# An Activity Monitoring System for Network System Using Generative Probabilistic Model

**AnigboguKenechukwu[1], Ernest-Okoye Ngozi[2].**

[1]Department of Computer Science Technology,
AnambraState Polytechnic, Mgbakwu, AnambraState.
Email: anigbogukenechukwu@gmail.com

[2]Department of Computer Engineering Technology,
AnambraState Polytechnic, Mgbakwu, AnambraState.
Email: ernestokoyengozi@gmail.com

## Abstract

*As organizations grow, the need to manage the network grows. This work studies the activity of a network, how they can be monitored to avoid corporate fraud, which is a type of fraud that is common to financial institutions most especially. Protecting your organization from advanced threats, compliance violations and operational issues is an ongoing process. It requires broad visibility, continuous monitoring, intelligent countermeasure capabilities, and ongoing adaptation to new and evolving issues and threats. A key component of that process is having the extended visibility to correlate what's happening at the host level to event data throughout the network. This paper delivers extended visibility and protection via fully integrated Activity Monitoring and presents an image capturing sensor network for intrusive observations in the network and show the potential of generative hidden Markov model and discriminative conditional models in activity recognition.*

## 1. Introduction

Attacks on computer by outside intruder are more publicized but the ones perpetrated by insiders are very common and often more damaging. Insiders represent the greatest threat to computer security because they understand their organization's business and how their computer systems work. They have both the confidentiality and access to perform these attacks. An inside attack will have a higher probability of successfully breaking into the system and extracting critical information. The insiders also represent the greatest challenge to securing the company network because they have authorized level of access to the file system.

In a quest for maximum profitability in a network, there is need to monitor the activities performed such that the network activity in a real time would be tracked, confidential information safeguarded and control over the daily activities of every staff established.

Activity monitoring system is used to detect inside threats by monitoring file access and process activity [1]. It is a powerful tool that allows one to track any local area network [1], giving you the most detailed information on when, how and what your network users do on daily basis. If it is a library public network, university or commercial organization network, Activity Monitor offers efficient control.

Activity monitoring system has become one of the vital tools in providing evidence in cases such as computer misuse and fraud. Computers and other devices are being used increasingly to commit, enable or support unwanted activity perpetrated against individuals, organizations or assets. Although it is most often associated with the investigation of a wide variety of computer crime, network activity monitoring system may also be used in civil proceedings [2]. The discipline involves similar techniques and principles to data recovery and a lot of information is retained on the computer than most people realize. It's also more difficult to

`

completely remove information than it is generally thought. For these reasons (and many more), Activity monitoring system can often find evidence or even completely recover lost or deleted information, even if the information was intentionally deleted.
The work in this paper consist of two tier application – server and client whereby the activity monitoring server can be installed in any computer in the entire local area network and the client which is the remote spy software is installed on all the computers on the network to be monitored.

## 2.General Review and Problem Definition

The failure of statutory audit to prevent and reduce mis-appropriation of corporate fund and an increase in corporate crime has put pressure on organizations, schools, banks and other firms that make maximum use of computer network.

### Independent Process Monitor

Problem
Enterprise IT systems have a constant flow of processes starting and stopping, but they are inconsistently logged. This makes detecting individual events, such as critical processes restarting properly after routine maintenance, challenging without an independent record of the event.

### Network Connection Monitor

Problem
Access to host-level detail surrounding network behaviour is a critical component of real time monitoring and forensic analysis. This can be difficult in an enterprise environment due to a lack of connection-specific log data or limited access to flow data.

### Data Loss Defender

Problem
Many advanced threats that result in data breaches employ physical means of transferring information using removable media sources such as UDB thumb drives and CD/DVD-RW devices to remove sensitive data from the network.

### Windows Registry Monitor

Problem
Changes to the Windows Registry are not natively logged, making it difficult for organizations to detect changes to the registry, including the addition of malicious software. Once embedded in the registry, malware can easily propagate by controlling processes, downloading payloads, and infecting additional systems.

### User Activity Monitor

Problem
Knowing who is logged into a particular host when malicious activity or a critical operations failure happens is a key component to comprehensive understanding of a specific event.

### Detecting Insider Threats by Monitoring System Call Activity

[3] analyzed the detection of inside threats by monitoring system call activities. They analysed there results using system call traces to see if it is possible to detect insider threats by monitoring file access and process activity, raw data are looked at in a different manner: the relationships between users and files, users and processes, and processes and files. By analysing these models and relationships, they want to learn whether it is possible to build an effective insider threat detection system for each of these relationships. If any of the models do not work, they want to discover the reasons and all technical difficulties behind the problem. Furthermore, they want to discover any characteristics or promising approaches that can help to build good profiles for users and processes. As a proof of concept, they implemented a small detection system that use one of these profiles to detect a large set of buffer-overflow attacks.
To analyse file access and process execution, they has a log of system activity, they already had a large database of system call traces, collected for the project using software developed for Seer. The traces were collected from ten machines with twenty users over two years. Their approach for analysing file access was to develop patterns for two models: user-

`

oriented and process oriented. When analysing patterns for each user, the authors decided to categorize the users into two sets: system users and normal users.

At the end of the result, the authors where able to design a system that detects insider misbehaviour, monitor system call activity and watch for danger signs or unusual behaviour. The authors describe an experimental system designed to test this approach. They tested the system's ability to detect common insider misbehaviour by examining file system and process-related system calls management.The basic problems and weaknesses confronting the paper was the modes in which they monitor the activities on the network with dates only with regards to memory efficient, the speed of processing and the organization of logs in the server system.

## 3. Notation

The time series data obtained from the activity are divided in time slices of constant length $\Delta t$. We denote an activity reading for time $t$ as $xit$, indicating whether activity$i$ fired at least once between time $t$ and time $t+\Delta t$, with $xit \in \{0,1\}$ . In an organization with $N$ computers installed, we define a binary observation vector $\mathbf{x}t=(x1t,x2t,\ldots,xNt)T$. An activity at time slice $t$ is denoted by $y_t$ with $yt \in \{1\ldots K\}$, therefore, the recognition task is to find a sequence of labels $y1{:}T=\{y1,y2,\ldots,yT\}$ that best explains the sequence of observations $\mathbf{x}1{:}T=\{\mathbf{x}1,\mathbf{x}2,\ldots,\mathbf{x}T\}$ for a total of $T$ time steps.

### 3.1 Hidden Markov model

The hidden Markov Model (HMM) is a generative probabilistic model consisting of a hidden variable $y$ and an observable variable $x$ at each time step. In our case the hidden variable is the activity performed, and the observable variable is the vector of image capture. Generative models provide an explicit representation of dependencies by specifying the factorization of the joint probability of the hidden and observable variables $p(y_{1:T}, \mathbf{x}_{1:T})$. In the case of HMMs there are two dependency assumptions that define this model.

- The hidden variable at time $t$, namely $y_t$, depends only on the previous hidden variable $y_{t-1}$ (*Markov assumption* [4]).
- The observable variable at time $t$, namely $x_t$, depends only on the hidden variable $y_t$ at that time slice.

The joint probability therefore factorizes as follows

$$p(y1{:}T,\mathbf{x}1{:}T)=p(y1)p(\mathbf{x}1|y1)\prod t=2Tp(yt|yt-1)p(\mathbf{x}t|yt).(1)$$

The different factors further specify the workings of the model. The initial state distribution $p(y_1)$ is a probability table with individual values denoted as follows

$$p(y1=i)\equiv\pi i.(2)$$

The observation distribution $p(\mathbf{x}_t|y_t)$ indicating the probability that the state $y_t$ would generate observation **x_t**. In our case each activity capturing is modelled as an independent Bernoulli distribution, giving

$$p(\mathbf{x}t|yt)=\prod n=1Np(xnt|yt) \quad (3)$$
$$p(xnt=v|yt=i)=(\mu in)v(1-\mu in)1-v. \quad (4)$$

The transition probability distribution $p(y_t|y_{t-1})$ represents the probability of going from one state to the next. This is given by a conditional probability table where individual transition probabilities are denoted as follows:

$$p(yt=j|yt-1=i)\equiv aij.(5)$$

Our HMM is therefore fully specified by the following parameters $A = \{a_{ij}\}$, $B = \{\mu_{in}\}$ and $\pi = \{\pi_i\}$. The parameters are learned from training data using maximum likelihood. This is done for each user separately. Using these parameters we can find the sequence of

`

activities performed by various users in organizations. This is done by calculating $p(y_{1:T}|\mathbf{x}_{1:T})$, which can be easily calculated from the joint probability [5].

## 4. Activity Monitoring System

Correlating network-wide event data with activities occurring at the client level is often hindered by the fact that critical client-based activities may not be consistently logged, often requiring multiple solutions to fill in the information gaps. Activity Monitoring provides independent awareness and insight into what's happening on client systems, providing a critical layer of protection from a broad spectrum of problems, ranging from important operational events such as system and application failures to security and compliance violations tied to unauthorized or malicious activity. Centrally monitored and managed as a fully integrated component of the Activity Monitoring includes:

**Independent Process Monitor**

Detects and records process and service activity that may not otherwise be reported. This can identify and alert on important behavior such as clients running blacklisted processes (such as peer-to-peer clients), critical processes stopping or any non-approved process starting up.

**Windows Registry Monitor**

Monitors the Windows Registry for additions, modifications, deletions, permission changes, and ownership changes. This visibility provides greater insight into changes or manipulations of Windows operating systems, including the addition of new startup processes, to detect advanced threats and compromised client systems.

**Network Connection Monitor**

Independently records network connection activity to and from the client systems, providing a detailed, independent log of all network connections opened and closed on a client system. It detects and alarms on critical events such as unauthorized web or FTP servers.

**Data Loss Defender**

Monitors and prevents data transfers to and from removable media such as CD/DVD-RW devices and USB drives. Data Loss Defender logs, alerts on, and audits all data transfers to removable media ports and can optionally block transfers on selected machines and devices.

**Clients Activity Monitoring**

Logs any user or process that authenticates to the server, This independently records an audit trial that can be used to either supplement local auditing systems or to validate that system logs have not been modified on the server.

Activity Monitoring combined with comprehensive Security Intelligence Platform provides tremendous visibility into what's going on where, and when it happens throughout the IT Environment.

This work can independently

i. Detect and alert whenever a blacklisted process starts or when a critical process stops or fails to restart following a specific event, such as a reboot.

ii. Creates an independent log of every network connection on a monitored client system, including relevant detail such as ID port, communication direction and the process that opened the connection.

iii. Detect the use of removable media directly on the client, generating an alarm before data can be transferred either to or from a removable media device.

iv. log who is logged in and for how long, correlating user audit activity with other log and event data, creating a comprehensive audit of user behavior throughout the IT environment.

v. Monitor the Windows Registry to detect changes, including the introduction of malicious software and new startup processes.

By harnessing the power of Smart Response, this paper can provide extensive, active protection at the client level from advanced threats, compliance violations, and operational issues. The smart response can also;

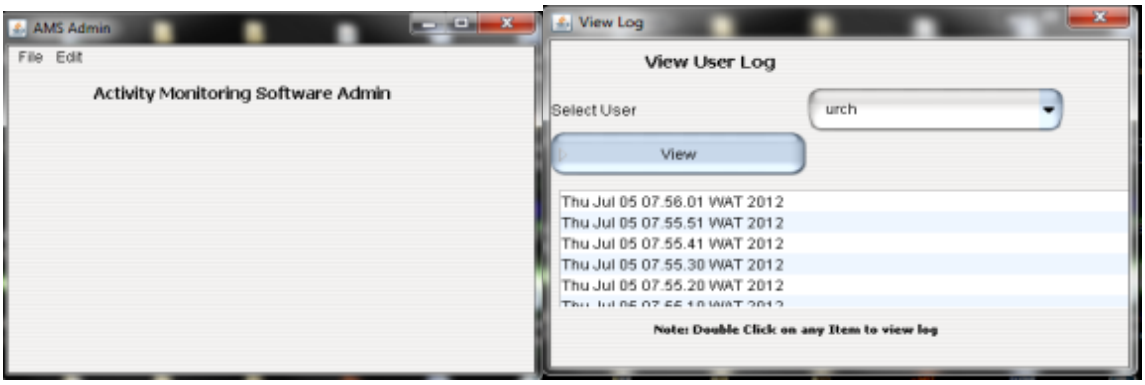
`

i. Restart individual processes, pulling all relevant information, such asthe process name and impacted host, directly from the alarm.
ii. Be configured to automatically close an unauthorized port or shut down a suspicious network connection in response to any alarm.
iii. With the help of Data Loss Defender can automatically prevent data from being transferred to or from removable media directly on the host, by immediately ejecting the device, for real time protection
iv. Remove from a network unauthorized users or those who engage in suspicious or detrimental behavior – either with or without requiring authorization before taking action.

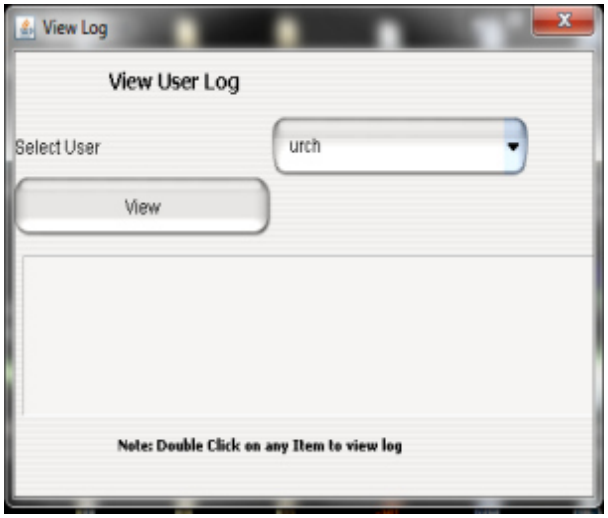
Work when changes to the Windows Registry are detected, an alarm is triggered alerting IT and security personnel to the activity. If the change is determined to be malicious, administrators can approve a SmartResponseplug-in that automatically disables the startup processes to prevent the spread of Malware.

## 5. EXPERIMENTAL RESULTS

Figures shows the results of activity monitoring from server and client side by using exemplar based monitoring algorithm. Fig. 1(a) shows the original image of the admin side. (b) shows a sample of view user log. (c) shows each user log account.




{a} (b)



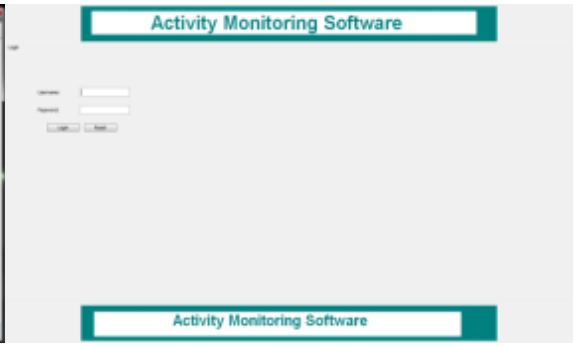

(c)

Fig. 2 Shows the client side (d) shows sample of captured screen and (e) shows the main log in interface of the client side

.

`

(d)


Activity Monitoring Software
Activity Monitoring Software

(e)

`

## REFERENCES


T. L. M. van Kasteren, G. Englebienne, B. J. A. Kröse, "An activity monitoring system for elderly care using generative and discriminative models", September 2010, Volume 14, Issue 6, pp 489–498

Jaeyeon Jung (June 2006): "*Real time detection of malicious network activity using stochastic models*", http://www.valley.org.uk/realtimedetection/mis.it/networkmontoring/mis./html. Retrived on [24/03/2012].

Nguyen, N., Reiher, P. and *Kuenning*, G."*Detecting insider threats by monitoring system call activity*", Information Security Journal: A Global Perspective Volume 21, 2012 - Issue 4

Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286.

Duong TV, Bui HH, Phung DQ, Venkatesh S (2005) Activity recognition and abnormality detection with the switching hidden semi-Markov model. In: CVPR '05: Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). IEEE Computer Society 1:838–845


`