing the cybernet is parameterized by the set $\lambda \equiv \{(\lambda_{v,v',\sigma_v}) : v, v' \in V, v; \neq v, \sigma_v \in \mathbb{B}\}$ giving the total rates at which v sends messages to v' when v is in state σ_v. (The far larger rate constants for message absorption are irrelevant to our analysis.) We write the rate parameter for just emission of malicious message from v to v' as $\Delta_{v,v'} \equiv \lambda_{v,v',0} - \lambda_{v,v',1}$. For simplicity, in this paper we take λ fixed and greater than zero — in a full analysis we would average over it according to a prior.

Suppose we observe the traffic on a net for a time interval $[0, T]$, resulting in a dataset $D = \{(\tau_i, v_i, v'_i)\}$, where each $\tau_i \in [0, T]$ and each $(v_i, v'_i) \in V^2$. We interpret any $(\tau, v, v') \in D$ as the observation that a message was added at time τ to the edge from v to v'. We assume that the observation process is noise-free, i.e., that all messages are recorded and no spurious messages are.

For all $1 \leq k \leq N$, define $\mathbb{S}^k$ as the set of vectors $s \in V^{|k|}$ such that for all $i, j \neq i$, $s_i \neq s_j$. Define $\mathbb{S} = \cup_{k=1}^{N} \mathbb{S}^k$. Below we will interpret any $s \in \mathbb{S}$ as a time-ordered sequence of all node infections that occur in $[0, T]$ (though others might occur later). Also define the space $Z \equiv [0, T] \cup \{*\}$ and write elements of any associated space Z^m as $z = (z_{v_1}, z_{v_2}, ... z_{v_m})$, i.e., index elements of z by the elements of V. Below we will interpret any component $z_v = *$ to mean that node v does not get infected during $[0, T]$ (though it might get infected later), and every real-valued $z_v \leq T$ as the time that v gets infected.[1] So z_{s_i} is the time that the i'the infection occurs.

For each pair (v, v'), it will be useful to define an associated function $\kappa_{v,v'}(z, D)$ that equals the number of messages recorded in D as going from v to v' before z_v if $z_v \neq *$, and that equals the total number of such messages in the window otherwise. Similarly define $\underline{\kappa}_{v,v'}(z, D)$ as the number of messages after v gets infected, or 0 if it never gets infected.

For any $k \in \mathbb{N}$, $\tau > 0$, τ^k is the subset of $[0, \tau)^k$ such that $x \in \tau^\kappa \Rightarrow x_i \leq x_j \; \forall i, j > i$. We use "$P(\ldots)$" to refer to either probabilities or probability densities, with the context making the meaning clear.

3.2 The two likelihoods

Our likelihood ratio detector is based on comparing the probability of D under the Poisson process where there is no attack to the probability under the process in which there such an attack at node v_1 at time 0. An anomaly detector only considers the first of these probabilities. Whether or not there is an attack, the probability of our dataset

[1] For some of the equations below, we could treat the event that v does not get infected during $[0, T]$ as equivalent to the event $z_v = T$. However $z_v = T$ and $z_v = *$ are not the same event, and so have different statistical behavior. For example, the marginal probability that $z_v = *$ for any node v is a nonzero number; it is $1 -$ {the probability that no nodes pointing to v_i send a malicious message to v at any time during $[0, T]$}. However the marginal probability that $z_v = T$ is zero, since it is the probability that one of the nodes pointing to v send a malicious message to v at the exact moment T. (It is the density of that marginal that is non-zero.)

conditioned on z is

$$P(D \mid z) \quad = \quad \prod_{v \in V} \prod_{v' \in V, v' \neq v} \Big[(1 - \delta_{z_v,*}) \frac{e^{-z_v \lambda_{v,v',1}} (z_v \lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!} \frac{e^{-(T - z_v)\lambda_{v,v',0}} ((T - z_v)\lambda_{v,v',0})^{\underline{\kappa}_{v,v'}(z,D)}}{\underline{\kappa}_{v,v'}(z,D)!} + (\delta_{z_v,*}) \frac{e^{-T\lambda_{v,v',1}} (T\lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!}\Big] \tag{1}$$

where $\delta_{a,b}$ indicates the Kronecker delta function. (Note that $\delta_{z_v,*}$ equals 1 if node v is not infected in the window $[0, T]$, 0 otherwise.) In particular, the probability of D given that there is no attack is

$$P(D \mid z = \vec{*}) \quad = \quad \prod_{v \in V} \prod_{v' \in V, v' \neq v} \frac{e^{-T\lambda_{v,v',1}} (T\lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!} \tag{2}$$

where $\vec{*}$ is the vector of all $*$'s. This is the only probability considered by an anomaly detector, and is the first of the two probabilities considered by our likelihood ratio detector.

In our initial project, we assume that if an attacker is ever present in the observation window, at time 0 they have infected a particular node v_1 and no other node. (In a full analysis we would average over such infection times and the nodes where they occur according to some prior probability, but for simplicity we ignore this extra step in this paper.) Accordingly, $z_{v_i} > 0 \; \forall i > 1$ (whether there is an attacker or not), and the second of the two probabilities we wish to compare is $P(D \mid z_{v_1} = 0)$.

Unfortunately, our Markov process model gives us $P(D \mid z)$, not $P(D \mid z_{v_1} = 0)$. So we have to evaluate our desired likelihood using a hidden Markov model:

$$P(D \mid z_{v_1} = 0) \quad = \quad \sum_{s \in \mathbb{S}} \int_{T^{|s|}} d\bar{z} \; P(D \mid \bar{z}, s) P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) \tag{3}$$

The first probability in Eq. (3), $P(D \mid \bar{z}, s)$, is given by writing $z_{s_i} = \bar{z}_i$ for all $i \leq |s|$, all other $z_v = *$, and plugging into Eq. (1). (N.b., $\bar{z}$ is indexed by integers, and z by nodes.) The second probability equals 1 if $|s| = 1$. For other s's we can evaluate by iterating the Gillespie algorithm [8]:

Proposition 1. *As shorthand write "$v \notin s$" to mean $\forall i \leq |s|, s_i \neq v$. For any $s, \bar{z} \in T^{|s|}$ where $|s| > 1$,*

$$P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) \quad = \quad \prod_{v \notin s} e^{-(T - z_{|s|}) \sum_{i \leq |s|} \Delta_{s_i,v}} \prod_{j=1}^{|s|-1} \Delta'_{s,j+1} e^{\lambda'_{s,j}(\bar{z}_{j+1} - \bar{z}_j)}$$

where $\lambda'_{s,k} \equiv \sum_{i=1}^{k} \sum_{v \notin \cup_{j=1}^{k} \{s_j\} : (s_i,v) \in E} \Delta_{s_i,v}$ *and* $\Delta'_{s,k} \equiv \sum_{i=1}^{k-1} \Delta_{s_i,s_k}$.

Proof. To begin, expand

$$P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) \quad = \quad P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1) \times P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) \; \times \tag{4}$$

To evaluate the first term on the RHS, $P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1)$, expand the aggregate rate of a malicious message leaving node s_1 if that node is infected as $\lambda'_{s,1}$.

The probability that node s_1 sends a malicious message to s_2 before sending one to any other node is $\frac{\Delta'_{s,2}}{\lambda'_{s,1}}$. Also, the probability that s_1 sends its first infected message at time $\bar{z}_2$ is $\lambda'_{s,1} e^{-\lambda'_{s_1}(\bar{z}_2 - \bar{z}_1)}$. Therefore

$$\begin{aligned} P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1) &= P(s_2 \mid \bar{z}_2, \bar{z}_1 = 0, s_1 = v_1) P(\bar{z}_2 \mid \bar{z}_1 = 0, s_1 = v_1) \\ &= P(s_2 \mid s_1 = v_1) P(\bar{z}_2 \mid \bar{z}_1 = 0, s_1 = v_1) \\ &= \Delta'_{s,2} e^{-\lambda'_{s,1}(\bar{z}_2 - \bar{z}_1)} \end{aligned} \quad (5)$$

Next we similarly expand $P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) = P(s_3 \mid s_2, s_1) P(\bar{z}_3 \mid \bar{z}_2, s_2, s_1)$. The set of edges that lead from either s_1 or s_2 to some novel (currently uninfected) node is $\cup_{v \neq s_1, s_2 \,:\, (s_1, v) \in E \text{ or } (s_2, v) \in E}$. The sum of the malicious message rates of those edges is $\lambda'_{s,2}$ Therefore we have $P(s_3 \mid s_2, s_1) = \Delta'_{s,3} / \lambda'_{s,2}$ and $P(\bar{z}_3 \mid \bar{z}_2, s_2, s_1) = \lambda'_{s,2} e^{\lambda'_{s,2}(\bar{z}_3 - \bar{z}_2)}$, so that

$$P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) \quad = \quad \Delta'_{s,3} e^{-\lambda'_{s,2}(\bar{z}_3 - \bar{z}_2)}$$

Iterating through the remaining components of s gives the second product term on the RHS in the claimed result. The first product term then arises by considering the time interval between $\bar{z}_{|s|}$ and T, during which no nodes v not listed in s receive a malicious message from any of the nodes that are listed in s. □

To evaluate our likelihood ratio attack detector we need to plug the results of Prop. 1 and Eq. (1) into (3), evaluate that integral, and then divide by the likelihood given in Eq. (2).

4 Computational approximations

To use our likelihood ratio attack detector, we need to evaluate Eq. (3). To do this we express it as the expected value of $P(D \mid \bar{z}, s)$ over all $\bar{z}$ and s, evaluated under the multivariate distribution $P(\bar{z}, s \mid \bar{z}_1 = 0, s = v_1)$. We then reformulate that expectation value, in a way that allows us to approximate it via simple sampling Monte Carlo [16].

To begin, we consider a new network (V, E') created from our original network (V, E) by adding enough new edges to those in E so that V contains a (directed) path from v_1 to every node in V. Leave rates of both benign and malicious edges on all of the old edges (i.e., on all $e \in E \subseteq E'$) unchanged. Define some strictly positive value $\tilde{\lambda}$ so that both $T\tilde{\lambda}N^2$ is infinitesimal on the scale of 1 and so that $\tilde{\lambda}$ is infinitesimal on the scale of the smallest rate in the original network. This ensures that the probability that any non-empty data set D' generated with our new net has a message traverse one of the new edges before time T is infinitesimal. This in turn means that the likelihood of any non-empty D generated with the new net is the same as its likelihood with the original net, whether we condition on there being an attacker or on there not being one.

However we are still considering Poisson processes with the new net, and both Poisson rates are greater than zero on all edges in the new net. Combining this with the fact that there is a path in E' from v_1 to every node $v \in V$, we see that if v_1 is infected in the new net, then every node in the new net gets infected at some finite time, with probability 1. This allows us to re-express Eq. (3) as

$$\int_{\infty^N} d\overline{z} \sum_{s \in \mathbb{S}} \delta_{|s|,R(\overline{z})} P(D \mid \overline{z}, s) P(\overline{z}, s \mid \overline{z}_1 = 0, s_1 = v_1) \tag{6}$$

where $R(\overline{z})$ is the number of components of $\overline{z}$ that are less than or equal T. It is this expectation value that we approximate with simple sampling.

Since it is the product $\delta_{|s|,R(\overline{z})} P(\overline{z}, s \mid \overline{z}_1 = 0, s_1 = v_1)$ that is a normalized distribution for this new integral's regions of integration, we must sample from that. To do this, we iterate the expansion of $P(\overline{z}, s \mid \overline{z}_1 = 0, s_1 = v_1)$ in Eq. (4), multiplying by the Kronecker delta function at each step. Note that due to that Kronecker delta function, whenever we reach an iteration i where the sample $\overline{z}_i$ we generate is greater than T, before evaluating $P(D \mid \overline{z}, s)$ we first pad all components of $\overline{z}$ at i or later to be "*", and set s to be the current list. After evaluating $P(D \mid \overline{z}, s)$ for that $\overline{z}$ and s, we break out, and form a new sample of $P(\overline{z}, s \mid \overline{z}_1 = 0, s_1 = v_1)$.

As an illustration, to sample the term $\delta_{|s|,R(\overline{z})} P(\overline{z}_2, s_2 \mid \overline{z}_1 = 0, s_1 = v_1)$, we first set $s_1 = v_1, \overline{z}_1 = 0$, and then sample $\lambda'_{s,1} e^{-\lambda'_{\hat{s}_1} (\overline{z}_2 - \overline{z}_1)}$ to get a value of $\overline{z}_2$. If that $\overline{z}_2 > T$, then we break and start generating a new sample. Otherwise we sample s_2 according to $\frac{\Delta'_{\hat{s},2}}{\lambda'_{\hat{s},1}}$, and then iterate to generate a sample of $P(\overline{z}_3, s_3 \mid \overline{z}_2, s_2, \overline{z}_1 = 0, s_1 = v_1)$. (Pseudocode of this is presented in the Supplemental Materials.)

5 Experimental results

We now present receiver operating characteristics (ROC) curves for various network topologies, message transmission rates and observation windows. The results cover a wide range of typical network structure and attacker behavior. These experimental results provide strong evidence that our detector significantly outperforms state-of-the-art techniques based on anomaly detection, irrespective of network topology and attacker strategy.

An ROC curve is a two-dimensional plot that compares the true positive and false positive rates of a binary classifier.. For a given threshold, the true positive rate is calculated as $\frac{\text{True positives}}{\text{Total positives}}$ and the false positive rate as $\frac{\text{False Positives}}{\text{Total Negatives}}$. These values are plotted for different threshold choices to create a curve. When comparing ROCs, the curve *higher* the curve, the better.

In each of our experiments, we generated 400 realizations of network activity over an observation window length T. There are 200 cases with an attacker and 200 cases without an attacker. For each realization, we compute the likelihood that the observed message transmissions come from a system with no attacker and the likelihood that the observed message transmissions come from a system

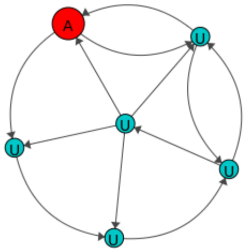

Figure 1: A small star network. The Red "A" node is the attacker. The blue "U" nodes are the normal users of the network.

with an attacker (proposition 1). The classifier is the ratio of the two likelihoods. The ROC curve for anomaly detection uses the likelihood of no attacker as the classifier.

In our first experiment, we analyze a small "star" network as shown in Fig. 1 for $T = 1500$. Normal message traffic has rates in the interval $[0, 2]$ (clustered around 1) that comes from data collected from real networks. Malicious message transmission was set to 3% or 4% of the normal rate (randomly), which models a relatively slow attack. Under this scenario, the attacker remains on the network for long periods of time and traverse the network sporadically.

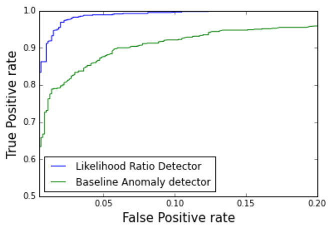

Figure 2: ROC curves for the network topology shown in figure 1

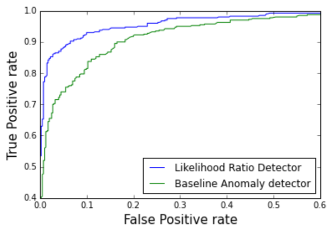

Figure 3: ROC curve under the "enter slowly, traverse quickly" specification

Fig. 2 shows that the likelihood ratio detector outperforms the baseline anomaly detector.

In our second experiment, we consider the same network but adopt an "enter slowly, traverse quickly" strategy for the attacker and use $T = 50$. In this scenario, the attacker initially sends messages at a rate of 10% of the initial infected node's normal transmission rate. Once inside the network, the attacker traverses the network rapidly by sending messages at a rate of 25% of the normal message rate. Fig. 3 once again shows that the likelihood ratio detector dominates the simple anomaly detector.

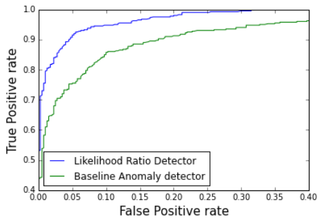

Figure 4: ROC with $T = 10$

In our third experiment, using the same network and $T = 10$, we model an attacker that makes no attempt to "hide" from the detectors, but instead tries to traverse the network fast enough so that by the time an alarm sounds, the network has already been compromised. Malicious message rates are set to half the normal traffic rate. Fig. 4 provides the third validation that the likelihood ratio detector performs better than the anomaly detector and indicates that it is possible to detect the attacker before he reaches his goal.

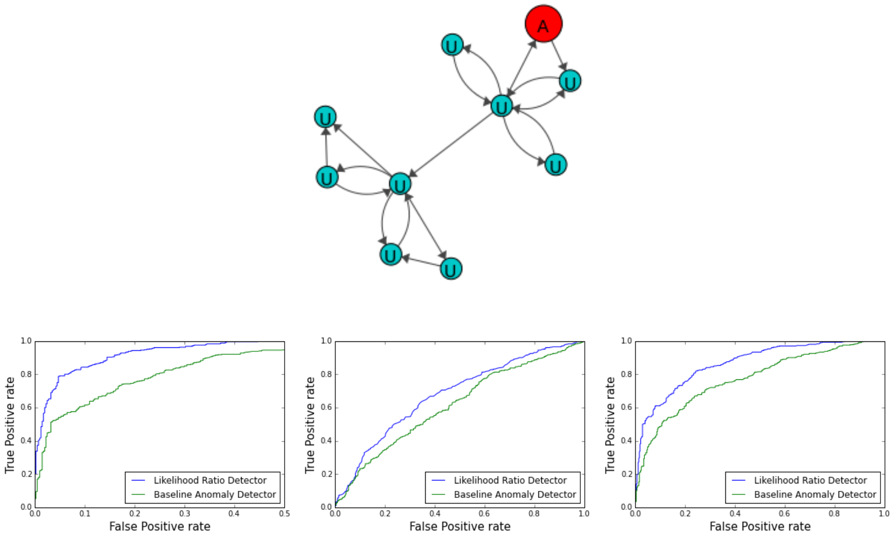

Figure 5: Network topology and ROC curves for various sets of parameters: Left: $T = 2000$ and rates 1–5% normal rates. Middle: $T = 50$ and rates 10–12% normal rates. Right: $T = 10$, rates 25% normal infection rate, except for original attacker node which sends at 50% normal rate.

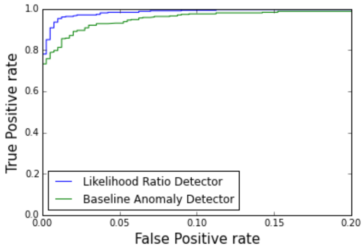

Figure 6: ROC curve under the scenario where the attacker becomes more aggressive as he approaches the goal

In our fourth experiment, we considered the same three attacker strategies for the larger network described in Fig. 5. The ROC curves indicate that the likelihood ratio detector outperforms the anomaly detector under all three specifications in this case as well.

Finally, we tested the likelihood ratio detector when the attacker's strategy is to increase in aggression as he approaches his end target. Fig. 7 shows a cybernet where the attacker moves toward a goal (green node 'G'). All nodes send normal messages at rate 1. At node 'A', the attacker sends malicious messages at rate .05. For each subsequent node infection, the malicious message rate increases by .05. We simulate this scenario for $T = 125$. Once again, Fig. 6 shows that the likelihood ratio detector outperforms the simple anomaly detector.

Model misspecification The preceding results assumed that the rate of malicious message transmission and attacker strategy is known. In reality this is never the case.

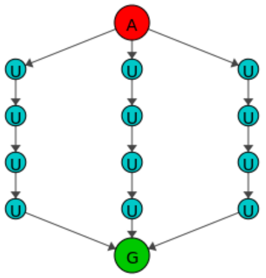

Figure 7: Network topology with misspecified attacker strategy

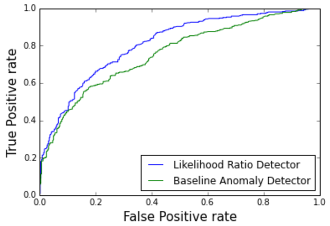

Figure 8: ROC curve when the model allows the attacker to take all paths, but he only takes one.

Therefore, to test the performance of the likelihood ratio detector in a more realistic scenario we assume that the attacker's strategy is misspecified and test the performance of our approach. In this scenario, we used the network described in Fig. 7 with $T = 10$ and we set all normal message rates equal to 1 and compute the likelihood ratio as if the rate of malicious messages is .5 for all infected nodes. This model of the attacker allows the detector to hedge for an attacker that could choose any path from A to G. In our experiments, the actual attacker only traverses the center path (using rate .5). Fig. 8 shows that even under misspecification, likelihood ratio detector is superior to the baseline anomaly detector.

6 Future work

There are many ways that our hidden Markov process model of the behavior of a cybernet can be extended. Most straightforwardly, by incorporating a loss function for incorrect alerts and specifying a prior probability of there being an attacker, we should be able to construct a Bayesian decision-theoretic extension of our anomaly detector.

Other future work involves applying our modeling approach for more realistic cybernets. This will require us to consider other approaches to evaluating our likelihood, e.g., importance sampling or MCMC, rather than simple sampling. Indeed, it may even be possible to do closed form evaluation of our integrals, using the Laplace convolution theorem [19].

It should be possible to use our model to make predictions for *any* scenario in which humans interact with technical systems in continuous time, and our observational data is limited. In other future work we will apply our models to make predictions in such scenarios. (See also [4].) This should allow us to address any statistical question concerning such scenarios, not just for anomaly detection.

Finally, we have started to extend our approach involving hidden Markov processes to model not just a single human interacting with a technology system, but a set of humans, interacting with one another as well as that underlying technology system [18].

This extension can be viewed as an "event-driven" noncooperative game theory, distinct from both differential games (in which player moves are real-valued, and chosen continually, at all times) and Markov games (which lack hidden variables). Future work involves investigating this event-driven game theory, especially for models of human strategic behavior grounded in real-world data, e.g., the quantal response equilibrium.

Acknowledgments

DHW and TB acknowledge the Santa Fe Institute.

References

[1] Karsten M Borgwardt, H-P Kriegel, and Peter Wackersreuther, *Pattern mining in frequent dynamic subgraphs*, Data Mining, 2006. ICDM'06. Sixth International Conference on, IEEE, 2006, pp. 818–822.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar, *Anomaly detection: A survey*, ACM Computing Surveys (CSUR) **41** (2009), no. 3, 15.

[3] Mihai Christodorescu and Somesh Jha, *Static analysis of executables to detect malicious patterns*, Tech. report, DTIC Document, 2006.

[4] Wolpert D.H. and J.W. Bono, *Distribution-valued solution concepts*, Review of Behavioral Economics (in press).

[5] Hristo Djidjev, Gary Sandine, Curtis Storlie, and Scott Vander Wiel, *Graph based statistical analysis of network traffic*, Proceedings of the Ninth Workshop on Mining and Learning with Graphs, 2011.

[6] William Eberle, Jeffrey Graves, and Lawrence Holder, *Insider threat detection using a graph-based approach*, Journal of Applied Security Research **6** (2010), no. 1, 32–81.

[7] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez, *Anomaly-based network intrusion detection: Techniques, systems and challenges*, computers & security **28** (2009), no. 1, 18–28.

[8] Daniel T Gillespie, *Exact stochastic simulation of coupled chemical reactions*, The journal of physical chemistry **81** (1977), no. 25, 2340–2361.

[9] Nicholas A Heard, David J Weston, Kiriaki Platanioti, David J Hand, et al., *Bayesian anomaly detection methods for social networks*, The Annals of Applied Statistics **4** (2010), no. 2, 645–662.

[10] C Hummel, *Why crack when you can pass the hash*, SANS Institute InfoSec Reading Room **21** (2009), 2009.

[11] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski, *Overview of sequential monte carlo methods for parameter estimation on general state space models*, Proc. 15th IFAC Symposium on System Identification (SYSID) 2009, Saint-Malo, France, 2009.

[12] Alexander D Kent and Lorie M Liebrock, *Differentiating user authentication graphs*, Security and Privacy Workshops (SPW), 2013 IEEE, IEEE, 2013, pp. 72–75.

[13] Wenke Lee and Salvatore J Stolfo, *Data mining approaches for intrusion detection*, Defense Technical Information Center, 2000.

[14] Joshua Neil, Curtis Hash, Alexander Brugh, Mike Fisk, and Curtis B Storlie, *Scan statistics for the online detection of locally anomalous subgraphs*, Technometrics **55** (2013), no. 4, 403–414.

[15] Roberto Perdisci, David Dagon, Wenke Lee, Prahlad Fogla, and Monirul Sharif, *Misleading worm signature generators using deliberate noise injection*, Security and Privacy, 2006 IEEE Symposium on, IEEE, 2006, pp. 15–pp.

[16] C. P. Robert and G. Casella, *Monte Carlo statistical methods*, Springer-Verlag, New York, 2004.

[17] Stuart Staniford-Chen, Steven Cheung, Richard Crawford, Mark Dilger, Jeremy Frank, James Hoagland, Karl Levitt, Christopher Wee, Raymond Yip, and Dan Zerkle, *Grids-a graph based intrusion detection system for large networks*, Proceedings of the 19th national information systems security conference, vol. 1, Baltimore, 1996, pp. 361–370.

[18] David H. Wolpert, Tanmoy Bhattacharya, Russell Bent, Joshua Neil, Alexander Kent, and Annarita Giani, *Event-driven non-cooperative games, la-ur-13-22238*, Tech. report, Los Alamos National Laboratory, 2013.

[19] David H Wolpert and Simon DeDeo, *Estimating functions of distributions defined over spaces of unknown size*, Entropy **15** (2013), no. 11, 4668–4699.

HMMs for Optimal Detection of Cybernet Attacks

Justin Grana
Economics Department
American University
Washington, DC
jg3705a@student.american.edu

David Wolpert
Santa Fe Institute
1399 Hyde Park Rd.
Santa Fe, NM 87501
davidwolpert.weebly.com

Joshua Neil
Los Alamos National Laboratory
PO Box 1663 MS B264
Los Alamos, NM
jneil@lanl.gov

Dongping Xie
Economics Department
American University
Washington, DC
xdp668@gmail.com

Tanmoy Bhattacharya
Santa Fe Institute
1399 Hyde Park Rd.
Santa Fe, NM 87501
tanmoy@santafe.edu

Russell Bent
Los Alamos National Laboratory
PO Box 1663 MS B264
Los Alamos, NM
rbent@lanl.gov

June 20, 2014

Abstract

The rapid detection of attackers within firewalls of computer networks is of paramount importance. Anomaly detectors address this problem by quantifying deviations from baseline statistical models of normal network behavior. However anomaly detectors have many false positives, severely limiting their practical utility. To circumvent this problem we need to evaluate both the likelihood of observed network behavior given that no attacker is present (as in anomaly detectors) and the likelihood given that an attacker *is* present. Any realistic stochastic model for behavior of a compromised network must work in continuous time, with many latent variables. Here we develop such a stochastic model of a compromised network's behavior, and show how to use Monte Carlo methods to integrate over its latent variables. This allows us to evaluate the likelihood of observed behavior in a compromised network. We then present computer experiments showing that a likelihood ratio detector that combines our attacker model with a model of normal network behavior has far better ROC curves than an anomaly detector that only uses the model of normal network behavior.

1 Introduction

Many existing systems for detecting intrusions into cybernetworks monitor data streams only at the perimeter of the network. There is no examination of the behavior of communicating computers within the network that might reveal the penetration of the firewall once it has occurred. In addition, perimeter-focused tools are typically based upon matching previously known attack signatures to current data, and so are unable to detect attacks that avoid the current database of signatures [3, 15]. The extremely fast development time of new attack vectors, resulting in "zero-hour exploits", makes such matching increasingly problematic. For these two reasons, the rapid detection of attackers *within* network perimeters, without reliance on signature matching, is of paramount importance.

Machine learning provides some of the most promising approaches to such detection within the network. An example is anomaly detectors, which quantify deviations from baseline statistical models of normal network behavior when the network has not been penetrated [14, 9]. However in practice, many reported anomalies end up being false, reflecting behavior that is unusual but benign, severely limiting the usefulness of anomaly detectors. The underlying problem is that anomaly detectors do not exploit any model for the alternative to normal network behavior, i.e. for the behavior of the network behavior once it has been penetrated. Since our goal is to distinguish benign behavior from behavior indicative of an attack, we should be able to achieve far better performance if we could evaluate the likelihood of a given set of data under models for both types of behavior.

The challenge is how to model behavior of a network that has been penetrated without pre-supposing attacker methods, since these methods evolve so rapidly. To see how this might be done, consider the movement of an attacker through a network. Often in order to traverse the network the attacker will steal administrator credentials [12], using techniques such as pass-the-hash [10]. However *no matter what attack method they use*, typically they will conduct reconnaissance to guide their movement, perhaps to insert malware, or perhaps to collect increasingly valuable data for later exfiltration.

This means that there is a definite sequence in the movement of the attacker across the net, from computers with low value (for any of the goals of inserting malware, extracting data, or stealing credentials) to computers with higher value. This will be true *no matter what precise methods the attacker uses* to achieve that movement. Moreover, it will leave a trace of increasing network traffic going from low value computers to progressively higher value ones. Accordingly, this trace of the attacker's movement within the net — an inherently global property of the data traffic — can be used as the basis of a model of network behavior once it has been penetrated.

Since this trace has a definite time-ordering, we must model with a Markov process, not an IID process. However evaluating the likelihood of a given dataset of observed network traffic under such a statistical model of an attack is challenging. One of the main issues is that traffic occurs so rapidly that an accurate model must treat time as a continuous variable. Another is that traffic monitoring equipment does not detect the most important variables governing the traffic, e.g., the infection states of the computers in the network at a given time. The challenge then is to evaluate the likelihood of observed traffic under two hidden Markov process models, one for the case of no attacker on the network, and one for the case where there is an attacker, but their locations in the network at any given moment are unknown.

In this paper we show how one can do this, using Monte Carlo techniques to approximate the relevant integrals. We then present computer experiments on toy scenarios that show that a likelihood ratio detector which combines our attacker model with a model of normal network behavior has far better ROC curves than an anomaly detector that only uses the model of normal network behavior.

In Section 2, a brief discussion of past approaches to model-based anomaly detection in computer networks is given, using the Generalized Likelihood Ratio Test (GLRT) to produce anomaly scores. We follow that in Section 3 with a detailed exposition of our Markov process model for traffic between pairs of computers, both for a network not under attack and for one that is. Section 4 discusses the Monte Carlo integral estimates required to evaluate the associated likelihoods for any given dataset of traffic patterns. Section 5 then presents experimental results showing remarkable improvement in detection performance when attack models are taken into account. Finally, we discuss future directions and conclude in Section 6.

2 Background

Model-based anomaly detection proceeds by first estimating the parameters of a null model for expected behavior. We denote these historical estimates as $\hat{\theta}$. Next, given a data set $\mathbf{X}$ under question, the likelihood of the parameters given that data can be evaluated: $\mathcal{L}(\hat{\theta} \mid \mathbf{X})$. One can test whether a more likely alternative parameterization is present given $\mathbf{X}$, by calculating the GLRT:

$$\lambda = \frac{\mathcal{L}(\hat{\theta} \mid \mathbf{X})}{\sup_{\theta \in \mathbf{\Theta}} \mathcal{L}(\theta \mid \mathbf{X})}$$

where $\mathbf{\Theta}$ is an alternative parameter space.

Typically, we choose what data $\mathbf{X}$ to collect to facilitate statistical discovery of security breaches. The associated likelihood model may involve a graph connecting computers (nodes) with edges representing time-series of traffic. Since attacks typically cover multiple nodes and edges, subgraphs can be used to group data from multiple nodes and edges into $\mathbf{X}$ for increased detection power. Graph based methods include [1, 5, 6, 14, 17]. However, in no work identified is the stochastic behavior of attackers as they traverse the network collecting reward discussed.

To include this behavior, we will introduce an attack model. If an attacker is present and behaving according to an alternative parameterization, $\theta_{\mathbf{A}}$, then the uniformly most powerful test for rejecting the null hypothesis that {no attacker is present} is the test where $\theta_{\mathbf{A}}$ is used in the denominator:

$$\tilde{\lambda} = \frac{\mathcal{L}(\hat{\theta} \mid \mathbf{X})}{\mathcal{L}(\theta_{\mathbf{A}} \mid \mathbf{X})}.$$

We will use this fact to design optimal attack detectors.

A good survey of general anomaly detection is provided by [2]. Specifically for cyber security applications, machine learning and statistical approaches have shown promise in detecting malicious behavior in computer networks. The first robust statistical approach to this was done by Lee and Stolfo [13], and a good survey of the modern literature is given in [7]. The underlying problem of using partial observations to estimate the parameters of a system undergoing stochastic dynamics is also well studied, see [11] for a review.

3 Model

We model a cybernet as a directed graph, potentially with cycles, where each node represents either a computer or a human, either inside the firewall or outside it. Each node has an associated state. Examples of human nodes are users, system administrators, and hackers, whose states can represent their knowledge, their strategies, etc. Each directed edge represents a potential communication directly connecting one node (human or computer) to another node (human or computer). These edges have associated states, which represent communication messages. So the cybernet evolves according to a Markov process across all possible joint states of every node and every edge. (See Supplementary Material for a review of Markov processes over discrete state spaces.)

In this initial project, we only consider computer nodes, treating the human using a particular computer as part of that computer. We also only consider those computers that are inside the firewall. Each node can be in one of two states, "normal" or "infected". Similarly, each edge can be in one of two states, "no message", or "message in transit". When a node is in a normal state, it sends benign messages along any of its directed edges according to an underlying Poisson process with a pre-specified rate. When a node is infected, it still sends benign messages at the same rate as when it is not infected, but now it superimposes malicious messages. These are generated according to another Poisson process, with a much lower rate.

For simplicity we assume that if an edge from an infected node to a non-infected node gains a new malicious message at time t, then with probability 1.0 the second node becomes infected and the new malicious message disappears immediately, leaving a trace on our net-monitoring equipment that that message traveled down that edge at t. (Formally, we model by this by having the Markov rate constants for message absorptions all be much larger than the rate constants for message emissions.) No node can become infected spontaneously, and no node can become uninfected.

3.1 Definitions

Let $G = (V, E)$ be the directed graph of a cybernet where $V = \{v_1, v_2 ... v_N\}$ is the set of nodes. Use 1 to represent the normal state of a given node and 0 to represent the infected state. Let $\sigma \in \mathbb{B}^N$ denote the state of all nodes in the network and σ_{v_i} denote the state of node v_i. The Markov process governing the cybernet is parameterized by the set $\lambda \equiv \{(\lambda_{v,v',\sigma_v}) : v, v' \in V, v; \neq v, \sigma_v \in \mathbb{B}\}$ giving the total rates at which v sends messages to v' when v is in state σ_v. (The far larger rate constants for message absorption are irrelevant to our analysis.) We write the rate parameter for just emission of malicious message from v to v' as $\Delta_{v,v'} \equiv \lambda_{v,v',0} - \lambda_{v,v',1}$. For simplicity, in this paper we take λ fixed and greater than zero — in a full analysis we would average over it according to a prior.

Suppose we observe the traffic on a net for a time interval $[0, T]$, resulting in a dataset $D = \{(\tau_i, v_i, v'_i)\}$, where each $\tau_i \in [0, T]$ and each $(v_i, v'_i) \in V^2$. We interpret any $(\tau, v, v') \in D$ as the observation that a message was added at time τ to the edge from v to v'. We assume that the observation process is noise-free, i.e., that all messages are recorded and no spurious messages are.

For all $1 \leq k \leq N$, define $\mathbb{S}^k$ as the set of vectors $s \in V^{|k|}$ such that for all $i, j \neq i$, $s_i \neq s_j$. Define $\mathbb{S} = \cup_{k=1}^N \mathbb{S}^k$. Below we will interpret any $s \in \mathbb{S}$ as a time-ordered sequence of all node infections that occur in $[0, T]$ (though others might occur later). Also define the space $Z \equiv [0, T] \cup \{*\}$ and write elements of any associated space Z^m as $z = (z_{v_1}, z_{v_2}, ... z_{v_m})$, i.e., index elements of z by the elements of V. Below we will interpret any component $z_v = *$ to mean that node v does not get infected during $[0, T]$ (though it might get infected later), and every real-valued $z_v \leq T$ as the time that v gets infected.[1] So z_{s_i} is the time that the i'the infection occurs.

For each pair (v, v'), it will be useful to define an associated function $\kappa_{v,v'}(z, D)$ that equals the number of messages recorded in D as going from v to v' before z_v if $z_v \neq *$, and that equals the total number of such messages in the window otherwise. Similarly define $\underline{\kappa}_{v,v'}(z, D)$ as the number of messages after v gets infected, or 0 if it never gets infected.

For any $k \in \mathbb{N}$, $\tau > 0$, τ^k is the subset of $[0, \tau)^k$ such that $x \in \tau^\kappa \Rightarrow x_i \leq x_j \; \forall i, j > i$. We use "$P(\ldots)$" to refer to either probabilities or probability densities, with the context making the meaning clear.

3.2 The two likelihoods

Our likelihood ratio detector is based on comparing the probability of D under the Poisson process where there is no attack to the probability under the process in which there such an attack at node v_1 at time 0. An anomaly detector only considers the first of these probabilities. Whether or not there is an attack, the probability of our dataset

[1] For some of the equations below, we could treat the event that v does not get infected during $[0, T]$ as equivalent to the event $z_v = T$. However $z_v = T$ and $z_v = *$ are not the same event, and so have different statistical behavior. For example, the marginal probability that $z_v = *$ for any node v is a nonzero number; it is $1 -$ {the probability that no nodes pointing to v_i send a malicious message to v at any time during $[0, T]$}. However the marginal probability that $z_v = T$ is zero, since it is the probability that one of the nodes pointing to v send a malicious message to v at the exact moment T. (It is the density of that marginal that is non-zero.)

conditioned on z is

$$P(D \mid z) = \prod_{v \in V} \prod_{v' \in V, v' \neq v} \Big[(1 - \delta_{z_v,*}) \frac{e^{-z_v \lambda_{v,v',1}} (z_v \lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!} \frac{e^{-(T-z_v)\lambda_{v,v',0}} ((T - z_v)\lambda_{v,v',0})^{\underline{\kappa}_{v,v'}(z,D)}}{\underline{\kappa}_{v,v'}(z,D)!} + (\delta_{z_v,*}) \frac{e^{-T\lambda_{v,v',1}} (T\lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!}\Big] \tag{1}$$

where $\delta_{a,b}$ indicates the Kronecker delta function. (Note that $\delta_{z_v,*}$ equals 1 if node v is not infected in the window $[0, T]$, 0 otherwise.) In particular, the probability of D given that there is no attack is

$$P(D \mid z = \vec{*}) = \prod_{v \in V} \prod_{v' \in V, v' \neq v} \frac{e^{-T\lambda_{v,v',1}} (T\lambda_{v,v',1})^{\kappa_{v,v'}(z,D)}}{\kappa_{v,v'}(z,D)!} \tag{2}$$

where $\vec{*}$ is the vector of all $*$'s. This is the only probability considered by an anomaly detector, and is the first of the two probabilities considered by our likelihood ratio detector.

In our initial project, we assume that if an attacker is ever present in the observation window, at time 0 they have infected a particular node v_1 and no other node. (In a full analysis we would average over such infection times and the nodes where they occur according to some prior probability, but for simplicity we ignore this extra step in this paper.) Accordingly, $z_{v_i} > 0 \; \forall i > 1$ (whether there is an attacker or not), and the second of the two probabilities we wish to compare is $P(D \mid z_{v_1} = 0)$.

Unfortunately, our Markov process model gives us $P(D \mid z)$, not $P(D \mid z_{v_1} = 0)$. So we have to evaluate our desired likelihood using a hidden Markov model:

$$P(D \mid z_{v_1} = 0) = \sum_{s \in \mathbb{S}} \int_{T^{|s|}} d\bar{z} \; P(D \mid \bar{z}, s) P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) \tag{3}$$

The first probability in Eq. (3), $P(D \mid \bar{z}, s)$, is given by writing $z_{s_i} = \bar{z}_i$ for all $i \leq |s|$, all other $z_v = *$, and plugging into Eq. (1). (N.b., $\bar{z}$ is indexed by integers, and z by nodes.) The second probability equals 1 if $|s| = 1$. For other s's we can evaluate by iterating the Gillespie algorithm [8]:

Proposition 1. *As shorthand write "$v \notin s$" to mean $\forall i \leq |s|, s_i \neq v$. For any $s, \bar{z} \in T^{|s|}$ where $|s| > 1$,*

$$P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) = \prod_{v \notin s} e^{-(T - z_{|s|}) \sum_{i \leq |s|} \Delta_{s_i,v}} \prod_{j=1}^{|s|-1} \Delta'_{s,j+1} e^{\lambda'_{s,j}(\bar{z}_{j+1} - \bar{z}_j)}$$

where $\lambda'_{s,k} \equiv \sum_{i=1}^{k} \sum_{v \notin \cup_{j=1}^{k}\{s_j\}:(s_i,v) \in E} \Delta_{s_i,v}$ and $\Delta'_{s,k} \equiv \sum_{i=1}^{k-1} \Delta_{s_i,s_k}$.

Proof. To begin, expand

$$P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) = P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1) \times P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) \times \tag{4}$$

To evaluate the first term on the RHS, $P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1)$, expand the aggregate rate of a malicious message leaving node s_1 if that node is infected as $\lambda'_{s,1}$.

The probability that node s_1 sends a malicious message to s_2 before sending one to any other node is $\frac{\Delta'_{s,2}}{\lambda'_{s,1}}$. Also, the probability that s_1 sends its first infected message at time $\bar{z}_2$ is $\lambda'_{s,1} e^{-\lambda'_{s_1}(\bar{z}_2 - \bar{z}_1)}$. Therefore

$$\begin{aligned} P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1) &= P(s_2 \mid \bar{z}_2, \bar{z}_1 = 0, s_1 = v_1) P(\bar{z}_2 \mid \bar{z}_1 = 0, s_1 = v_1) \\ &= P(s_2 \mid s_1 = v_1) P(\bar{z}_2 \mid \bar{z}_1 = 0, s_1 = v_1) \\ &= \Delta'_{s,2} e^{-\lambda'_{s,1}(\bar{z}_2 - \bar{z}_1)} \end{aligned} \tag{5}$$

Next we similarly expand $P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) = P(s_3 \mid s_2, s_1) P(\bar{z}_3 \mid \bar{z}_2, s_2, s_1)$. The set of edges that lead from either s_1 or s_2 to some novel (currently uninfected) node is $\cup_{v \neq s_1, s_2 \,:\, (s_1, v) \in E \text{ or } (s_2, v) \in E}$. The sum of the malicious message rates of those edges is $\lambda'_{s,2}$ Therefore we have $P(s_3 \mid s_2, s_1) = \Delta'_{s,3} / \lambda'_{s,2}$ and $P(\bar{z}_3 \mid \bar{z}_2, s_2, s_1) = \lambda'_{s,2} e^{\lambda'_{s,2}(\bar{z}_3 - \bar{z}_2)}$, so that

$$P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1) \quad = \quad \Delta'_{s,3} e^{-\lambda'_{s,2}(\bar{z}_3 - \bar{z}_2)}$$

Iterating through the remaining components of s gives the second product term on the RHS in the claimed result. The first product term then arises by considering the time interval between $\bar{z}_{|s|}$ and T, during which no nodes v not listed in s receive a malicious message from any of the nodes that are listed in s. □

To evaluate our likelihood ratio attack detector we need to plug the results of Prop. 1 and Eq. (1) into (3), evaluate that integral, and then divide by the likelihood given in Eq. (2).

4 Computational approximations

To use our likelihood ratio attack detector, we need to evaluate Eq. (3). To do this we express it as the expected value of $P(D \mid \bar{z}, s)$ over all $\bar{z}$ and s, evaluated under the multivariate distribution $P(\bar{z}, s \mid \bar{z}_1 = 0, s = v_1)$. We then reformulate that expectation value, in a way that allows us to approximate it via simple sampling Monte Carlo [16].

To begin, we consider a new network (V, E') created from our original network (V, E) by adding enough new edges to those in E so that V contains a (directed) path from v_1 to every node in V. Leave rates of both benign and malicious edges on all of the old edges (i.e., on all $e \in E \subseteq E'$) unchanged. Define some strictly positive value $\tilde{\lambda}$ so that both $T \tilde{\lambda} N^2$ is infinitesimal on the scale of 1 and so that $\tilde{\lambda}$ is infinitesimal on the scale of the smallest rate in the original network. This ensures that the probability that any non-empty data set D' generated with our new net has a message traverse one of the new edges before time T is infinitesimal. This in turn means that the likelihood of any non-empty D generated with the new net is the same as its likelihood with the original net, whether we condition on there being an attacker or on there not being one.

However we are still considering Poisson processes with the new net, and both Poisson rates are greater than zero on all edges in the new net. Combining this with the fact that there is a path in E' from v_1 to every node $v \in V$, we see that if v_1 is infected in the new net, then every node in the new net gets infected at some finite time, with probability 1. This allows us to re-express Eq. (3) as

$$\int_{\infty^N} d\bar{z} \sum_{s \in \mathbb{S}} \delta_{|s|,R(\bar{z})} P(D \mid \bar{z}, s) P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1) \tag{6}$$

where $R(\bar{z})$ is the number of components of $\bar{z}$ that are less than or equal T. It is this expectation value that we approximate with simple sampling.

Since it is the product $\delta_{|s|,R(\bar{z})} P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1)$ that is a normalized distribution for this new integral's regions of integration, we must sample from that. To do this, we iterate the expansion of $P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1)$ in Eq. (4), multiplying by the Kronecker delta function at each step. Note that due to that Kronecker delta function, whenever we reach an iteration i where the sample $\bar{z}_i$ we generate is greater than T, before evaluating $P(D \mid \bar{z}, s)$ we first pad all components of $\bar{z}$ at i or later to be "*", and set s to be the current list. After evaluating $P(D \mid \bar{z}, s)$ for that $\bar{z}$ and s, we break out, and form a new sample of $P(\bar{z}, s \mid \bar{z}_1 = 0, s_1 = v_1)$.

As an illustration, to sample the term $\delta_{|s|,R(\bar{z})} P(\bar{z}_2, s_2 \mid \bar{z}_1 = 0, s_1 = v_1)$, we first set $s_1 = v_1, \bar{z}_1 = 0$, and then sample $\lambda'_{s,1} e^{-\lambda'_{\hat{s}_1}(\bar{z}_2 - \bar{z}_1)}$ to get a value of $\bar{z}_2$. If that $\bar{z}_2 > T$, then we break and start generating a new sample. Otherwise we sample s_2 according to $\frac{\Delta'_{\hat{s},2}}{\lambda'_{\hat{s},1}}$, and then iterate to generate a sample of $P(\bar{z}_3, s_3 \mid \bar{z}_2, s_2, \bar{z}_1 = 0, s_1 = v_1)$. (Pseudocode of this is presented in the Supplemental Materials.)

5 Experimental results

We now present receiver operating characteristics (ROC) curves for various network topologies, message transmission rates and observation windows. The results cover a wide range of typical network structure and attacker behavior. These experimental results provide strong evidence that our detector significantly outperforms state-of-the-art techniques based on anomaly detection, irrespective of network topology and attacker strategy.

An ROC curve is a two-dimensional plot that compares the true positive and false positive rates of a binary classifier.. For a given threshold, the true positive rate is calculated as $\frac{\text{True positives}}{\text{Total positives}}$ and the false positive rate as $\frac{\text{False Positives}}{\text{Total Negatives}}$. These values are plotted for different threshold choices to create a curve. When comparing ROCs, the curve *higher* the curve, the better.

In each of our experiments, we generated 400 realizations of network activity over an observation window length T. There are 200 cases with an attacker and 200 cases without an attacker. For each realization, we compute the likelihood that the observed message transmissions come from a system with no attacker and the likelihood that the observed message transmissions come from a system

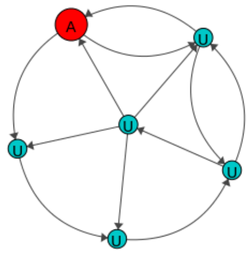

Figure 1: A small star network. The Red "A" node is the attacker. The blue "U" nodes are the normal users of the network.

with an attacker (proposition 1). The classifier is the ratio of the two likelihoods. The ROC curve for anomaly detection uses the likelihood of no attacker as the classifier.

In our first experiment, we analyze a small "star" network as shown in Fig. 1 for $T = 1500$. Normal message traffic has rates in the interval $[0, 2]$ (clustered around 1) that comes from data collected from real networks. Malicious message transmission was set to 3% or 4% of the normal rate (randomly), which models a relatively slow attack. Under this scenario, the attacker remains on the network for long periods of time and traverse the network sporadically.

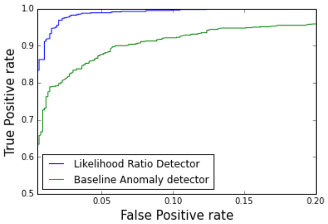

Figure 2: ROC curves for the network topology shown in figure 1

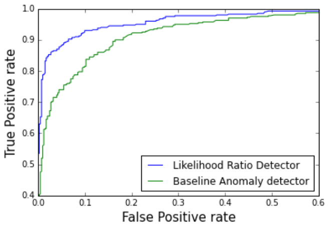

Figure 3: ROC curve under the "enter slowly, traverse quickly" specification

Fig. 2 shows that the likelihood ratio detector outperforms the baseline anomaly detector.

In our second experiment, we consider the same network but adopt an "enter slowly, traverse quickly" strategy for the attacker and use $T = 50$. In this scenario, the attacker initially sends messages at a rate of 10% of the initial infected node's normal transmission rate. Once inside the network, the attacker traverses the network rapidly by sending messages at a rate of 25% of the normal message rate. Fig. 3 once again shows that the likelihood ratio detector dominates the simple anomaly detector.

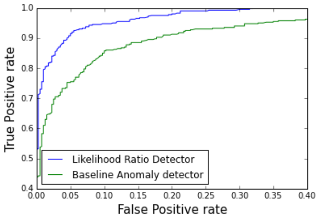

Figure 4: ROC with $T = 10$

In our third experiment, using the same network and $T = 10$, we model an attacker that makes no attempt to "hide" from the detectors, but instead tries to traverse the network fast enough so that by the time an alarm sounds, the network has already been compromised. Malicious message rates are set to half the normal traffic rate. Fig. 4 provides the third validation that the likelihood ratio detector performs better than the anomaly detector and indicates that it is possible to detect the attacker before he reaches his goal.

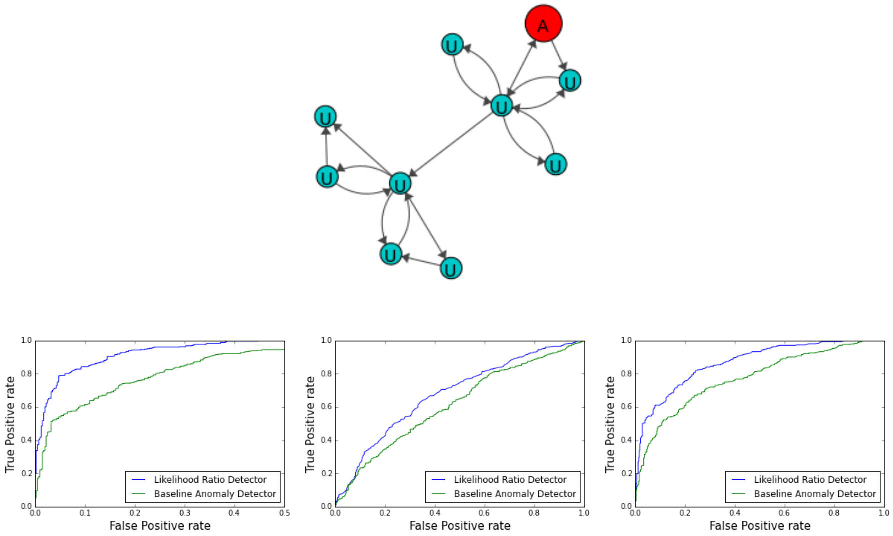

Figure 5: Network topology and ROC curves for various sets of parameters: Left: $T = 2000$ and rates 1–5% normal rates. Middle: $T = 50$ and rates 10–12% normal rates. Right: $T = 10$, rates 25% normal infection rate, except for original attacker node which sends at 50% normal rate.

In our fourth experiment, we considered the same three attacker strategies for the larger network described in Fig. 5. The ROC curves indicate that the likelihood ratio detector outperforms the anomaly detector under all three specifications in this case as well.

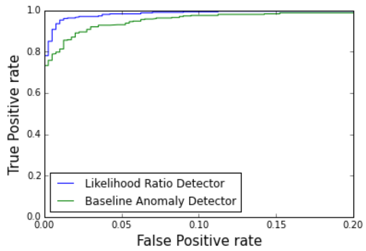

Figure 6: ROC curve under the scenario where the attacker becomes more aggressive as he approaches the goal

Finally, we tested the likelihood ratio detector when the attacker's strategy is to increase in aggression as he approaches his end target. Fig. 7 shows a cybernet where the attacker moves toward a goal (green node 'G'). All nodes send normal messages at rate 1. At node 'A', the attacker sends malicious messages at rate .05. For each subsequent node infection, the malicious message rate increases by .05. We simulate this scenario for $T = 125$. Once again, Fig. 6 shows that the likelihood ratio detector outperforms the simple anomaly detector.

Model misspecification The preceding results assumed that the rate of malicious message transmission and attacker strategy is known. In reality this is never the case.

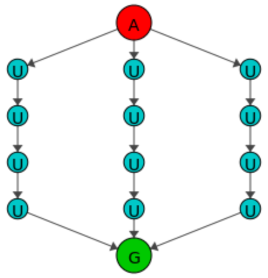

Figure 7: Network topology with misspecified attacker strategy

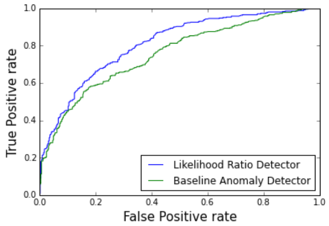

Figure 8: ROC curve when the model allows the attacker to take all paths, but he only takes one.

Therefore, to test the performance of the likelihood ratio detector in a more realistic scenario we assume that the attacker's strategy is misspecified and test the performance of our approach. In this scenario, we used the network described in Fig. 7 with $T = 10$ and we set all normal message rates equal to 1 and compute the likelihood ratio as if the rate of malicious messages is .5 for all infected nodes. This model of the attacker allows the detector to hedge for an attacker that could choose any path from A to G. In our experiments, the actual attacker only traverses the center path (using rate .5). Fig. 8 shows that even under misspecification, likelihood ratio detector is superior to the baseline anomaly detector.

6 Future work

There are many ways that our hidden Markov process model of the behavior of a cybernet can be extended. Most straightforwardly, by incorporating a loss function for incorrect alerts and specifying a prior probability of there being an attacker, we should be able to construct a Bayesian decision-theoretic extension of our anomaly detector.

Other future work involves applying our modeling approach for more realistic cybernets. This will require us to consider other approaches to evaluating our likelihood, e.g., importance sampling or MCMC, rather than simple sampling. Indeed, it may even be possible to do closed form evaluation of our integrals, using the Laplace convolution theorem [19].

It should be possible to use our model to make predictions for *any* scenario in which humans interact with technical systems in continuous time, and our observational data is limited. In other future work we will apply our models to make predictions in such scenarios. (See also [4].) This should allow us to address any statistical question concerning such scenarios, not just for anomaly detection.

Finally, we have started to extend our approach involving hidden Markov processes to model not just a single human interacting with a technology system, but a set of humans, interacting with one another as well as that underlying technology system [18].

This extension can be viewed as an "event-driven" noncooperative game theory, distinct from both differential games (in which player moves are real-valued, and chosen continually, at all times) and Markov games (which lack hidden variables). Future work involves investigating this event-driven game theory, especially for models of human strategic behavior grounded in real-world data, e.g., the quantal response equilibrium.

Acknowledgments

DHW and TB acknowledge the Santa Fe Institute.

References

[1] Karsten M Borgwardt, H-P Kriegel, and Peter Wackersreuther, *Pattern mining in frequent dynamic subgraphs*, Data Mining, 2006. ICDM'06. Sixth International Conference on, IEEE, 2006, pp. 818–822.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar, *Anomaly detection: A survey*, ACM Computing Surveys (CSUR) **41** (2009), no. 3, 15.

[3] Mihai Christodorescu and Somesh Jha, *Static analysis of executables to detect malicious patterns*, Tech. report, DTIC Document, 2006.

[4] Wolpert D.H. and J.W. Bono, *Distribution-valued solution concepts*, Review of Behavioral Economics (in press).

[5] Hristo Djidjev, Gary Sandine, Curtis Storlie, and Scott Vander Wiel, *Graph based statistical analysis of network traffic*, Proceedings of the Ninth Workshop on Mining and Learning with Graphs, 2011.

[6] William Eberle, Jeffrey Graves, and Lawrence Holder, *Insider threat detection using a graph-based approach*, Journal of Applied Security Research **6** (2010), no. 1, 32–81.

[7] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez, *Anomaly-based network intrusion detection: Techniques, systems and challenges*, computers & security **28** (2009), no. 1, 18–28.

[8] Daniel T Gillespie, *Exact stochastic simulation of coupled chemical reactions*, The journal of physical chemistry **81** (1977), no. 25, 2340–2361.

[9] Nicholas A Heard, David J Weston, Kiriaki Platanioti, David J Hand, et al., *Bayesian anomaly detection methods for social networks*, The Annals of Applied Statistics **4** (2010), no. 2, 645–662.

[10] C Hummel, *Why crack when you can pass the hash*, SANS Institute InfoSec Reading Room **21** (2009), 2009.

[11] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski, *Overview of sequential monte carlo methods for parameter estimation on general state space models*, Proc. 15th IFAC Symposium on System Identification (SYSID) 2009, Saint-Malo, France, 2009.

[12] Alexander D Kent and Lorie M Liebrock, *Differentiating user authentication graphs*, Security and Privacy Workshops (SPW), 2013 IEEE, IEEE, 2013, pp. 72–75.

[13] Wenke Lee and Salvatore J Stolfo, *Data mining approaches for intrusion detection*, Defense Technical Information Center, 2000.

[14] Joshua Neil, Curtis Hash, Alexander Brugh, Mike Fisk, and Curtis B Storlie, *Scan statistics for the online detection of locally anomalous subgraphs*, Technometrics **55** (2013), no. 4, 403–414.

[15] Roberto Perdisci, David Dagon, Wenke Lee, Prahlad Fogla, and Monirul Sharif, *Misleading worm signature generators using deliberate noise injection*, Security and Privacy, 2006 IEEE Symposium on, IEEE, 2006, pp. 15–pp.

[16] C. P. Robert and G. Casella, *Monte Carlo statistical methods*, Springer-Verlag, New York, 2004.

[17] Stuart Staniford-Chen, Steven Cheung, Richard Crawford, Mark Dilger, Jeremy Frank, James Hoagland, Karl Levitt, Christopher Wee, Raymond Yip, and Dan Zerkle, *Grids-a graph based intrusion detection system for large networks*, Proceedings of the 19th national information systems security conference, vol. 1, Baltimore, 1996, pp. 361–370.

[18] David H. Wolpert, Tanmoy Bhattacharya, Russell Bent, Joshua Neil, Alexander Kent, and Annarita Giani, *Event-driven non-cooperative games, la-ur-13-22238*, Tech. report, Los Alamos National Laboratory, 2013.

[19] David H Wolpert and Simon DeDeo, *Estimating functions of distributions defined over spaces of unknown size*, Entropy **15** (2013), no. 11, 4668–4699.