Randomly Roving Agents for Intrusion Detection*

Ira S. Moskowitz†, Myong H. Kang‡, LiWu Chang, & Garth E. Longdon§
Information Technology Division, Mail Code 5540
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, D.C. 20375

March 2001

Abstract

Agent based intrusion detection systems (IDS) have advantages such as scalability, reconfigurability, and survivability. In this paper, we introduce a mobile-agent based IDS, called ABIDE (Agent Based Intrusion Detection Environment). ABIDE is comprised of various types of agents, all of which are mobile, lightweight, and specialized. The most common form of agent is the DMA (Data Mining Agent), which randomly moves around the network and collects information. The DMA then relays the information it has gathered to a DFA (Data Fusion Agent) which assesses the likelihood of intrusion. As we show in this paper, there is a quantifiable relationship between the number of DMA and the probability of detecting an intrusion. We study this relationship and its implications.

*NRL CHACS Tech. Report 5540-TM/02/003. An abbreviated version of this paper appears under the same title in: Proc. 15th IFIP WG 11.3 Working Conference on Database and Application Security, Niagra on the Lake, Canada, July 2001, Kluwer Press

†contact author: moskowitz@itd.nrl.navy.mil

‡present address: Mitretek Systems, 7525 Colshire Dr., McLean, VA 22102

§ITT Industries

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAR 2001		**00-00-2001 to 00-00-2001**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Randomly Roving Agents for Intrusion Detection	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Center for High Assurance Computer Systems,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

1 Introduction

An intrusion to a computer system may be indicated by abnormal network traffic, anomalous user activity, or application misbehavior. Intrusion detection systems (IDS)[1] which focus on detecting abnormal network activity are called *network-based* IDS, whereas intrusion detection systems that focus on detecting abnormal host activity are called *host-based* IDS. In addition, some "hybrid" IDS have sensors which collect both host and network data.

Traditional IDS which use a monolithic architecture (i.e., a centralized architecture of data collection and analysis) have a variety of problems. These problems include introducing a single point of failure (which is bad for survivability), lack of scalability, and in addition traditional IDS may be difficult to reconfigure. To overcome these shortcomings, agent based IDS which are distributed, scalable, and re-configurable have become popular [1],[2]. To take advantage of this agent based IDS idea, the US Naval Research Laboratory is designing a host-based intrusion detection system called *ABIDE (agent based intrusion detection environment)*,[2]. that uses mobile agent technology. ABIDE differs from other agent-based IDS, which usually introduce some level of coordinated communications among IDS components, in the following way:

> To avoid a targeted attack to disable the IDS, all agents randomly move around monitoring hosts. There is no fixed infrastructure, except that each host needs to be monitored, and has an agent-platform that can host agents when they decide to move in. There is neither a central site for analysis, nor a scheduler for agents in ABIDE. Also, to make the agent lightweight (i.e., using a small amount of code, which reduces network overhead associated with agent movement), tasks are split among different kinds of agents that perform different functions.

In ABIDE, there are four different kinds of agents. These agents have an implied hierarchy for the purpose of data and command flow.

1. A data mining agent (DMA) roams around (i.e., randomly chooses hosts and moves to the hosts) and acquires environmental information. It is small, lightweight, and specialized. For example, a DMA may be tasked to verify a checksum on an import system binary such as the Unix PS binary. If the agent finds suspicious data, it will acquire it for further analysis.

2. A data fusion agent (DFA) roams around and randomly interacts with the various DMA. It receives the DMA data payload and builds a larger picture of events from this data. As the DFA collects data, it can apply classic IDS techniques to determine whether an intrusion is taking place. Of course, when the DMA and DFA meet up is a function of time and the size of the network.

[1] Abbreviations can be taken as either singular or plural depending upon the context.
[2] The ABIDE idea grew out of the work of Michael Reed while he was at NRL [3, 4]

3. A probe agent (PA) is dispatched by a DFA to perform a test to confirm intrusion.

4. Once the DFA has decided that a system has been compromised, a corrective agent (CA) can be dispatched to take actions. The CA is the only agent empowered to change system state on the host systems.

In this paper, we focus on the first two types of agents about which ABIDE is concerned. We study the probabilistic behavior of the DMA reporting to a DFA. Specifically, we are concerned with two questions:

- Q1 — Given a network of a fixed number of hosts and a fixed number of DMA, what is the probability of detecting an intrusion?
- Q2 — Given a network of a fixed number of hosts, if we want to detect an intrusion with a certain confidence, how many DMA have to be deployed?

2 Special Case

In this section, we consider the situation of K DMA randomly visiting nodes of a network to discover various pieces of information and report this information back to one DFA. Each individual piece of information that a DMA obtains may not be in itself, enough to alert the DFA to an intrusion; however an aggregate of the individual pieces of information collected by the DMA may alert the DFA to an intrusion. It is this threshold criteria with which we are concerned. Once this threshold is reached, the DFA deploys a PA. Our analysis stops at the decision to deploy a PA. We refer to each host which a DMA visits as a node $\mu_i, i = 1, \ldots, M$. We assume that, as each DMA randomly travels from node to node, it picks up a unique atom of information α_i at each node μ_i. In our special case, a DMA never visits the same node twice. (In reality, a DMA may visit the same node more than once, due to the randomness of its travels, but it must visit a given fixed number of unique nodes during its sojourn. We examine the simple case, which is equivalent.) For simplicity, we assume that, at a specific time, the DMA transfers its atoms to the DFA. (In reality both the DMA and the DFA randomly travel the network. When a DMA meets up with a DFA, it then transfers its atoms to the DFA.) For simplicity, we assume that there is only one DFA. If the DFA has sufficient atoms, it declares that the intrusion threshold Θ has been reached and therefore it deploys a PA.

This is similar to the threshold schemes discussed in [5], in that below the threshold level of Θ, one can assume no knowledge, but at or above Θ, the game is up. In this paper we do not discuss how Θ is determined, nor do we discuss the case where, below Θ, the DFA has no knowledge of an intrusion. In addition, we have made further simplifying assumptions and will discuss the general situation in future work. What is salient about our work in this paper is that even with the assumptions made for simplification, the mathematics are quite difficult

to derive and computationally quite expensive to perform. We are presently investigating approximations to the formulas presented in this paper to speed up the computation and to develop "rules of thumb."

2.1 Formalism

We will now formally present our problem.

- The network is made up of M nodes μ_i , $i = 1, \ldots, M$.
- There are K DMA A_k, $k = 1, \ldots, K$.
- Each A_k visits n and only n nodes, and each node is distinct. A_k obtains a unique atom from each node. Every DMA that visits the same node μ_i receives the same atom α_i.
- After A_k has visited n nodes, it gives the n (unique) atoms α_{k_i}, $i = 1, \ldots, n$ to (the single) DFA.

Note that even though A_k has n unique atoms, $A_{k'}$ might have some of the same atoms as A_k. Therefore, when all of the A_k have reported to the DFA, we can then view the DFA as a bag of atoms, i.e., an atom might be in DFA more than once. We are only interested in the unique number of atoms in the DFA. Note that since visiting the node μ_i is equivalent to obtaining the atom α_i, so we will sometimes blur the distinction.

Let $P_K(M, n : T)$ be the probability that the DFA contains exactly T unique atoms, given that K agents have searched through M nodes, picking n (distinct) nodes per agent. Let us consider a simple example first. Keep in mind the actual probabilistic term of interest, when a threshold Θ is given, is the more complicated $\sum_{T \geq \Theta} P_K(M, n : T)$. This allows us to answer Q1 in this special case.

Example 1: Say that we have a network of 5 nodes, 2 agents, and each agent visits one node. The only non-trivial choices for T are 1 or 2, since we can never have 2 agents, each visiting one node, together visit more than 2 distinct nodes. Each run of the experiment results in an ordered pair of nodes $(N_i, N_j), i = 1, \ldots, 5,\ j = 1, \ldots, 5$. There are 25 equally likely ways to pick these pairs. We easily see that there are 5 pairs of the form (N_i, N_i), there are 20 "distinct" pairs. Thus $P_2(5, 1 : 1) = 5/25 = .2$ and $P_2(5, 1 : 2) = 20/25 = .8$

Example 2a: What happens now if we have 5 nodes, 2 agents, $T = 2$, but each agent visits 2 distinct nodes (hence 2 distinct atoms per agent). Thus, we wish to determine the probability $P_2(5, 2 : 2)$. Consider the first agent A_1 (note that we have arbitrarily called one agent the "first"). The 2 nodes visited by A_1 are represented as the unordered pair (ij) (thus (ij) = (ji)). Consider the 5x5 matrix $a_{i,j}$, $i = 1, \ldots, 5,\ j = 1, \ldots, 5$. The visits of A_1 are represented by the upper triangular matrix of $a_{i,j}$. These are the 10 unordered pairs (12),

(13), (14), (15), (23), (24), (25), (34), (35), (45). The only way to achieve $T = 2$ is for A_2 to visit exactly the same nodes as A_1. Since there are 10 ways for A_1 and A_2 to agree out of a total of 100 different possible visits for A_1 and A_2 (10 for each DMA when unconstrained), we see that $P_2(5, 2 : 2) = 10/100 = .1$.

Example 2b: Now we are in the same situation as Ex. 2, except that we have $T = 4$. To achieve this the visits from A_1 and A_2 must have a null intersection. Given any visit of A_1 there are always exactly 3 ways for the A_2 visit to have a null intersection with the given A_1 pick. Since there are 10 possible A_1 picks, there are 30 ways to achieve $T = 4$, thus $P_2(5, 2 : 4) = 30/100 = .3$. Note that since $P_2(5, 2 : 1) = 0$, and we know that $P_2(5, 2 : 2) = .1$ and $P_2(5, 2 : 4) = .3$, and $P_2(5, 2 : T) = 0$, for $T > 4$, we have that $P_2(5, 2 : 3) = .6$.

We see that calculating the probabilistic terms $P_K(M, n : T)$ quickly becomes quite complicated. Therefore we present a closed form solution. Each agent is considered a draw. Without any restrictions there are $\binom{M}{n}$ ways for a DMA to pick n nodes out of the total of M nodes. Since there are 2 draws in Ex. 1 and Ex. 2, let us start with $K = 2$. The total number of draws, *without restriction*, are $\binom{M}{n}^2$, which is the number of elements in the sample space. Now let us consider the event under question — this is where the combined number of distinct nodes picked by both agents is T. A_1 has no restriction so there are $\binom{M}{n}$ ways for A_1 to pick n nodes. Now A_2 has to pick nodes so that there are exactly T distinct nodes between the two nodes. Since A_1 has chosen n distinct nodes $M - n$ nodes are left unchosen. Thus, A_2 must pick $T - n$ nodes from the $M - n$. A_2 still has $n - (T - n) = 2n - T$ nodes to pick from the n that A_1 has chosen. Therefore there are $\binom{M-n}{T-n}\binom{n}{2n-T}$ ways for A_2 to choose. Combining this with the $\binom{M}{n}$ ways for A_1 to pick, we see that $P_2(M, n : T) = \frac{\binom{M}{n}\binom{M-n}{T-n}\binom{n}{2n-T}}{\binom{M}{n}^2} = \frac{\binom{M-n}{T-n}\binom{n}{2n-T}}{\binom{M}{n}}$. Of course for things to make sense we must have that $n \leq T \leq \min(M, 2n)$. Therefore we have that

$$P_2(M, n : T) = \begin{cases} \binom{M}{n}^{-1} \binom{M-n}{T-n}\binom{n}{2n-T} & n \leq T \leq \min(M, 2n) \ , \\ \\ 0 & \text{otherwise.} \end{cases}$$

To simplify terminology we use the extended definition of the binomial coefficient $\binom{a}{b}$ as:

$$\binom{a}{b} = \begin{cases} \frac{a!}{(a-b)!b!} & a \geq b \geq 0, \ a \text{ and } b \text{ are integers} \\ \\ 0 & \text{otherwise.} \end{cases}$$

So we see that

$$P_2(M, n : T) = \binom{M}{n}^{-1} \binom{M-n}{T-n}\binom{n}{2n-T} \tag{1}$$

Now what happens if we have 3 agents? As before the size of the sample space is $\binom{M}{n}^3$, which is the total number of ways that 3 agents may pick n nodes each. A_1 is unconstrained so it has $\binom{M}{n}$ ways to pick n nodes. The second agent A_2 has $n_2 = 0, 1, \ldots, n$ nodes in

common with A_1. Therefore $n - n_2$ nodes picked by A_2 are in the remaining $M - n$ nodes left after A_1 picked. Therefore, for n_2 fixed, there are $\binom{n}{n_2}\binom{M-n}{n-n_2}$ ways for A_2 to choose nodes. Of course we must sum over all the different values that n_2 may achieve. So all together there are $\sum_{n_2=0}^{n}\binom{n}{n_2}\binom{M-n}{n-n_2}$. For the third agent A_3, $n+(n-n_2)$ distinct nodes have already been picked by A_1 and A_2 from the M nodes. Therefore $T-(n+(n-n_2))$ nodes are picked from the remaining $M-(n+(n-n_2))$, which accounts for a factor of $\binom{M-2n+n_2}{T-2n+n_2}$. But there are the $n-(T-n-(n-n_2))$ nodes that A_3 shares with the picks of A_1 and A_2. This results in a factor of $\binom{2n-n_2}{3n-T-n_2}$. Putting all three agents together and dividing by the number of elements in the sample space results in

$$P_3(M,n:T) = \binom{M}{n}^{-2} \sum_{n_2=0}^{n} \left\{ \binom{n}{n_2}\binom{M-n}{n-n_2}\binom{2n-n_2}{3n-T-n_2}\binom{M-2n+n_2}{T-2n+n_2} \right\} . \quad (2)$$

Of course this will only result in non-zero values for $n \le T \le \min(M, 3n)$.

Similarly for 4 agents we can derive the following formula for $P_4(M,n:T)$.

$$\begin{aligned} P_4(M,n:T) = & \\ & \binom{M}{n}^{-3} \sum_{n_2,n_3=0}^{n} \left\{ \binom{n}{n_2}\binom{M-n}{n-n_2}\binom{2n-n_2}{n_3}\binom{M-2n+n_2}{n-n_3} \right. \\ & \left. \cdot \binom{3n-n_2-n_3}{4n-T-n_2-n_3}\binom{M-3n+n_2+n_3}{T-3n+n_2+n_3} \right\} . \end{aligned}$$

In general, for K picks of n distinct things from a total out of M the probability of picking T unique items is:

$$\begin{aligned} P_K(M,n:T) = & \\ & \binom{M}{n}^{-(K-1)} \sum_{n_2,\ldots,n_{K-1}=0}^{n} \left\{ \binom{n}{n_2}\binom{M-n}{n-n_2}\binom{2n-n_2}{n_3}\binom{M-2n+n_2}{n-n_3} \cdots \right. \\ & \cdots \binom{(K-2)n-n_2-\ldots-n_{K-2}}{n_{K-1}}\binom{M-(K-2)n+n_2+\ldots+n_{K-2}}{n-n_{K-1}} \\ & \left. \cdot \binom{(K-1)n-n_2-\ldots-n_{K-1}}{Kn-T-n_2-\ldots-n_{K-1}}\binom{M-(K-1)n+n_2+\ldots+n_{K-1}}{T-(K-1)n+n_2+\ldots+n_{K-1}} \right\}, K \ge 4. \quad (3) \end{aligned}$$

Thus Eqs. (1), (2), and (3) give us $P_K(M,n:T)$ for all $K > 1$. As discussed before, concentrating solely upon the probability $P_K(M,n:T)$ is not sufficient. $P_K(M,n:T)$ is the probability of getting exactly T unique atoms of information. If the information that the agents are attempting to retrieve is revealed when $T = C$, then the correct probabilistic term of interest (as previously discussed with respect to the threshold) is defined as:

$$P_K(M,n:C^+) \stackrel{\text{def}}{=} \sum_{T \ge C} P_K(M,n:T) .$$

This is the probability of K DMA obtaining at least C unique atoms.

Let us consider $P_K(M,n:T)$ and its limiting behavior for some small values of M, n, and K. The only non-zero probabilities are $P_K(M,n:T)$, for $n \le T \le \min(M, K \cdot n)$. Now

let us consider how $P_K(M, n : T)$ behaves as $M \to \infty$. This is the situation when the agents are searching over a large network.

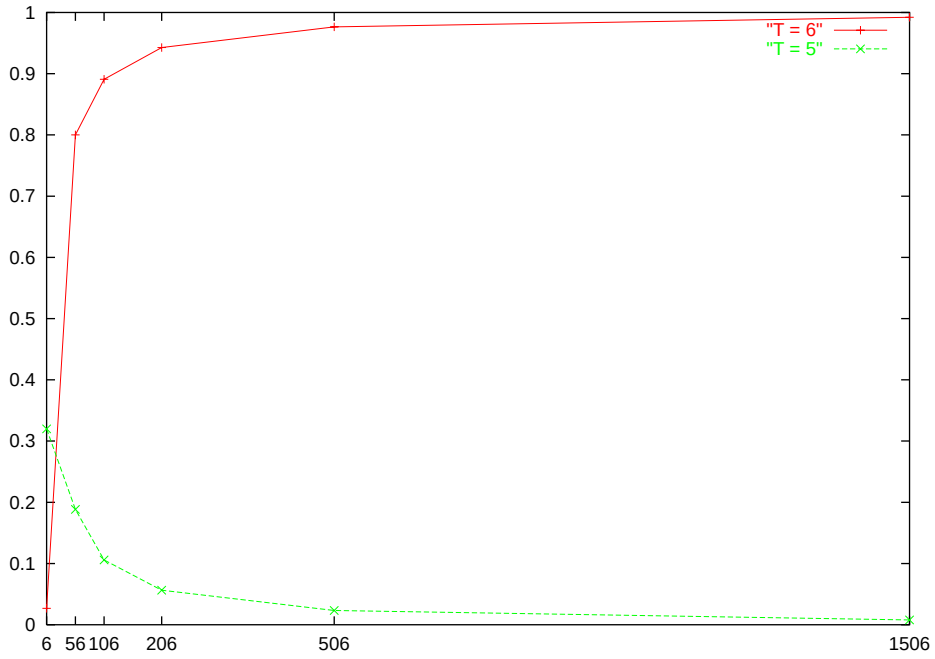

Figure 1: Limiting behavior, as M grows, of $P_3(M, 2 : 5)$ and $P_3(M, 2 : 6)$

Let M be very large with respect to Kn. The larger M is the smaller the chance of intersection between nodes picked by different agents. In Figure 1 we see a plot of the probability $P_3(M, 2 : 6)$ (approaches 1) and $P_3(M, 2 : 5)$ (approaches 0) against M. Of course Figure 1 is only dealing with a very few picks of a small number of nodes. The total number of nodes M must be several orders of magnitude larger that Kn before the limiting behavior becomes apparent. We will return to limiting behavior in the next subsection.

2.2 Some Simulation Results

In this subsection we study the behavior of $P_{30}(M, 20 : T)$. Simulations are used since the time to run the closed form solution is on the order of n^K, and thus closed form calculations are only feasible for very small values of the various terms. Simulations of 1000 were sufficient for Figure 2 (in later plots we use much larger simulations). Of course one should keep in mind that theoretically $P_K(M, n : T)$ is never 0, for $M \leq T \leq Kn$, and that $P_K(M, n : T)$ is never 1, for $M \leq T \leq Kn$. The simulations might have a value of 0 or 1, but this is because in reality the probability is either extremely small, or large, respectively. Thus we will often say that a probability is "essentially" 0 or "essentially" 1. In Figure 2 we see what happens when $K = 30$ and $n = 20$. Figure 2 shows the plots of $P_{30}(M, 20 : 381)$, $P_{30}(M, 20 : 599)$, and $P_{30}(M, 20 : 600)$ for $M = 600$,1000, 10000, 10^5, 10^6, 10^7,10^8,10^9. For the $P_{30}(M, 20 : T)$

case, M must be at 10^9 before we start seriously approaching the limiting probabilities.

With respect to the given M values we see the following:

1. When $T = 381$, the only probability $P_{30}(M, 20 : 381)$ that is not essentially 0, is when $M = 600$. (We used $T = 381$ because it is a generic "intermediate" value for M when $K = 30$ and $n = 20$.)

2. When $T = 599$ the probability $P_{30}(M, 20 : 599)$ is essentially 0 for $M = 600, 1000, 10000$, then the probability increases, but it decreases again as M grows very large.

3. When $T = 600$, which is Kn, the probability $P_{30}(M, 20 : 600)$ is essentially 0, for $M = 600, 1000, 10000$. The probability then increases until it essentially reaches its limiting value of 1 around $M = 10^9$

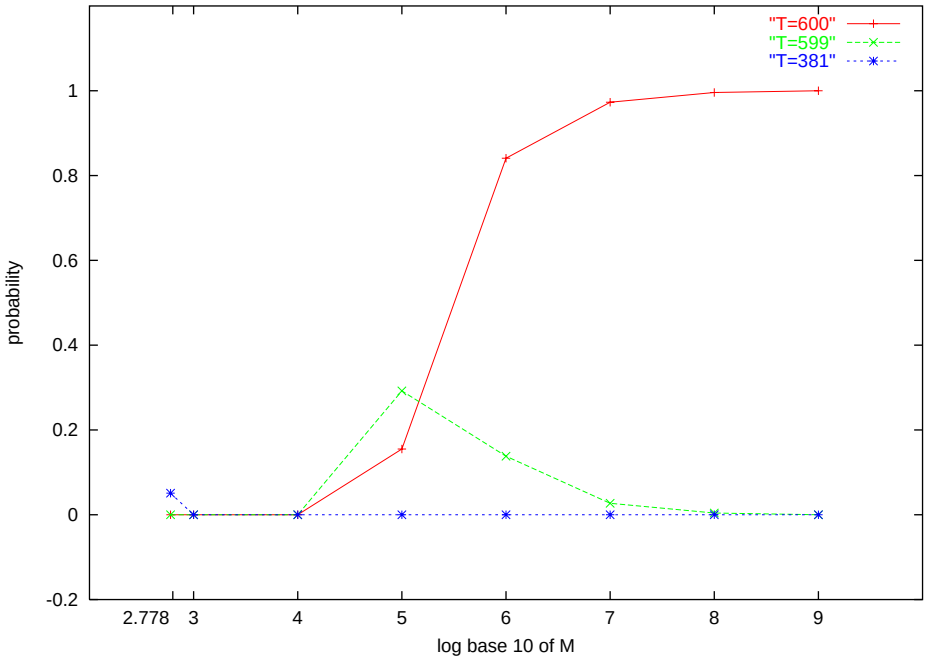

Figure 2: Simulated 1000 times limiting behavior, as M grows, of $P_{30}(M, 20 : 381)$, $P_{30}(M, 20 : 599)$ and $P_{30}(M, 20 : 600)$

We see that the distribution of the T values indexed by M, $\mathbf{T}_M(T)$ (index over M and let T run through its values in $P_K(M, n : T)$ with K, n fixed.) behaves like $I_{Kn}(T)$, which is the distribution that has probability 1 when $T = Kn$ and is zero elsewhere, as M grows. To be precise:

Given $\epsilon > 0$ and for any value of T, there exists a Ψ such that $|\mathbf{T}_M(T) - I_{Kn}(T)| < \epsilon$, for all $M > \Psi$.

We need not discuss the various types of probabilistic convergence for our needs. It suffices that $\mathbf{T}_M(T)$ behave like $I_{Kn}(T)$ for large m. We can also heuristically state this as

$$P_K(\infty, n : T) = \begin{cases} 1 & T = Kn \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The limiting behavior of $P_K(M, n : T)$ determines the limiting behavior of $P_K(M, n : C^+)$ which we may also state this heuristically as

$$P_K(\infty, n : C^+) = \begin{cases} 1 & C \leq Kn \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Let us continue to use $P_{30}(M, 20 : T)$ as an example. Above we have shown that for M large the only value of T of interest is the limiting value of $600 = 30 \cdot 20$. This agrees with our intuition. If the "universe" of the network is essentially infinite, then the different DMA do not have to be concerned with visiting the same nodes — probabilistically, it will not happen. Therefore, $T = Kn$ is the only non-zero probability, and it is of course 1. Now let us look at M values near the minimum limiting value of $M = 20$. The smallest M can be, and for the problem to still make sense, is that M is bounded from below by n. Of course, when $M = n$ the probability collapses to

$$P_K(n, n : T) = \begin{cases} 1 & T = n \\ 0 & \text{otherwise} \end{cases}$$

$M = n$ is the smallest that M can be. What happens when M is small, but not at its minimum value of n. Here we have 30 DMA, and each DMA randomly travels through a network of M nodes, and each DMA selects 20 distinct nodes from the network, and then transfers the atoms of information to the DFA.

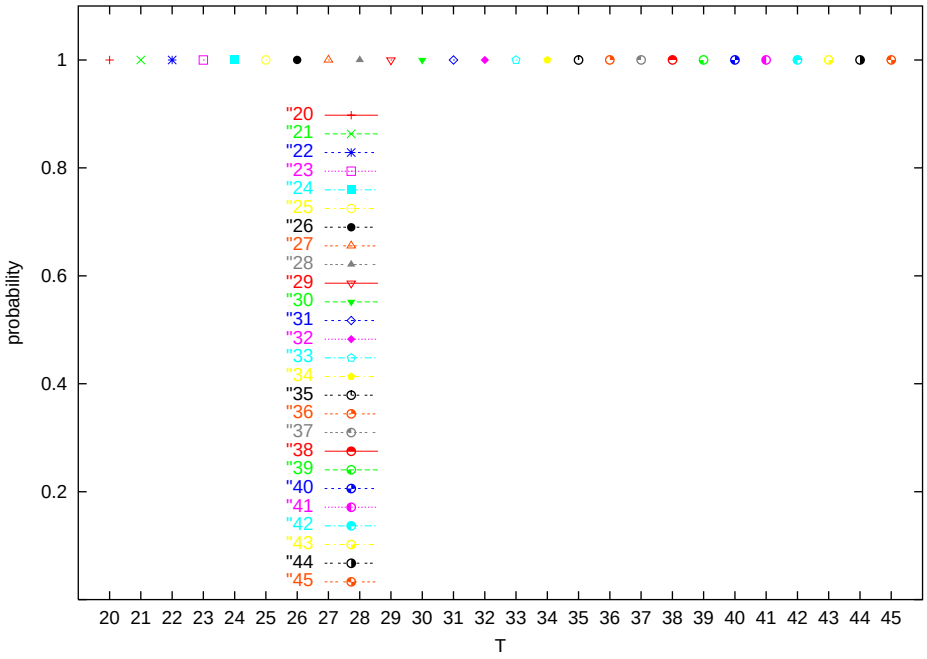

Figure 3: Plots of essentially non-zero values of (simulated 100000 times/M) of $P_{30}(M, 20 : T)$, for $M = 20, 21, \ldots, 45$. Note the simulated distributions have all their mass at $T = M$.

We wish to investigate how $P_K(M, n : T)$ behaves as $M \to n^+$. Figure 3 shows the results of simulations, run 100000 times each, of $P_{30}(M, 20 : T)$ for $M = 20, 21, \ldots, 45$. We see that, for small M, we have

$$\text{For } M \text{ "near and greater than" } n, \quad P_K(M, n : T) = \begin{cases} \text{essentially 1} & T = M \\ \text{essentially 0} & \text{otherwise} \end{cases}$$

This is because the universe is so small when M is small that, with probability very close to 1, all of the nodes are chosen by the 30 DMA. The question is how "near" is "near." In our example, the above property holds approximately for $M \leq 2n$, however, it does not hold much beyond. In Figure 4 we see what happens as M increase from 45 to 165 in steps of 10. For $M = 55$, $P_K(M, n : T)$ has two essentially non-zero values. We stay with two values in the simulations until $M = 85$. As M increases the number of essentially non-zero probabilities increase, and by hooking the values up with a curve they start to slide into a bell shape. The bell shape is very obvious in Figure 5, where we are investigating M in the intermediate range of 200 to 1000, in increments of 100. As M increases greatly, as shown in Figure 6, the bell shape slowly "hits the wall" at $T = 600$ and finally we have the limiting behavior as discussed with respect to Eq. 4. From this analysis we see that $P_K(M, n : T)$ behaves like a uni-valued distribution for M small ($P_K(\text{small } M, n : T)$.

$$\text{For } P_K(\text{small } M, n : T) = \begin{cases} \text{essentially 1} & T = M \\ \text{essentially 0} & \text{otherwise} \end{cases}$$

and it is essentially uni-valued for M large as given by Eq. 4. In the intermediate range the graph of $P_K(M, n : T)$ slides into a bell shape from the right as M increases, then behaves like as a bell shape (normal distribution), and then slides into a uni-valued distribution from the left as $M \to \infty$.

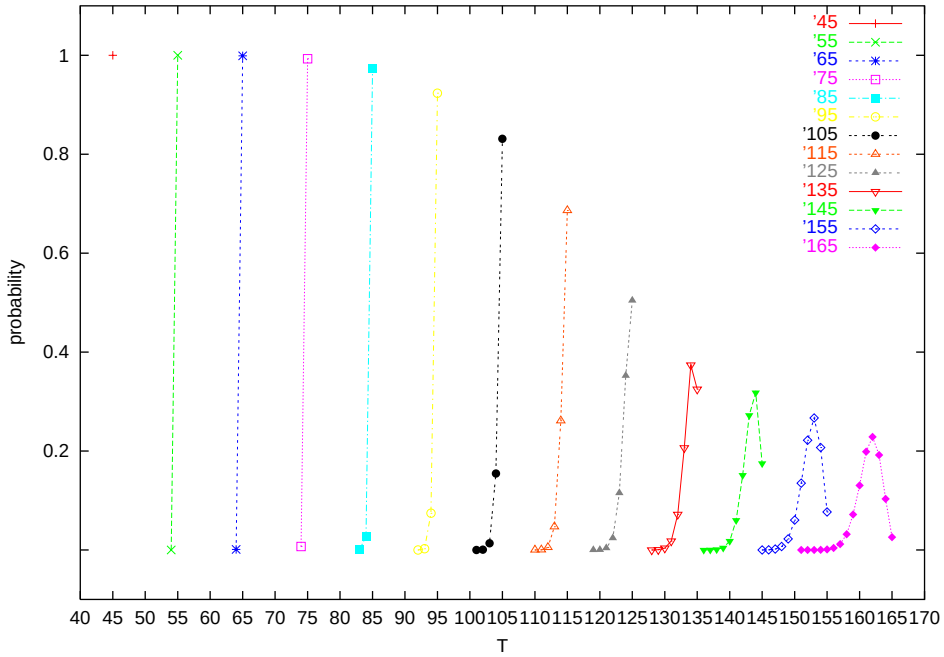

Figure 4: Plots of essentially non-zero values of (simulated 100000 times/M) $P_{30}(M, 20 : T)$ for $M = 45, 55, \ldots 165$.

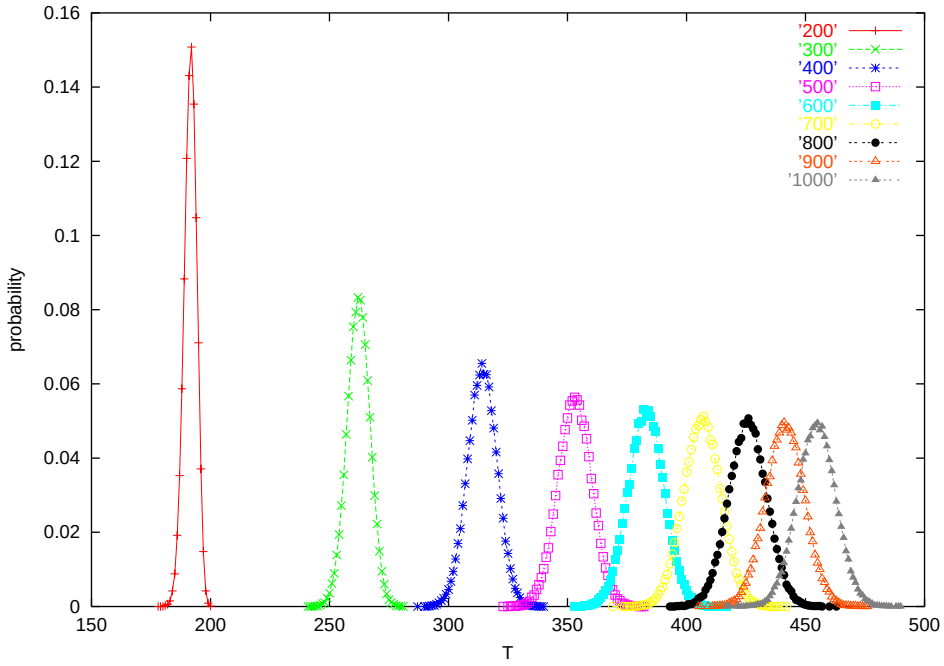

Figure 5: Plots of essentially non-zero values of (simulated 100000 times/M) of $P_{30}(M, 20 : T)$, as M grows.

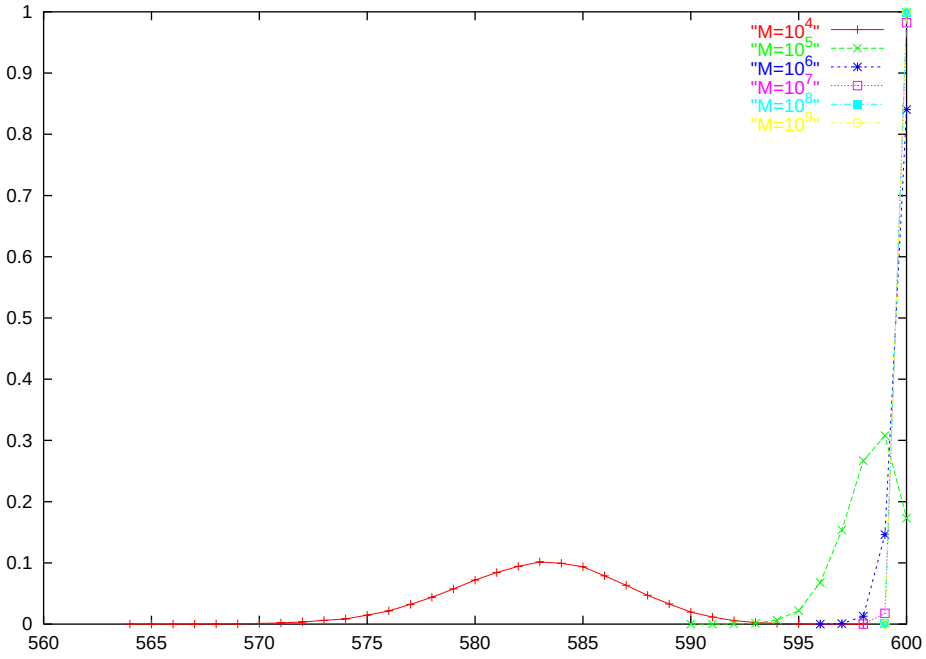

Figure 6: Plots of essentially non-zero values of (simulated 100000 times/M) of $P_{30}(M, 20 : T)$ for $M = 10^4, 10^5, \ldots, 10^9$.

2.3 Cumulative Distributions from the Simulations

Recall that the actual term of interest is $P_K(M, n : C^+)$. We could of course just sum the results from the simulations for the T values that are greater than or equal to C. However we wish to exploit the bell shape of the distribution for M in the intermediate range.

We do not know why the distribution has a bell shape. (We hypothesize that it is related to the normal approximation to the binomial distribution.) We are presently investigating it and we hope to discuss it with the workshop participants. With knowledge of the mean of T, μ and variance of T, σ^2 we could easily compute the probability, for intermediate M, by

$$P_K(M, n : C^+) = \sum_{T \geq C} P_K(M, n : T) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \int_C^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \tag{6}$$

We are viewing , with M, n, K fixed, $P_K(M, n : T)$ as a random variable T. Of course this approximation introduces error by approximating a discrete mass function by a continuous density function. If $C = \mu$ then we have, independent of the value of the variance σ^2, that

$$P_K(M, n : \mu^+) \approx \frac{1}{\sqrt{2\pi\sigma^2}} \int_\mu^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1/2$$

Note that this only holds in the intermediate range of M values, which is a relative term with respect to the size of K and n. We cannot determine how to get a computable term

for the mean T value from the closed form Eqs. (1),(2), and (3). However, we are able to theoretically determine an approximation to the mean. In fact, when we compare it with our simulations it seems to be better than an approximation! The problem is that the same approach does not work for the variance. In our problem a DMA must pick n unique nodes. There is nothing probabilistic about the number n, it is a hard constraint. However, if we pick one particular node and ask "What is the probability that a particular DMA picked this node (given no other information)?" one would answer n/M. In our problem knowledge of certain nodes being picked affects the conditional probability. For example if we know that a particular DMA did not pick any of the first $M-n$ nodes it picks node μ_{M-n+1} with probability 1. In other words we cannot assume independence. Now we perform our approximation, *assuming* independence. For a given node, we say that a DMA has a probability of picking that node equal to n/M, and all of the nodes are independent. (We see that on the average, independence does not matter. We note though that the variances derived assuming independence are larger than sample simulation variances.) Therefore the probability that a node is not picked by a DMA is $1-(n/M)$. Hence, the probability that no DMA picks a particular node is $(1-(n/M))^K$. So the probability that at least one DMA picks the node is $1-(1-(n/M))^K$. Now we are in the situation of a binomial random variable, with M trials, where the probability of a success is $1-(1-(n/M))^K$. Therefore the mean is $M \cdot \left(1-(1-(n/M))^K\right)$. Hence, we use this for our approximation of the mean T value, we call the approximation F, thus $F \approx$ mean of T, where

$$F = M \cdot \left(1-(1-(n/M))^K\right) \tag{7}$$

Table 1: *mean values*

distribution	*# simulation*	*sample mean*	F
$P_{42}(500, 17 : T)$	10^4	383	383
$P_{42}(1000, 17 : T)$	10^4	513	513
$P_{30}(600, 20 : T)$	10^5	383	383
$P_{30}(950, 20 : T)$	10^5	448	448
$P_{30}(1000, 20 : T)$	10^5	455	455
$P_{30}(10^4, 20 : T)$	10^5	583	583
$P_{30}(10^5, 20 : T)$	10^5	598	598
$P_{30}(10^6, 20 : T)$	10^5	600	600

Based on Table 1, and other data we have obtained, it seems that the approximation might actually be an equality, but we cannot prove it. There are slight differences between the F values and the sample means derived from our simulations. Of course, simulation sample means are only approximations themselves. Unfortunately, since the closed form for $P_K(M, n : T)$ is so computationally expensive, we cannot use it to compare F to the actual mean μ of $P_K(M, n : T)$. We note in Table 1, that Eq. 7 agrees with the limiting value of the distributions, as M grows, and the distributions collapse to a single non-trivial value. This is because $(1-(n/M))^K = (1-(K\frac{n}{M}/K))^K$. Since $e^x = \lim_{K\to\infty}\left(1+\frac{x}{K}\right)^K$ we have for large K that $(1-(K\frac{n}{M}/K))^K \approx e^{\frac{-Kn}{M}}$. Therefore, for large K, $F \approx M \cdot \left(1-e^{\frac{-Kn}{M}}\right)$.

By using the Taylor series for e^x we have for very large M that $F \approx Kn$.

The usefulness of F is that it gives us a way of determining if the probability associated with a given threshold is more or less than 50%. Of course, if we find a way of approximating the variance we could use any probability, not just 1/2.

In Figure 7 we see the plot of F against different K values (only the integers make sense) for $M = 1000$ and $n = 20$. If the threshold value is above (below) F, then there is less (greater) than a 50% chance of detecting the intrusion.

Hence, we have developed a useful rule of thumb, for intermediate M, that is easily calculated from only knowing M, n, and K. Of course, one should keep in mind that M being in the intermediate range is relative to the sizes of K and n. For very large M, with moderate n, one would need to deploy a large amount of DMA to use the cut-off regions. For non-intermediate M values we can use the our previous limiting results to handle the case where M is either very small or very large. Thus we have some handle on the probabilistic behavior of $P_K(M, m : C^+)$.

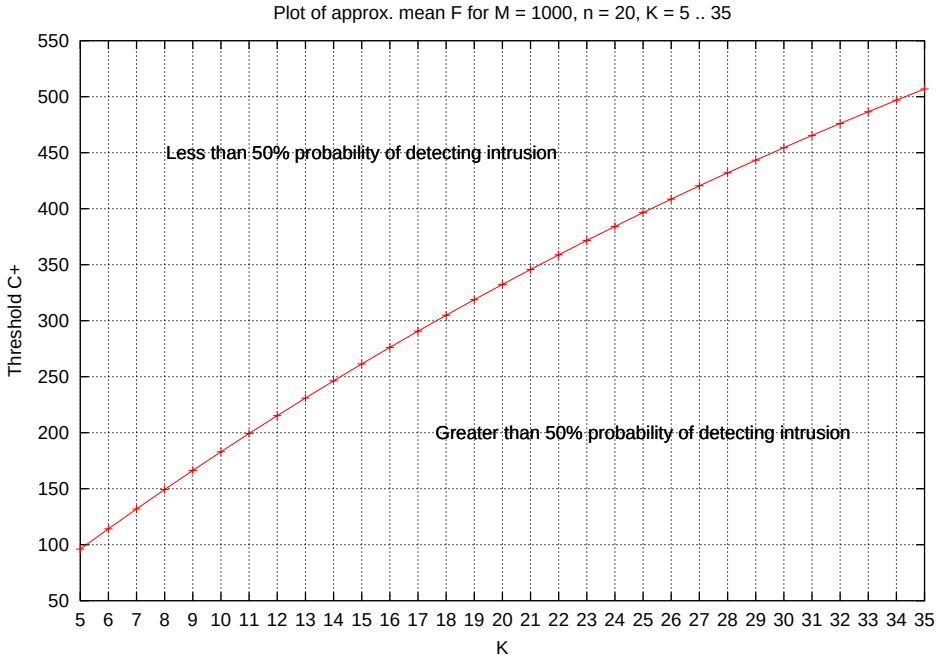

Figure 7: 50% cut-off regions

Let us go through a specific example using F, Figure 7, and Table 1. Consider a network of size $M = 1000$, $K = 30$ DMA, and each DMA visits $n = 20$ nodes, and we assume that an intrusion is detected as soon as the DFA has $\Theta = 400$ atoms of information. Since $400 < F = 455$, the probability of detecting the intrusion is greater than 1/2. If we use a different Θ that is less than 400, the probability of detecting the intrusion is even greater. On the other hand if $\Theta = 500$, we have less than a 50% chance of detecting the intrusion.

3 More General Scenarios

We have seen in the previous section that even for the simple scenario put forward we can derive a closed form solution for the probability, but it is not computationally feasible. Then why did we derive it? Intellectual integrity demands that we attempt to solve the problem. We do not have the tools to simplify the closed form but we are working on it. The terms making up the closed form are special functions and one can do approximations with them. We have also used the closed form to verify our simulations in simple cases. Another important reason that we presented the closed form solution was to show that if the solution is so computationally complex, even in the simple scenario put forward, how can we expect to derive and use a closed form solution in more complex scenarios? With this in mind, until we can approximate the special functions in $P_K(M, n : T)$, we suggest only simulations for the more general scenarios.

3.1 Future Work

Previously every DMA chose the same number of nodes. This may be relaxed and the number of nodes chosen by each DMA can be variable. If this is the case the results from the previous section can be used to bound the probabilities in this more general scenario.

We also presented a scenario where all the DMA report to the DFA at a set time. What if the times are variable, this certainly will affect the number of nodes visited. One can also look at the probabilistic terms as a stochastic process where the results change in time. Certainly, if this is the case and the DMA are traveling around the network the limiting probabilities would eventual collapse because enough nodes would have been visited.

It is not necessary that every atom of information have the same value. Perhaps some nodes atoms should be weighted more than others? Perhaps interactions between different nodes results in different types of information.

4 Conclusion

We have presented a model for a mobile-agent based IDS, called ABIDE. Using ABIDE as a framework we have analyzed a probabilistic scenario for determining if an intrusion alarm should be sounded. We have presented the closed form solution and detailed simulation results for a simple scenario. A rule of thumb has been obtained for determining certain probabilistic regions of interest. We have also discussed how our results can be used and extended to more complex scenarios.

5 Acknowledgements

We appreciate helpful suggestions from Allen Miller and the anonymous referees.

References

[1] D. J. Ingram, H. S. Kremer, & N. C. Rowe: *Distributed Intrusion Detection for Computer Systems Using Communicating Agents*, The 2000 Command and Control Research and Technology Symposium (CCRTS), 2000.

[2] E. H. Spafford & D. Zamboni: *Intrusion Detection Using Autonomous Agents*, Computer Networks, 34(4): 547-570, October 2000.

[3] M. Reed: *Agent Based Intrusion Detection Environment Architecture*, NRL Technical Report 5540/TM/117, 18 July, 2000.

[4] M. Reed: *ABIDE: Scalability*, NRL Technical Report 5540/TM/118, 6 September, 2000.

[5] A. Shamir: *How to Share a Secret*, Communications of the ACM, 22(11): 612-613, November 1979.