Temporal and Spatial Distributed Event Correlation for Network Security

Guofei Jiang, *Member, IEEE* and George Cybenko, *Fellow, IEEE*

Abstract - **Computer networks produce large amount of event-based data that can be collected for network security and management analysis. Computer networks are dynamic systems and network events are the observable of their dynamic activities. Evidence of attacks against a network and its resources is often scattered among these distributed events. Therefore a critical challenge is to correlate these events across observation space and time to detect various attack scenarios. This paper analyzes how control and estimation methods can be applied to correlate distributed events for network security. Based on those methods, a Process Query System has been implemented which can scan and correlate distributed network events according to users' high-level description of dynamic processes.**

I. Introduction

Computer networks produce large amount of event-based data that can be collected for network security analysis. These data include alerts from firewalls and Intrusion Detection Systems (IDS), log files of various software systems, routing information from the Internet and so on. Network events are instantaneous occurrences of certain types of network activity at a point in time and location. If we regard computer networks as dynamic systems, network events are the observable of their dynamic state transitions. Given the distributed nature of networks, evidence of attacks against a network and its resources is often embedded within the totality of events of the distributed systems. Moreover, attacks against a network may also involve multiple steps so that evidence of attacks is also typically distributed over time as well. With large amount of event data originating from the distributed systems in a network, a critical challenge is how to correlate these events across observation space and time to detect and track various attack scenarios.

Many traditional IDS only use single event as the signature to detect attacks, which leads to high false alarm rate. It's essential to exploit more evidence from large number of network events to get better detection accuracy. In this paper, we discuss how control and estimation methods can be applied to correlate distributed events for network security. For example, Bayesian estimation can be used to correlate events across observation space while Kalman Filter can be used to correlate events along observation time. Based on these approaches, we have developed the notion of a Process Query System (PQS) and have implemented a PQS in software, which is able to scan and correlate distributed events according to users' high-level process description.

II. Scenario Signature

A computer network consists of many components such as routers, switches, web servers, mail servers, database servers, DNS servers, IDS and firewalls. A large network like the Internet can have millions of these components. Moreover, computer networks are dynamic systems and each time interval these components produce large amount of event-based data. All these events can, in principle, be collected by network data analysis centers. The trace of an attack is often scattered in these ad-hoc events. Without efficient correlation algorithms, identifying the trace of an attack in this large and noisy event space is essentially intractable. Like other pattern recognition problems, an attack scenario signature (or pattern) is needed to distinguish the attack from other attacks and normal network activities. The detection accuracy relies on the accuracy of scenario signature as well as the accuracy of collected events. Therefore, a critical challenge is how to characterize various attack scenarios.

Figure 1 illustrates how the evidence of an attack is distributed over space and time. Based on cause-effect relationship, an attack could affect the events of multiple observation spaces at the same time t. For example, computer worms like CodeRed and Nimda generate and scan random IP numbers to search for vulnerable targets in the IP space. Since many IP numbers are not assigned to or used by a network, this active probing process could generate large volume of ICMP unreachable messages [1] in network routing. The intensive worm propagation process could also affect the latency of the Internet. Moreover, it is known that a worm breakout could also lead to unstable Internet Border Gateway Protocol (BGP) routing [2]. Based on causal relationship, here we have at least three independent observation spaces to sense the worm breakout: the volume of ICMP unreachable messages, network latency and BGP routing stability. Therefore, instead of using single event for worm detection,

This work was partially supported by: ARDA Grant F30602-03-C-0248, DARPA projects F30602-00-2-0585 and F30602-98-2-0107; National Institute of Justice, Department of Justice award number 2000-DT-CX-K001.

Guofei Jiang is with the Institute for Security Technology Studies, Dartmouth College, 45 Lyme Road, Suite 200, Hanover, NH 03755. (e-mail: gfj@dartmouth.edu).

George Cybenko is with the Thayer School of Engineering, Dartmouth College, 8000 Cummings Hall, Hanover, NH 03755. (e-mail: gvc@dartmouth.edu).

0-7803-8335-4/04/$17.00 ©2004 AACC

we can use these three indicators as a combined signature to correlate events spatially and detect worm breakout.

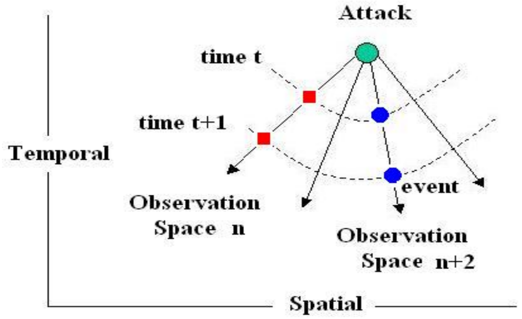

Figure 1: Temporal and spatial evidence distribution of an attack

Furthermore, an attack affects events across time as well. For example, a computer worm propagates across the Internet following an epidemic model and goes through multiple stages during its lifetime: breakout; propagation; and eradication. During this dynamic process, the volume of ICMP unreachable message follows a temporal pattern, which can be used as a temporal signature to detect the worm. In fact each observation space discussed above could sense the worm's breakout independently with its specific temporal pattern. Later in this paper, we will use a process model to characterize the temporal signature. Therefore temporal events in each observation space can be correlated with a process model to detect attacks.

As shown in Figure 1, instead of using single event as the signature, we can use a joint scenario signature combined with spatial and temporal patterns to characterize and distinguish various attacks. With more evidence exploited from the distributed events, we believe that this approach should result in better detection accuracy, especially in a noisy network environment. A challenging problem is how to get enough knowledge to build exact signatures for various attack scenarios. Mainly there are two approaches to address this issue: One is to use expert knowledge to build scenario signatures, based on causality analysis as discussed above. Another is to use data-mining technology [3] to extract signatures from large amount of training data. Though these technologies are both very important for data analysis, this issue is beyond the scope of this paper. Instead, given a scenario signature, we analyze how distributed events can be correlated according to the signature.

III. SPATIAL-BASED CORRELATION

Correlation speed and accuracy are two important performance aspects of event correlation systems. A classical approach to event correlation is rule-based analysis. That is, a correlation system constantly uses a set of predefined rules to evaluate incoming observations until a conclusion is reached. Therefore the correlation ability depends solely on the depth and capability of the rule set. Large amount of expert knowledge are required to design correct rule sets. Following the rigid paths of rule sets, observations may have to be checked against numerous conditional logics so that rule-based systems usually do not scale well. Meanwhile, rule-based systems are inherently stateless and do not handle dynamic data correlation very well. In the following sections, we discuss how control and estimation methods can be applied to improve the speed and accuracy in event correlation.

Spatial-based correlation correlates events from multiple observation spaces or sensors at the same time to detect attack scenarios. Denote an attack scenario as s and assume we have a set of m attack scenarios $S=\{s_1,s_2,\cdots,s_m\}$. Denote an observation space as O and assume we have a set of n observation spaces $O=\{O_1,O_2,\cdots,O_n\}$. Each observation space could be an independent indicator of attack scenarios. Spatial-based event correlation is about how to correlate n indicators to detect and distinguish these m attack scenarios.

A. Deterministic Correlation

The *codebook* approach [4] is a simple event correlation approach in network management. The principle of this correlation approach is based on the causal relationship of events. We believe that this approach can also be applied in network intrusion detection. Figure 2 illustrates a causality graph with three attack scenarios and four observation spaces. The directed edges in the figure represent causality. For example, if the attack s_1 occurs, it causes abnormal observations in O_1 and O_3. Conversely, this attack doesn't affect observations in O_2 and O_4. Based on these causal relationships, we can build a codebook correlation matrix as shown in Table 1, where one and zero represent "abnormal" and "normal" observations classified with specific thresholds. Therefore we can compare events from multiple observation spaces with the correlation matrix to detect and distinguish these attacks. Every attack scenario must have a distinguishable scenario signature in this correlation matrix. Expert knowledge is needed to build the scenario signature and correlation matrix. The size of the correlation matrix could be reduced but scenario signatures have to be a minimum Hamming distance apart in order to be distinguishable [4].

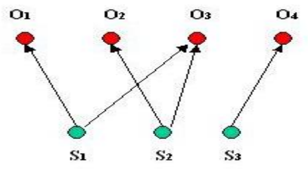

Figure 2: A causality graph

	S1	S2	S3
O1	1	0	0
O2	0	1	0
O3	1	1	0
O4	0	0	1

Table 1: Correlation matrix

Define the correlation matrix as $OS = \{os_{ij}\}$ and os_{ij} is an element of this matrix for $1 \le i \le n$ and $1 \le j \le m$. In the codebook approach, $os_{ij} = 1$ or 0, i.e. the observations are put into two categories: "abnormal" or "normal". This binary representation doesn't give much information on the intensity of "abnormal" observations. Instead, os_{ij} could be a real value, such as the volume of ICMP unreachable messages or the number of a specific system calls. In this case, we believe that the correlation problem can be formulated as the following Integer Programming problem:

$$\min_{H} \|O - OS * H\|$$

$$\text{Subject to: } \forall 1 \le i \le n,\ \sum_{j=1}^{m} os_{ij} \cdot h_j \le o_i \ ;$$

$$h_j \ge 0 \text{ and integer.} \qquad (1)$$

Here H is the hypothesis vector, $H = (h_1, h_2, \cdots, h_m)^T$. O is the observation vector from m observation spaces, $O = (o_1, o_2, \cdots, o_m)^T$. The above Integer Programming problem is about how to combine attack scenarios so that the real observations can be interpreted. For example, $H = (0,1,1,0,\cdots,0)^T$ means that attack scenarios s_2 and s_3 occurred at the same time. The codebook approach cannot detect such combination of attack scenarios. If we regard an observation space as a signal channel, the observed events usually include both signal from attacks and noise from network environment. Our integer programming approach could detect multiple instances of attack scenarios at the same time and work in the environment where the Signal/Noise ratio of observations is strong. Nonetheless, expert knowledge is needed to get os_{ij} values and these values have to be normalized across various attack scenarios. Integer programming algorithm has been well analyzed in many literatures.

B. Probabilistic Correlation

As we mentioned above, deterministic correlation approaches don't work well in a noisy environment. Network noise originates from normal network activities. For example, a major router failure could generate many ICMP unreachable messages; an alert of multiple login failures could result from a forgotten password. The question is how to detect attack scenarios based on biased observations? Denote the observation value of the observation space O as o_i and $o_i \in V$ ($1 \le i \le n$), where V is the whole set of possible o_i value. Based on expert knowledge and statistics, assume that we know the prior probabilities:

$$p(o_i | s_j) = Pr(O = o_i | S = s_j) \qquad (2)$$

for $1 \le j \le m$ and $1 \le i \le n$. That is, the distribution of observation values caused by an attack is known. According to Bayesian theorem, we can compute the posterior distribution:

$$p(s_j | o_i) = \frac{p(o_i | s_j) \cdot p(s_j)}{p(o_i)} \qquad (3)$$

Now the question is how to correlate observations from multiple observation spaces. Assume that we have observations from O and Q, we can have the joint posterior probability:

$$p(s_j | o_i, q_k) = \frac{p(o_i, q_k | s_j) \cdot p(s_j)}{p(o_i, q_k)} \qquad (4)$$

If observation space O and Q are independent, that is, events in one observation space don't cause events in another and vise versa, Equation (4) can be written as:

$$p(s_j | o_i, q_k) = \frac{p(s_j | o_i) \cdot p(s_j | q_k)}{p(s_j)} \qquad (5)$$

In fact, if we only want to identify the most likely attack that causes the current observations, we can use the right side of Equation (6) to compare the likelihood of different attack scenarios:

$$\frac{p(s_j | o_i, q_k)}{p(s_l \mid o_i, q_k)} = \frac{p(o_i, q_k | s_j) \cdot p(s_j)}{p(o_i, q_k \mid s_l) \cdot p(s_l)}. \qquad (6)$$

However, in most case, probabilities like $p(s_j)$ and $p(s_l)$ are unknown and we have to assume that they have the same distribution. Under this assumption, the ratio of the prior probabilities in Equation (6) can be evaluated against a selected threshold to determine the attack scenario. Based on this threshold, Neyman-Pearson detection theory [5] can be used to conclude the related false alarm rate and misdetection rate. Straightforwardly, if we increase the number of observation spaces, we can make the attack scenarios more distinguishable.

Multi-level causal relationships of events can be expressed with Bayesian network [6]. A Bayesian network is a directed acyclic graph in which nodes are random variables and the edges indicate that the source exerts direct causal influence on the destination. In a Bayesian network, a joint probability is factored into a set of conditional probabilities, which can be computed sequentially along the causality path in the network. Abouzakhar et.al.[7] have used a model of Bayesian networks to detect Distributed Denial of Service (DDOS) attacks, for example. Another approach for spatial correlation is to use Dempster-Shafer theory, which can combine the beliefs from multiple observation spaces.

Probabilistic correlation can work well in noisy environments. However, it is difficult to get the prior probabilities and conditional probabilities so that this approach is not as feasible as deterministic correlation methods in reality.

IV. TEMPORAL-BASED CORRELATION

In this section, we discuss how distributed events can be correlated over observation time to detect attack scenarios. Many attacks involve multiple steps and the evidence of attacks is often scattered over events in time. A computer network itself is a dynamic system and network events are observable of its dynamic activity. The temporal signature of an attack or a normal network behavior could be described as a dynamic process, deterministic or stochastic. A process model describes the state transitions of an object, which evolves with time according to specific known laws. For example, a process model can be described with a state transition equation, a Markov model, a finite state machine and so on. "State" is an important concept in temporal-based correlation.

Temporal-based correlation strives to correlate observed events in time to detect attacks and it can be formalized as a target-tracking problem. Target tracking algorithms from radar and sonar signal processing can be applied to temporal-based event correlation. If the dynamic process of an attack is known, temporal-based correlation could detect this attack by tracking whether the events follow the process of the attack. Otherwise, if the process of normal network behavior is known, temporal-based correlation could detect unknown attacks by tracking whether the events follow the process of the normal network behavior. This second approach is named "anomaly detection" in the network security literature.

A. Deterministic Correlation

Much previous work uses a finite state machine to describe the deterministic process of an attack or a software behavior. Events are evaluated against the sequence of state transitions to detect attacks. Ilgun, Kemmerer and Porras [8] used state transition diagrams to identify precisely the stages of a penetration and present only the critical events that must occur for the successful completion of the penetration. Kumar and Spafford [9] used Colored Petri-Nets to describe the temporal signatures of attacks. All these approaches modeled temporal signatures or penetration processes of attacks.

Conversely, much "anomaly detection" works have modeled the process of normal software behavior or network behavior to detect unknown attacks. Hofmeyr, Forrest and Somayaji [10] used a short sequences of system calls executed by running programs as a temporal signature to detect abnormal software behavior. Ko [11] used audit logs to capture the behavior of a program, and used that specification as an oracle against which the behavior is checked. It is known that eighty percent of a program's execution usually occurs in only 20 percent of its code. The hot paths in a program usually represent the major behavior of that program.

The first approach needs expert knowledge of attacks to build the temporal signature. The second approach could build the temporal signature of software behavior automatically based on a training process. However, the "anomaly detection" approach cannot detect the type of attacks.

B. Probabilistic Correlation

In deterministic correlation, the states of a dynamic process are observed and tracked without noise. In a noisy environment, observations are often tainted by network noise. Denote the state of a dynamic process as X and the observation as O. Denote the state X up to time t as $x_{1:t} = x_1, x_2, \cdots, x_t$ and the related observation O as $o_{1:t} = o_1, o_2, \cdots, o_t$. Since several states in a dynamic process could lead to a same observation and there is noise, the state itself is unobservable and we can only estimate the state based on observations. At time t, one task of temporal-based correlation is to correlate the observations up to time t to estimate the current state x_t, i.e. $Pr(x_t \mid o_{1:t})$. We can compute this posterior probability recursively with the Bayesian filter [12],

$$p(x_t \mid o_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid o_{1:t-1}) dx_{t-1},$$

$$p(o_t \mid o_{1:t-1}) = \int p(o_t \mid x_t) p(x_t \mid o_{1:t-1}) dx_t,$$

$$p(x_t \mid o_{1:t}) = \frac{p(o_t \mid x_t) p(x_t \mid o_{1:t-1})}{p(o_t \mid o_{1:t-1})}, \tag{7}$$

if the following assumptions about the process hold: 1. the state transition of the process model has the Markovian property, i.e., the current state S_t is only dependent on previous state S_{t-1} but not any earlier states; 2. The observation O_t is only dependent on the current state S_t but not any earlier states and observations.

Linear Kalman Filter [13] models and Hidden Markov Models (HMM) [14] are two powerful models that satisfy these two assumptions. Efficient correlation algorithms such as Kalman Filter and Viterbi algorithm [14] can be derived from Equation (7) for these specific models. The linear model used in the Kalman Filter can be described by the following equations:

$$x_{t+1} = D \cdot x_t + w \tag{8}$$

$$o_t = H \cdot x_t + v \tag{9}$$

where w and v are Gaussian noise, D and H are constant matrices. Kalman filter uses observed $o_{1:t}$ to estimate the underlying unknown x_t. In discrete case, hidden Markov model uses a state transition matrix and an emission matrix to replace Equation (8) and (9), respectively.

Denote an attack scenario as s and assume we have a set of m attack process models $S = \{s_1, s_2, \cdots, s_m\}$. The detection problem here is to determine which attack is generating these observations $o_{1:t} = o_1, o_2, \cdots, o_t$. Based on our early analysis on the Equation (6), we can compare the likelihood $p(o_{1:t} \mid s_j)$ of various attack scenarios and identify the attack with the following inequalities:

$$B < r^t = \frac{p(o_{1:t} \mid s_j)}{p(o_{1:t} \mid s_k)} < A, \tag{10}$$

where A and B are two thresholds. If the ratio r^t is bigger than A, we conclude that the attack is s_j. Conversely, if r^t is smaller than B, we conclude that the attack is s_k. If r^t is smaller than A but bigger than B, we continue to receive new observations until the ratio passes across the threshold A or B. Though the probability $p(o_{1:t} \mid s_j)$ can be recursively computed and derived from Equation (7), in most case, we don't know its analytical form of probability distribution (For example, how to compute $p(o_{1:t} \mid s_j)$ was referred as "Problem 1" of HMM in [14]). Therefore we cannot use Neyman-Pearson detection theory to conclude the related false alarm rate and misdetection rate. Denote the false alarm rate as $\alpha = p_{s=s_k}(r^t > A)$, i.e. the attack is s_k but the ratio r^t is bigger than A. Similarly denote the misdetection rate as $\beta = p_{s=s_j}(r^t < B)$. Based on the result of sequential analysis [15], we can have the following inequalities: $1-\beta \geq A\alpha$ and $\beta \leq (1-\alpha)B$.

Both the Kalman Filter linear model and HMM have been applied to model the dynamic process of attacks or normal software behaviors. Based on epidemic models and observations data of a fast-spreading worm, Zou et.al.[16] use a linear model to describe the dynamic process of worm propagation and deploy a Kalman Filter to predict worm propagation in real-time. Warrender and Forrest [17] use training data to learn a HMM to represent normal software behavior. However, usually it's difficult to get accurate parameters for these models and we are developing nonparametric weak models and algorithms for temporal-based event correlation [18].

V. Joint Temporal and Spatial Correlation

As mentioned in Section II, evidence of attacks against a network are scattered over events across observation space and time. As illustrated in Figure 3, it is important to integrate spatial and temporal event correlation together for intrusion detection. Assume that an attack process can be observed in three observation spaces. Each observation space can correlate its events along the time with a process model. At each time t, the events from these three observation spaces should be correlated spatially. There are several approaches to integrate the temporal and spatial correlation methods.

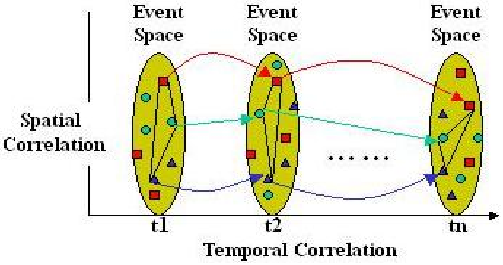

Figure 3: Temporal and spatial correlation

A. Deterministic Correlation

As shown in Figure 3, multiple observation spaces can correlate their temporal events along the time independently. The result of each temporal correlation could indicate "normal" or "abnormal" behavior of that specific observation space. With the results from multiple observation spaces, a codebook or Integer Programming approach can be used to correlate these results from temporal correlation spatially as described in Section III.

Several states in a dynamic process could lead to a same observation. Therefore the hidden state underlying an observation is unobservable. For example, HMM has an emission matrix. In temporal-based correlation, a sequence of observations could originate from many hypotheses of the hidden state sequences. With multiple observation spaces, at each time t, we can use a codebook approach to distinguish hidden states instead of attack scenarios. Theoretically as long as we add enough observation spaces with distinguishable features in the correlation matrix as shown in Table 1, we can make each state observable. In this case, a temporal correlation process can directly map a sequence of observations to a sequence of states. However, in most case, we don't need to distinguish each state for every observation since we can conclude the sequence of hidden states based on the state transition property of the process model [18]. Currently we are developing theory to address how to configure observation spaces to make hypothesis size manageable (not exponential).

B. Probabilistic Correlation

Denote the state of a dynamic attack process as X. Denote the state X up to time t as $x_1, x_2, \cdots, x_t$. Assume that we have two observation spaces O and Q to detect this attack process. Denote the observations of O up to time t as $o_{1:t} = (o_1, o_2, \cdots, o_t)$ and the observations of Q up to time t as $q_{1:t} = (q_1, q_2, \cdots, q_t)$. At each time t, spatial and temporal events can be correlated together if we have the posterior probability $p(s_t | o_{1:t}, q_{1:t})$. However, according to Equation (4) and (7), usually it's very difficult to compute the joint probabilities. But if the two observation spaces are independent, we can compute $p(s_t | o_{1:t}, q_{1:t})$ with the following three steps:

Step 1: At each time t, for each observation space, according to Equation (7), we correlate temporal events and compute $p(s_t | o_{1:t})$ and $p(s_t | q_{1:t})$, respectively.

Step 2: According to Equation (5), we correlate spatial events and compute $p(s_t | o_{1:t}, q_{1:t})$ with $p(s_t | o_{1:t})$ and $p(s_t | q_{1:t})$ from Step 1. $p(s_t)$ can be recursively computed.

Step 3: $p(s_t | o_{1:t}, q_{1:t})$ replaces $p(s_t | o_{1:t})$ and $p(s_t | q_{1:t})$. $t = t + 1$ and go to Step 1.

Theoretically even if the observation spaces O and Q are dependent, we can add another dimension to the measurement equation in (9) and correlate events by one temporal correlation process, i.e.

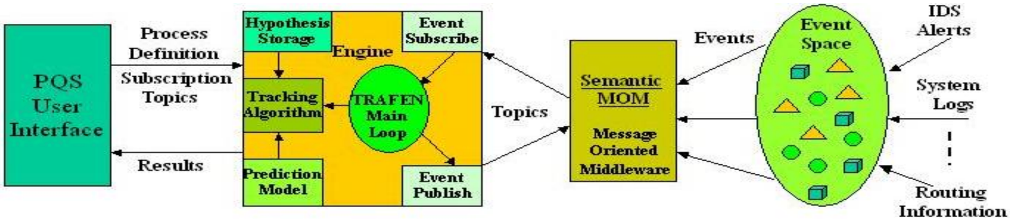

Figure 4: The architecture of PQS

$$\begin{pmatrix} o_t \\ r_t \end{pmatrix} = H \cdot x_t + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \qquad (11)$$

In discrete case, as we discussed in subsection A, multiple observation spaces could help to distinguish hidden states and lead to a sparser emission matrix in HMM. For detection problem, we can use the same approach as we discussed in inequality (10).

With more evidence exploited from distributed events, we would expect that a joint signature with temporal and spatial patterns should lead to a better detection accuracy and a lower false-alarm rate. The challenge of this approach is that we need enough knowledge to build the joint signature. Currently we are developing ten types of sensors and twelve attack scenarios to verify the proposed concept here. These sensors include Internet-based, local network-based and host-based sensors.

VI. Process Query System

A Process Query System (PQS) has been implemented using these correlation methods to scan and correlate distributed events. The PQS system allows users to fine process signatures at a high level of abstraction and submit the signatures as queries to the correlation system. The system scans and correlates distributed events according to the signatures in real time. Our current PQS only supports temporal-based correlation.

As shown in Figure 4, the PQS consists of three major components: User Interface, TRAFEN correlation engine and Message Oriented Middleware (MOM). Network events are published into a MOM with specific topics such as “Network Latency”. With a front-end user interface, users can define a process signature with high-level abstraction such as a HMM. The process signature and the topics of event subscription are submitted to the back-end TRAFEN correlation engine. TRAFEN engine parses the query and subscribes events from MOM with user-specified topics. Then MHT algorithms are invoked with user-defined process models to scan and correlate incoming events. During event correlation, MHT algorithms recursively calculate the probability of how likely the new event is associated with existing hypotheses. The new event is added into the hypothesis with the maximum likelihood and the set of hypotheses are updated. The submitted process model is used to compute the conditional probability of how likely a new event is associated with the existing hypotheses.

Based on ICMP unreachable messages collected from several routers, our PQS has been successfully used for Internet worm detection [19]. However, the current PQS implementation only supports temporal-based event correlation. In future work, the temporal and spatial event correlation technology analyzed in this paper will be implemented in the next version of PQS.

References

[1] Vince Berk, Robert Gray and George Bakos, “Using sensor networks and data fusion for early detection of active worms”, *Proc. of 2003 SPIE Aerosense Conference*, Orlando, FL, April, 2003.

[2] James Cowie, Andy T. Ogielski, BJ Premore and Yougu Yuan, “Internet worms and global routing instabilities”, *Proceedings of SPIE, Vol. #4868*, July/August 2002.

[3] Herbert A. Edelstein, *Introduction to Data Mining and Knowledge Discovery*, Two Crows Corporation, 1999.

[4] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini and D. Ohsie, “High speed and robust event correlation”, *IEEE Communications Magazine*, May, 1996.

[5] Vincent Poor, *An Introduction to Signal Detection and Estimation*, Springer-Verlag, 1994.

[6] C. Howson and P. Urbach, *Scientific Reasoning: the Bayesian Approach*, Open Court Publishing Company, La Salle, 1989.

[7] N.S. Abouzakhar and A. Gani et.al., “bayesian leaning networks approach to cybercrime detection”, *PGNet 2003*, June 16-17, 2003, Liverpool, UK.

[8] Koral Ilgun, Richard A. Kemmerer and Phillip A. Porras, **“**State transition analysis: a rule-based intrusion detection approach”, *IEEE Trans. On Software Engineering*, Vol. 21, No.3, March, 1995.

[9] Sandeep Kumar and Eugene H. Spafford, *“*A pattern-matching model for instrusion detection*”*, in *Proceedings of the National Computer Security Conference*, pp. 11-21, Oct. 1994.

[10] S. Hofmeyr, S. Forrest, and A. Somayaji "Intrusion detection using sequences of system calls." *Journal of Computer Security*, Vol. 6, pp. 151-180, 1998.

[11] C.C.W. Ko , *Execution Monitoring of Security-Critical Programs in a Distributed System: A Specification-Based Approach*, *Ph.D. Thesis*, University of California at Davis, August 1996

[12] Lawrence D. Stone, Carl A. Barlow, and Thomas L. Corwin, *Bayesian Multiple Target Tracking*, Artech House, Norwood, MA, 1999.

[13] R.E. Kalman, “A new approach to linear filtering and prediction problems,*” J. Basic Emg.,* vol. 82-D, 1969.

[14] Lawrence R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition”, *Proceedings of The IEEE*, Vol. 77, No. 2, February 1989.

[15] A. Wald, *Sequential Analysis*, John Wiley & Sons, 1947.

[16] Cliff Changchun Zou, Lixin Gao, Weibo Gong, and Don Towsley. "Monitoring and early warning for internet worms", *10th ACM Conference on Computer and Communication Security (CCS'03*), Oct. 27-31, WashingtonDC, USA, 2003.

[17] B.P.C. Warrender, S. Forrest, “Detecting intrusion using system calls: alternative data models”, *1999 IEEE Symposium on Security and Privacy*,1999.

[18] Guofei Jiang, “Weak model for robust process detection”, *SPIE Symposium on Defence and Security,* Florida, April, 2004.

[19] Vince Berk and Wayne Chung et.al., “Process query system for surveillance and awareness”, *7th World Multi-conference on Systemics, Cybernetics and Informatics (SCI2003)*, July 27-39, Orlando, FL, 2003.